Dynamic Scheduling in Heavy Traffic

Eurandom YEQT Workshop
Eindhoven, The Netherlands

November 4-5, 2013

Michael Harrison
Graduate School of Business
Stanford University

Abstract

This tutorial will focus on single-hop stochastic processing networks. That is, in the network models to be considered, arriving jobs of various types each require a single service before departing, but that service may be obtainable from several different servers, or may require capacity allocations from several servers simultaneously. In the heavy traffic parameter regime, the dynamic scheduling problem for the processing network is formally approximated by a corresponding Brownian control problem (BCP). In all cases the approximating BCP is more tractable than the conventional scheduling problem that it replaces. In particular, the approximating BCP may have a smaller effective dimension than the original problem, and if its effective dimension is one, then the approximating BCP can be solved explicitly. Many open problems remain, especially concerning the translation of Brownian solutions back into the conventional model context.

Key References

Harrison, J. M., and Lopez, M. J. (1999). Heavy Traffic Resource Pooling in Parallel-Server Systems, *Queueing Systems*, **33** 339-368.

Massoulié, L. and Roberts, J. (2000). Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, **15** 185–201.

Stolyar, A. L. (2004). MaxWeight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Ann. Appl. Probab.*, **14** 1–53.

Mandelbaum, A., and Stolyar, A. L. (2004). Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Operations Research*. **52** 836–855.

Kang, W. N., Kelly, F. P., Lee, N. H. and Williams, R. J. (2009). State space collapse and diffusion approximation for a network operating under a fair bandwidth sharing policy. *Ann. Appl. Probab*. **19** 1719–1780.

Outline

→ 1. Stochastic Processing Networks

2. Multi-Mode Resource Sharing Models

    2.1 General Formulation
    2.2 Reduction to a Standard Model
    2.3 Heavy Traffic Parameter Regime
    2.4 Approximating Brownian Control Problem

3. One-Dimensional Brownian Control Problems

    3.1 Pathwise Solution
    3.2 Achieving the Solution via MaxWeight
    3.3 A Complex Parallel-Server Example
    3.4 Asymptotic Optimality with Linear Costs

4. Achieving HGI Performance in Bandwidth Sharing

    4.1 Local Traffic Assumption
    4.2 The BCP with all Resources Critical
    4.3 HGI Performance
    4.4 Open Problem

The Processing Network Paradigm

A processing network model involves the following primitive elements. (A processing network might be called by a more specific name, like “transport system” or “service supply chain” if the speaker wants to emphasize a particular application domain.)

- *m* materials (jobs, customers, applications, requests, etc.)
- *n* activities (elemental units of technology)
- *p* resources (endowed with capacities, not consumed by activities)

Resources engage in activities. Activities create, destroy or transform materials.

References

T. C. Koopmans (1951), *Activity analysis of production and allocation*, Wiley, New York.

L. V. Kantorovich, *Mathematical methods in the organization and planning of production* (1960), *Management Science*, 6, 366–422.

G. B. Dantzig (1963), *Linear programming and extensions*, Princeton U. Press.

Stochastic Processing Networks

Queueing and inventory models, the central focus of stochastic OR, are suggestive of something quite fundamental, but the standard textbook treatment of those subjects is narrow in its applicability. Can one place conventional stochastic models in a broader framework that has greater conceptual scope? Yes. Generalize the framework proposed by Koopmans *et al.* to allow the amount of input flow or output flow produced by an activity, and the magnitude of exogenous input flows and output flows, to be stochastic.

General References

Harrison, J. M. (2000). Brownian models of open processing networks: Canonical representation of workload. *Ann. Appl. Probab.*, **10** 75-103. Correction **13** (2003) 390-393.

Harrison, J. M. (2002). Stochastic Networks and Activity Analysis, in Yu. Suhov (ed.), *Analytic Methods in Applied Probability. In Memory of Fridrih Karpelevich.* American Mathematical Society, Providence, RI, 53-76.

http://www.math.ucsd.edu/~williams/talks/belz/belz1.pdf

Outline

1. Stochastic Processing Networks

→ 2. Multi-Mode Resource Sharing Models

2.1 Examples and General Formulation
2.2 Reduction to a Standard Model
2.3 Heavy Traffic Parameter Regime
2.4 Approximating Brownian Control Problem

3. One-Dimensional Brownian Control Problems

3.1 Pathwise Solution
3.2 Achieving the Solution via MaxWeight
3.3 A Complex Parallel-Server Example
3.4 Asymptotic Optimality with Linear Costs

4. Achieving HGI Performance in Bandwidth Sharing

4.1 Local Traffic Assumption
4.2 The BCP with all Resources Critical
4.3 HGI Performance
4.4 Open Problem

A Multi-Mode Resource Sharing Model
(Bandwidth Sharing Model with Multi-Path Routing)

- Jobs of types 1, …, m arrive via independent renewal processes at rates $\lambda_1, \ldots, \lambda_m$.

- The processing system is composed of resources (or servers) numbered 1, …, p. The capacity of each resource is 1 by convention.

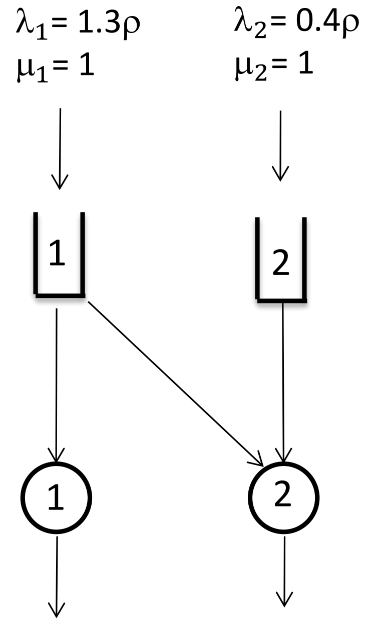
- Processing activities are indexed by $j = 1, \ldots, n$. We assume that each activity serves just one job type. By definition, one unit of activity j provides one unit of *service* to the associated job type. Define $M_{ij} = 1$ if activity j serves type i, and $M_{ij} = 0$ otherwise.

- Each type i job has a *size* that is drawn from a type-specific distribution with mean μ_i^{-1}. When the cumulative amount of service provided to a job equals its size, the job departs.

- An activity may, in general, consume the capacity of more than one resource. We denote by A_{kj} the rate at which activity j consumes the capacity of resource k.

- Thus the n-vector x of activity levels (or activity rates) that is chosen at any given time must satisfy $Ax \leq e$, where A is the $p \times n$ matrix (A_{kj}) and e is the p-vector of ones.

