Optimization of Buffers, Service Rates, & Population in Closed Finite Queueing Networks

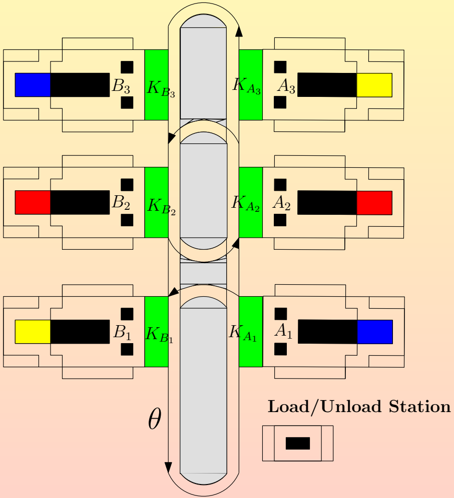
J. MacGregor Smith

Department of Mechanical and Industrial Engineering, Amherst MA, 01002, USA

June 1, 2017

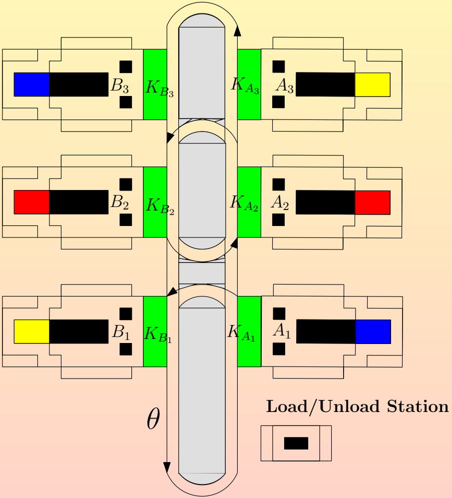
Outline of Lecture Topics

- *A.* Motivation
- *B.* Background
- *C.* Literature Review
- *D.* Optimization Models
- *E.* Performance & Optimization Algorithms
- *F.* Experimental Results
- *G.* Summary & Conclusions

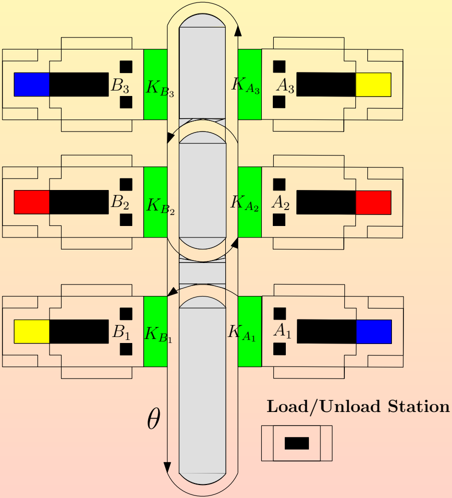


Outline of Lecture Topics

- *A.* Motivation
- *B.* Background
- *C.* Literature Review
- *D.* Optimization Models
- *E.* Performance & Optimization Algorithms
- *F.* Experimental Results
- *G.* Summary & Conclusions

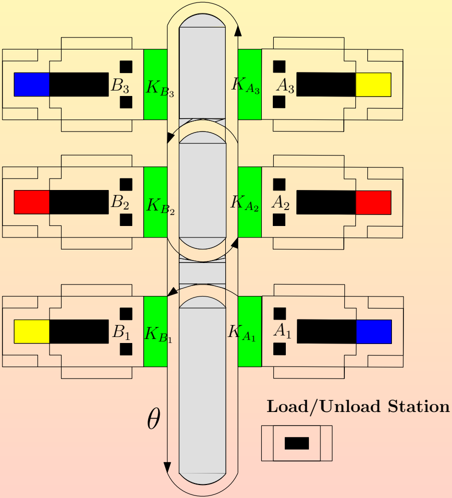


- *A.* Motivation
- *B.* Background
- *C.* Literature Review
- *D.* Optimization Models
- *E.* Performance & Optimization Algorithms
- *F.* Experimental Results
- *G.* Summary & Conclusions



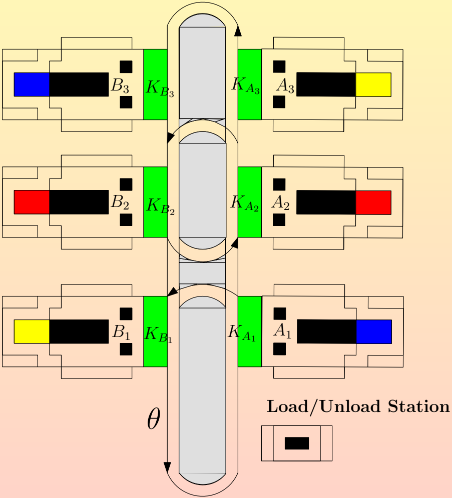
Outline of Lecture Topics

- *A.* Motivation
- *B.* Background
- *C.* Literature Review
- *D.* Optimization Models
- *E.* Performance & Optimization Algorithms
- *F.* Experimental Results
- *G.* Summary & Conclusions



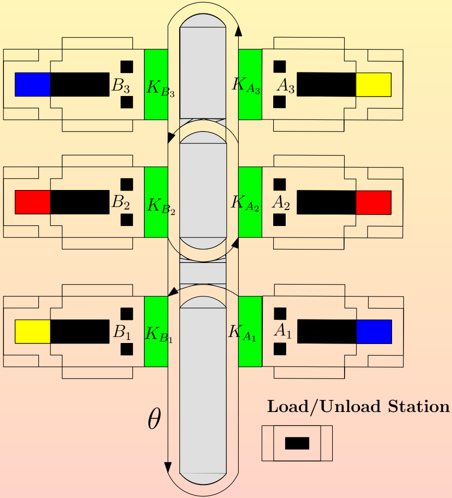
Outline of Lecture Topics

- *A.* Motivation
- *B.* Background
- *C.* Literature Review
- *D.* Optimization Models
- *E.* Performance & Optimization Algorithms
- *F.* Experimental Results
- *G.* Summary & Conclusions



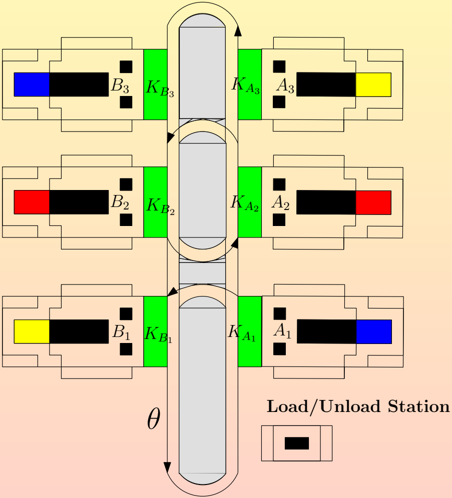
Outline of Lecture Topics

- *A.* Motivation
- *B.* Background
- *C.* Literature Review
- *D.* Optimization Models
- *E.* Performance & Optimization Algorithms
- *F.* Experimental Results
- *G.* Summary & Conclusions



