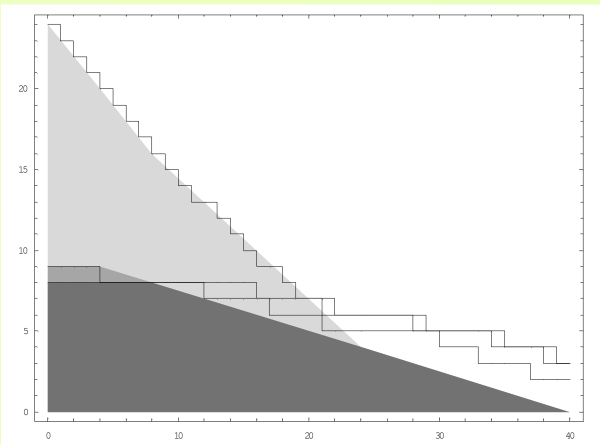
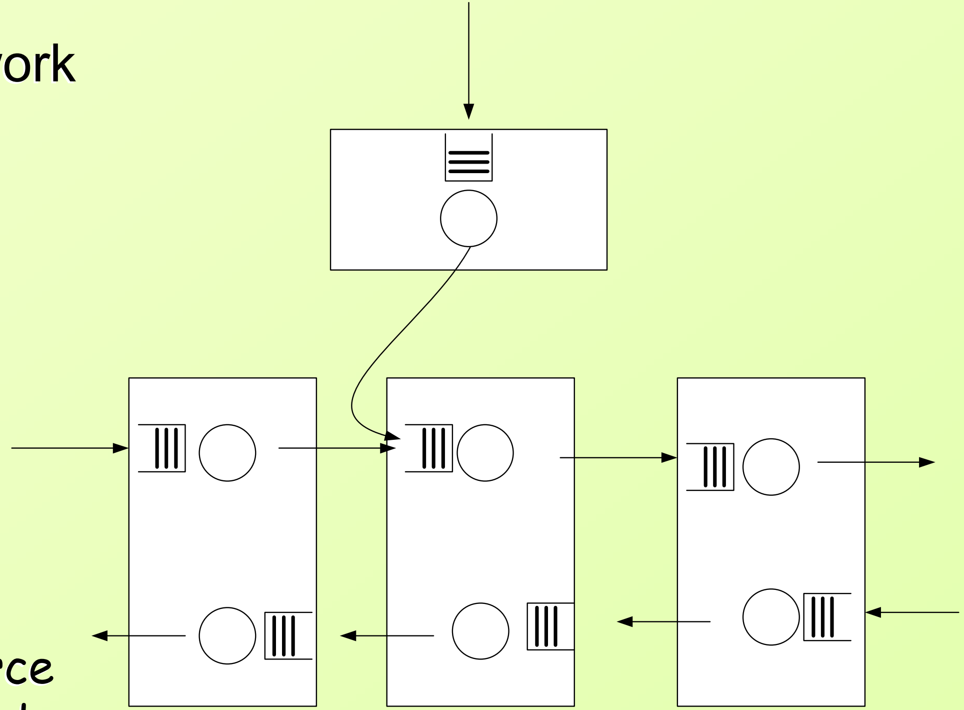
Near Optimal Control of a Multi-Class Queueing Network over a Finite Time Horizon with Applications to Communication Networks

Yoni Nazarathy and Gideon Weiss
University of Haifa



MCQN Modeling of a Communication Network



Multi Class Queueing Networks:

•Jobs (Messages)

•Queues (Buffers)

•Resources (Processors)

•Activities (Operations)

•Routes

Activities share resources → Need resource allocation and scheduling

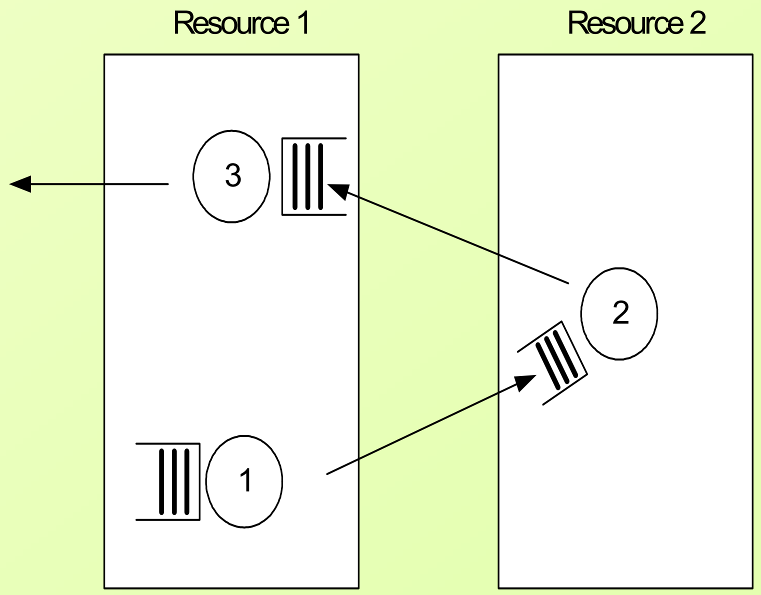
In this work:

•Finite time horizon (T)

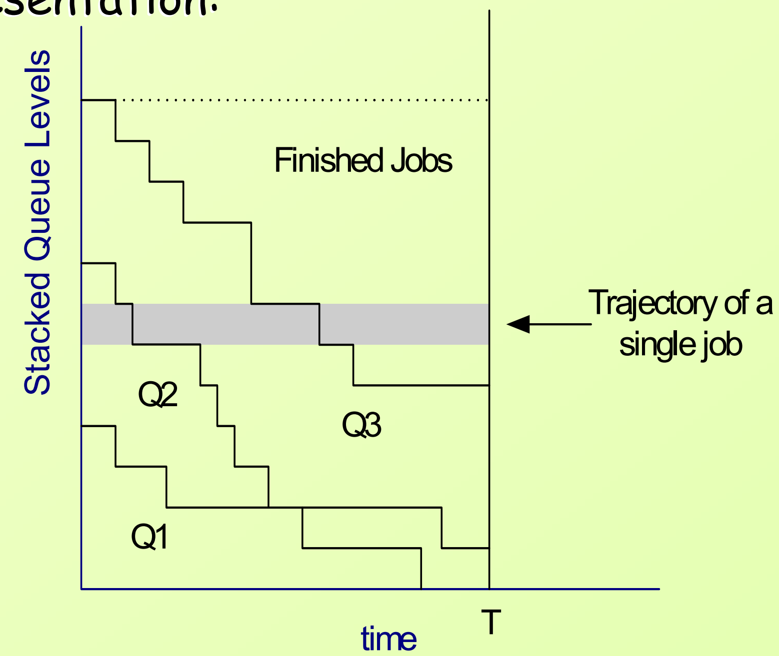
•Centrally controlled optimization

Example Network

- 3 queues
- 2 resources
- Processing rates, μ_1, μ_2, μ_3
- Initial queue amounts, $Q_1(0)$, $Q_2(0)$, $Q_3(0)$
- Resource allocation essentially needed for resource 1.



