COLUMBIA
BUSINESS
SCHOOL

Spring 1999
Professor Costis Maglaras
B9801-33 Stochastic Processing Networks
409 Uris Hall

Part IV: Discrete-Review Policies for Stochastic Network Control

Summary. We have seen how to formulate and solve a fluid optimal control problem associated with a stochastic network control problem. In this note we describe a family of discrete-review policies that can be used as a general mechanism to translate the solution of a fluid optimal control problem back into the original stochastic network in a way that guarantees several desirable properties. Namely, stability and fluid-scale asymptotic optimality; the latter tests the implemented policy for asymptotic optimality in the limiting regime where the model approximation is valid and the stochastic problem reduces to one of transient optimization. We describe the general structure of these policies, their basic characteristics, and finally sketch a proof of stability and fluid-scale asymptotic optimality. A brief remark about elementary large deviations bounds is included. Another family of continuous-review policies will be described that achieve the same stability and fluid-scale asymptotic optimality properties. We conclude with some extensions to this framework and a short discussion on recent bibliography on the topic.

IV.1 The Fluid Model Approach to Network Control Problems

Dynamic control problems for stochastic processing networks are both analytically and computationally hard. While most often one relies on the use of heuristics that are validated through simulation studies, one approach that has emerged from research over the past 10-15 years is based on a hierarchy of approximate models that provide tractable "relaxations" of these problems as a framework for analysis and synthesis. In particular, we have seen that the analytical theory associated with fluid approximations has produced important insights in understanding how the performance of a multiclass network depends on different design and control parameters. This is our starting point here. Specifically, the approach taken here is based on approximating (or replacing) the stochastic network by its fluid analog -this is a model with deterministic and continuous dynamics-, solving an associated fluid optimal control problem, and then using the derived fluid control policy in order to define an implementable policy in the stochastic network. This procedure is summarized below:

1. Consider a dynamic control problem for the original stochastic network;
2. Form fluid analog of stochastic network and solve the associated fluid optimal control problem;
3. Translate/implement the optimal fluid control in original stochastic network;
4. Consider fluid limit of stochastic network under implemented policy;

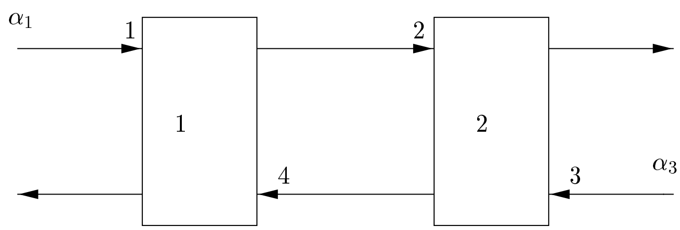

Figure 1: The *Rybko-Stolyar* network

5. Verify fluid-scale asymptotic optimality and stability.

Stages 1 to 3 are clear. Stages 4 and 5 describe a criterion for performance analysis under the implemented policy that is consistent with the model approximation adopted at stage 2, in the following sense: the implementation is tested for asymptotic optimality in the limiting regime where the model approximation is valid. This criterion is referred to as *fluid-scale asymptotic optimality* (FSAO). To be more precise, fluid limits are derived through a Law of Large Numbers (LLN) type of scaling, where one observes the system behavior starting form a large initial condition for a proportionally long time horizon, which essentially yields a deterministic transient response model. Thus, the proposed criterion tests whether in the fluid limit regime the system's limiting performance achieves that of the optimal fluid (transient) response that was used in stage 3 in designing the policy under investigation. This is a "minimal" requirement for the implemented policy. In comparison to the original problem at hand, it provides a relaxed notion of optimality that appears to be simpler and one that hopefully could be achieved even for the general class of multiclass networks. Finally, apart from FSAO, we also require that the original stochastic network is stable under the implemented policy provided that the traffic intensity parameter at each station is less than one. This note analyzes stages 3 to 5.

IV.1.1 Rybko-Stolyar Revisited

Recall the Rybko-Stolyar network shown in Figure 1 for the parameters: $\alpha_1 = \alpha_3 = 1, \quad \mu_1 = \mu_3 = 6$ and $\mu_2 = \mu_4 = 1.5$. The fluid model associated with the Rybko-Stolyar network is as follows.

$$\dot{z}(t) = \alpha - Rv(t), \quad z(0) = q, \tag{IV.1}$$

$$v(t) \geq 0, \quad v_1(t) + v_4(t) \leq 1, \quad v_2(t) + v_3(t) \leq 1, \quad z(t) \geq 0, \tag{IV.2}$$

where

$$\alpha = \begin{bmatrix} \alpha_1 \\ 0 \\ \alpha_3 \\ 0 \end{bmatrix}, \quad R = \begin{bmatrix} \mu_1 & 0 & 0 & 0 \\ -\mu_1 & \mu_2 & 0 & 0 \\ 0 & 0 & \mu_3 & 0 \\ 0 & 0 & -\mu_3 & \mu_4 \end{bmatrix}.$$

Fluid optimal control policy. The optimal control for the fluid model can be characterized as *Last-Buffer-First-Served (LBFS)* with server splitting whenever an exiting class is emptied at the other server. That is, each server has responsibility for one incoming buffer and one exit buffer; the exit buffer is given priority unless the other server's exit buffer is empty, in which case server splitting occurs. For an explanation of the latter situation, let us focus on the behavior of server 1 when buffer 2 (the exit buffer for server 2) is empty and buffer 1 is non-empty. In that circumstance, given the system parameters, server 1 devotes 25% of its effort to buffer 1 (its own incoming buffer) so that server 2 can remain fully occupied with class 2 jobs, and devotes the other 75% of its effort to draining buffer 4 (its own exit buffer). This policy is myopic in the sense that it removes fluid from the system at the fastest possible instantaneous rate, regardless of future considerations, and it is optimal regardless of the horizon length T.

Policy translation. Given the solution to the fluid optimization problem, which has the structure of a static priority rule together with some "boundary modifications," we now seek to translate the derived policy back into the original stochastic network. The following natural alternatives arise.

LBFS. The simplest candidate policy is to use the static priority rule that emerges from the optimal fluid control law, which gives priority to exiting classes in each server. As we know by now, this policy is unstable! That is, the static priorities derived from the optimal fluid control policy by neglecting "boundary behavior" have catastrophic performance: they cause instability!

LBFS with priority reversal (LBFS/PR). Here each server uses LBFS as its "default," but switches to the opposite priority when the other server's exit buffer is empty. This policy is stable, but its asymptotic performance is not satisfactory, as will be explained below.

LBFS with server splitting (LBFS/SS). Here we implement exactly the optimal policy derived form the fluid model, splitting server effort in the percentages prescribed by the fluid optimal control policy. Whenever one of the queue lengths is empty, any positive server utilization predicted by the optimal fluid control policy for this class will not be implementable due to the discrete dynamics of the stochastic network. In such cases, this percentage of server utilization will be reallocated to the other class waiting to be processed at that server, if this is non-empty. For example, when $q = (0, 0, +, +)$ the optimal server allocation from the fluid control problem is $(0.25, 1, 0, 0.75)$, yet the implemented server allocation in the stochastic network will be $(0, 0, 1, 1)$; a "+" denotes positive buffer content. Again, this policy is stable, but its asymptotic performance is not satisfactory.

Performance analysis of candidate policies. Using any one of the control policies just described, it is natural to consider system behavior under a sequence of initial conditions $\{q^n\}$, such that $|q^n| \to \infty$ as $n \to \infty$, keeping all other system parameters fixed. Consider, for example, the case where $q^n = n[1, 0, 0.5, 1]$. Let $Z^n(\cdot)$ be the four dimensional queue length process with initial state $Z^n(0) = q^n$, define its fluid scaled version as

$$\bar{Z}^n(t) = \frac{Z^n(nt)}{n}, \quad 0 \le t \le T, \tag{IV.3}$$

and ask whether $\bar{Z}^n$ converges to a limit trajectory as $n \to \infty$ that is optimal in the fluid model. Essentially, we are testing whether the system behavior under one of the implemented policies approaches -as n grows and the stochastic problem approaches one of fluid, or transient, optimization- the optimal performance that one started with.