A Simple Parallel-Server Model

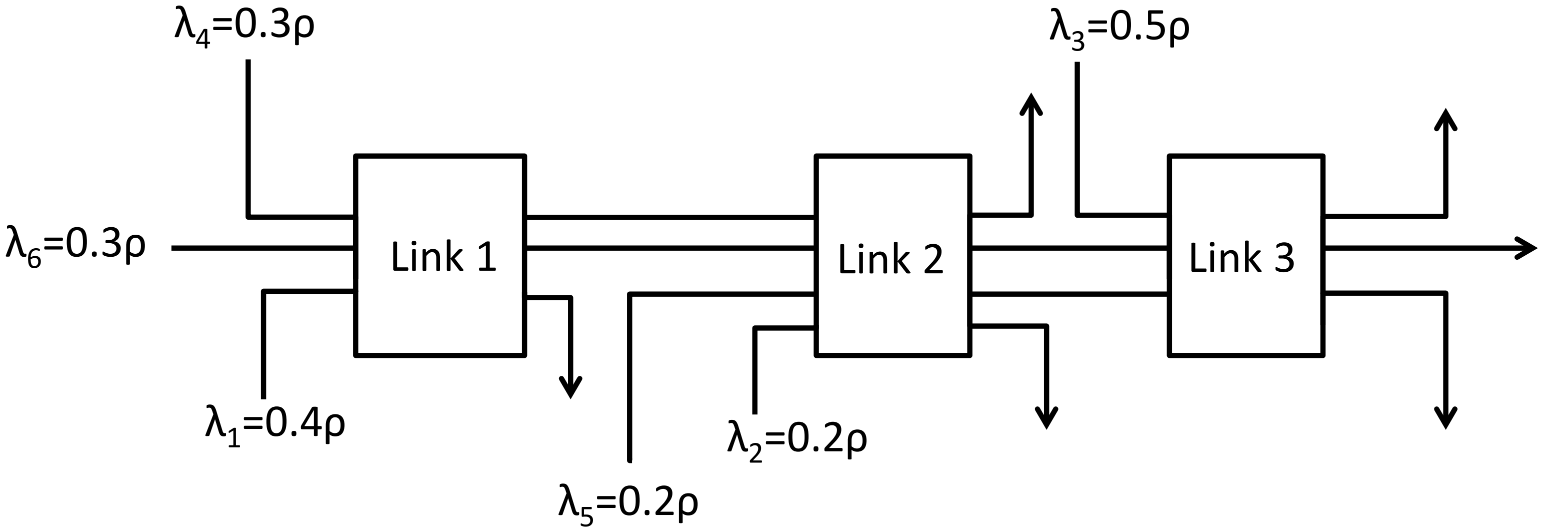


$$M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

A Bandwidth Sharing Model

Mean file size is 1 for each job type



At ρ=1, all three resources (links) are critically loaded

Mathematical Formulation

Taken as primitive are a non-decreasing *normalized input process* $E_i = \{E_i(t), t \geq 0\}$ and a non-decreasing *output processes* $F_i = \{F_i(t), t \geq 0\}$ for each job type $i = 1, \ldots, m$. These are renewal processes with $\mathbf{E}(E_i(t)) \sim t$ and $\mathbf{E}(F_i(t)) \sim \mu_i t$ as $t \to \infty$.

Also given are a strictly positive m-vector $\lambda = (\lambda_i)$ of *average arrival rates*, an $m \times n$ matrix $M = (M_{ij})$ having a single 1 in each column and the rest zeros, and a non-negative $p \times m$ matrix $A = (A_{kj})$ of *capacity consumption rates*.

A *control policy* is an adapted, non-negative, n-dimensional process $x = \{x(t), t \geq 0\}$ whose components $x_j(t)$ specify activity levels as functions of time.

$S_i(t) = \sum_{j=1}^{n} M_{ij} \int_0^t x_j(u)\, du$ for $i = 1, \ldots, m$ and $t \geq 0$ (cumulative service levels)

$J_i(t) = E_i(\lambda_i t) - F_i(S_i(t))$ for $i = 1, \ldots, m$ and $t \geq 0$ (job count process)

$Ax(t) \leq e$ (the p-vector of ones) $t \geq 0$ (capacity constraints)

$J(t) \geq 0$ $t \geq 0$ (state space constraint)

Cost Structure

We are also given a strictly convex holding cost function h: $\mathbf{R}_{+}^{m} \rightarrow \mathbf{R}$ of the form

$$h(z) = \sum_{i=1}^{m} c_i z_i^{1+\alpha}, \text{ where } c_1, \ldots, c_m > 0 \text{ and } \alpha > 0,$$

and we define the cumulative cost process

$$\xi(t) = \int_0^t h(J(u))\, du, \quad t \geq 0.$$

For concreteness, let's say that the system manager's objective is to

$$\text{minimize } \mathbf{E}[\xi(T)] \, ,$$

where the time horizon T is large.

Reduction to a Standard Model

Let us define the *achievable region* $R = \{y \in \mathbf{R}_+^m : y = Mx, Ax \leq e, x \geq 0\}$. That is, R consists of all service rate vectors $y = Mx$ that are available to the system manager.

Theorem. $R = \{y \in \mathbf{R}_+^m : \hat{A}y \leq e\}$, where $\hat{A}$ is a non-negative $\hat{p} \times m$ matrix ($\hat{p}$ is a positive integer) having at least one positive element in each column, and e is the $\hat{p}$-vector of ones.

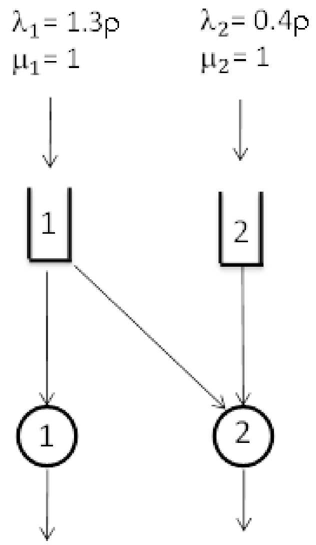
Proof. This is Proposition 5.1 of Kang *et al*. (2009), but essentially the same result appeared in Section 3.3 of Kelly, F.P. (1991). Loss networks. *Ann*. *Appl*. *Probab*. **1** 319-378.

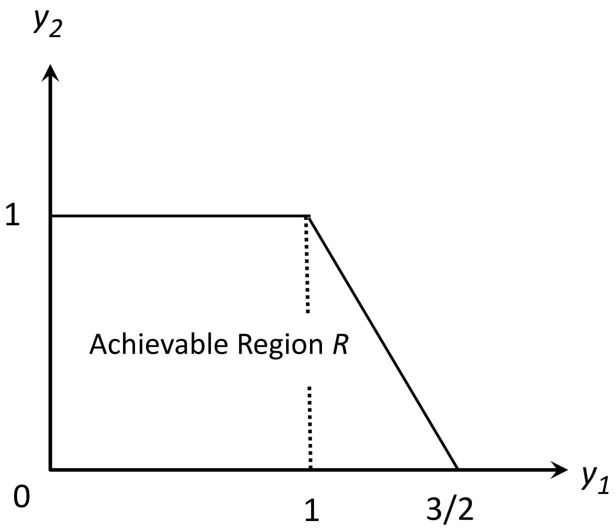
Thus we see that the original model is *completely equivalent* to another one whose service rate matrix $\widehat{M}$ is the $m \times m$ identity matrix. That is, in the equivalent model each activity $i = 1, \ldots, m$ simply serves job type i at unit rate. The equivalent model has $\hat{p}$ resources, each with unit capacity, and has capacity consumption matrix $\hat{A}$. We call the equivalent model *standard* if $\hat{p}$ is minimal.