Stacked Queue level representation:

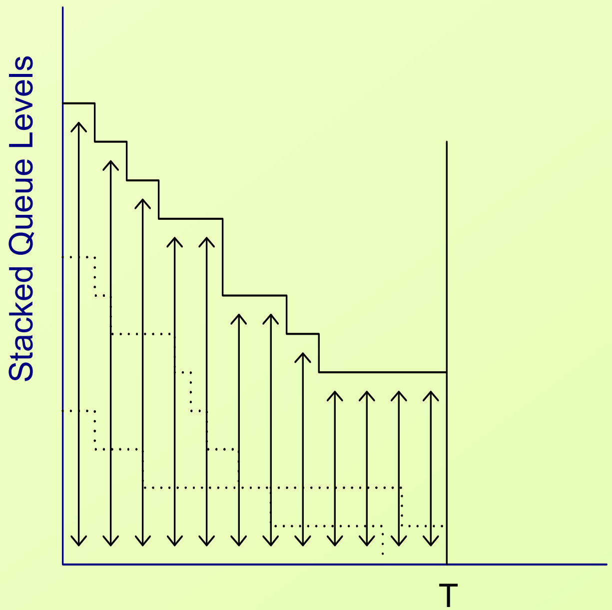


What to optimize?

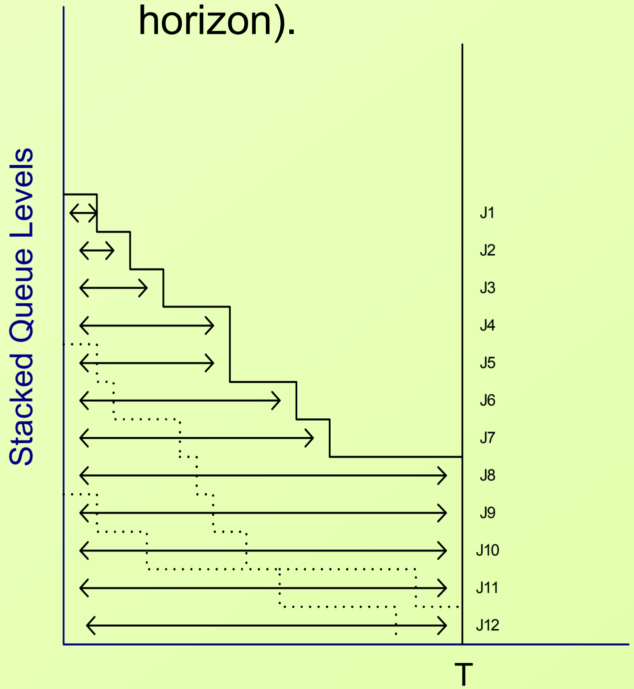
Attempt to minimize: $$\int_0^T \sum_{k=1}^{3} Q_k(t)\,dt$$

This corresponds to:

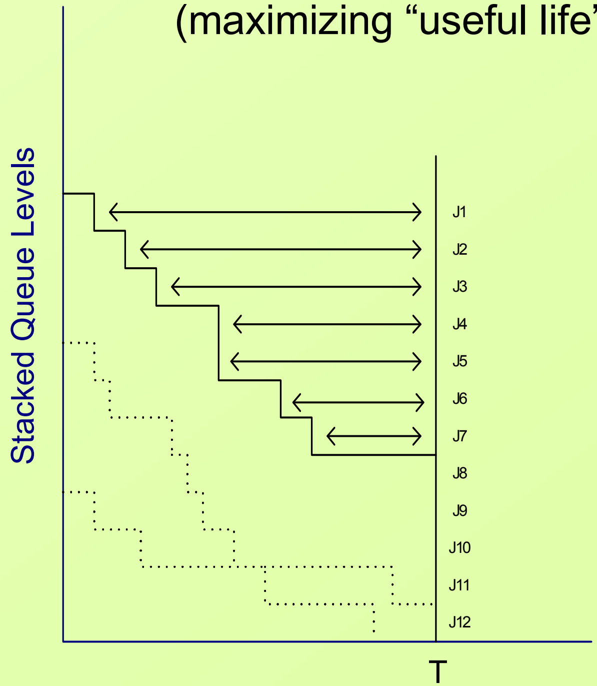
Minimizing inventory costs.



Minimizing the total job waiting time. (truncated to time horizon).



Maximizing the total time from job completion to the time horizon. (maximizing “useful life”)



Applications to Communication networks

Control of highly dynamic, yet heavily congested MANETs.

High Dynamics → Varying unpredictable link conditions.

Heavy Congestion → Long sojourn times.

→ Many messages encounter changes in link capacities and routes during transit.

Possibly integrate with a predictive location based routing scheme (e.g. Shah and K. Nahrstedt 2002):

- Predict the state and relative stability of link conditions.
- Set appropriate duration of Finite time horizon.
- Use near optimal control for time horizon.
- Repeat for next time horizon...

Framework

- Finite Horizon Control
- Stochastic and Discrete System

→ Solution is intractable

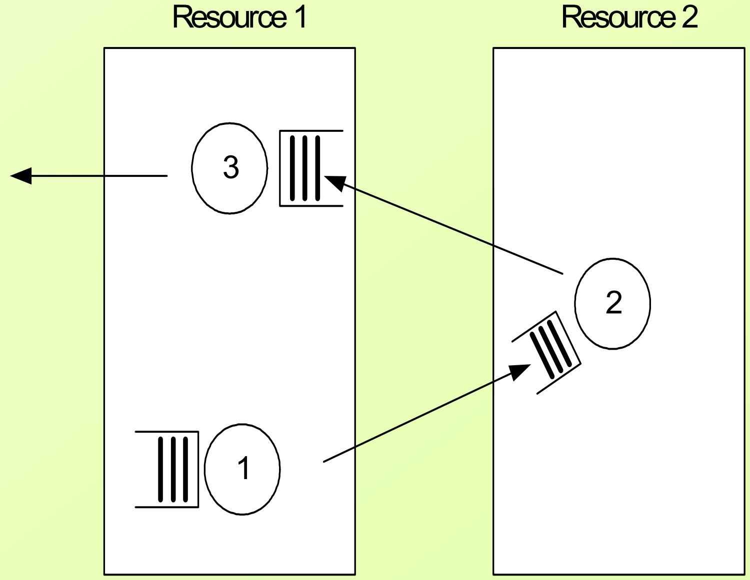
Approach:

1) Approximate the problem using a fluid system.
2) Solve the fluid system (SCLP).
3) Track the fluid solution on-line.
4) Under proper scaling, our approach is asymptotically optimal.