For both LBFS/PR and LBFS/SS, the scaled processes $\bar{Z}^n$ do converge to a deterministic limit as $n \to \infty$, but that limit does not coincide with the optimal fluid trajectory. That is, although these policies may be intended as implementations of the optimal fluid control policy, they do not in fact achieve as their fluid limits a trajectory that is optimal in the fluid model. In detail, both LBFS/PR and LBFS/SS will introduce undesirable idling periods at server 2 while waiting for new class 1 jobs to complete service at station 1. This behavior will not change as n grows, since queue 2 will always have either one or no jobs waiting, and this will lead to the suboptimal behavior claimed above. Both policies fail because the servers are too slow in switching from myopically draining cost out of the system to guarding against idleness that will prevent optimal cost draining in future times. Following this argument one would expect that performance under LBFS/SS will be worse than that under LBFS/PR. Do you agree? Why?

Finally, some concluding remarks. Despite the apparently modest objective of FSAO, the meaning of the fluid policy in the original network is subtle. This will be demonstrated in the next section through the analysis of the Rybko-Stolyar network, where although the associated fluid optimization problem is "trivial," each of three "obvious" interpretations in the stochastic network is "wrong." In fact, given the solution of the associated fluid optimal control problem, which is easily computable, it is surprisingly difficult to translate the optimal fluid control policy into an implementable policy in the original stochastic network, due to the finer structure of the original network model.

IV.2 Fluid-scale Asymptotic Optimality

Fluid-scale asymptotic optimality is a relaxed notion of optimality in comparison to the original criterion in the stochastic network, consistent with the policy design procedure considered here. The following definition is adapted from Meyn [Mey97b]:

Definition IV.2.1 *Consider any sequence of initial conditions $\{x_n\} \subset \mathbf{X}$ such that $|x_n| \to \infty$ as $n \to \infty$ and assume that for every converging subsequence $\{x_{n_j}\}$ and some random variable $\bar{Z}(0)$, $\bar{Z}^{n_j}(0) \to \bar{Z}(0)$ almost surely. Then a policy π^* is said to be asymptotically optimal under fluid scaling if for all admissible scheduling policies π*

$$\liminf_{n\to\infty} \left[\mathbf{E}^{\pi^*}_{x_n} \int_0^T g(\bar{Z}^n(t))dt - \mathbf{E}^{\pi}_{x_n} \int_0^T g(\bar{Z}^n(t))dt \right] \le 0. \tag{IV.4}$$

Remark. Meyn [Mey97b] stated this definition for the case of linear costs, where he considered the limit as $T \to \infty$, and restricted attention to stable scheduling policies. By focusing on a finite horizon cost, one need not impose this stability restriction, which is difficult to check. Furthermore, the finite horizon criterion remains meaningful even when the traffic intensity at every station is not restricted to be strictly less than one, which allows the study of networks in heavy-traffic. Choosing T according to (III.28), we exactly recover Meyn's criterion in this finite horizon setting. Finally, the assumption regarding the a.s. convergence of the initial fluid scaled queue length vector appears to be a mild one and it will be motivated shortly.

Fluid limit equivalent conditions. One would like to establish a criterion of fluid-scale asymptotic optimality that depends on the fluid limit trajectories and not the prelimit of fluid scaled sequences as in (IV.4). To gain intuition we only consider the simpler sequence of initial conditions $\bar{Z}^n(0) = nq$, for which $\lim_n \bar{Z}^n(0) = q$ a.s.. In this case, the property of FSAO is reduced to checking whether starting from an arbitrary initial condition z the fluid limit under a candidate policy achieves the optimal cost of $\bar{V}^g(z)$; this was the check performed in the Rybko-Stolyar example. Let's see why.

Given that g is non-negative and all processes $\bar{Z}^n$ are defined in the same probability space, $(\mathbf{X}, \mathcal{B}_X)$ equipped with the probability measure P^π, we have by Fatou's Lemma,

$$\liminf_n \mathbf{E}^\pi_{x_n} \left[\int_0^T g(\bar{Z}^n(t))dt \right] \geq \mathbf{E}^\pi \left[\liminf_n \int_0^T g(\bar{Z}^n(t))dt \right]. \tag{IV.5}$$

Note that since the limiting initial condition is deterministic, the expectation operator on the RHS can be omitted; this was with respect to the distribution of the limiting initial condition that is a proper random variable by assumption -see the definition. Under the sequence of initial conditions considered here, the limit of these sequences exists a.s., and it is equal to

$$\lim_j \int_0^T g(\bar{Z}^n(t,\omega))dt = \int_0^T g(z(t))dt,$$

where $z(0) = q$. Clearly,

$$\int_0^T g(z(t))dt \geq \bar{V}^g(q), \tag{IV.6}$$

which implies that

$$\liminf_n \mathbf{E}^\pi_{x_n} \left[\int_0^T g(\bar{Z}^n(t))dt \right] \geq \bar{V}^g(q), \tag{IV.7}$$

and the FSAO criterion has now been reduced to checking whether for all limiting initial conditions q, the fluid limit under a specified policy achieves satisfies (IV.6) with equality; i.e., achieves asymptotically optimal performance under fluid scaling.

> **Remark.** For a more general sequence of initial conditions one simply needs to be more careful with the this limiting procedure, but the fluid limit condition that needs to be checked is the same. We have restricted attention to sequences of initial conditions that yield *undelayed* fluid limits. This is intuitive when studying fluid optimal control problems, and moreover greatly simplifies the required analysis for asymptotic optimality.

Interpretation of the FSAO criterion. Fluid model optimality is a property about transient response. In the stochastic network, FSAO says that the transient recovery of the system when initialized from a large initial backlog and observed over a proportional time horizon will again be optimal. It will mimic the optimal response computed in the fluid model.

IV.3 Discrete-Review Policies for RS Network

Discrete-review (DR) policies, and specifically, policies that

1. step through time in large intervals of time, and

2. within each such interval a deterministic planning logic is employed,

have also been proposed by other researchers in the areas of applied probability and network control. The same idea will be used here, where in particular,

3. the deterministic planning logic employed is based on the solution of the associated fluid optimal control problem.

Policy mechanics. A discrete-review policy is defined by or is derived from a function $l : \mathbf{R}_+ \to \mathbf{R}_+$, the function $r^g : \mathbf{R}_+^K \to \mathbf{R}^K$ derived in section III.3, plus a K-dimensional vector β that satisfy the following restrictions. First, $l(\cdot)$ is real valued, strictly positive, concave, and further satisfies

$$\frac{l(x)}{\log(x)} > c_0 \quad \text{and} \quad \frac{l(x)}{\log(x)} \to \infty \quad \text{as } x \to \infty, \tag{IV.8}$$

and

$$\frac{l(x)}{x} \to 0 \quad \text{as } x \to \infty. \tag{IV.9}$$

The significance of the growth condition (IV.8) will become apparent later on. Second, β is a vector in $\mathbf{R}_+^K$ that satisfies

$$\beta > \mu, \tag{IV.10}$$

where μ is the K-vector of service rates.