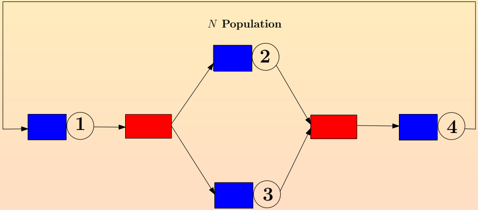
Outline of Lecture Topics

- *A.* Motivation
- *B.* Background
- *C.* Literature Review
- *D.* Optimization Models
- *E.* Performance & Optimization Algorithms
- *F.* Experimental Results
- *G.* Summary & Conclusions



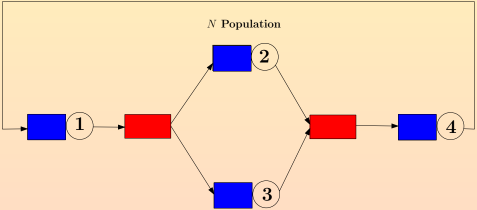
Background

- Assumptions
  - Unpaced, asynchronous Flow line or FMS
  - Finite Buffers & Production Blocking
  - Closed Network Models, finite population, single-servers
- Approximate Mean Value Analysis (MVA) Model
  - Two-Moment General Service Time Distributions
  - Two-Moment Blocking Probability
- Integrated Material Handling System



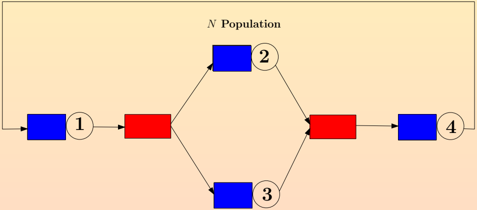
Background

- Assumptions
  - Unpaced, asynchronous Flow line or FMS
  - Finite Buffers & Production Blocking
  - Closed Network Models, finite population, single-servers
- Approximate Mean Value Analysis (MVA) Model
  - Two-Moment General Service Time Distributions
  - Two-Moment Blocking Probability
- Integrated Material Handling System



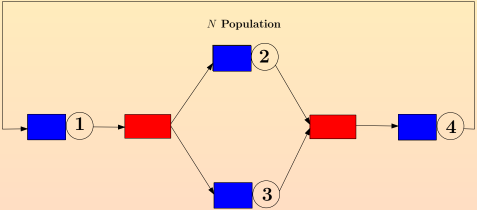
Background

- Assumptions
  - Unpaced, asynchronous Flow line or FMS
  - Finite Buffers & Production Blocking
  - Closed Network Models, finite population, single-servers
- Approximate Mean Value Analysis (MVA) Model
  - Two-Moment General Service Time Distributions
  - Two-Moment Blocking Probability
- Integrated Material Handling System



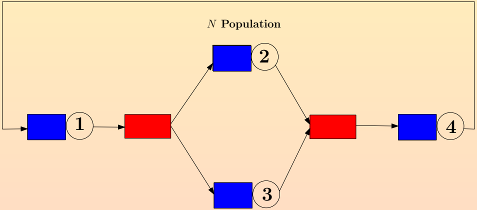
Background

- Assumptions
  - Unpaced, asynchronous Flow line or FMS
  - Finite Buffers & Production Blocking
  - Closed Network Models, finite population, single-servers
- Approximate Mean Value Analysis (MVA) Model
  - Two-Moment General Service Time Distributions
  - Two-Moment Blocking Probability
- Integrated Material Handling System



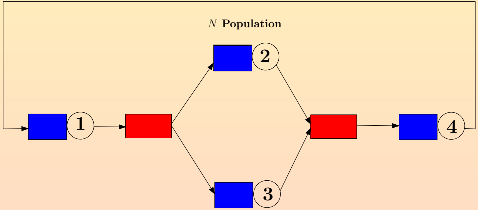
Background

- Assumptions
  - Unpaced, asynchronous Flow line or FMS
  - Finite Buffers & Production Blocking
  - Closed Network Models, finite population, single-servers
- Approximate Mean Value Analysis (MVA) Model
  - Two-Moment General Service Time Distributions
  - Two-Moment Blocking Probability
- Integrated Material Handling System



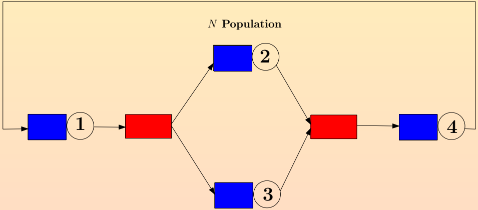
Background

- Assumptions
  - Unpaced, asynchronous Flow line or FMS
  - Finite Buffers & Production Blocking
  - Closed Network Models, finite population, single-servers
- Approximate Mean Value Analysis (MVA) Model
  - Two-Moment General Service Time Distributions
  - Two-Moment Blocking Probability
- Integrated Material Handling System



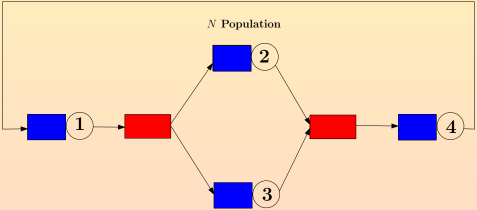
Background

- Assumptions
  - Unpaced, asynchronous Flow line or FMS
  - Finite Buffers & Production Blocking
  - Closed Network Models, finite population, single-servers
- Approximate Mean Value Analysis (MVA) Model
  - Two-Moment General Service Time Distributions
  - Two-Moment Blocking Probability
- Integrated Material Handling System



Background

- Assumptions
  - Unpaced, asynchronous Flow line or FMS
  - Finite Buffers & Production Blocking
  - Closed Network Models, finite population, single-servers
- Approximate Mean Value Analysis (MVA) Model
  - Two-Moment General Service Time Distributions
  - Two-Moment Blocking Probability
- Integrated Material Handling System



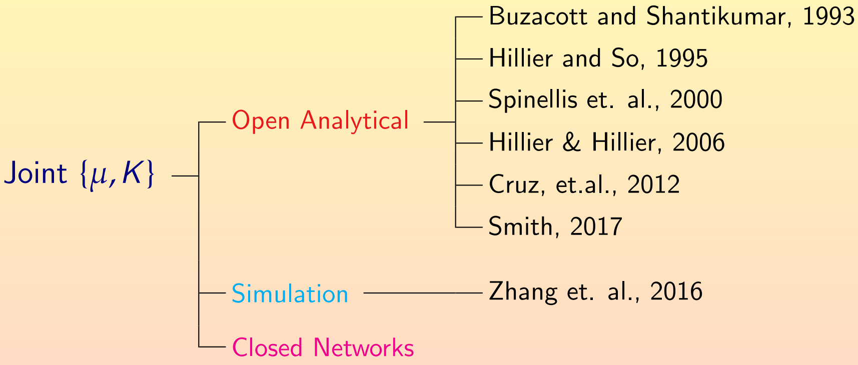
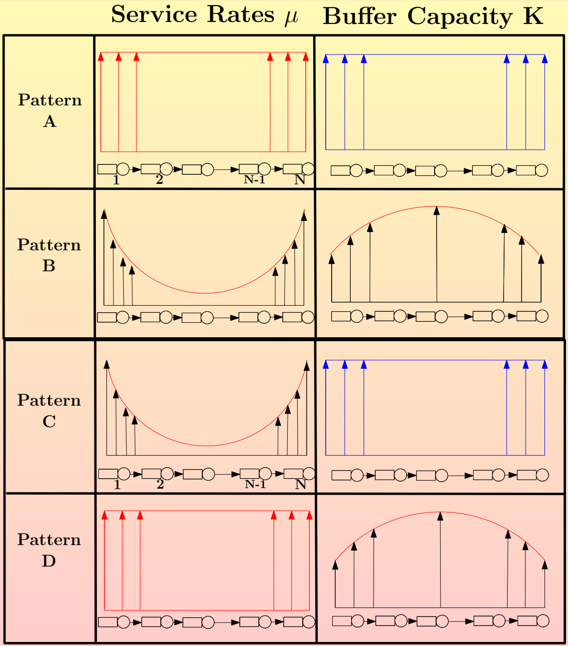