Reference

Kang, W. N., Kelly, F. P., Lee, N. H. and Williams, R. J. (2009). State space collapse and diffusion approximation for a network operating under a fair bandwidth sharing policy. *Ann*. *Appl*. *Probab*. **19** 1719–1780.

The Simple Parallel-Server Example





$$M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\widehat{M} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

$$\hat{A} = \begin{pmatrix} 2/3 & 1/3 \\ 0 & 1 \end{pmatrix}$$

A Last Bit of Notation

Let $D = \text{diag}(\mu_1, \ldots, \mu_m)$ and

$$\Delta = \{\delta \in \mathbf{R}_+^m \colon \delta = Dy,\ y \in R\}.$$

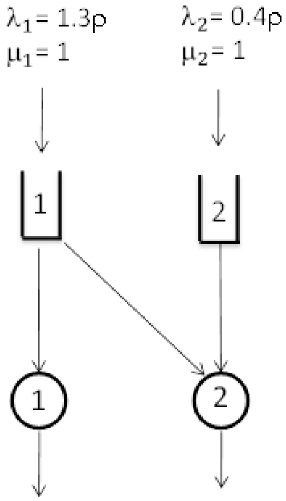
Thus Δ is the set of all long-run average *departure rates* for the various job types that are achievable given our resource capacity constraints.

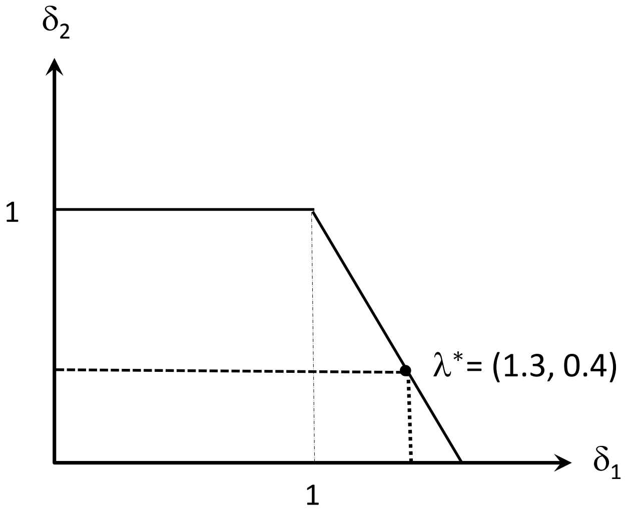
For our simple parallel server example, $\Delta = R$.

The Heavy Traffic Parameter Regime

Consider a standard model with data (λ, μ, A), and define the set Δ of achievable departure rates as above. We say that the network is *in heavy traffic* if (*i*) the arrival rate vector λ is close to a vector $\lambda^* > 0$ that is on the boundary of Δ, and (*ii*) the relevant time horizon T is large. More precisely, we define heavy traffic as follows: there exists $\lambda^* > 0$ on the boundary of Δ, and a large parameter $r > 0$, such that

- the time horizon T is of order r^{2} or larger, and
- all components of the m-vector $\theta = r\,(\lambda - \lambda^*)$ are moderate





Approximating Brownian Control Problem

Consider a standard model in heavy traffic, as above, and suppose that $AD^{-1}\lambda^{*}= e$. (That is, to match the nominal arrival rates λ_i^* one must use the full capacity of *all* p resources, or to put it another way, all p resources are *critical* in the heavy traffic limit under consideration.) The system manager's problem is then plausibly well approximated by the following Brownian control problem (BCP).

Here $X = \{X(t), 0 \leq t \leq \tau\}$ is an m-dimensional Brownian motion with drift vector θ and a certain covariance matrix Σ. The chosen control $Y = \{Y(t), 0 \leq t \leq \tau\}$ must be continuous and adapted to X with $Y(0) = 0$, and must further satisfy constraints (1) and (2) below:

(1) $Z(t) \equiv X(t) + Y(t) \geq 0$ (normalized job counts must remain non-negative)

(2) $U(\cdot) \equiv \mathrm{A}Y(\cdot)$ is $\uparrow$ (cumulative idleness is non-decreasing for each resource)

(3) $\mathbf{E}\{\int_0^\tau h(Z(t)dt\}$ objective to be minimized

Interpretations:

$$
\begin{aligned}
Z(t) &= r^{-1}J(r^2t) \\
Y(t) &= r^{-1}[\lambda^* r^2 t - DS(r^2 t)] \\
X(t) &= r^{-1}[E(r^2t) - F(\bar{S}(t))] \\
\bar{S}(t) &= r^{-2}[\bar{S}(r^2 t) \\
\tau &= r^{-2}T
\end{aligned}
$$

Equivalent Workload Formulation of the BCP

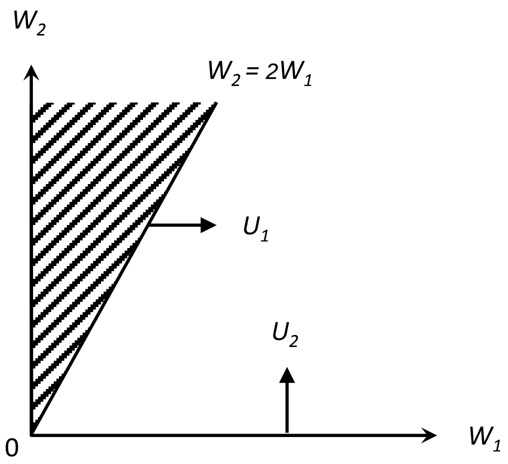
Defining the p-dimensional workload process $W(t) \equiv AD^{-1}Z(t)$, and the p-dimensional Brownian motion $B(t) \equiv AD^{-1}X(t)$, we can restate problem (1)-(3) as follows: choose a control $U = \{U(t), 0 \le t \le \tau\}$ to satisfy

(4) minimize $\mathbf{E}\{\int_0^\tau h(Z(t)dt\}$ where

(5) $W(t) = B(t) + U(t),\ 0 \le t \le \tau$

(6) $U(\cdot)$ is $\uparrow$ with $U(0) = 0$

(7) $W(t) = AD^{-1}Z(t),\ 0 \le t \le \tau,$
for some continuous process $Z(t) \ge 0$.



Of, course, (7) requires that $U(\cdot)$ be chosen so as to keep $W(\cdot)$ in the column span of AD^{-1} (see picture above).