Under any of the policies to be considered, system status will be observed at a sequence of times $0 = t_0 < t_1 < t_2 < \ldots$; we call t_j the j^{th} review point and the time interval between t_j and t_{j-1} the j^{th} planning period. Define $l_0 = al(|Z(0)|)$, where a is a small ($\ll 1$) positive constant; the value of this constant is not crucial to the operation of the proposed discrete-review policy and for that reason it is not included as one of the defining quantities of these policies. Given that the queue length vector $z = Z(t_j)$ is observed at t_j, server activities over the next planning period are determined by solving a linear program, the data for which involve $l(\cdot)$, $r^g(\cdot)$, and β. To be specific, having observed z the controller sets $\tilde{z} = z/|Z(0)|$,

$$l = l_0 \vee l(|z|), \;\; r = r^g(\tilde{z}), \;\text{and}\; \theta = l\beta, \tag{IV.11}$$

and then solves the following linear program: choose a K-vector x of time allocations to

$$\text{maximize} \quad r'x \tag{IV.12}$$

$$\text{subject to} \quad z + l\alpha - Rx \geq \theta, \quad x \geq 0, \quad Cx \leq l\mathbf{1}. \tag{IV.13}$$

Interpretation.

- *Overall structure.* The basic structure is the following: at every review point we observe the work-in-process (WIP) in the network, and based on that we decide on the length of the next planning horizon. At every review period we plan on what to do based on the solution of a linear program, the data for which is derived using the solution of the associated fluid optimization problem. In order to ensure smooth operation of the system we enforce safety stock requirements.

- *Magnitude of planning horizon and safety stocks.* At first pass, it is convenient to think that when $|Z(t)| \sim \mathcal{O}(n)$ then $l \approx c\log(n)$. The safety stock requirements are $\theta = \beta l$. The condition (IV.10) ($\beta > \mu$) ensures that every vector of time allocations computed from the solution of the planning LP will be implementable from stock in hand upon review.

  (This way we will try to avoid the problems we had in the Rybko-Stolyar and the criss-cross networks, where the system incurred undesirable idleness because the servers were starved while waiting for work to be forwarded to them.)

- *The constraints of the planning LP.* First, an interpretation of this planning logic will be provided assuming that this linear program is feasible; the case where (IV.12)-(IV.13) is infeasible will be dealt with shortly. Intuitively, the controller first computes the nominal length of the planning period $l(|z|)$, and a target safety stock θ to be maintained upon completion of this planning period, as a function of the observed queue length vector. In general, $l_0 \ll l(|z|)$, unless $|z|$ is very small in which case l_0 provides a lower bound on the planning horizon which is in the appropriate time scale. Then the nominal time allocations for the ensuing planning period are computed using the linear program (IV.12)-(IV.13): the decision variable x_k represents the nominal amount of time that will be devoted to serving class k jobs over this planning period. The constraint $z + l\alpha - Rx \geq \theta$ implies that the target ending queue length vector will be above a specified threshold requirement, while $Cx \leq l\mathbf{1}$ states that the total time allocation for each of the servers cannot exceed its capacity. It is implicit in this formulation that the planning problem involves a deterministic "continuous variable" approximation.

- *The LP objective.* The linear programming form of this planning algorithm is motivated by the structure of the optimal fluid control policy of equation (III.29). The transformation from z to $\tilde{z}$ simply reduces the planning phase for each review period to a common *normalized* problem. Therefore, planning decisions are roughly taken according to the fluid control policy one started with, and safety stocks are enforced so that the corresponding processing plans will be implementable in the stochastic network.

- *Processing plan.* Given the vector of nominal time allocations x, a plan expressed in units of jobs of each class to be processed over the ensuing period, and a nominal idleness plan expressed in units of time for each server to remain idle over the same period are formed as follows:

$$p(k) = \left\lfloor \frac{x_k}{m_k} \right\rfloor \wedge q_k \quad \text{for } k = 1, \ldots, K, \quad \text{and} \quad u_i = l - (Cx)_i \quad \text{for } i = 1, \ldots, S. \tag{IV.14}$$

  The execution of these decisions is as follows. First, the plan p is implemented in open-loop fashion; that is, each server i processes sequentially $p(k)$ jobs for each class $k \in C_i$. The construction of the processing plan p using equation (IV.14) ensures that it will be implementable from jobs present at the beginning of this review period. Let d_i denote the time taken to complete processing of these jobs at server i and let $\delta_i = (l - d_i)^+$ be the nominal time remaining until the end of the ensuing period. In the second phase of execution, each server i will idle for $u_i \wedge \delta_i$ time units. The completion of both execution phases for all servers signals the beginning of the $(j+1)^{st}$ review period.

- *Infeasible planning logic.* An alternative logic is employed when (IV.12)-(IV.13) is infeasible in order to steer the state above the desired threshold levels and resume proper operation. To avoid unnecessary details, we will not give the precise description of this step here. Instead, we will work through the required actions for the Rybko-Stolyar network (still with the same parameters). Assume that the system is currently empty. We now need to accumulate the sufficient amount of safety stock in the four buffers which is given by the vector θ. Consider the following methodology: idle server 1 and 2 until we have $\theta_1 + \theta_2$ jobs into buffer 1 and $\theta_3 + \theta_4$ jobs in buffer 3; process θ_2 jobs in server 1 and θ_4 at server 2. How long do you think this will take? The claim is that when $|Z(0)| = n$ is large, this plan will take at most a few multiples of $|\theta|$. A similar -but slightly more complex- argument will do the trick for general multiclass networks that may have Markovian routing.

Hereafter, the notation $\mathbf{DR}(r^g, l, \beta)$ will denote the discrete-review policy derived from the functions $r^g(\cdot)$, $l(\cdot)$, and the vector β that satisfy (IV.8)-(IV.9) and (IV.10). In the sequel, we will use a subscript to differentiate between different review periods.

Markov state descriptor. For a multiclass network under any policy in the proposed family the underlying continuous time Markov chain is defined as follows. Assume that $t_j \leq t < t_{j+1}$ and define the parameter $N(t)$ to be equal to 1 if the linear program (IV.12)-(IV.13) is feasible or otherwise set it equal to the number of remaining executions of the processing plan $\hat{p}$ (the latter is relevant info for the infeasible planning logic). Let $p(t)$ be a K-vector, where $p_k(t)$ is the number of class k jobs that remain to be processed at time t according to the processing plan p_j or $\hat{p}_j$, depending on whether the planning LP was feasible. Let $u(t)$ be the S-vector of remaining idling times for each of the servers for the ensuing planning period. Finally, let $R_a(t)$ be the $|\mathcal{E}|$-vector and $R_s(t)$ be the K-vector associated with the residual arrival and service time information. The Markovian state descriptor will then be

$$X(t) = [Z(t);\ N(t);\ p(t);\ u(t);\ R_a(t);\ R_s(t);\ |Z(0)|], \tag{IV.15}$$

and $\mathbf{X}$ will represent the underlying state space. Imitating Dai's argument [Dai95] and using the strong Markov property for piecewise deterministic processes of Davis [Dav84], it is easy to show that the process $\{X(t),\ t \geq 0\}$ is a strong Markov process with state space $\mathbf{X}$. The associated norm will be

$$\|X(t)\| = |Z(t)| + N(t) + |p(t)| + |u(t)| + |R_s(t)| + |R_a(t)|.$$

Behavior of DR policies. We know review the basic properties for these policies.

- Enforcing the safety stocks renders the implementation of the processing plans to be trivial.

- The planning logic of (IV.12)-(IV.13) will mimic the allocations that one gets by the direct solution of the fluid optimal control problem. As the review horizons become longer the accuracy of the fluid approximation embodied in (IV.13) increases.