Complementary ideas and methods:

- Solving SCLP using the simplex based algorithm (Weiss).
- MCQNs with Infinite Virtual Queues (Weiss et. al.).
- Maximum Pressure Policies (Dai and Lin 2005).

Fluid formulation



$$\min \int_0^T q_1(t) + q_2(t) + q_3(t)dt$$

s.t.

$$q_1(t) = q_1(0) - \int_0^t u_1(s)ds$$

$$q_2(t) = q_2(0) + \int_0^t u_1(s)ds - \int_0^t u_2(s)ds$$

$$q_3(t) = q_3(0) + \int_0^t u_2(s)ds - \int_0^t u_3(s)ds$$

$$\frac{1}{\mu_1}u_1(t) + \frac{1}{\mu_3}u_3(t) \le 1 \qquad t \in (0,T)$$

$$\frac{1}{\mu_2}u_2(t) \le 1$$

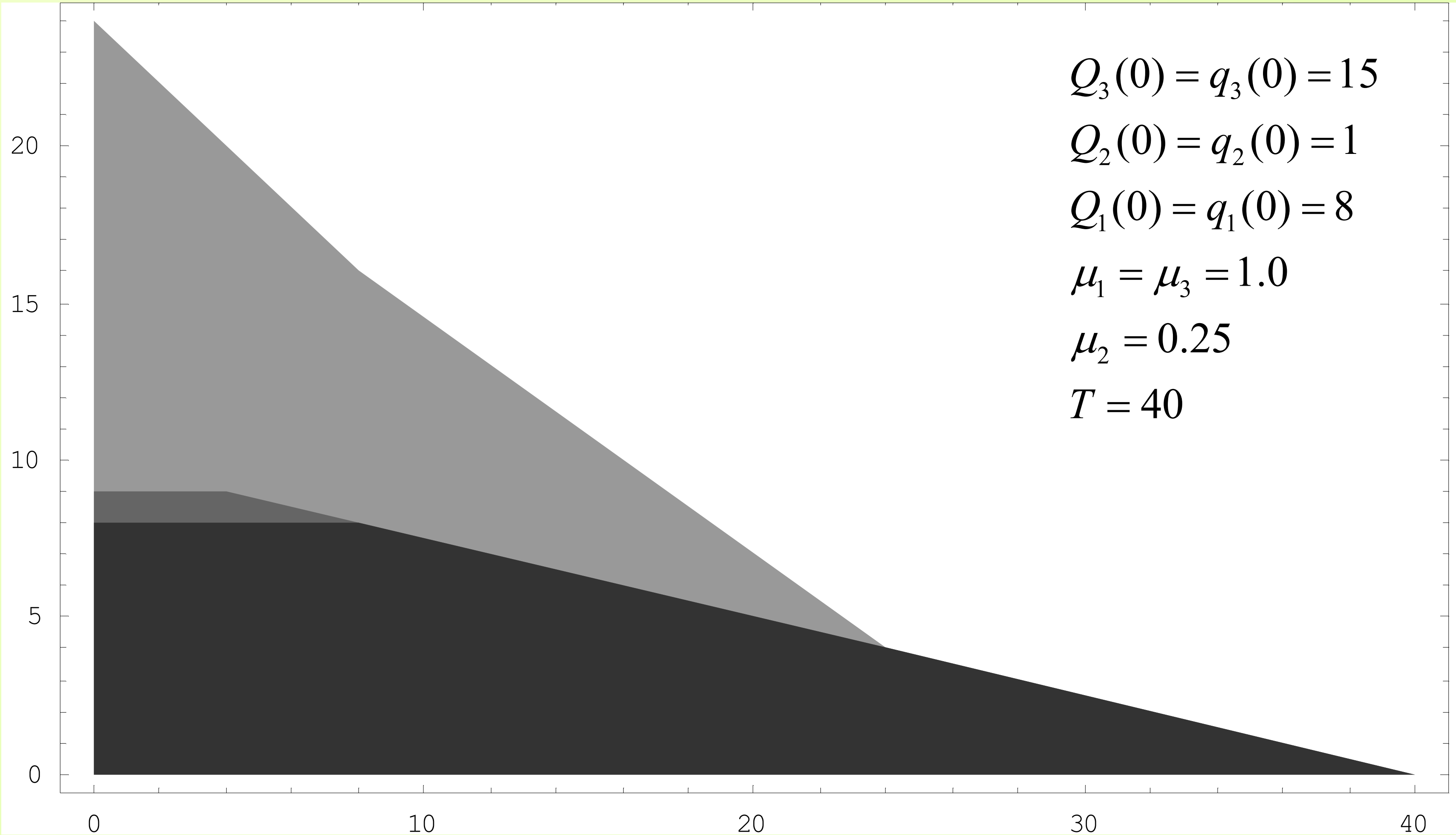
$$u, q \ge 0$$

This is a Separated Continuous Linear Program (SCLP)

Fluid solution

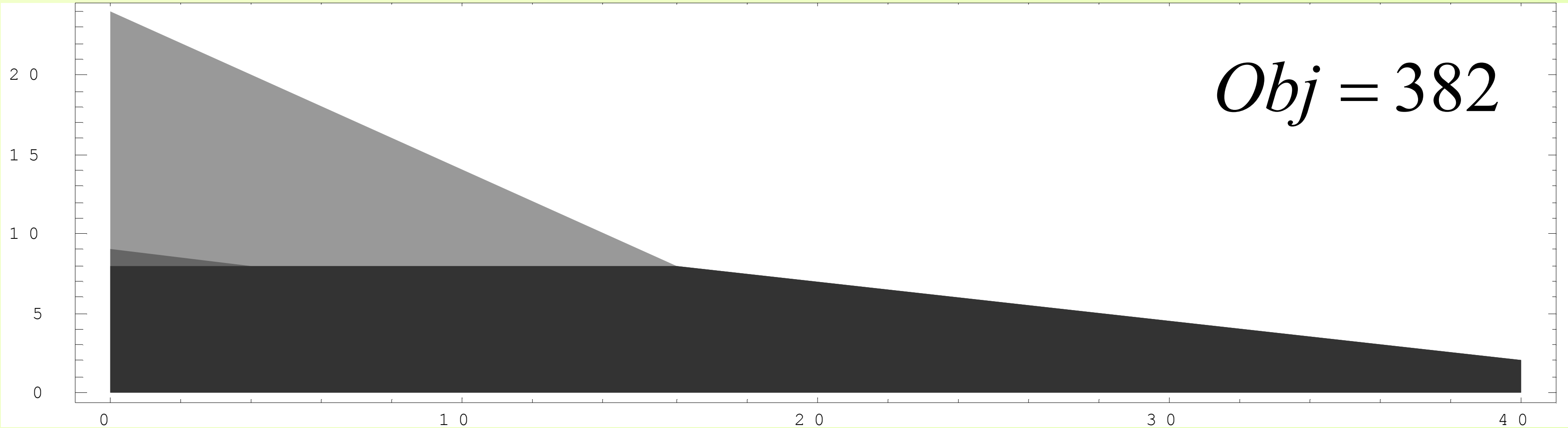
- SCLP – Bellman, Anderson, Pullan, Weiss.
- Solution is piecewise linear with a finite number of breakpoints.
- Simplex based algorithm, finds the optimal solution in a finite number of steps (Weiss).