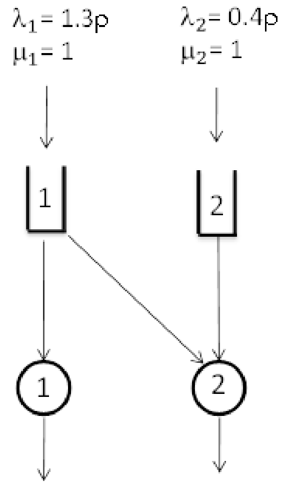
Reference

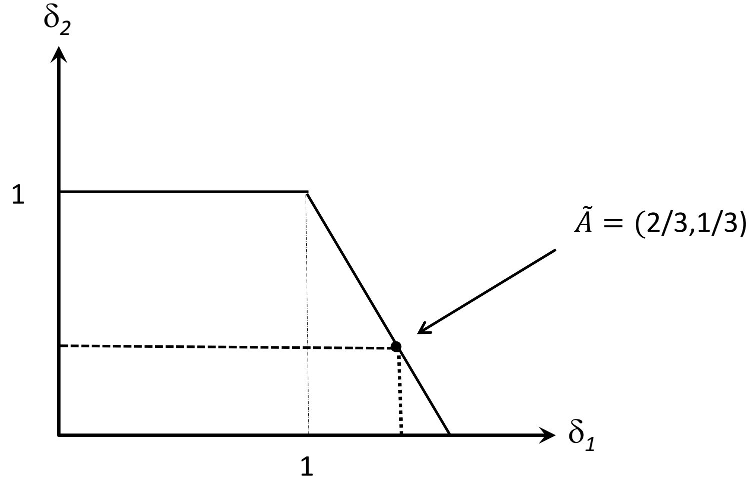
Harrison, J. M. (2000). Brownian models of open processing networks: Canonical representation of workload. *Ann. Appl. Probab.*, **10** 75-103. Correction **13** (2003) 390-393.

Heavy Traffic Limit with Just One Critical Resource

If some, but not all, of the standard model's p capacity constraints are binding with the nominal arrival rate vector λ^*, we simply drop non-critical resources from the approximating BCP, and similarly drop any job classes that require only non-critical resources for their processing. We denote by $\tilde{p}$ the number of critical resources, and by $\tilde{m}$ the number of critical job classes that are retained in the approximating BCP.

The extreme scenario, illustrated by our simple parallel-server example (see below), is that with a *single critical resource* (that is, $\tilde{p} = 1$), which means that λ^* lies on just one of the hyperplanes that form the outer boundary of the polytope Δ.





When $\tilde{p} = 1$ and non-critical resources are dropped from the model, the resulting capacity consumption matrix $\tilde{A}$ is a $1\times\tilde{m}$ vector normal to the (unique) boundary hyperplane of Δ on which λ^* lies.

Outline

1. Stochastic Processing Networks

2. Multi-Mode Resource Sharing Models

   2.1 Examples and General Formulation
   2.2 Reduction to a Standard Model
   2.3 Heavy Traffic Parameter Regime
   2.4 Approximating Brownian Control Problem

→ 3. One-Dimensional Brownian Control Problems

   3.1 Pathwise Solution
   3.2 Achieving the Solution via MaxWeight
   3.3 A Complex Parallel-Server Example
   3.4 Asymptotic Optimality with Linear Costs

4. Achieving HGI Performance in Bandwidth Sharing

   4.1 Local Traffic Assumption
   4.2 The BCP with all Resources Critical
   4.3 HGI Performance
   4.4 Open Problem

Equivalent Workload Formulation of a One-Dimensional BCP

Assuming that $\tilde{p} = 1$, define the one-dimensional Brownian motion $B(t) \equiv \tilde{A}X(t)$. The problem is to choose a one-dimensional control $U = \{U(t), 0 \leq t \leq \tau\}$ to

(4) minimize $\mathbf{E}\{\int_0^{\tau} h(Z(t)dt\}$ where

(5) $W(t) = B(t) + U(t)$, $0 \leq t \leq \tau$

(6) $U(\cdot)$ is $\uparrow$ with $U(0) = 0$

(7) $W(t) = \tilde{A}Z(t)$ for some continuous process $Z(t) \geq 0$.

Pathwise Solution of a One-Dimensional BCP

The optimal solution is

$$U^*(t) = -\min_{0\le u\le t} B(u),\ \ 0 \le t \le \tau,$$

which minimizes cumulative idleness of the one critical resource, and hence also minimizes workload for that resource, at all times *t* simultaneously, together with

$$Z^*(t) \in \operatorname{argmin}\{h(z)\colon \tilde{A}z = W^*(t),\ \ z\in\mathbf{R}_+^m\},$$

where $W^*(t) = B(t) + U^*(t)$. That is, the optimal control strategy minimizes workload in the pathwise sense, and holds the workload at each time *t* in a least-cost job count vector.

Further Detail on the Pathwise Solution

With a cost function of the assumed form $h(z) = \sum_{i=1}^{\tilde{m}} c_i z_i^{1+\alpha}$, the optimal choice $Z^*(t)$ given $W^*(t)$ is

$$Z_i^*(t) = \beta_i W^*(t) \ \ \text{where}\ \ \beta_i = c_i^{-1/\alpha} / \sum_{k=1}^{\tilde{m}} \tilde{A}_{1i} c_k^{-1/\alpha} \ \text{ for } i = 1, \ldots, \tilde{m}.$$

MaxWeight (MW) Scheduling

For our standard model with p resources, m job classes, arrival rate vector λ, and capacity consumption matrix A, the MW scheduling rule is as follows: at each time t choose the m-vector y of service rates so as to

$$\text{maximize } \sum_{i=1}^{m} c_i \mu_i y_i J_i^{\alpha}(t) \quad \text{subject to } Ay \leq e,\ y \geq 0.$$

This is the vector y which maximizes the expected rate of decrease in the cost rate $h(J(t))$.

Theorem. Consider a standard model with capacity consumption matrix A. Let there be given a sequence of arrival rate vectors $\{\lambda^r, r = 0, 1, \ldots\}$ that approach a limit $\lambda^* > 0$ that lies on just one boundary hyperplane of the polytope Δ. Let there also be given a sequence of time horizons $\{T^r, r = 0, 1, \ldots\}$. Assume that $r(\lambda^r - \lambda^*) \to \theta \in \mathbf{R}^m$ and $r^{-2}T^r \to \tau > 0$ as $r \to \infty$.

Let $Z^r = \{Z^r(t), 0 \leq t \leq \tau\}$ be the corresponding sequence of diffusion-scaled job count processes using MaxWeight scheduling, and let $Z^* = \{Z^*(t), 0 \leq t \leq \tau\}$ be the optimal solution of the one-dimensional BCP described earlier. Then Z^r converges weakly (that is, converges in distribution) to Z^* as $r \to \infty$.

References

Stolyar, A. L. (2004). MaxWeight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Ann. Appl. Probab.*, **14** 1–53.

Mandelbaum, A., and Stolyar, A. L. (2004). Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Operations Research*. **52** 836–855.

How MW Scheduling Works in Heavy Traffic

Assuming that $\tilde{p}$ = 1 (just one resource constraint binding in the standard model), let us define the one-dimensional workload process $w(t) = \tilde{A}J(t)$. In heavy traffic (that is, when λ is close to λ^*), the value of $w(t)$ changes slowly.