- Suppose that the WIP in the system is of $\mathcal{O}(n)$. There are three relevant time scales in the system. The macroscopic behavior of the system evolves in a time scale which is of order n. The relevant time scale for control actions is of order $l(n)$. Finally, the microscopic scale of individual events is of order of magnitude given by mean interarrival or service times. Conditions (IV.8)-(IV.9) ensure the separation of these three time scales. This is a important feature for DR policies. In detail, (IV.8) ensures that $l(n)$ is increasing with n. As a result as n becomes larger the accuracy of the execution of each processing plan will become more accurate, while relative to the macroscopic time scale that describes the overall system performance, it appears as if we are reviewing status at an ever increasing rate (this follows from (IV.9)). Therefore, the fluid scaled (or transient) performance of the system will approach the optimal transient response computed in the fluid model. Finally, the safety stock requirements are becoming negligible when compared to the queue lengths that are of order n.

- Scheduling complexity of DR policies during each review period is low. This is due to the fact that the execution of a discrete-review policy is insensitive to the precise processing sequence followed and thus, the overall complexity is that of a linear program of size equal to K, the number of classes in the network, which scales very gracefully with the size of the network. That is, the computational effort required in each planning phase is constant as a function of the review period length, the load in the network, and the amount of work to be scheduled. This is an important feature, for if the scheduling complexity had a superlinear growth rate as a function of $|z|$, then the associated computational delay would become significant relative to the time allocated to processing jobs, which would degrade performance and could affect the stability of the controlled network.

IV.4 A Basic Result from Large Deviations

Let $\{Y_i\}$ be a sequence of iid zero mean random variables. The SLLN says that

$$\frac{\sum_{i=1}^n Y_i}{n} \to 0 \quad a.s..$$

A more detailed argument can reveal how fast does the sample mean $\sum_i Y_i/n$ converges to 0 as a function of n. Using Markov's (or Chebyshev's) inequality we get that

$$\begin{aligned}\mathbf{P}\left(\sum_i Y_i \geq n\epsilon\right) &= \mathbf{P}\left(e^{\theta\sum_i Y_i} \geq e^{\theta n\epsilon}\right)\\ &\leq e^{-\theta n\epsilon}\mathbf{E}e^{\theta\sum_i Y_i}\\ &= e^{-\theta n\epsilon}\left(\mathbf{E}e^{\theta Y_1}\right)^n\\ &= \left(e^{-\theta\epsilon}\mathbf{E}e^{\theta Y_1}\right)^n\end{aligned}$$

If $\mathbf{E}e^{\theta Y_1} < \infty$ for some $\theta > 0$, then for every $\epsilon > 0$ we get a bound of the form

$$\mathbf{P}\left(\sum_i Y_i \geq n\epsilon\right) \leq e^{-h(\epsilon)n},$$

where $h(\epsilon) = \sup_\theta(\theta\epsilon - \log \mathbf{E}e^{\theta Y_1})$ is a convex function and $h(\epsilon) > 0$ ($h(\cdot)$ is called the rate function). (Similar results can be obtained for sample path properties when we analyze a sequence of processes rather than a sequence of random variables.)

In words the bound derived above says that the probability that the sample average of n iid random variables differs by more that ϵ from its true mean, decays exponentially in n and the rate in the exponent is given by $h(\epsilon)$.

Suppose that $\sum_i Y_i/n$ converges to α with rate function $h_y(\cdot)$ and $\sum_i X_i/n$ converges to μ with rate function $h_x(\cdot)$, then $\sum_i (X_i+Y_i)/n$ converges to $\mu+\alpha$ with rate function $h_{x+y}(\cdot) = \min\{h_x(\cdot), h_y(\cdot)\}$; that is, the rate of convergence is the minimum of the two.

Large deviations analysis of a DR policy. We want to bound the difference between the ending state after the execution of a processing plan and the nominal ending state computed using the planning LP. The ending state upon completion of the processing plan will be

$$z_{j+1} = z_j + E(t_{j+1} - t_j) - p_j + \sum_k \Phi^k(p_j(k)).$$

First, a bound is obtained for the duration of execution of the processing plan. Second, a bound is obtained for differences due to the external arrival process; this is proved using the previous result on the duration of execution of each plan and the large deviations fact we developed earlier for all sequences of service time random variables. Finally, we bound the differences due to the Bernoulli routing random variables. (The details will be filled in later... HW?) The end result is that

$$\mathbf{P}\left(|z_{j+1} - (z_j + \alpha l_j - Rx_j)| > l_j\epsilon\right) \leq e^{-f(\epsilon)l_j},$$

where $f(\cdot)$ is the appropriate rate function. That is, large exceedances relative to the length of the planning horizon occur with probabilities that decay exponentially in l. Moreover, it is easy to now argue that

$$\mathbf{P}\left(z_{j+1} \not\geq (1-\epsilon)\theta_{j+1}\right) \leq e^{-h(\epsilon)l_j},$$

Identical bounds can be derived for the bahavior of the policy under the infeasible logic with the exception that the time taken to complete the corresponding processing plan is longer, but bounded above by a small multiple of l with high probability.

Choice of $l(\cdot)$. We can now provide a justification for conditions (IV.8)-(IV.9). Suppose that the WIP in the system is of $\mathcal{O}(n)$. If $l \approx \frac{c}{f(\epsilon)}\log(n)$, then errors of order at least as large as ϵl occur with probability $1/n^c$; alternatively, we are tracking the desired trajectory computed through the planning LP to within ϵl with probability $1-1/n^c$. This will ensure that

- the planning LP for the next review period is feasible
- there is enough stock in hand for the next review period

and moreover, asymptotically as $n \to \infty$

- the infeasible planning logic is never invoked.

This will allow us to obtain the desired fluid limits. Here is why: as n gets large the infeasible planning logic is never used, so it suffices to analyze the behavior of the policy under the planning LP; for this we know that the errors in our execution will be of order $l(n)$; for the fluid scaled processes -or when we focus to macroscopic transient behavior- that implies that the DR policy mimics the optimal fluid model behavior to within errors of order $l(n)/n$ that become negligible as n grows. Hence, the desired result will follow. Finally, the penalty we are paying due to the safety stock requirement is of order $\log(n)/n$, which is again small.

Remark. The tool of large deviations has found extensive use over the recent past in many areas of management science and applied probability in general. The idea presented here, even though simple, it is quite powerful. It is worth investing some time getting familiar with its basic usage.

IV.5 Asymptotic Analysis of Discrete-Review Policies

The main theorem proved in this section is the following:

Theorem IV.5.1 *Consider a multiclass open queueing network under the discrete-review policy* $\mathbf{DR}(r^g, l, \beta)$. *For almost all sample paths* ω *and any sequence of initial conditions* $\{x_n\} \subset \mathbf{X}$ *such that* $|x_n| \to \infty$ *as* $n \to \infty$, *there is subsequence* $\{x_{n_j}(\omega)\}$ *with* $|x_{n_j}(\omega)| \to \infty$ *such that*

$$\bar{Z}^{n_j}(0,\omega) \to z(0,\omega) \tag{IV.16}$$

$$(\bar{Z}^{n_j}(\cdot,\omega), \bar{T}^{n_j}(\cdot,\omega)) \to (z(\cdot,\omega), \bar{T}(\cdot,\omega)) \ \ u.o.c., \tag{IV.17}$$

$$(\bar{N}^{n_j}(\cdot,\omega), \bar{p}^{n_j}(\cdot,\omega), \bar{u}^{n_j}(\cdot,\omega)) \to (0,0,0) \ \ u.o.c., \tag{IV.18}$$

and the cumulative allocation process can be expressed in the form

$$\bar{T}(t,\omega) = \int_0^t v(\tau,\omega)d\tau \quad \text{for } t \geq 0. \tag{IV.19}$$

The pair (z, v) *satisfies equations*

$$\dot{z}(t) = \alpha - Rv(t), \quad z(0) = q, \tag{IV.20}$$

$$z(t) \geq 0, \quad v(t) \geq 0, \quad Cv(t) \leq \mathbf{1}, \tag{IV.21}$$

and the policy specific equation

$$v(t) \ \in \ \underset{v \in \mathcal{V}(z(t))}{\text{argmax}} \ r^g(z(t))'v. \tag{IV.22}$$

Remark. In particular, it easily follows that the proposed policy is asymptotically optimal under fluid scaling. (The limit trajectory starting from any limiting initial condition is the one that satisfies the optimality conditions in the fluid model.)