Figure: Simultaneous Optimization Literature Morphology

Simultaneous Optimization Problem Methodology



Simultaneous Optimization Problem Methodology

Basic Issues:

- How can we develop a closed network approximation for generally distributed finite blocking processes?
- How can we account for blocking from General distributions?
- Can we create an efficient running time performance and optimization algorithm?
- What will be the service rate and buffer allocation patterns for series, merge, and split topologies.

Simultaneous Optimization Problem Methodology

Basic Issues:

- How can we develop a closed network approximation for generally distributed finite blocking processes?
- How can we account for blocking from General distributions?
- Can we create an efficient running time performance and optimization algorithm?
- What will be the service rate and buffer allocation patterns for series, merge, and split topologies.

Simultaneous Optimization Problem Methodology

Basic Issues:

- How can we develop a closed network approximation for generally distributed finite blocking processes?
- How can we account for blocking from General distributions?
- Can we create an efficient running time performance and optimization algorithm?
- What will be the service rate and buffer allocation patterns for series, merge, and split topologies.

Simultaneous Optimization Problem Methodology

Basic Issues:

- How can we develop a closed network approximation for generally distributed finite blocking processes?
- How can we account for blocking from General distributions?
- Can we create an efficient running time performance and optimization algorithm?
- What will be the service rate and buffer allocation patterns for series, merge, and split topologies.

Optimization Formulation

Primal : Maximize $\theta(\mathsf{K}, \mu, \mathsf{N})$

s.t.:

$$\sum_{j}^{m} b_j \mu_j = m$$

$$\sum_{j}^{m} d_j K_j \leq D$$

$$N \leq \frac{\left\lfloor \sum_j K_j + m \right\rfloor}{2}$$

$$K_j \leq L_q^j \ \forall j$$

$$\mu_j^\ell > 0$$

$$K_j \geq 1 \ \forall \ j$$

Dual : Minimize $\sum_{j} d_j K_j$

s.t.:

$$\theta \geq \theta^{\min}$$

$$\sum_{j}^{m} b_j \mu_j = m$$

$$N \leq \frac{\left\lfloor \sum_j K_j + m \right\rfloor}{2}$$

$$K_j \leq L_q^j \ \forall j$$

$$\mu_j^\ell > 0$$

$$K_j \geq 1 \ \forall \ j$$

Optimization Formulation

Primal : Maximize $\theta(\mathrm{K}, \mu, \mathrm{N})$

s.t.:

$$\sum_{j}^{m} b_j \mu_j = m$$

$$\sum_{j}^{m} d_j K_j \leq D$$

$$N \leq \frac{\left\lceil \sum_j K_j + m \right\rceil}{2}$$

$$K_j \leq L_q^j \ \forall j$$

$$\mu_j^\ell > 0$$

$$K_j \geq 1 \ \forall \ j$$

Dual : Minimize $\sum_{\mathrm{j}} \mathrm{d_j K_j}$

s.t.:

$$\theta \geq \theta^{\ \mathrm{min}}$$

$$\sum_{j}^{m} b_j \mu_j = m$$

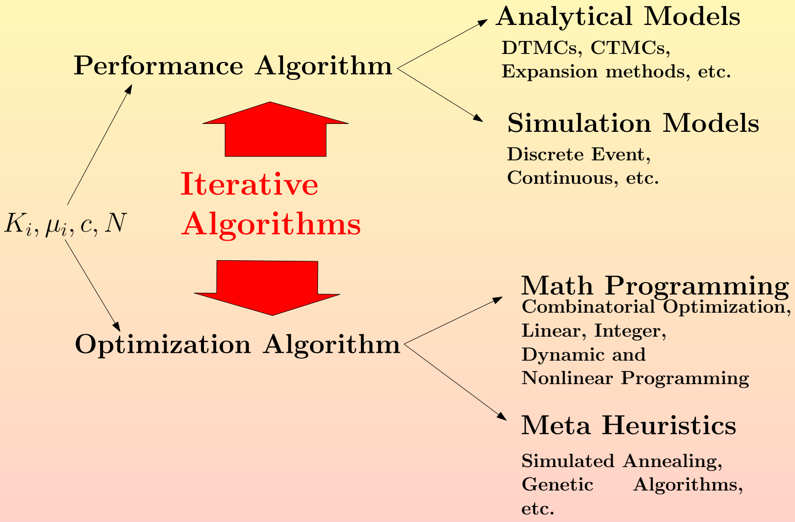
$$N \leq \frac{\left\lceil \sum_j K_j + m \right\rceil}{2}$$

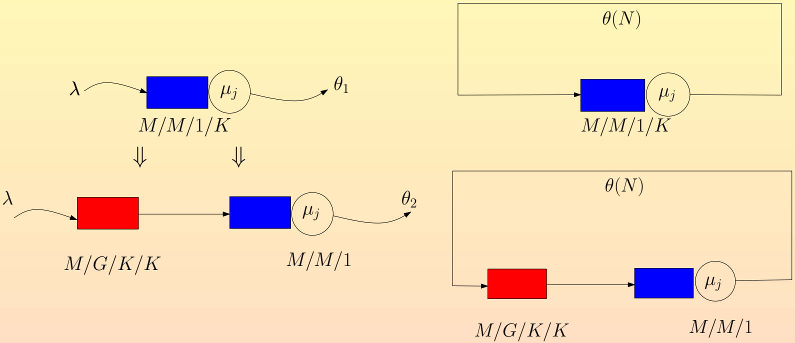
$$K_j \leq L_q^j \ \forall j$$

$$\mu_j^\ell > 0$$

$$K_j \geq 1 \ \forall \ j$$

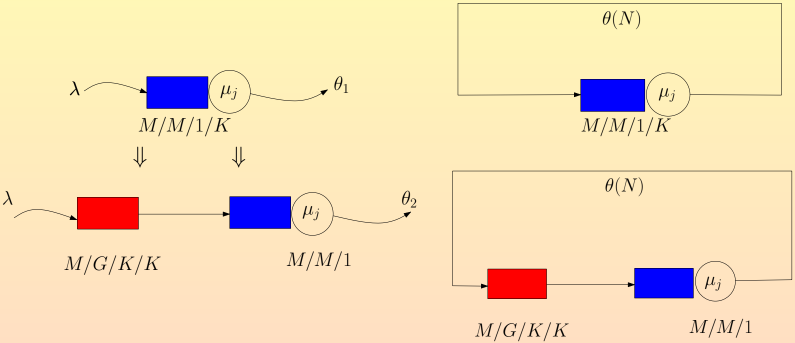
Iterative Performance and Optimization Algorithm