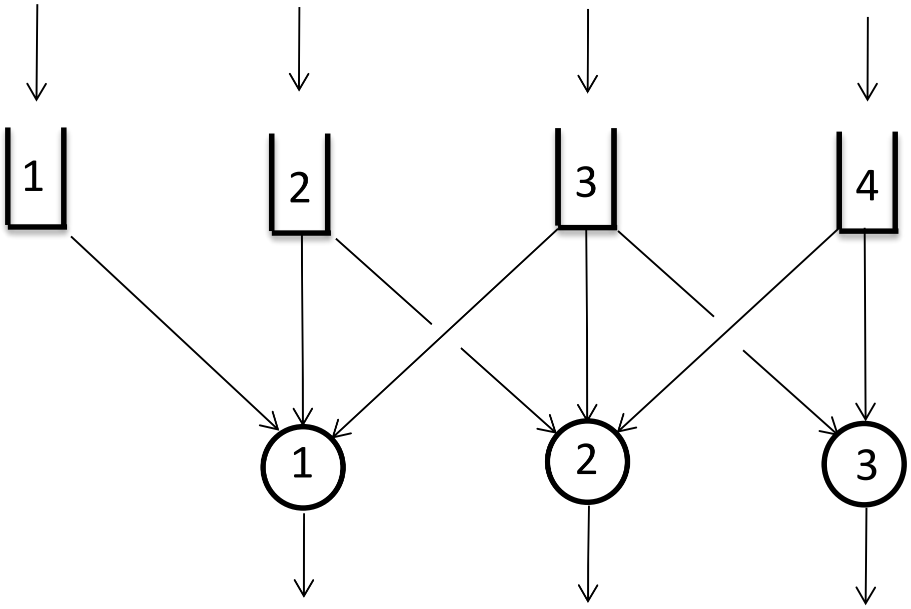
Because MW continually strives to maximize the expected rate of decrease in the cost rate $h(J(t))$, at any given workload level $w(t) = w$, it drives $J(\cdot)$ toward the vector z^* which minimizes $h(z)$ subject to the constraints $\tilde{A}z = w,\ z \geq 0$. The solution to that optimization problem is $z^* = \beta w$, where $\beta = (\beta_i)$ is the strictly positive m-vector identified above. Thus the processes $J(t)$ and $\beta w(t)$ are indistinguishable in the heavy traffic limit. This is *called state space collapse*.

Given that job types are held in essentially fixed ratios, the MW rule selects very nearly the same service rate vector y at all times, namely, the vector

$$y = (\frac{\lambda_1^*}{\mu_1}, \dots, \frac{\lambda_m^*}{\mu_m}).$$

That service rate vector fully utilizes the one critical resource, provided that there is work in the system for the critical resource to do. Thus, cumulative idleness of the critical resource is minimized, or to put it another way, the associated workload process $w(\cdot)$ is minimized in the *pathwise* sense.

A Complex Parallel-Server Example



Is this system in heavy traffic?

If so, does the corresponding standard model have a single critical resource?

The Static Planning Problem

Find average activity rates $x_1, \ldots, x_n$ and a scalar ρ to

$$\begin{aligned} &\text{minimize} \quad \rho \\ &\text{subject to the constraints} \quad DMx = \lambda\,,\ \ Ax \leq \rho e,\ \ \text{and}\ \ x \geq 0 \end{aligned}$$

Interpret ρ as an upper bound on utilization rates for the various resources under processing plan x.

The Dual of the Static Planning Problem

Find vectors $u = (u_1, \ldots, u_m)$ and $v = (v_1, \ldots, v_p)$ to

$$\begin{aligned} &\text{maximize} \quad u'\lambda \\ &\text{subject to the constraints} \quad u'M \leq v'A,\ \ v'e = 1,\ \ \text{and}\ \ v \geq 0. \end{aligned}$$

Interpret u_i as the *total work content* ascribed to a type i job, and v_k as the *relative capacity* of resource k.

The Complex Parallel-Server Example in Harrison-Lopez (1999)

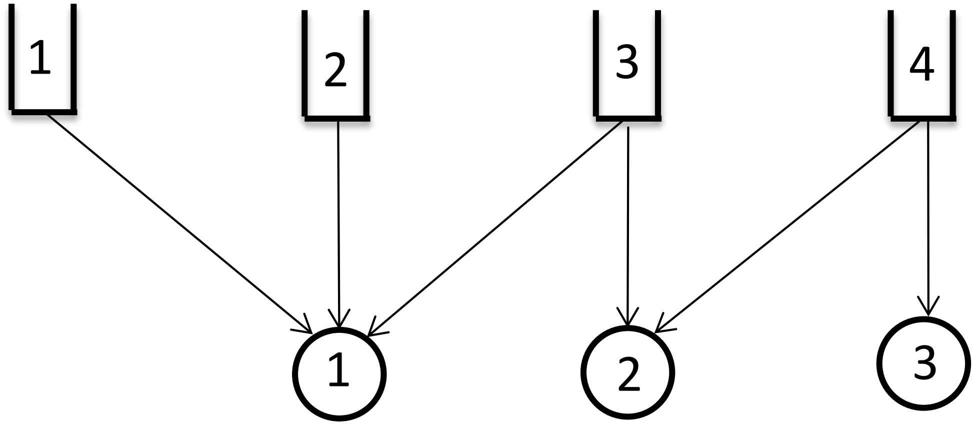
- The static planning problem has a unique optimal solution (x^*, ρ^*). Moreover, $\rho^* = 1$ and $Ax^* = e$. It follows that λ lies on the boundary of the achievable region R.

- The dual of the static planning problem has a unique optimal solution (u^*, v^*). It follows that the corresponding standard model has a single critical resource (that is, just one resource constraint is binding in the equivalent standard model), and v^* is normal to the single boundary hyperplane on which λ lies.

- Thus, assuming that the time horizon T is large, this is a system in heavy traffic, its approximating Brownian control problem is one-dimensional, and the MaxWeight scheduling rule is asymptotically optimal in the heavy traffic limit.

Why does this stochastic processing network behave essentially like a single-resource system? That is, in what sense do its three processing resources, each critically loaded under the given arrival rate vector λ, constitute a single pool of fungible capacity? (This phenomenon is called *complete resource pooling*.)

The Complex Parallel-Server Example (Continued)



The arrows in this diagram correspond to activities j which are used at positive levels in the nominal processing plan x^*.

The system is effectively one-dimensional because *all servers communicate*.

References

Harrison, J. M., and Lopez, M. J. (1999). Heavy Traffic Resource Pooling in Parallel-Server Systems, *Queueing Systems*, **33** 339-368.

Stolyar, A. L. (2004). MaxWeight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Ann. Appl. Probab.*, **14** 1–53.