- If $\rho < \mathbf{1}$, the policy $\mathbf{DR}(r^g, l, \beta)$ will also be stable. Why?
- Given a cost rate function $g(\cdot)$ and the solution r^g of the associated fluid optimal control problem, we now have a way to translate this fluid control policy in the stochastic network in a way that guarantees FSAO and stability.

- If we replace r^g by any reward rate function $r(\cdot)$, we would get that the fluid limits under $\mathbf{DR}(r,l,\beta)$ satisfy the conditions of this theorem with r^g replaced by r in (IV.22). Thus we can also translate the family of reward maximizing fluid control policies described in note III. For example, any priority rule implemented in this setting (for the approriate choice of r) will result in a stable scheduling policy.
- For example, in the Rybko-Stolyar network LBFS is unstable, but LBFS implemented in a DR structure (e.g., with $r = [1,5,1,5]$) will result in a stable policy. The DR structure acts as a stabilization mechanism that prevents f the system from entering undesirable stravation periods.

We now provide an overview for the proof of this theorem. We start by establishing that the probability that the infeasible will have to be employed is vanishingly small in the following sense. Given the sequence of review points $t_0, t_1, \dots$ and any time $t \geq 0$, let $j_{max} = \min\{j : t_j \geq nt\}$.

Proposition IV.5.1 *Define the sequence of events $\{A_n\}$, where $A_n = \{\omega : \exists j \leq j_{max},$ such that $z_j^n \not\geq (1-\epsilon)\theta_j^n\}$. Then for any $\epsilon > 0$, $\mathbf{P}\left(\limsup_n A_n\right) = 0$.*

Proof. Recall that $|x_n| = n$. Given the definition of l_0 and the growth condition (IV.8), for any $\epsilon > 0$ and any constant $\kappa > 0$, there exists an $N(\epsilon,\kappa) > 0$ such that for any $n \geq N(\epsilon,\kappa)$ we have that

$$l_0^n > \kappa \log(n).$$

Set $N(\epsilon,\kappa) = \max(N(\epsilon,\kappa), N_1, N_2)$. Using the bounds mentioned in the previous section one gets that

$$\begin{aligned}
\mathbf{P}(A_n) &= \mathbf{P}\left(z_1^n \not\geq (1-\epsilon)\theta_1^n\right) + \\
&\quad + \sum_{j=1}^{j_{max}} \mathbf{P}\left(z_{j+1}^n \not\geq (1-\epsilon)\theta_{j+1}^n,\ z_i^n \geq (1-\epsilon)\theta_i^n,\ i \leq j\right) \\
&\leq e^{-\kappa\log(n)} + \\
&\quad + \sum_{j=1}^{j_{max}} \mathbf{P}\left(z_{j+1}^n \not\geq (1-\epsilon)\theta_{j+1}^n \mid z_i^n \geq (1-\epsilon)\theta_i^n,\ i \leq j\right) \times \\
&\qquad\qquad \mathbf{P}\left(z_i^n \geq (1-\epsilon)\theta_i^n,\ i \leq j\right) \\
&\leq \frac{1}{n^\kappa} + \sum_{j=1}^{j_{max}} \mathbf{P}\left(z_{j+1}^n \not\geq (1-\epsilon)\theta_{j+1}^n \mid z_i^n \geq (1-\epsilon)\theta_i^n,\ i \leq j\right) \\
&\leq \frac{j_{max}}{n^\kappa}.
\end{aligned}$$

For $\kappa \geq 3$,

$$\begin{aligned}
\sum_n \mathbf{P}(A_n) = &\leq \sum_{n \leq N(\epsilon,\kappa)} \mathbf{P}(A_n) + \sum_{n > N(\epsilon,\kappa)} \frac{j_{max}}{n^\kappa} \\
&\leq N(\epsilon,\kappa) + c \sum_{n > N(\epsilon,\kappa)} \frac{1}{n^{\kappa-1}} \\
&< \infty.
\end{aligned}$$

The desired result follows from the Borel-Cantelli Lemma. □

Theefore it suffices to study system behavior for planning decisions obtained through the solution of (IV.12)-(IV.13). For any sample path ω, let $\bar{X}^n(t,\omega)$ be the (scaled) nominal allocation process, which is equal to the sum of all planned allocation times over all review periods up to time t, along the sample path ω. Similarly, let $\bar{T}^n(t,\omega)$ be the (scaled) actual allocation process, which is equal to the sum of all actual time allocations observed during execution of the processing plans for all review periods up to time t. The second part of the proof bounds the difference between $\bar{X}^n(\cdot)$ and $\bar{T}^n(\cdot)$ as a consequence of the FSLLN.

Proposition IV.5.2 *$|\bar{T}^n(t) - \bar{X}^n(t)| \to 0$ a.s., as $n \to \infty$.*

Therefore, to obtain the fluid limit under a DR policy we just need to study the limits under the nominal allocations computed from the planning LP in (IV.12)-(IV.13). Once again, fix again time at some $t \geq 0$. Choose j such that $t_j \leq nt < t_{j+1}$ and let x_j^n denote the nominal allocation over the j^{th} planning period, which is of length $l^n(|z_j^n|)$. It is easy to see that $\bar{l}_j^n = l_j^n/n \to 0$ as $n \to \infty$. Let $\bar{x}^n(t,\omega) = x_j^n/l_j^n$ and observe that

$$\bar{X}^n(t,\omega) = \sum_j \bar{x}^n(t,\omega)\bar{l}_j^n \to \int_0^t v(\tau,\omega)d\tau;$$

that is, as $n \to \infty$ the Riemann sum converges to this definite integral for some $v(\cdot,\omega)$ not yet specified. From Proposition IV.5.2 and the basic theorems regarding fluid limit convergence, it follows that $\bar{X}^n(t,\omega)$ converges to some limit, $\bar{X}(t,\omega)$, which is absolutely continuous; this is also the limit of $\bar{T}^n(t,\omega)$. It follows that

$$v(t,\omega) = \frac{d\bar{X}(t,\omega)}{dt}$$

almost everywhere on the real line and thus, it is sufficient to study the limit of $\bar{x}^n(t,\omega)$ along the sequence $\{y^n\}$ in order to establish the fluid limit of the nominal allocation process.

Proposition IV.5.3 *For a.e. ω, $\bar{x}^n(t,\omega) \to v_\Psi(q(t,\omega))$*

To prove this proposition one first expresses the optimality conditions (including the dual of the planning LP and the corresponding complementarity conditions) that the nominal allocation should satisfy. These conditions are linear constraints on several variables that can be treated as, for example, the non-idling conditions that we have seen eralier. The limiting conditions are the optimality conditions of the LP in (III.29), and hence we are done.

The desired theorem now easily follows by combining these results with the classical arguments that guarantee existense of fluid limits studied earlier.