The Optimal Solution:



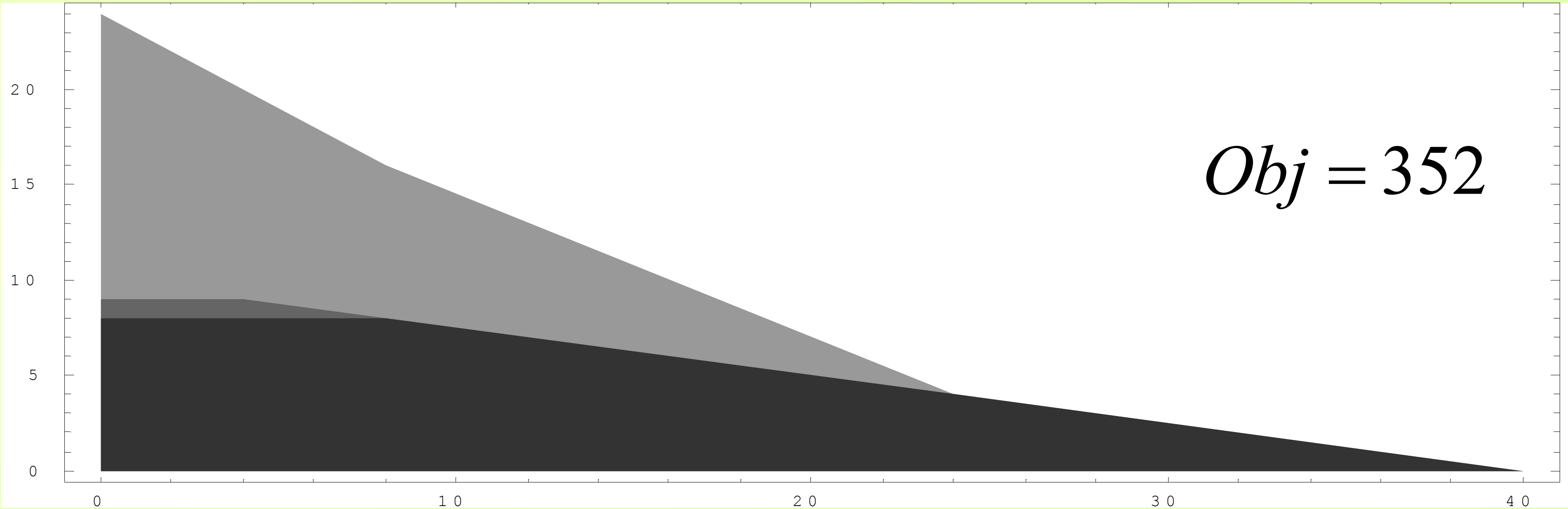
Structure of the optimal solution – comparison to LBFS

Last Buffer First Server (LBFS):



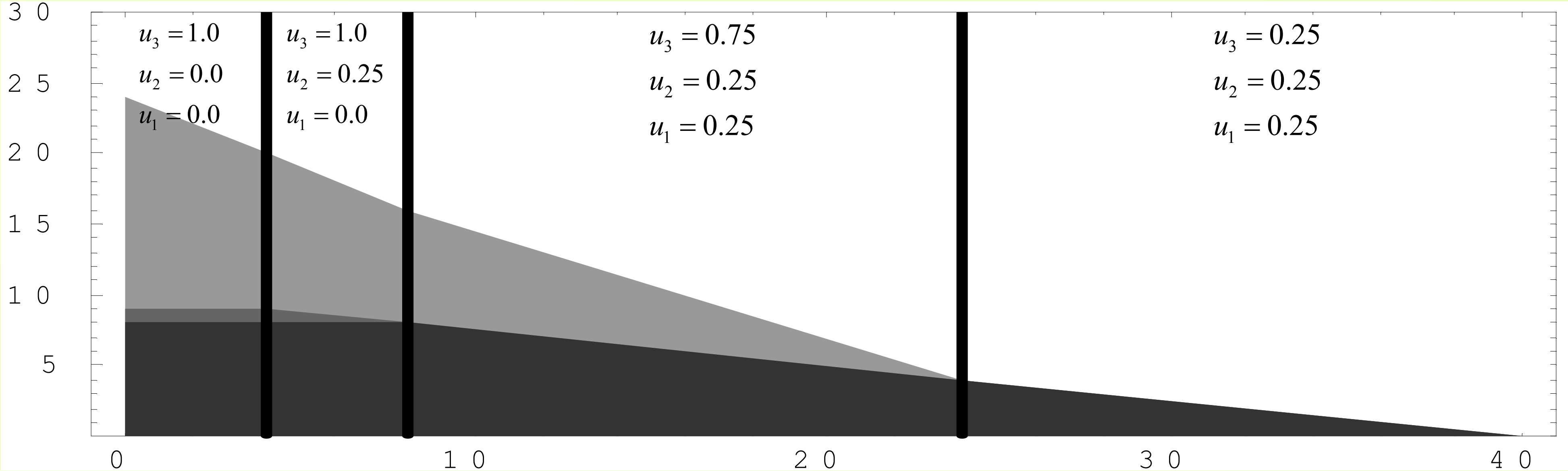
Improve: Don't wait with the emptying of buffer 1 until 3 is empty…

The optimal solution:



Time intervals

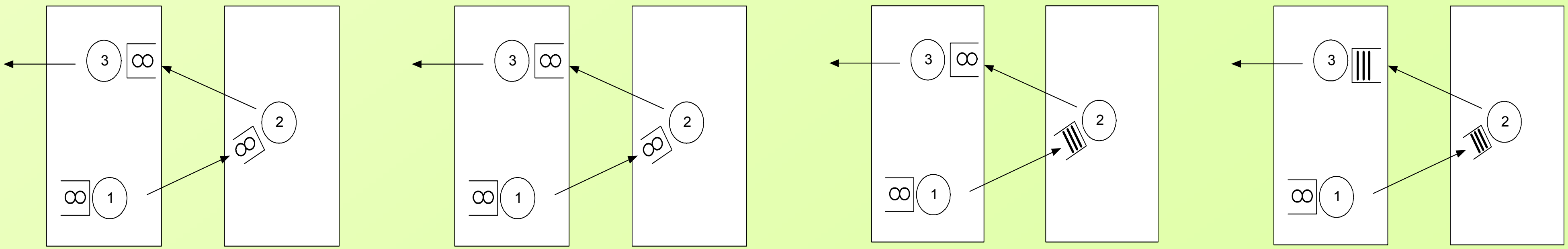
$\tau_1 = [0,4)$ $\tau_2 = [4,8)$ $\tau_3 = [8,24)$ $\tau_4 = [24,40)$



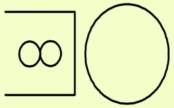
Define:

$$K_0^{\,n} = \{k \mid q_k(t) = 0, \quad t \in \tau_n\}$$

$$K_\infty^{\,n} = \{k \mid q_k(t) \geq 0, \quad t \in \tau_n\}$$

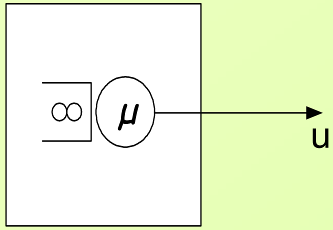


Infinite Virtual Queues

What does 

mean?

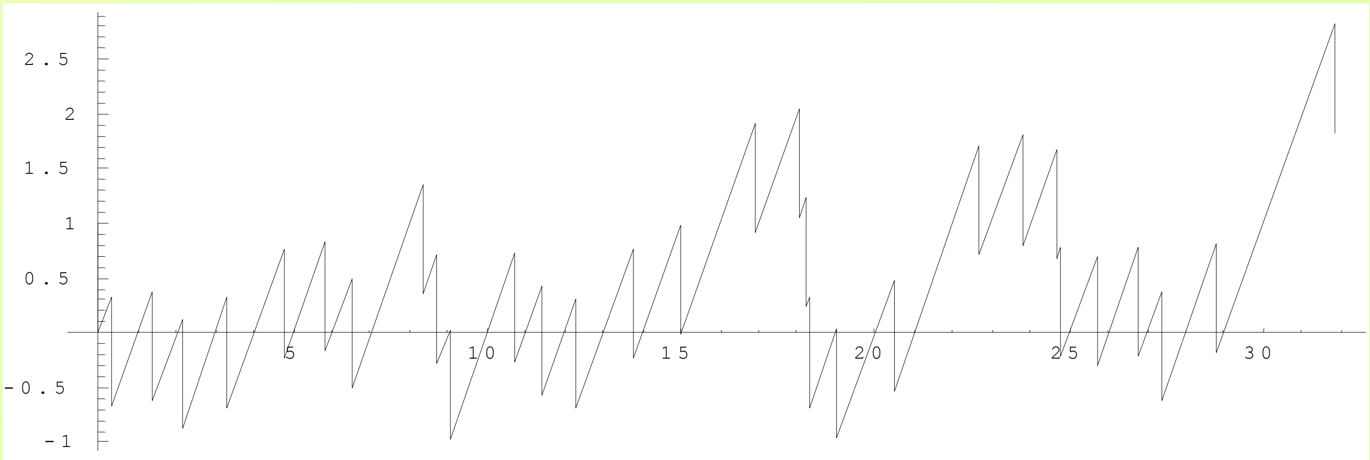
Parameterize with μ and u (nominal flow rate):



Define the state:

$$V(t) = V(0) + u\,t - S(T(t))$$

Example realization:



MCQNs with Infinite Virtual Queues

- Generalize the typical MCQN model.
- Examples analyzed by Weiss, Kopzon and Adan.

- Queues are either finite or infinite virtual (K_0, K_∞).
- Infinite virtual queues provide all inputs.
- Processing rate parameters:

$$\cancel{\left(\overline{\alpha} \quad \overline{\mu}\right)} \Longrightarrow \left(\overline{u} \quad \overline{\mu}\right)$$

- Network dynamics:

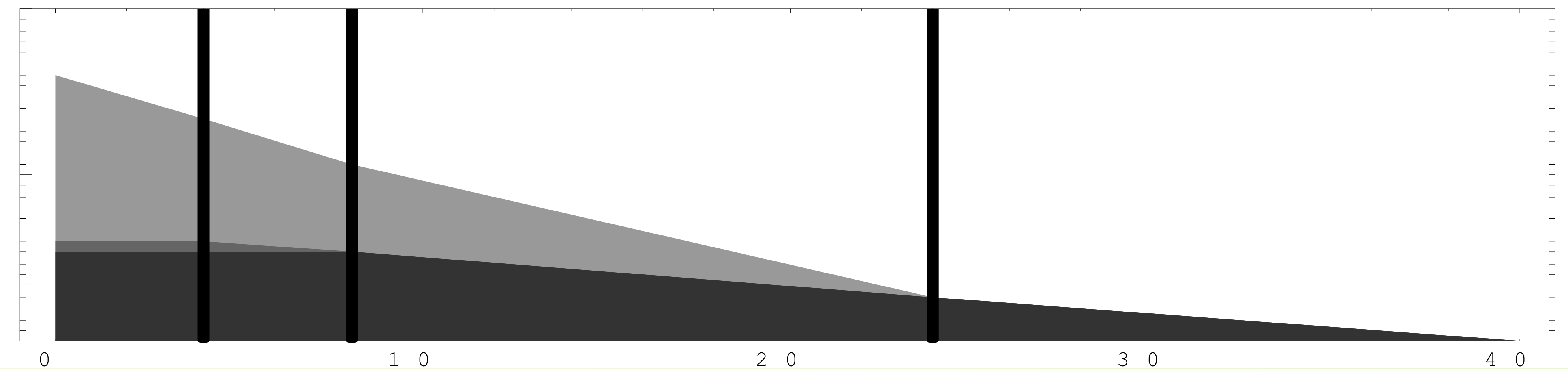
$$Z_k(t) = \begin{cases} Q_k(t) = Q_k(0) - S_k(T_k(t)) + \sum_{l \neq k} \Phi_{lk}(S_l(T_l(t)) \geq 0 & k \in K_0 \\ V_k(t) = V_k(0) + u_k t - S_k(T_k(t)) & k \in K_\infty \end{cases}$$