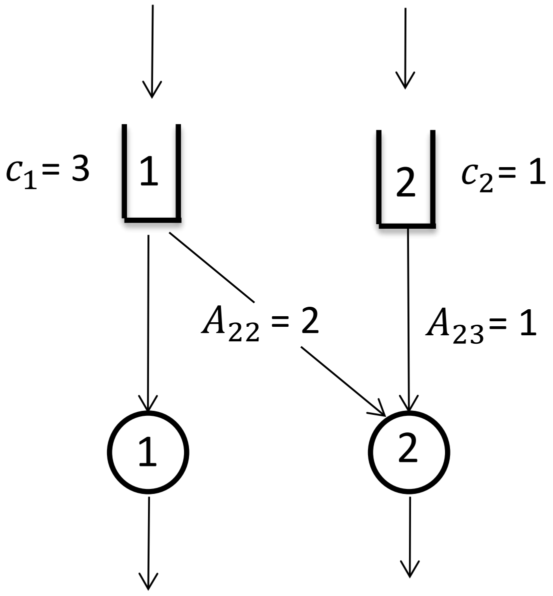
Dynamic Scheduling with Linear Costs

If we take $\alpha = 0$ in our cost function, then the MW scheduling rule amounts to the following: choose the vector y of service rates to

$$\text{maximize} \quad \sum_{i=1}^{m} c_i \mu_i y_i \quad \text{subject to} \quad y = Mx, \ Ax \leq e, \ x \geq 0.$$

This is the classical cμ rule. It directs us to choose instantaneous service rates greedily, striving to drive down the current cost rate $h(J(t)) = \sum_{i=1}^{m} c_i \, J_i(t)$ as fast as possible.

For our simple parallel server model, this means the following: server 2, given a choice between serving job type 1 at rate ½ and serving job type 2 at rate 1, will always choose the former.



This seems consistent with the optimal solution of the approximating BCP, which has the following naïve verbal paraphrase: never let either server be idle if there is work for it in the system, and never allow a queue of type 1 jobs to develop.

What is the effect of this priority scheme over the long run?

Greedy Scheduling is Potentially Disastrous with Linear Costs

The graph at right shows the backlog of type 1 jobs (very few) and of type 2 jobs (growing linearly) for the simple parallel-server example with greedy scheduling and ρ = 0.95.

Two modifications of greedy scheduling have been analyzed in the literature, each restoring asymptotic optimality if its policy parameters are chosen properly.

- discrete-review policies
- threshold policies

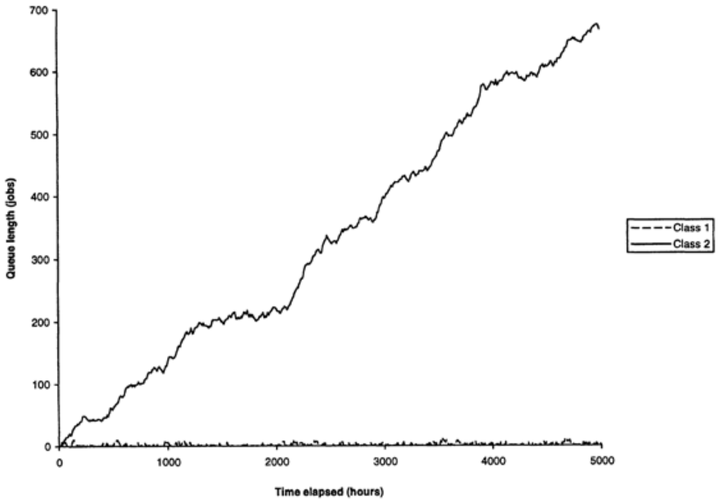


FIG. 2. *System behavior with greedy scheduling* (ρ = 0.95).

References

Harrison, J. M. (1998), Heavy traffic analysis of a system with parallel servers: Asymptotic analysis of discrete-review policies, *Ann. Appl. Probab.*, **8** 822-848.

Bell, S. L. and Williams, R. J. (2005). Dynamic scheduling of a parallel server system in heavy traffic with complete resource pooling: Asymptotic optimality of a threshold policy. *Electronic J. Probab.* **10** 1044-1115.

Outline

1. Stochastic Processing Networks

2. Multi-Mode Resource Sharing Models

   2.1 Examples and General Formulation
   2.2 Reduction to a Standard Model
   2.3 Heavy Traffic Parameter Regime
   2.4 Approximating Brownian Control Problem

3. One-Dimensional Brownian Control Problems

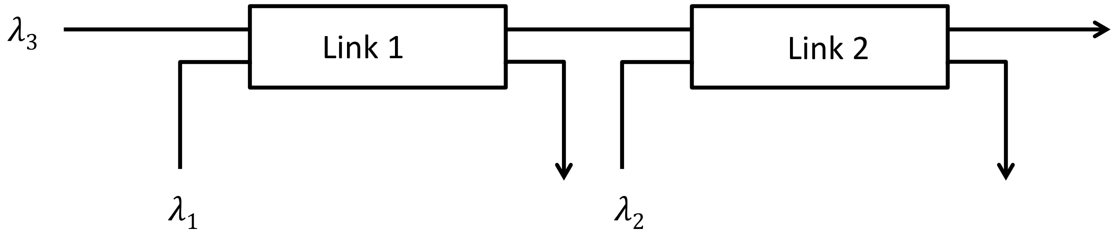
   3.1 Pathwise Solution
   3.2 Achieving the Solution via MaxWeight
   3.3 A Complex Parallel-Server Example
   3.4 Asymptotic Optimality with Linear Costs

→ 4. Achieving HGI Performance in Bandwidth Sharing

   4.1 Local Traffic Assumption
   4.2 The BCP with all Resources Critical
   4.3 HGI Performance
   4.4 Open Problem

The Bandwidth Sharing Model of Massoulie and Roberts

Files (jobs) of different types arrive according to independent Poisson processes. File sizes for type j are random, drawn from an exponential distribution with mean $1/\mu_j$. Transmission of a file is treated as pumping of a fluid; amount of fluid to be pumped = size of file. Network resources are transmission links (or servers). Link i has capacity C_i (see below for the meaning of this). **Assume there is "local traffic" on every link.**



References

Massoulié, L. and Roberts, J. (2000). Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, **15** 185–201.

Kang, W. N., Kelly, F. P., Lee, N. H. and Williams, R. J. (2009). State space collapse and diffusion approximation for a network operating under a fair bandwidth sharing policy. *Ann. Appl. Probab.*, **19** 1719–1780.