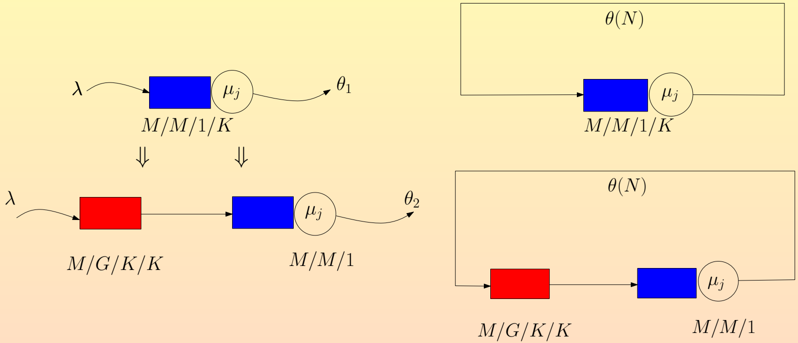
- Underlying logic behind Queue Decomposition idea:
  - $M/G/K/K$ queue acts as a holding node for the parts.
  - As the population increases, the congestion (blocking) increases as a function of the # of parts within the system.
  - Effective service rates decay as a function of the blocking in the system.

Performance Mathematical Models



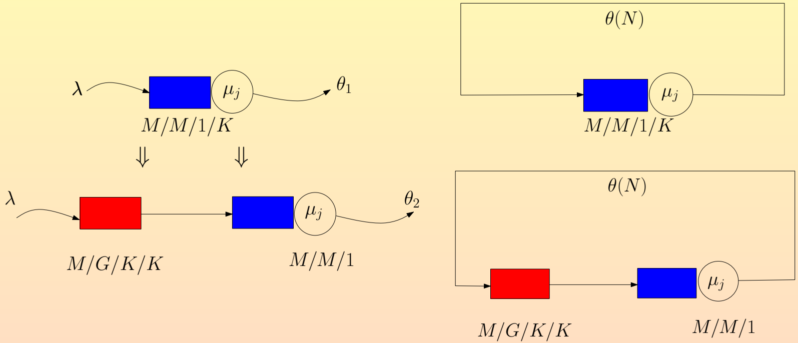
- Underlying logic behind Queue Decomposition idea:
  - *M/G/K/K* queue acts as a holding node for the parts.
  - As the population increases, the congestion (blocking) increases as a function of the # of parts within the system.
  - Effective service rates decay as a function of the blocking in the system.

Performance Mathematical Models



- Underlying logic behind Queue Decomposition idea:
  - *M/G/K/K* queue acts as a holding node for the parts.
  - As the population increases, the congestion (blocking) increases as a function of the # of parts within the system.
  - Effective service rates decay as a function of the blocking in the system.

Performance Mathematical Models



- Underlying logic behind Queue Decomposition idea:
  - $M/G/K/K$ queue acts as a holding node for the parts.
  - As the population increases, the congestion (blocking) increases as a function of the # of parts within the system.
  - Effective service rates decay as a function of the blocking in the system.

Queue Decomposition Algorithm

- Step 1.0: Add a pair of nodes $M/G/K/K$ and $M/M/1$ for each finite buffer queue. Estimate System population.

$$N^{*} \leq \frac{\left\lceil \sum_j K_j + m \right\rceil}{2}$$

- Step 2.0: Adjust the free-flow speed and state dependent service rate.

$$V_1(\ell) = V_1(\ell)(1 - p_K(\ell + 1) \quad (1)$$

$$\mu_n = n \frac{V_1}{\mathcal{L}} \exp\left[-\left(\frac{n-1)}{\beta} \right)^{\gamma} \right] \quad (2)$$

- Step 3.0: Calculate the fundamental output measures of residence time $w_\ell(N)$, throughput $\theta_\ell(N)$, and work-in-process n_ℓ from the Mean Value Analysis algorithm.

Queue Decomposition Algorithm

- Step 1.0: Add a pair of nodes $M/G/K/K$ and $M/M/1$ for each finite buffer queue. Estimate System population.

$$N^* \leq \frac{\left\lceil \sum_j K_j + m \right\rceil}{2}$$

- Step 2.0: Adjust the free-flow speed and state dependent service rate.

$$V_1(\ell) = V_1(\ell)(1 - p_K(\ell + 1) \quad (1)$$

$$\mu_n = n \frac{V_1}{\mathcal{L}} \exp\left[-\left(\frac{n-1)}{\beta} \right)^{\gamma} \right] \quad (2)$$

- Step 3.0: Calculate the fundamental output measures of residence time $w_\ell(N)$, throughput $\theta_\ell(N)$, and work-in-process n_ℓ from the Mean Value Analysis algorithm.

Queue Decomposition Algorithm

- Step 1.0: Add a pair of nodes $M/G/K/K$ and $M/M/1$ for each finite buffer queue. Estimate System population.

$$N^* \leq \frac{\left[\sum_j K_j + m\right]}{2}$$

- Step 2.0: Adjust the free-flow speed and state dependent service rate.

$$V_1(\ell) = V_1(\ell)(1 - p_K(\ell + 1) \tag{1}$$

$$\mu_n = n\frac{V_1}{\mathcal{L}} \exp\left[-\left(\frac{n-1)}{\beta}\right)^{\gamma}\right] \tag{2}$$

- Step 3.0: Calculate the fundamental output measures of residence time $w_\ell(N)$, throughput $\theta_\ell(N)$, and work-in-process n_ℓ from the Mean Value Analysis algorithm.