- Resource allocation:

$$T_k(0) = 0,\ \ T_k(t) \nearrow$$

$$\sum_{k \in C_i} (T_k(t) - T_k(s)) \leq t - s$$

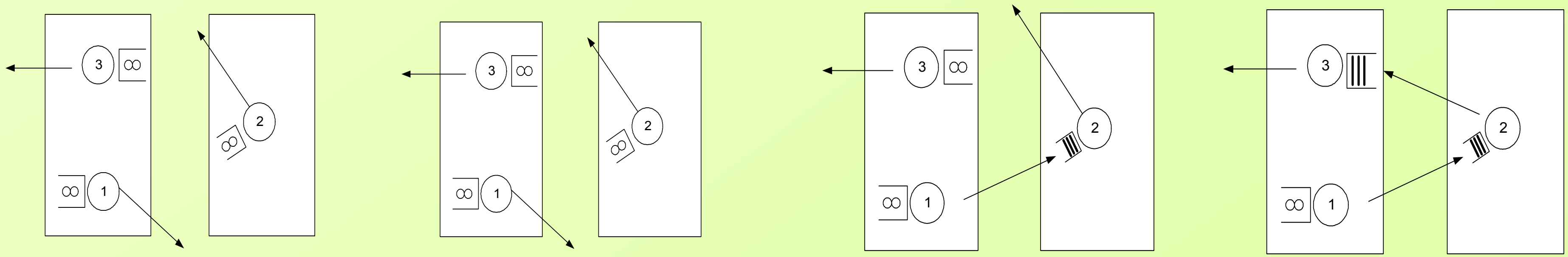
Back to the time intervals of the fluid solution…



$$K_0^n = \{\} \quad \{\} \quad \{2\} \quad \{2,3\}$$

$$K_\infty^n = \{1,2,3\} \quad \{1,2,3\} \quad \{1,3\} \quad \{1\}$$

For each time interval, set a MCQN with Infinite Virtual Queues.



$$u = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 1/4 & 1 \end{pmatrix} \quad \begin{pmatrix} 1/4 & - & 3/4 \end{pmatrix} \quad \begin{pmatrix} 1/4 & - & - \end{pmatrix}$$

- Control the queueing network of each time interval, keeping Z small.
- For some resources
  in some time intervals $\rho = 1$.
- Near optimal control for each time interval yields efficient tracking of the whole fluid solution.
- The objective function value is close to the desired optimum.

“Near optimal” in the asymptotic sense…

Keep the time horizon fixed, scale by N:

$$Q_k^{\;N}(0) = NQ_k(0)$$

$$\mu_k^{\;N} = N\mu_k$$

(Alternatively, keep rates fixed and increase time horizon.)

The goal is to achieve:

$$Q_k^{\;N}(t) - Nq_k(t) = O(\sqrt{N})$$

Thus the larger N, the smaller the relative error:

$$\lim_{N\to\infty} \frac{Q_k^{\;N}(t) - Nq_k(t)}{Nq_k(t)} = 0$$

For large N, “Near optimality” of the objective function follows:

$$\lim_{N\to\infty} \int_0^T \sum_{k=1}^{3} Q_k^{\;N}(t)dt = opt^*$$

Maximum Pressure Policies (Tassiulas, Stolyar, Dai & Lin)

<u>Input-output matrix R</u>

- R_{ij} is the average depletion of buffer i per one unit of work on buffer j.
- $R_{ii} = \mu_i$
- Treating Z and T as fluid we obtain: $\dot{Z}(t) = -R\,\dot{T}(t)$

<u>Allocations</u>

- Values of $\dot{T}(t)$
- At any given time E is the set of available allocations.
- $0 \in E$ so there is always some allocation.

<u>"Energy" minimization</u>

- Lyapunov function: $f(t) = Z(t) \cdot Z(t)$
- Find allocation that reduces it as fast as possible: $\frac{d}{dt} f(t) = 2Z(t) \cdot \dot{Z}(t) = -2Z(t) \cdot R\,\dot{T}(t)$

<u>The resulting policy:</u>

Choose $$\arg\max_{a \in \mathrm{E}} Z(t)^T R \cdot a$$

Path-wise stability when $\rho \le 1$

- Path-wise stability: $$\lim_{t\to\infty}\frac{Z(t)}{t}=0$$

- Theorem (Dai and Lin 2005):

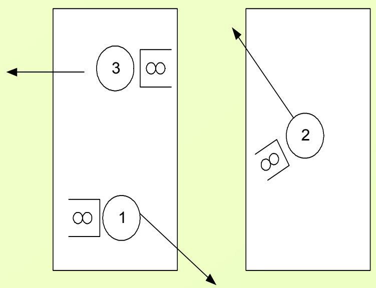
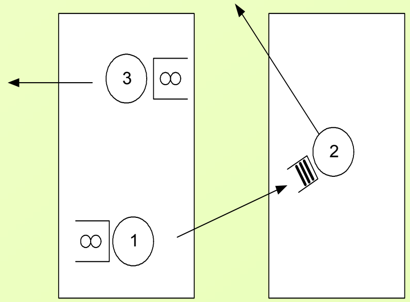
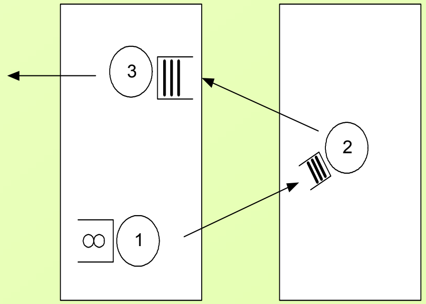
  For a broad class of models (stochastic processing networks), Maximum Pressure policy is path-wise stable if:

  - EAA assumption is satisfied.
  - $\rho \le 1$.

- Any MCQN satisfies EAA (Extreme allocation available).

- Same holds for MCQNs with Infinite Virtual Queues.