E(Tot) as a Performance Measure

For purposes of this presentation, let us consider a linear cost structure with $c_i = 1$ for all i, and further focus on steady-state performance. That is, the performance measure on which we focus is

$$E(Tot) = \sum_{i=1}^{m} \mathbf{E}[J_i(\infty)]$$

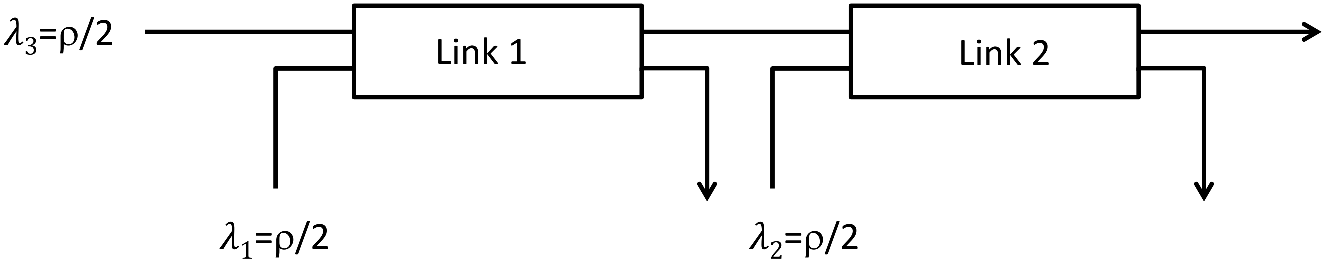
Bandwidth Sharing Networks versus Conventional Queueing Networks

- No internal buffering
- Simultaneous resource possession
- Each job has just one “service time” (same for all resources involved in its processing)
- Services can be interrupted without penalty or efficiency loss

Simple Two-Link Linear Network (S2LLN)

λ_i = rate of Poisson arrival process for jobs of type i
ρ = common load factor for the two links

File size distribution is exponential with mean 1 for each type



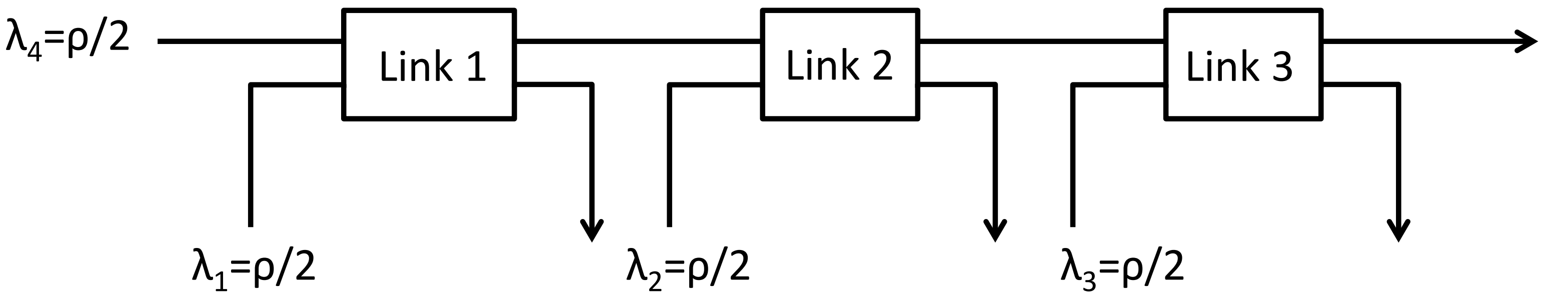
Reference

Verloop, I. M. and Núñez Queija, R. (2009). Assessing the efficiency of resource allocations in bandwidth-sharing networks. *Performance Evaluation*, **66** 59-77.

Simple Three-Link Linear Network (S3LLN)

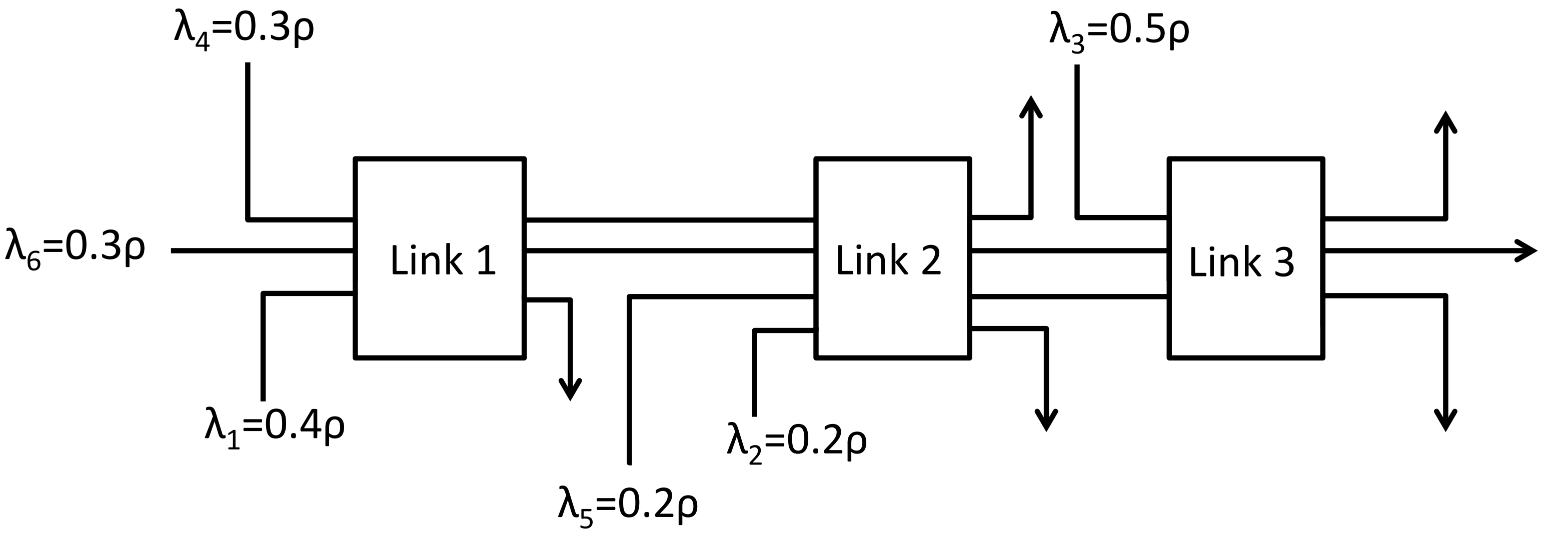
λ_i = rate of Poisson arrival process for jobs of type i
ρ = common load factor for the three links

File size distribution is exponential with mean 1 for each type



Complex Three-Link Linear Network (C3LLN)

File size distribution is exponential with mean 1 for each type



Proportionally Fair (PF) Resource Allocations

Let $J(t) = n = (n_i)$ be the vector of job counts for different types at a given time t. The vector $y = (y_i)$ of *proportionally fair* service rates solves the following problem:

$$(1) \qquad \underset{\substack{Ay \leq e \\ y \in \Phi(n)}}{maximize} \quad \sum_i n_i \log y_i \ ,$$

where $\Phi(n)$ is the set of allocation vectors $y \geq 0$ such that $y_i = 0$ if $n_i = 0$.

PF allocations are *non-extremal*: every job that is currently active gets *some* bandwidth allocation, regardless of network status.

Can We Improve on Proportional Fairness?