Blocking Probability (Two moment estimation)

If one fixes the number of servers, one can solve for the blocking probability of the $M/M/1/K$ system.

$$p_K = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}} \Rightarrow K = \left\lceil \frac{\ln(p_K/(1-\rho+p_K\rho))}{\ln(\rho)} \right\rceil \tag{3}$$

$$B = \frac{\left(\ln\left(\frac{p_K}{1-\rho+p_K\rho}\right) - \ln(\rho)\right)\left(2 + \sqrt{\frac{\rho}{e^{s^2}}}s^2 - \sqrt{\frac{\rho}{e^{s^2}}}\right)}{2\ln(\rho)} \tag{4}$$

In the case of $c = 1$, the following expression is obtained for the blocking probability:

$$p_K = \frac{\rho^{\frac{\sqrt{\rho}s^2 - \sqrt{\rho} + 2K}{2+\sqrt{\rho}s^2 - \sqrt{\rho}}}(\rho - 1)}{\left(\rho^{2\frac{1+\sqrt{\rho}s^2-\sqrt{\rho}+K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}} - 1\right)} \tag{5}$$

Blocking Probability (Two moment estimation)

If one fixes the number of servers, one can solve for the blocking probability of the $M/M/1/K$ system.

$$p_K = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}} \Rightarrow K = \left\lceil \frac{\ln(p_K/(1-\rho+p_K\rho))}{\ln(\rho)} \right\rceil \quad (3)$$

$$B = \frac{\left(\ln\left(\frac{p_K}{1-\rho+p_K\rho}\right) - \ln(\rho)\right)\left(2+\sqrt{\frac{\rho}{e^{s^2}}}s^2 - \sqrt{\frac{\rho}{e^{s^2}}}\right)}{2\ln(\rho)} \quad (4)$$

In the case of $c=1$, the following expression is obtained for the blocking probability:

$$p_K = \frac{\rho^{\frac{\sqrt{\rho}s^2-\sqrt{\rho}+2K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}}\left(\rho-1\right)}{\left(\rho^{2\frac{1+\sqrt{\rho}s^2-\sqrt{\rho}+K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}}-1\right)} \quad (5)$$

Blocking Probability (Two moment estimation)

If one fixes the number of servers, one can solve for the blocking probability of the $M/M/1/K$ system.

$$\mathrm{p_K} = \frac{(1-\rho)\rho^{\mathrm{K}}}{1-\rho^{\mathrm{K}+1}} \Rightarrow \mathrm{K} = \left\lceil \frac{\ln(p_K/(1-\rho+p_K\rho))}{\ln(\rho)} \right\rceil \tag{3}$$

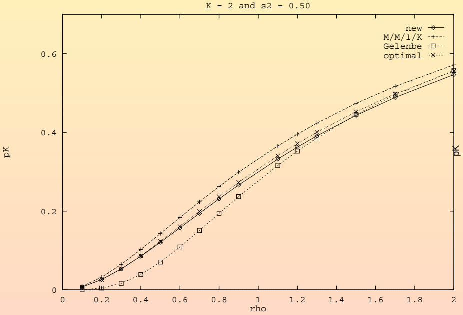
$$B = \frac{\left(\ln\left(\frac{p_K}{1-\rho+p_K\rho}\right) - \ln(\rho)\right)\left(2+\sqrt{\frac{\rho}{e^{s^2}}}s^2 - \sqrt{\frac{\rho}{e^{s^2}}}\right)}{2\ln(\rho)} \tag{4}$$

In the case of $c=1$, the following expression is obtained for the blocking probability:

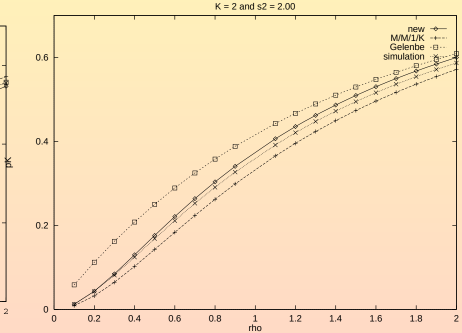
$$p_K = \frac{\rho^{\frac{\sqrt{\rho}s^2-\sqrt{\rho}+2K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}}\left(\rho-1\right)}{\left(\rho^{2\frac{1+\sqrt{\rho}s^2-\sqrt{\rho}+K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}}-1\right)} \tag{5}$$

Blocking Probability (Two moment estimation)

P_K Comparisons M/G/1/2 $s^2 = \frac{1}{2}$

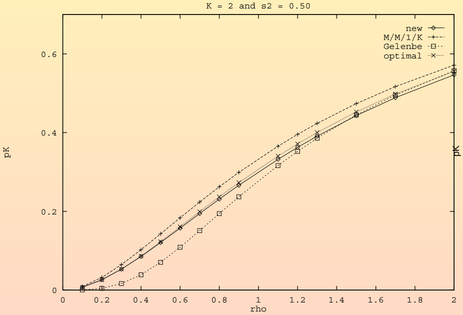


P_K Comparisons M/G/1/2 $s^2 = 2$

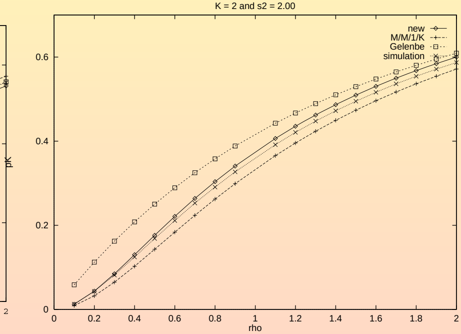


Blocking Probability (Two moment estimation)

P_K Comparisons M/G/1/2 $s^2 = \frac{1}{2}$



P_K Comparisons M/G/1/2 $s^2 = 2$



General Service Time Approximation

The standard Equation 6 in the MVA for the expected delay time at a queue is based upon the PASTA property that

$$w_\ell(N) = \tau_\ell[1 + n_\ell(N-1)] \tag{6}$$

Accounting for the remaining service time which is a function of the utilization of the queue, the full service time of the number of customers in the queue, and the full service time of the arriving customer:

$$w_\ell(N) = \rho_\ell(N-1)\frac{\tau_\ell(1+s^2)}{2} + (n_\ell(N-1) - \rho_\ell(N-1))\tau_\ell + \tau_\ell \tag{7}$$

General Service Time Approximation

The standard Equation 6 in the MVA for the expected delay time at a queue is based upon the PASTA property that

$$w_\ell(N) = \tau_\ell[1 + n_\ell(N-1)] \tag{6}$$

Accounting for the remaining service time which is a function of the utilization of the queue, the full service time of the number of customers in the queue, and the full service time of the arriving customer:

$$w_\ell(N) = \rho_\ell(N-1)\frac{\tau_\ell(1+s^2)}{2} + (n_\ell(N-1) - \rho_\ell(N-1))\tau_\ell + \tau_\ell \tag{7}$$

Mean Value Analysis (MVA) Algorithm

- Reiser and Lavenberg's modified property of product-form networks to estimate the delay or residence time at the queue:

$$w_\ell(N) = \rho_\ell(N-1)\frac{\tau_\ell(1+s^2)}{2} + (n_\ell(N-1) - \rho_\ell(N-1))\tau_\ell + \tau_\ell \qquad (8)$$

- Little's equation for product chains:

$$\lambda_\ell(N) = \frac{N}{[\sum_{\ell=1}^{m} w_\ell(N)\alpha_\ell]} \qquad (9)$$

- Little's equation for queues:

$$n_\ell(N) = \lambda_\ell(N) w_\ell(N) \qquad (10)$$

Mean Value Analysis (MVA) Algorithm

- Reiser and Lavenberg's modified property of product-form networks to estimate the delay or residence time at the queue:

$$w_\ell(N) = \rho_\ell(N-1)\frac{\tau_\ell(1+s^2)}{2} + (n_\ell(N-1) - \rho_\ell(N-1))\tau_\ell + \tau_\ell \qquad (8)$$