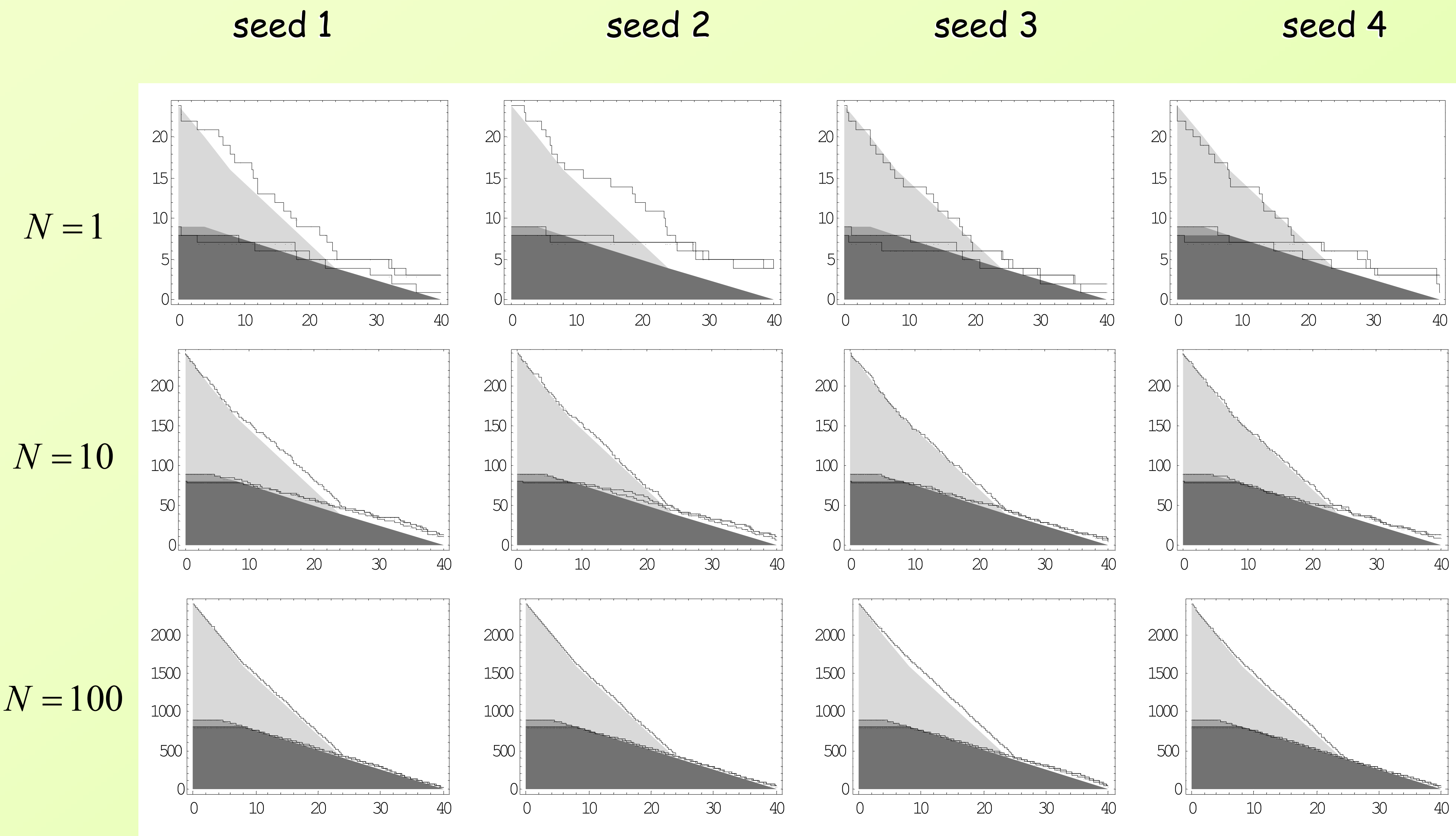
Input-Output matrixes when Infinite Virtual Queues exist

Time intervals 1 and 2	Time interval 3	Time interval 4
		
$R^1 = R^2 = \begin{pmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{pmatrix}$	$R^3 = \begin{pmatrix} \mu_1 & 0 & 0 \\ -\mu_1 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{pmatrix}$	$R^4 = \begin{pmatrix} \mu_1 & 0 & 0 \\ -\mu_1 & \mu_2 & 0 \\ 0 & -\mu_3 & \mu_3 \end{pmatrix}$

Pressure for buffer k based on k and k+1

current \ downstream	$k+1 \in K_0$	$k+1 \in K_\infty$
$k \in K_0$	$\mu_k(Q_k - Q_{k+1})$	$\mu_k Q_k$
$k \in K_\infty$	$\mu_k(u_k t - D_k - Q_{k+1})$	$\mu_k(u_k t - D_k)$