IV.6 Continuous-Review Policy for Rybko-Stolyar Network

Returning to the Rybko-Stolyar example, a fluid-scale asymptotically optimal discrete-review policy can now be defined using the optimal control description given earlier. The choices of threshold values

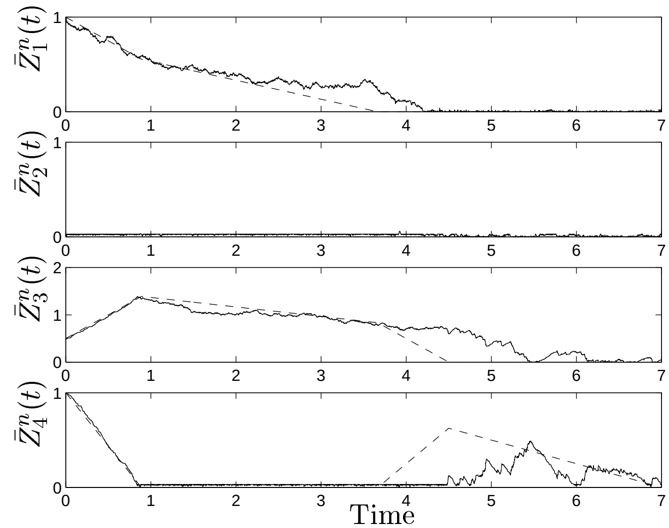

Figure 2: Trajectory tracking: optimal fluid trajectories (vs) state trajectories under $\mathbf{CR}(r^g,\theta)$: $Z(0) = 200[1,0,0.5,1]$, $(n = 200)$

and planning horizon lengths can be defined from any vector β and function $l(\cdot)$ that satisfy the appropriate conditions. These choices can be fine-tuned using a subsequent simulation study.

This policy can be further simplified by exploiting the structure of the Rybko-Stolyar network in order to form a threshold -or continuous-review- policy that achieves the same asymptotic performance. Denote by $\theta(\cdot) = \beta l(\cdot)$, the function that dynamically computes the desired safety stock requirements. The desired threshold policy will be one that gives priority to the exiting class at each server unless, the exit class at the other server is below the associated threshold requirement, in which case the incoming class gets higher priority. This is LBFS with priority reversal below the desired threshold and is denoted by $\mathbf{CR}(\Psi^g,\theta)$. Figure 2 depicting simulated trajectories for the continuous-review implementation for the case $n = 200$, illustrates the trajectory tracking of these policies; the optimal fluid trajectories overlaid for comparison.

This simple characterization of this CR policy hinges on the special structure of this network and the corresponding fluid optimal policy. In general, one cannot avoid the LP characterization of the nominal allocation at any time t. The following CR policy will have the desired asymptotic properties. Choose instantaneous server activities by solving a different linear program given below

$$v(t) \in \operatorname{argmax} \quad r(\tilde{Z}(t))'v \tag{IV.23}$$

$$\text{subject to} \quad Z(t) + (\alpha - Rv)l(t) \geq \theta(t), \quad v \geq 0, \quad Cv \leq \mathbf{1}. \tag{IV.24}$$

- This family of CR policies achieves exactly the same properties as their DR counterparts.
- The large deviations analysis described earlier needs to be modified, since we now need process convergence results. Eventually, through somewhat harder work, one gets stronger results that say that the network processes lie within a tube of diameter $l(n)$ around the desired fluid limiting trajectories.

- DR policies seem more natural to implement in most prctical scenarios. However, CR policies have improved performence in comparison to DR, and in some cases they can still provide reasonable models for practical systems of capacity or bandwidth allocation among many "small" users.

IV.7 Extensions

We list several extensions to the network models under investigation that can be readily incorporated within the proposed framework with only minor modifications both in the conceptual and implementation levels. This is possible because both fluid approximations and the proposed family of control policies are largely insensitive to many subtle modeling details that lie below their level of resolution. In contrast, within mainstream queueing network analysis, many of these extensions would normally venture into radically different domains of application and research, and would not be able to be treated in a unified framework.

- *Admission and routing control capability.* Routing control was added in the last HW. Admission control can be done in a similar manner. The fluid control problem is still one with deterministic and continuous dynamics with polytopic constraints on the instantaneous control. The DR policy can easily incorporate these changes in the definition of the processing plan. Now we also decide on routing decisions and on how many jobsto be admitted over each period.

- *Switchover times (and/or costs).* A practical feature of stochastic processing networks that is commonly omitted from mathematical models of these systems is that of setup delays/costs (or switchover delays) that are incurred when a server switches between processing different classes of jobs. There is an extensive literature for these problems mostly under the rubric of polling systems that focuses on single server systems and often restricts attention to simple classes of policies, most often some variant of *round-robin* or *serve-to-exhaustion* policies.

  In the context of discrete-review policies setup delays can be introduced at no extra cost in complexity or performance. This follows by exploiting the fact that planning horizons vary in a longer time-scale than that of setup delays incurred in switching between classes and thus, the cumulative time spent in setups will be small compared to the cumulative time spent in the actual processing of jobs.

  Specifically, let $d_{k,l}$ be the setup time required for server i to switch from processing class k jobs to processing class l jobs, where $k, l \in C_i$, and set $d_i = \max_{k,l \in C_i} d_{k,l}$. Then the maximum time spent by server i in setups within every review period is bounded above by $|C_i| d_i$, since only one setup need to be done per job class in executing the open-loop processing plans, which is constant as a function of $|Z(t)|$. As $|Z(t)|$ increases, the cumulative time spent in setups becomes negligible in comparison to the cumulative time spent in processing jobs at each station, and asymptotically under fluid scaling it vanishes. Using this last observation it should not be surprising that all the analysis and results provided so far will extend to the case of multiclass networks with setups.

- *Multiserver stations.* Such networks arise often within the context of computers systems, manufacturing systems, call centers and other applications in service operations.

  Assume that each server at a given station is capable of processing the same set of job classes and that the service rate of the j^{th} server at station $s(k)$ in processing class k jobs ($k \in C_i$) is $a_j^{s(k)} \tilde{\mu_k}$, where $\tilde{\mu_k}$ is a normalized service rate for class k jobs and the constants $a_j^{s(k)}$ describe the relative processing efficiencies of the various servers at station $s(k)$. Under these assumptions, each station may be comprised of servers that are allowed to have different processing capabilities and different distributional characteristics, but their average service rates cannot differ by more than just a constant of proportionality.

  Setting $\mu_k = \sum_j a_j^{s(k)} \tilde{\mu_k}$, all of the results derived so far immediately port to the case of multi-server stations. Once again, the main step required in this extension was to replace each station by a single-server that had the same average processing capability, which is then used in the definition of the fluid approximation and of the associated discrete-review policy.

- *Batch processing stations.* These are servers capable of handling more than one job simultaneously, which are common in some manufacturing systems such as semiconductor wafer fabs (well-drive furnaces for example).

  Each station is allowed to process jobs in batches and the maximum batch size at station i will be denoted by c_i. Let $\Gamma = \mathbf{diag}\{c_{s(1)}, \ldots, c_{s(k)}\}$ and $\hat{R} = (I - P')M^{-1}\Gamma$. The fluid model description is given by (IV.20)-(IV.21), where $\hat{R}$ has been substituted for R. Essentially, one has replaced the batch processing server by a faster server that processes one job at a time. In order for this substitution to be justified, one has to ensure that the station always serves jobs in full batches. This will be ensured within a discrete-review framework if at every review period the safety stock level is set to be $\theta = l\Gamma\beta$. That is, we enforce enough safety stock for one periods worth of work expressed in full batches.

IV.8 Notes and References

The focus on fluid approximations is primarily motivated by recent developments in the area of stability analysis of stochastic networks via fluid model analysis. The important breakthrough in this area was the theory developed by Dai [Dai95]; see also Chen and Mandelbaum [CM91], Rybko and Stolyar [RS92], Dai and Meyn [DM95], Dai and Weiss [DW96], Chen [Che95], Stolyar [Sto95], and Bramson [Bra98] for further discussions, refinements, and improvements. Simultaneously, there has been a growing interest in using fluid models in a synthesis framework as the one described here. Examples can be found in Chen and Yao [CY93], Atkins and Chen [AC95], Avram, Bertsimas and Ricard [ABR95], and Eng, Humphrey and Meyn [EHM96], where heuristic translation mechanisms based on fluid model optimization are described; in fact, related work can be traced back to Newell [New71]. The papers closest to the spirit of htis note are Meyn [Mey97b] and Chen and Meyn [CM98], where the authors try to solve directly the stochastic control problem (for the case of Poisson arrivals and exponential service times) by first approximating the value function with the one derived from the solution of the fluid optimization problem, and then using value or policy iteration. In this context, Meyn [Mey97b] first proposed the fluid-scale asymptotic optimality criterion. Related work can also be found in Bertsimas and Sethuraman [BS97].