- Little's equation for product chains:

$$\lambda_\ell(N) = \frac{N}{[\sum_{\ell=1}^{m} w_\ell(N)\alpha_\ell]} \qquad (9)$$

- Little's equation for queues:

$$n_\ell(N) = \lambda_\ell(N) w_\ell(N) \qquad (10)$$

Mean Value Analysis (MVA) Algorithm

- Reiser and Lavenberg's modified property of product-form networks to estimate the delay or residence time at the queue:

$$w_\ell(N) = \rho_\ell(N-1)\frac{\tau_\ell(1+s^2)}{2} + (n_\ell(N-1) - \rho_\ell(N-1))\tau_\ell + \tau_\ell \tag{8}$$

- Little's equation for product chains:

$$\lambda_\ell(N) = \frac{N}{[\sum_{\ell=1}^{m} w_\ell(N)\alpha_\ell]} \tag{9}$$

- Little's equation for queues:

$$n_\ell(N) = \lambda_\ell(N) w_\ell(N) \tag{10}$$

Sequential Quadratic Programming Problem

$$\text{QPP} : \text{Minimize } f(\mathrm{x}_\ell) = \nabla f(\mathrm{x}_\ell)^t \mathrm{p} + \frac{1}{2}\mathrm{p}^t \mathrm{H}(\mathrm{x}_\ell)\mathrm{p}$$

$$\text{subject to} : \mathrm{g}_i(\mathrm{x}_\ell) + \nabla \mathrm{g}_i(\mathrm{x}_\ell)^t \mathrm{p} \leq 0 \; \forall \ell \in \mathcal{M}$$

where for the network with a given population N:

- n_ℓ := is the expected length of queue ℓ,
- λ_ℓ := is the throughput products at queue ℓ,
- w_ℓ := is the expected delay products at queue ℓ,
- x_ℓ := is the decision vector which is a function of μ_ℓ, K_ℓ, N
- ρ_ℓ := utilization rates of each queue,
- p := is a direction vector,
- $\mathcal{M}$:= is the set of inequalities described in (1)-(6) or (7) through (12)

Optimization Integrated MVA Algorithm

Step 1.0 Given a starting solution $x = (\mu_\ell, K_\ell, N)$, formulate:

$$\text{SQP}(x_\ell)$$

Step 2.0 Solve $\text{SQP}(x_\ell)$ by calculating:

Step 2.1 Average delay at each queue

$$w_\ell(N) = \rho_\ell(N-1)\frac{\tau_\ell(1+s^2)}{2} + (n_\ell(N-1) - \rho_\ell(N-1))\tau_\ell + \tau_\ell$$

Step 2.2 Average throughput at each queue

$$\lambda_\ell = \frac{N}{\sum_{\ell=1}^{N} w_\ell y_\ell}$$

Step 2.3 Average number at each queue

$$n_\ell = \lambda_\ell w_\ell$$

Step 3.0 After solving $\text{QPP}(x_\ell)$, set $x_{\ell+1} = x_\ell + p$

Step 4.0 Check for convergence ($\epsilon = 1.0x10^{-7}$)

$$\text{Set k } \leftarrow k+1 \text{ and repeat Step 2.0}$$

Optimization Integrated MVA Algorithm

Step 1.0 Given a starting solution $x = (\mu_\ell, K_\ell, N)$, formulate:

$$\mathrm{SQP}(x_\ell)$$

Step 2.0 Solve $\mathrm{SQP}(x_\ell)$ by calculating:

Step 2.1 Average delay at each queue

$$w_\ell(N) = \rho_\ell(N-1)\frac{\tau_\ell(1+s^2)}{2} + (n_\ell(N-1) - \rho_\ell(N-1))\tau_\ell + \tau_\ell$$

Step 2.2 Average throughput at each queue

$$\lambda_\ell = \frac{N}{\sum_{\ell=1}^{N} w_\ell y_\ell}$$

Step 2.3 Average number at each queue

$$n_\ell = \lambda_\ell w_\ell$$

Step 3.0 After solving $\mathrm{QPP}(x_\ell)$, set $x_{\ell+1} = x_\ell + \mathrm{p}$

Step 4.0 Check for convergence ($\epsilon = 1.0x10^{-7}$)

Set k $\leftarrow k+1$ and repeat Step 2.0

Optimization Integrated MVA Algorithm

Step 1.0 Given a starting solution $x = (\mu_\ell, K_\ell, N)$, formulate:

$$\mathrm{SQP}(x_\ell)$$

Step 2.0 Solve $\mathrm{SQP}(x_\ell)$ by calculating:

Step 2.1 Average delay at each queue

$$w_\ell(N) = \rho_\ell(N-1)\frac{\tau_\ell(1+s^2)}{2} + (n_\ell(N-1) - \rho_\ell(N-1))\tau_\ell + \tau_\ell$$

Step 2.2 Average throughput at each queue

$$\lambda_\ell = \frac{N}{\sum_{\ell=1}^{N} w_\ell y_\ell}$$

Step 2.3 Average number at each queue

$$n_\ell = \lambda_\ell w_\ell$$

Step 3.0 After solving $\mathrm{QPP}(x_\ell)$, set $x_{\ell+1} = x_\ell + \mathrm{p}$

Step 4.0 Check for convergence ($\epsilon = 1.0x10^{-7}$)

Set k $\leftarrow k+1$ and repeat Step 2.0

Series Comparison

Primal Problem							
D	m	μ_1, μ_2	K_1, K_2	N	θ	W	B&B
8	2	(1,1)	(4,3)	5	0.833	6.00	37
9	2	(1,1)	(5,4)	6	0.857	7.00	24
13	2	(.983,.983)	(7,6)	8	0.874	9.15	22

Dual Problem						
m	μ_1, μ_2	K_1, K_2	N	θ	W	B&B
2	(1,1)	(4,3)	5	0.833	6.00	8
2	(1,1)	(4,5)	6	0.857	7.00	54
2	(1,1)	(5,6)	7	0.875	8.00	71

Table 1. Two-stage Primal and Dual Comparison Experiments

Series Comparison

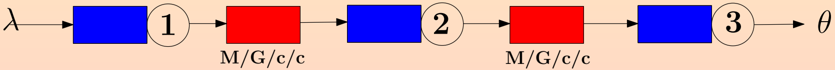
Primal Problem							
D	m	μ_1, μ_2	K_1, K_2	N	θ	W	B&B
8	2	(1,1)	(4,3)	5	0.833	6.00	37
9	2	(1,1)	(5,4)	6	0.857	7.00	24
13	2	(.983,.983)	(7,6)	8	0.874	9.15	22

Dual Problem						
m	μ_1, μ_2	K_1, K_2	N	θ	W	B&B
2	(1,1)	(4,3)	5	0.833	6.00	8
2	(1,1)	(4,5)	6	0.857	7.00	54
2	(1,1)	(5,6)	7	0.875	8.00	71