Example realizations, N={1,10,100}



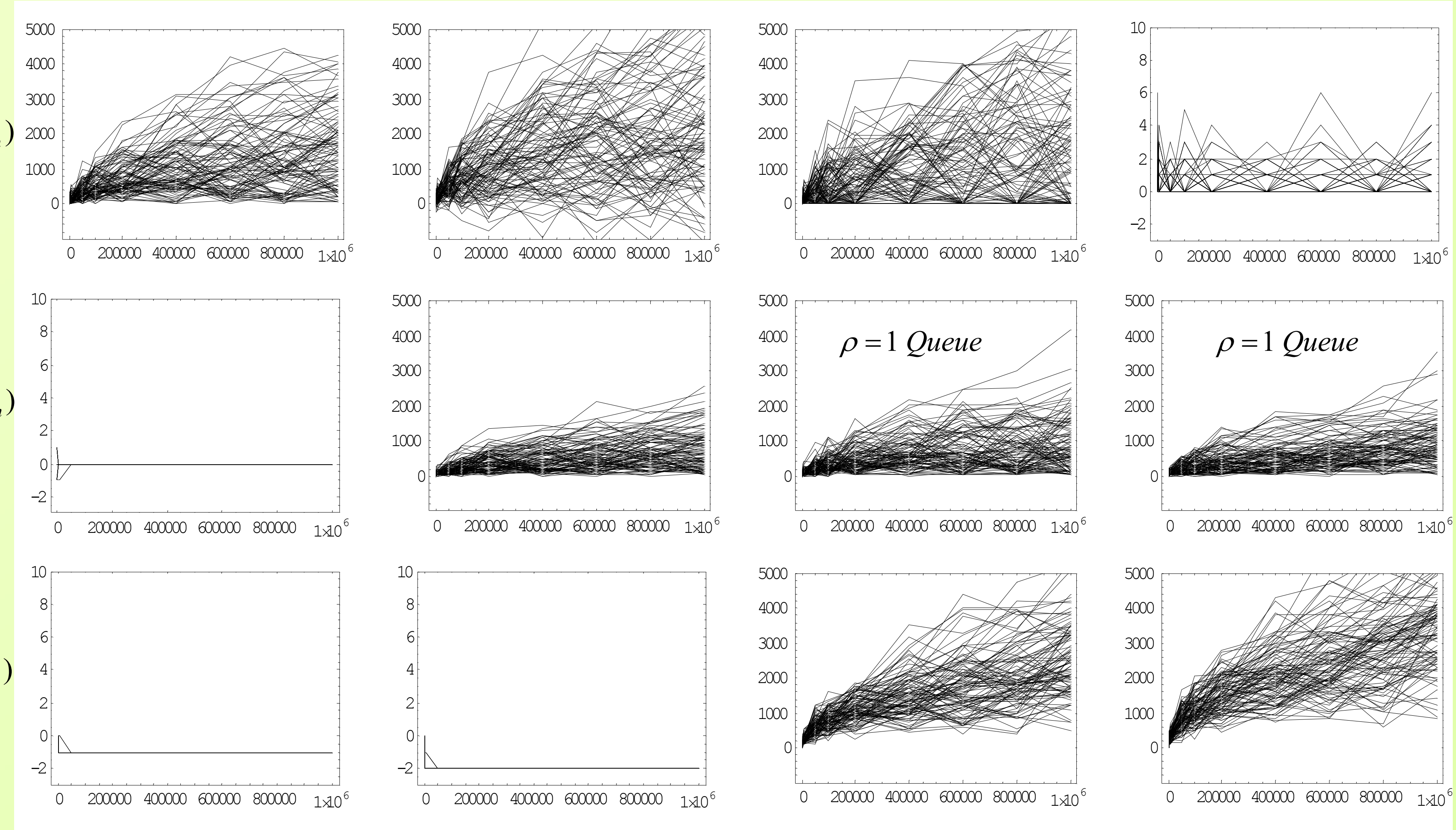
Empirical Asymptotics N = 1 to 10^6

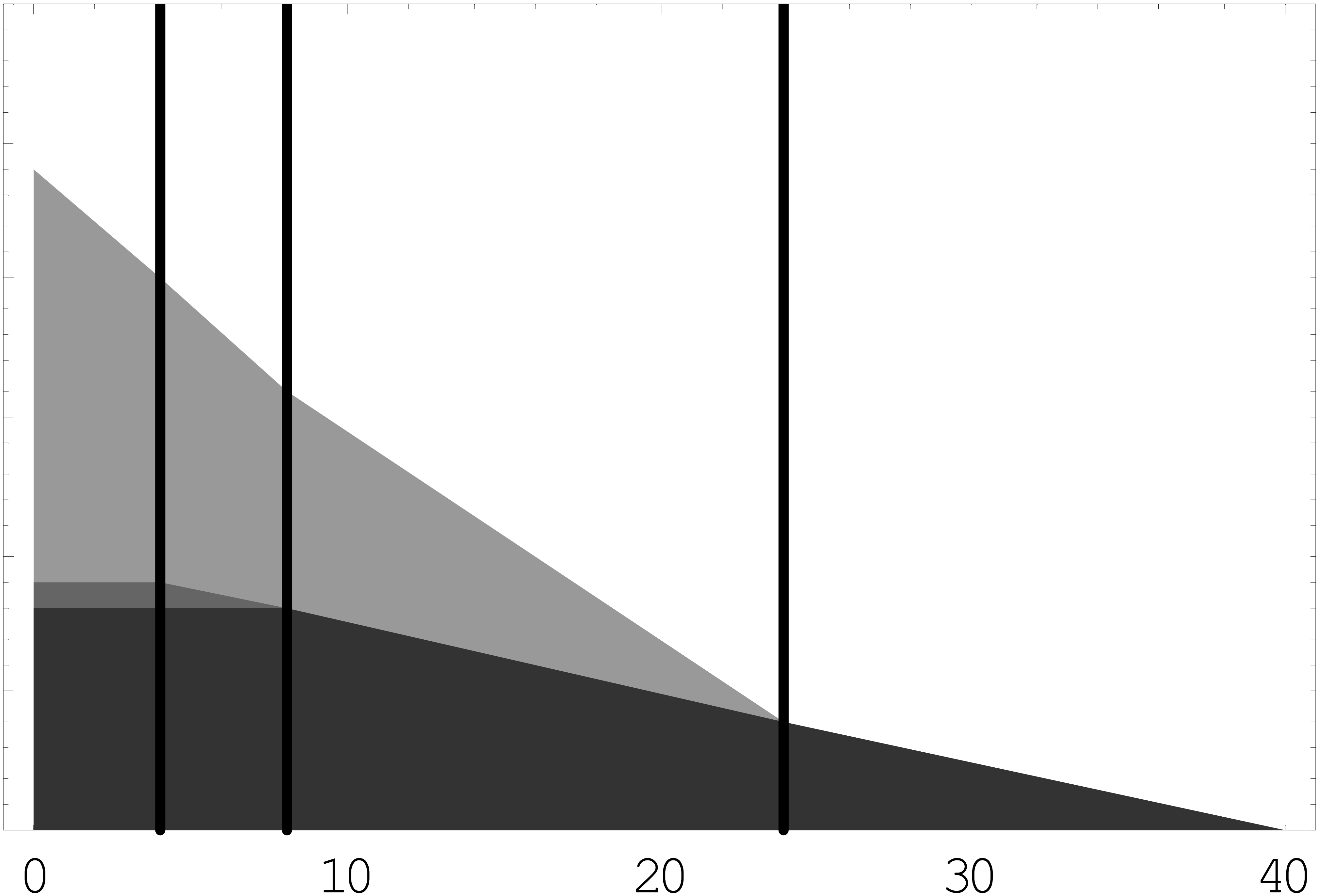
$K_0^{\ 1} = \{\}$	$K_0^{\ 1} = \{\}$	$K_0^{\ 1} = \{2\}$	$K_0^{\ 1} = \{2,3\}$
$K_\infty^{\ 1} = \{1,2,3\}$	$K_\infty^{\ 1} = \{1,2,3\}$	$K_\infty^{\ 1} = \{1,3\}$	$K_\infty^{\ 1} = \{1\}$
$u = (1,0,0)$	$u = (1,0.25,0)$	$u = (0.25,0.25,0.75)$	$u = (0.25,0.25,0.25)$

$Q_3^{\ N}(\tau_n) - Nq_3(\tau_n)$

$Q_2^{\ N}(\tau_n) - Nq_2(\tau_n)$

$Q_1^{\ N}(\tau_n) - Nq_1(\tau_n)$




0
10
20
30
40

Summary

- An efficient way to allocate resources over a network when optimizing over a finite time horizon.
- Solve SCLP optimization problems in finite time.
- An application of the fact that Maximum Pressure Policies exhibit path-wise stability even when $\rho=1$.
- MCQNs with infinite virtual queues extend the modeling capabilities of MCQNs.