Apart from the connections related to stability analysis, the use of fluid models is also motivated from the extensive literature on optimal control for deterministic systems with continuous dynamics (see, for example, the books Athans and Falb [AF66], Bryson and Ho [BH75], and Bertsekas [Ber95]), and from the vast computational simplifications they offer that has been exploited in developing efficient optimization algorithms (see Pullan [Pul93, Pul95, Pul96], Weiss [Wei95, Wei97], Luo and Bertsimas [LB96], and Maglaras [Mag97]). Recently, Betrsimas and Gamarnik developed very efficient algorithms for fluid optimal control problems for tandem networks. They first showed that for this class of networks there at most $2K$ breakpoints in the optimal trajectories, and then used this fact to propose a polynomial time algorithm for this class of networks. On a separate direction, Maglaras considered fluid optimal control problems with linear cost structures and characterized the class of problems that admit greedy optimal solutions in the form of a semidefinite program; this is equivalent to a closed-form solution of the fluid optimization problem. Efficient solution of fluid optimal control problems is an active topic of research.

Apart from these simplifications, there are positive results that suggest that the solution of the fluid optimal control problem retains significant information about the original stochastic network control problem. For example, policies such as the $c\mu$ rule and the *generalized* $c\mu$-rule, have been shown to be optimal for both the underlying stochastic networks and their associated fluid models; see Chen and Yao [CY93], Bertsimas, Paschalidis and Tsitsiklis [BPT95] and Haji and Newell [HN71], and Van Mieghem [VM95] respectively. In a separate direction, Meyn [Mey97a] showed that the optimal behavior in the stochastic network starting from a large initial condition and over a proportionally long horizon approaches the optimal behavior in the fluid model. In the spirit of positive results like these, the premise of the fluid model approach to network control problems is that although we are using a "weak" (FSAO) criterion for what constitutes a "good" control policy, this relaxed notion of optimality will guide us in designing "near-optimal" policies for the original stochastic network control problems.

The main issue that arises as part of this policy design framework is that of translating the fluid control policy in the stochastic network. A few simple examples have been analyzed in the papers cited above, but no general mechanism has been constructed that guarantees fluid-scale asymptotic optimality or even stability for the policy extracted from the fluid solution. The same problem of back-translation -only in more acute form- has been observed in the context of the heavy-traffic approach to network control problems; see Harrison [Har88, Har96a], Harrison and Wein [HW89, HW90], Williams [Wil96], Kushner and Martins [KM96]. (There, one would follow exactly the same synthesis procedure with the only difference that fluid models are now replaced by the Brownian approximating models and fluid scaling by diffusion -or Central Limit Theorem type of- scaling. More on this later.)

The first such general translation mechanism was proposed by Harrison [Har96a] in his BIGSTEP approach to dynamic control for stochastic networks; this was done in the context of Brownian approximations and heavy-traffic limits. Harrison [Har96b] rigorously proved that BIGSTEP is asymptotically optimal (in the heavy-traffic sense) for a simple two station network. The family of discrete-review policies we describe here is the first general translation mechanism that has been rigorously analyzed for a wide class of networks. It is an extension -or generalization- of BIGSTEP that hinges on the discrete-review structure proposed by Harrison. The results we present are from [Mag99b, Mag99a].

Discrete-review policies, and specifically, policies that step through time in large intervals within which a deterministic planning logic is employed, have been proposed by other researchers in the areas of applied probability and network control. Some examples that are closer to our work can be found in Bertsimas and Van Ryzin [BVR91], Bambos and Walrand [BW93], Tassiulas and Papavassiliou [TP95], and Gans and Van Ryzin [GVR97], but other related papers can be found as well.

Continuous review policies (or threshold policies like the ones considered in section IV.6 have also appeared as heuristic translation mechanisms in the literature; see Harrison and Wein [HW89] and Harrison [Har96b]. The general family of continuous-review policies we proposed at the end of section IV.6 is from [Mag98].

Some comments on the extensions mentioned in section IV.7. Systems with switchover delays (or setups) have been studied extensively in the literature mostly under the rubric of polling systems that focuses on single server systems and often restricts attention to simple classes of policies, most often some variant of *round-robin* or *serve-to-exhaustion* policies; see for example [Tak90, FK94, JPR98]. Systems with more general processing characteristics in the context of our work have been described by Harrison in [Har96a]. In particular, a simple parallel server network where the processing characteristics at each station are allowed to vary more substantially is studied in [HMJ98]. Finally, there is a significant literature on queueing networks with batch servers, but most of it is concentrated on simpler network models and results either of product form nature or of combinatoric optimization [Neu67, DS73, Med75, GW91, AADT92, Ser93, MT97]. Batch processing networks within the framework of fluid model analysis and DR policies have been studied in [MK97].

References

[AADT92] J. H. Ahmadi, R. H. Ahmadi, S. Dasu, and C. S. Tang. Batching and scheduling jobs on batch and discrete processors. *Oper. Res.*, 39(4):750–763, July 1992.

[ABR95] F. Avram, D. Bertsimas, and M. Ricard. Fluid models of sequencing problems in open queueing networks; an optimal control approach. In F. Kelly and R. Williams, editors, *Stochastic Networks*, volume 71 of *Proceedings of the IMA*, pages 199–234. Springer-Verlag, New York, 1995.

[AC95] D. Atkins and H. Chen. Performance evaluation of scheduling control of queueing networks: fluid model heuristics. *Queueing Systems*, 21:391–413, 1995.

[AF66] M. Athans and P. Falb. *Optimal Control*. McGraw-Hill, 1966.

[Ber95] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol.1*. Athena Scientific, Belmont, Massachusetts, 1995.

[BH75] A. E. Bryson and Y. C. Ho. *Applied Optimal Control*. Hemisphere Publishing, New York, 1975.

[BPT95] D. Bertsimas, I. Ch. Paschalidis, and J. N. Tsitsiklis. Branching bandits and Klimov's problem: Achievable region and side constraints. *IEEE Trans. Aut. Control*, 40(12):2063–2075, December 1995.

[Bra98] M. Bramson. Stability of two families of queueing networks and a discussion of fluid limits. *Queueing Systems*, 28:7–31, 1998.

[BS97] D. Bertsimas and J. Sethuraman. Optimal control of queueing networks: A singular pertubation approach. In *9th INFORMS Applied Probability Conference*, July 1997.

[BVR91] D. Bertsimas and G. Van Ryzin. A stochastic and dynamic vehicle routing problem in the euclidean plane. *Oper. Res.*, 39(4):601–615, July 1991.

[BW93] N. Bambos and J. Walrand. Scheduling and stability aspects of a general class of parallel processing systems. *Adv. Appl. Prob.*, 25:176–202, 1993.

[Che95] H. Chen. Fluid approximations and stability of multiclass queueing networks: work-conserving policies. *Ann. Appl. Prob.*, 5:637–655, 1995.

[CM91] H. Chen and A. Mandelbaum. Discrete flow networks: Bottleneck analysis and fluid approximations. *Math. Oper. Res.*, 16(2):408–446, 1991.