Table 1. Two-stage Primal and Dual Comparison Experiments

3-Stage Experiments

#	s^2	$\bar{\mu}^*$	$\bar{K}^*$	N^*	θ_α, θ_s	%	W_α, W_s	%	B&B
1	1	(1,1,1)	(6,7,7)	12	(0.8569,0.8431)	1.64	(14.003,14.233)	1.62	493
2	1	(1,1,1)	(6,8,8)	13	(0.8665,0.8514)	1.77	(15.004,15.269)	1.74	2310
3	1	(1,1,1)	(9,9,10)	16	(0.8887,0.8765)	1.39	(18.005,18.254)	1.36	408
4	1	(1,1,1)	(11,11,12)	19	(0.9045,0.8931)	1.28	(21.006,21.274)	1.26	433
5	1	(1.01,1.01,0.98)	(17,18,17)	28	(0.9307,0.9233)	0.80	(30.086,30.325)	0.79	386
6	1/4	(1,1,1)	(2,3,3)	6	(0.8466,0.8904)	4.92	(7.087,6.739)	5.16	86
7	1/2	(1,1,1)	(4,4,4)	8	(0.8532,0.8666)	1.55	(9.377,9.2318)	1.57	33
8	3/4	(1,1,1)	(6,5,5)	10	(0.8556,0.8554)	0.02	(11.687,11.690)	0.03	331
9	5/4	(1,1,1)	(8,8,6)	13	(0.8487,0.8614)	1.47	(15.3182,15.091)	1.51	190
10	3/2	(1,1,1)	(9,8,9)	15	(0.8505,0.8252)	3.07	(17.6362,18.177)	2.98	302
11	$\frac{1}{2}$,1, $\frac{1}{2}$	(1,1,1)	(4,5,5)	9	(0.8502,0.8533)	0.36	(10.586,10.547)	0.37	82



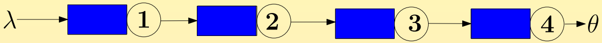
12	(1,1,1)	(1,1,1)	(7,7,6)	12	(0.8444,0.8240)	2.47	(14.212,14.562)	2.41	254
13	($\frac{1}{2}$,1, $\frac{1}{2}$)	(1,1,1)	(5,5,6)	10	(0.8510,0.8432)	0.93	(11.7503,11.859)	0.92	564

Table: Three-stage Experiments

SQP Optimization Experiment

```
--- FINAL CONVERGENCE ANALYSIS ---
    Objective function value:        F(X)  =  0.13333333D+01
    Approximation of solution:       X     =
   service rate -> 0.10000000D+01  0.10000000D+01  0.10000000D+01
         buffers-> 0.50000000D+01  0.50000000D+01  0.60000000D+01
      population-> 0.10000000D+02
    Constraint function values:      G(X)  =
         0.00000000D+00  0.14895618D+00  0.14895618D+00  0.14895618D+00
         0.50438204D-02  0.00000000D+00  0.12314750D+01  0.11708550D+01
         0.17979592D+01
    Distances from lower bounds:     XL-X  =
        -0.20000000D+00 -0.20000000D+00 -0.20000000D+00 -0.30000000D+01
        -0.30000000D+01 -0.40000000D+01 -0.50000000D+01
    Distances from upper bounds:     XU-X  =
         0.20000000D+01  0.20000000D+01  0.20000000D+01  0.20000000D+01
         0.20000000D+01  0.10000000D+01  0.30000000D+01
    Number of function calls:        NFUNC =      414
     - within TR method:             NF_TR =      119
     - integer derivatives:          NF_2D =      295
    Number of gradient calls:        NGRAD =       39
    Number of calls of QP solver:    NQL   =      179
     - 2nd order corrections:        NQL2  =       59
    Number of B&B nodes:             NODES =      564  <---------------
    Termination reason:              IFAIL =        0
```

4-Stage Experiments



#	s^2	$\bar{\mu}^*$	K^*	N^*	θ_α, θ_s	%	W_α, W_s	%	B&B
1	1	(1,1,1,1)	(6,7,6,6)	15	(0.8331,0.8165)	2.03	(18.004,18.371)	2.00	872
2	1	(1,1,1,1)	(6,7,9,9)	18	(0.8569,0.8373)	2.34	(21.004,21.497)	2.29	179
3	1	(1,1,1,1)	(9,10,11,11)	23	(0.8817,0.8697)	1.38	(26.085,26.446)	1.37	2974
4	1	(1,1,1,1)	(12,13,13,13)	28	(0.8999,0.8907)	1.03	(31.113,31.435)	1.02	2575
5	1	(1,1,1,1)	(19,20,19,19)	41	(0.9306,0.9221)	0.92	(44.057,44.462)	0.91	4224
6	1/4	(1,1,1,1)	(3,4,3,3)	9	(0.8348,0.8944)	6.66	(10.781,10.062)	7.15	4249
7	1/2	(1,1,1,1)	(4,4,5,4)	11	(0.8344,0.8545)	2.35	(13.184,12.872)	2.42	65
8	3/4	(1,1,1,1)	(6,5,5,5)	13	(0.8337,0.8347)	0.12	(15.5927,15.573)	0.13	4774
9	5/4	(1,1,1,1)	(8,7,7,7)	17	(0.8326,0.8394)	0.81	((20.4171,20.253)	0.81	2360
10	3/2	(1,1,1,1)	(8,8,9,8)	19	(0.8322,0.7960)	4.55	(22.8307,23.869)	4.35	2803
11	$\frac{3}{4}$,1,1,$\frac{3}{4}$	(1,1,1,1)	(6,5,7,5)	14	(0.8321,0.8242)	0.96	(16.825,16.986)	0.95	2313
12	(1,1,1,1)	(1,1,1,1)	(6,7,7,7)	16	(0.8301,0.8057)	3.03	(19.2741,19.858)	2.94	1523
13	($\frac{1}{2}$,1,1,$\frac{1}{2}$)	(1,1,1,1)	(6,7,6,6)	15	(0.8406,0.8289)	1.41	(17.8438,18.092)	1.37	506

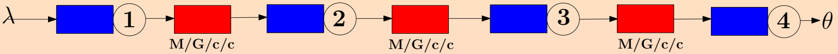
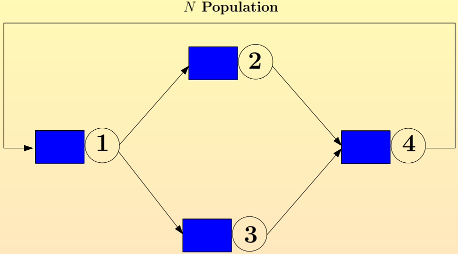