- S2LLN

Load (ρ)	$E(Tot)$ under PF	$E(Tot)$ under UFOS	Improvement
0.80	7.33	6.05	**17.4%**
0.90	17.17	13.26	**22.8%**
0.95	37.11	27.13	**26.9%**

- S3LLN

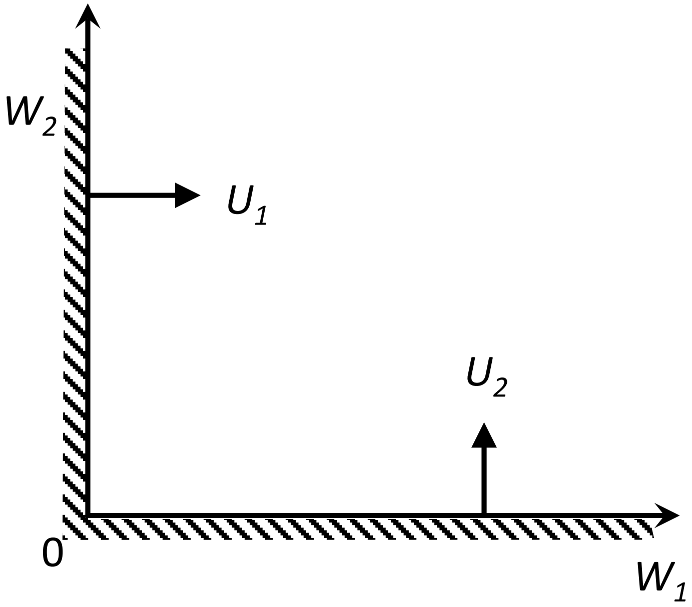
Load (ρ)	$E(Tot)$ under PF	$E(Tot)$ under UFOS	Improvement
0.80	10.56	9.03	**14.5%**
0.90	25.20	19.66	**22.0%**
0.95	53.97	41.71	**22.7%**

- C3LLN

Load (ρ)	$E(Tot)$ under PF	$E(Tot)$ under UFOS	Improvement
0.80	10.34	8.12	**21.5%**
0.90	24.75	17.57	**29.0%**
0.95	54.29	36.23	**33.3%**

Brownian Control Problem (BCP)

- formally approximates the dynamic bandwidth allocation problem
- approximating BCP is reduced to an *equivalent workload formulation*
- state descriptor is a Brownian analog W of the workload process w defined earlier
- components of control process U represent cumulative unused capacity for links
- main system equation is $W(t) = B(t) + U(t),\ t \geq 0,$ where B is a Brownian motion



W must be kept within a certain polyhedral cone, and *because of our local traffic assumption*, *that cone is the entire orthant*.

Thus there exists an admissible control U^* in the approximating BCP that gives a *minimal* workload process W^*,

W* is a reflected Brownian motion (RBM) living in the entire non-negative orthant, with normal reflection from each boundary.

Brownian Approximation (Continued)

In the approximating BCP, moreover, the Brownian analog of the job count process J can be any process $Z = \{Z(t), t \geq 0\}$ that is consistent with the chosen workload process W, meaning that $W(t) = AZ(t)$ and $Z(t) \geq 0$ for each $t \geq 0$. The best choice is $Z(t) = f(W(t))$, where

$$f(W) = argmin\ \{h(Z) : AZ = W,\ Z \geq 0\} \ \text{ for } \ W \geq 0.$$

Hierarchical Greedy Ideal (HGI Performance)

$$E\big[h\big(J(t)\big)\big] = \ E[f(w^*(t)\ )] \ \text{ for all } t \geq 0.$$

(a) Workload process closely approximates the minimal workload w^*. That is, we approach the ideal where no link's capacity is ever under-utilized while there is work for that link anywhere in the system

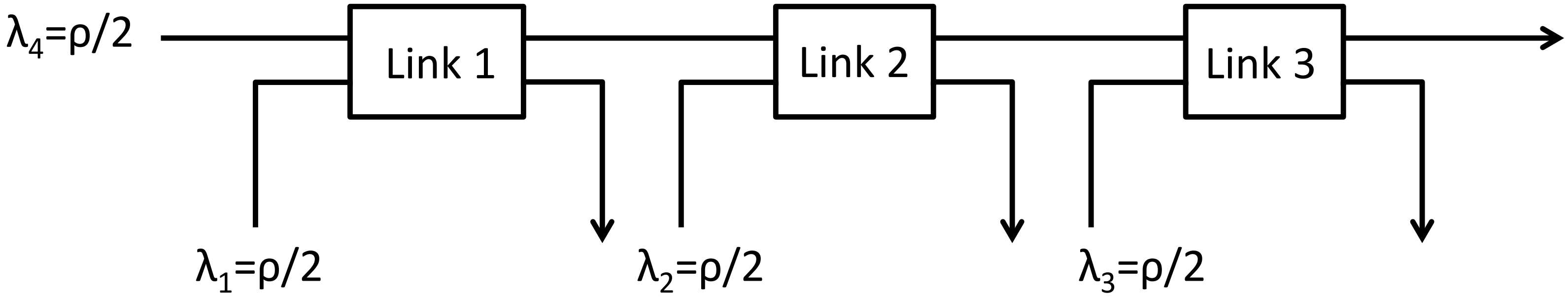
(b) Instantaneous cost rate $h\big(J(t)\big)$ at each time t is near its minimum possible value given the current workload vector. That is, the workload is held in a nearly-least-cost configuration.

Does HGI Performance = Optimal Performance?

For the S3LLN (pictured below) one has

$$f(1,0,1) = (1,0,1,0) \text{ with associated cost } h(1,0,1,0) = 2$$

$$f(1,1,1) = (0,0,0,1) \text{ with associated cost } h(0,0,0,1) = 1$$



Thus, in general, the function $f(\cdot)$ is not necessarily monotone, and hence HGI performance is *not* necessarily optimal performance.

Striving for HGI Performance via the UFOS Algorithm

Assume the ultimate goal is to minimize $E(Tot)$ for an M-R network. UFOS (*utilization first, output second*) is the following hierarchical greedy algorithm.

Let $n = (n_j)$ be the vector of current job counts for different types. We first identify the set of allocation vectors $x = (x_j)$ that solve the following problem (maximize total *utilization first*):

(1) $$\underset{\substack{Ax \le C \\ x \in \Phi(n)}}{maximize} \quad \sum_i (Ax)_i \ .$$

where $\Phi(n)$ is the set of allocation vectors $x \geq 0$ such that $x_j = 0$ if $n_j = 0$. Within the set of allocation vectors that achieve the max in (1), choose one that solves the following problem (*output second*):

(2) $$maximize \sum_j \mu_j x_j \ ,$$

where μ_j^{-1} is the mean file size for class j.

UFOS versus HGI for the Three Examples

		E(Tot)		
		UFOS	HGI	Difference
ρ=0.8	S2LLN	6.06	5.74	5.58%
	S3LLN	8.94	8.80	1.57%
	C3LLN	8.12	7.31	11.12%
ρ=0.9	S2LLN	13.19	12.66	4.20%
	S3LLN	19.81	19.06	3.94%
	C3LLN	17.57	16.00	9.84%
ρ=0.95	S2LLN	27.16	26.46	2.67%
	S3LLN	40.47	39.46	2.56%
	C3LLN	36.23	33.29	8.82%

Open Problem

How to achieve HGI performance in general.

Reference

Harrison, J.M., Mandayam, C., Shah, D. and Y. Yang (2013), Approaching HGI performance in resource sharing networks, submitted for publication September 2013.
http://faculty-gsb.stanford.edu/harrison/Documents/HGI_Paper_for_SSY.pdf