[CM98] R-R. Chen and S. P. Meyn. Value iteration and optimization of multiclass queueing networks. *Queueing Systems*, 1998.

[CY93] H. Chen and D. Yao. Dynamic scheduling of a multiclass fluid network. *Oper. Res.*, 41(6):1104–1115, December 1993.

[Dai95] J. G. Dai. On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *Ann. Appl. Prob.*, 5:49–77, 1995.

[Dav84] M. H. A. Davis. Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *J. R. Statist. Soc. Ser. B*, 46(3):353–388, 1984.

[DM95] J. G. Dai and S. Meyn. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Trans. Aut. Control*, 40(11):1889–1904, November 1995.

[DS73] R. K. Deb and R. F. Serfozo. Optimal control of batch service queues. *Ann. Appl. Prob.*, 5:340–361, 1973.

[DW96] J. G. Dai and G. Weiss. Stability and instability of fluid models for certain re-entrant lines. *Math. Oper. Res.*, 21:115–134, 1996.

[EHM96] D. Eng, J. Humphrey, and S. P. Meyn. Fluid network models: Linear programs for control and performance bounds. In *13th World Congress of International Federation of Automatic Control, San Francisco*, 1996.

[FK94] A. Federgruen and Z. Katalan. Approximating queue size and workload distributions in general polling systems. *Queueing Systems*, 18:353–386, 1994.

[GVR97] N. Gans and G. Van Ryzin. Optimal control of a multiclass, flexible queueing system. *Oper. Res.*, 45(5):677–693, September 1997.

[GW91] C. R. Glassey and W. W. Weng. Dynamic batching heuristics for simultaneous processing. *IEEE Trans. Semiconductor Manufacturing*, 4:77–82, 1991.

[Har88] J. M. Harrison. Brownian models of queueing networks with heterogeneous customer populations. In W. Fleming and P. L. Lions, editors, *Stochastic Differential Systems, Stochastic Control Theory and Applications*, volume 10 of *Proceedings of the IMA*, pages 147–186. Springer-Verlag, New York, 1988.

[Har96a] J. M. Harrison. The BIGSTEP approach to flow management in stochastic processing networks. In F. Kelly, S. Zachary, and I. Ziedins, editors, *Stochastic Networks: Theory and Applications*, pages 57–90. Oxford University Press, 1996.

[Har96b] J. M. Harrison. Heavy traffic analysis of a system with parallel servers: Asymptotic optimality of discrete-review policies. *Ann. Appl. Prob.*, 1996. To appear.

[HMJ98] J. M. Harrison and Lopez M. J. Heavy traffic resourse pooling in parallel-server systems. 1998. Preprint.

[HN71] R. Haji and G. Newell. Optimal strategies for priority queues with nonlinear costs of delay. *SIAM J. Appl. Math.*, 20:224–240, 1971.

[HW89] J. M. Harrison and L. M. Wein. Scheduling network of queues: Heavy traffic analysis of a simple open network. *Queueing Systems*, 5:265–280, 1989.

[HW90] J. M. Harrison and L. M. Wein. Scheduling networks of queues: Heavy traffic analysis of a two-station closed network. *Oper. Res.*, 38(6):1052–1064, December 1990.

[JPR98] E. G. Coffman Jr., A. A. Puhalskii, and M. I. Reiman. Polling systems in heavy traffic: A Bessel process limit. *Math. Oper. Res.*, 23(2):257–304, 1998.

[KM96] H. J. Kushner and L. F. Martins. Heavy traffic analysis of a controlled multiclass queueing network via weak convergence methods. *SIAM J. on Control and Optimization*, 34(5):1781–1797, September 1996.

[LB96] X. Luo and D. Bertsimas. A new algorithm for state constrained separated linear programs. Technical report, MIT, 1996.

[Mag97] C. Maglaras. A methodology for dynamic control policy design for stochastic processing networks via fluid models. In *Proc. IEEE Conf. on Decision and Control*, 1997.

[Mag98] C. Maglaras. A class of continuous-review policies for scheduling in stochastic processing networks. *Queueing Systems*, 1998. To be submitted.

[Mag99a] C. Maglaras. Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality. *Ann. Appl. Prob.*, 1999. Submitted.

[Mag99b] C. Maglaras. Dynamic scheduling in multiclass queueing networks: Stability under discrete-review policies. *Queueing Systems*, 1999. Forthcoming.

[Med75] J. Medhi. Waiting time in a Poisson queue with a general bulk service rule. *Management Science*, 21:777–782, 1975.

[Mey97a] S. P. Meyn. The policy improvement algorithm for Markov decision processes with general state space. *IEEE Trans. Aut. Control*, 42(12):1663–1680, 1997.

[Mey97b] S. P. Meyn. Stability and optimization of queueing networks and their fluid models. In G. G. Yin and Q. Zhang, editors, *Mathematics of Stochastic Manufacturing Systems*, volume 33 of *Lectures in Applied Mathematics*, pages 175–200. American Mathematical Society, 1997.

[MK97] C. Maglaras and S. Kumar. Capacity realization in stochastic batch-processing networks using discrete-review policies. *Manufacturing & Service Operations Management*, 1997. Submitted.

[MT97] M. Miyazawa and P. G. Taylor. A geometric product-form distribution for a queueing network with non standard batch arrivals and batch transfers. *Adv. Appl. Prob.*, 1997. To appear.

[Neu67] M. F. Neuts. A general class of bulk queues with Poisson input. *Ann. Math. Stat.*, 38:369–373, 1967.

[New71] G. F. Newell. *Applications of Queueing Theory.* Chapman and Hall, 1971.

[Pul93] M. C. Pullan. An algorithm for a class of continuous linear programs. *SIAM J. on Control and Optimization*, 31(6):1558–1577, November 1993.

[Pul95] M. C. Pullan. Form of optimal solutions for separated continuous linear programs. *SIAM J. on Control and Optimization*, 33(6):1952–1977, November 1995.

[Pul96] M. C. Pullan. A duality theory for separated continuous linear programs. *SIAM J. on Control and Optimization*, 34(3):931–965, May 1996.

[RS92] A. N. Rybko and A. L. Stolyar. Ergodicity of stochastic processes describing the operations of open queueing networks. *Problems of Information Transmission*, 28:199–220, 1992.

[Ser93] R. F. Serfozo. Queueing networks with dependent nodes and concurrent movements. *Queueing Systems*, 13:143–182, 1993.

[Sto95] A. L. Stolyar. On the stability of multiclass queueing networks: A relaxed sufficient condition via limiting fluid processes. *Markov Processes and Related Fields*, 1:491–512, 1995.

[Tak90] H. Takagi. Queueing analysis of polling models: an update. In H. Takagi, editor, *Stochastic Analysis of Computer and Communication Systems.* North Holland, Amsterdam, 1990.

[TP95] L. Tassiulas and S. Papavassiliou. Optimal anticipative scheduling with asynchronous transmission opprtunities. *IEEE Trans. Aut. Control*, 40(12):2052–2062, December 1995.

[VM95] J. A. Van Mieghem. Dynamic scheduling with convex delay costs: The generalized $c\mu$ rule. *Ann. Appl. Prob.*, 5:809–833, 1995.

[Wei95] G. Weiss. On optimal draining of re-entrant fluid lines. In F. Kelly and R. Williams, editors, *Stochastic Networks*, volume 71, pages 91–103. Proceedings of the IMA, 1995.

[Wei97] G. Weiss. Algorithm for minimum wait draining of two station fluid re-entrant line. Technical report, Department of Statistics, The University of Haifa, 1997.

[Wil96] R. J. Williams. On the approximation of queueing networks in heavy traffic. In F. Kelly, S. Zachary, and I. Ziedins, editors, *Stochastic Networks: Theory and Applications*, pages 35–56. Oxford University Press, 1996.