Table: Four-stage Experiments

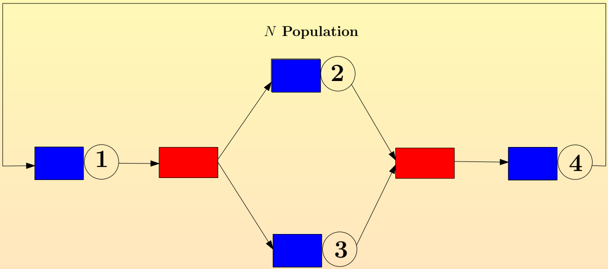
Four-Stage Split and Merge



s^2	$\bar{\mu}^*$	K^*	N^*	θ_α, θ_s	%	W_α, W_s	%	B&B
(1,1,1,1)	(1.33,0.67,0.67,1.33)	(5,6,6,6)	14	(1.0158,0.9913)	2.47	(13.782,14.122)	2.41	5946
(1,1,1,1)	(1.35,0.65,0.65,1.35)	(8,9,9,9)	20	(1.0818,1.0752)	0.61	(18.488,18.600)	0.60	9
(1,1,1,1)	(1.34,0.66,0.67,1.33)	(10,10,10,10)	22	(1.1019,1.1098)	0.71	(19.966,19.822)	0.73	568
(1,1,1,1)	(1.35,0.65,0.65,1.35)	(15,16,16,16)	34	(1.1488,1.1688)	1.71	(29.596,29.088)	1.75	3397
(1,1,1,1)	(1.35,0.65,0.65,1.35)	(24,23,24,24)	50	(1.1752,1.2143)	3.22	(42.546,41.173)	3.33	130
(1,$\frac{1}{2}$,$\frac{1}{2}$,1)	(1.35,0.65,0.65,1.35)	(23,23,23,24)	49	(1.1927,1.2337)	3.32	(41.085,39.716	3.45	22
(1,$\frac{3}{4}$,$\frac{3}{4}$,1)	(1.35,0.65,0.65,1.35)	(16,13,12,16)	31	(1.1504,1.1697)	1.65	(26.947,26.501)	1.68	3353
(1,$\frac{3}{2}$,$\frac{3}{2}$,1)	(1.35,0.65,0.65,1.35)	(11,14,13,11)	27	(1.1038,1.1019)	0.17	(24.462,24.502)	0.16	19
(1,2,2,1)	(1.35,0.65,0.65,1.35)	(10,10,10,10)	22	(1.0536,1.0428)	1.04	(20.882,21.096)	1.01	12

Table: Four-stage Split and Merge Experiments

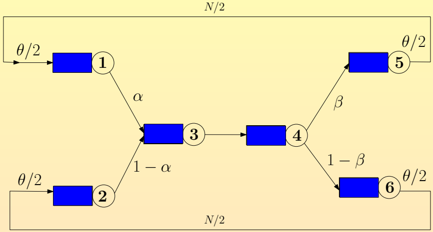
Four-Stage Split and Merge w/ Conveyors



s^2	$\bar{\mu}^*$	K^*	N^*	θ_α, θ_s	%	W_α, W_s	%	B&B
(1,1,1,1)	(1.35,0.65,0.65,1.35)	(5,6,6,6)	14	(1.0094,1.0403)	2.97	(13.869,13.457)	3.06	429
(1,1,1,1)	(1.34,0.66,0.66,1.34)	(6,7,7,7)	16	(1.0533,1.0760)	2.11	(15.190,14.869)	2.16	571
(1,1,1,1)	(1.23,0.77,0.71,1.29)	(8,8,8,9)	19	(1.1034,1.1088)	0.49	(17.219,17.134)	0.50	2321
(1,1,1,1)	(1.35,0.65,0.65,1.35)	(11,11,11,12)	25	(1.1514,1.1506)	0.07	(21.714,21.726)	0.06	337
(1,1,1,1)	(1.34,0.66,0.65,1.233)	(12,13,13,13)	28	(1.1714,1.1684)	0.26	(23.904,23.962)	0.24	1764
$(1,\frac{1}{2},\frac{1}{2},1)$	(1.35,0.65,0.65,1.35)	(6,7,6,6)	15	(1.0607,1.0952)	3.15	(14.141,13.696)	3.25	90
(1,2,2,1)	(1.19,0.81,0.81,1.19)	(7,9,8,9)	19	(1.0636,1.0404)	2.23	(17.863,18.261)	2.18	196

Table: Four-stage Split and Merge Experiments with Conveyors

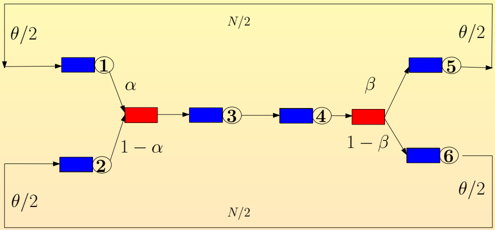
Six-Stage Split and Merge



s^2	$\bar{\mu}^*$	K^*	N^*	θ_α, θ_s	%
(1,1,1,1,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(4,5,5,5,5,5)	18	(1.0147,1.0055)	0.91
(1,1,1,1,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(5,5,6,6,5,6)	20	(1.0522,1.0570)	0.45
(1,1,1,1,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(6,7,7,7,7,7)	24	(1.1102,1.1182)	0.72
(1,1,1,1,1,1)	(0.74,0.74,1.51,1.51,0.76,0.75)	(8,8,8,8,8,9)	28	(1.1507,1.1657)	1.29
(1,1,1,1,1,1)	(0.76,0.75,1.49,1.49,0.75,0.75)	(8,10,10,10,10,9)	32	(1.1851,1.2014)	1.36
$(1,1,\frac{1}{2},\frac{1}{2},1,1)$	(0.76,0.76,1.49,1.49,0.76,0.74)	(5,6,6,6,7,7)	22	(1.1065,1.1273)	1.85
$(1,1,\frac{3}{4},\frac{3}{4},1,1)$	(0.74,0.74,1.50,1.49,0.77,0.72)	(5,7,9,8,6,6)	24	(1.1117,1.1184)	0.60
$(1,1,\frac{3}{2},\frac{3}{2},1,1)$	(0.75,0.75,1.50,1.50,0.75,0.75)	(6,7,8,8,6,8)	25	(1.1022,1.0978)	0.40
(1,1,2,2,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(7,9,8,9,6,8)	27	(1.1041,1.0849)	1.77

W_α, W_s	%	B&B
(35.480,35.799)	0.89	365
(38.0144,37.839)	0.46	1220
(43.234,42.925)	0.72	808
(48.667,48.038)	1.31	3287
(54.004,53.269)	1.38	9039
(39.764,39.027)	1.89	1775
(43.176,42.915)	0.61	5657
(45.364,45.543)	0.39	237
(48.908,49.773)	1.74	3543

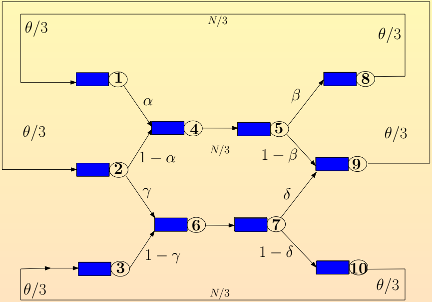
Six-Stage Split and Merge with Conveyors



s^2	$\bar{\mu}^*$	K^*	N^*	θ_α,θ_s	%
(1,1,1,1,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(3,4,4,4,4,4)	15	(1.0210,1.0599)	3.67
(1,1,1,1,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(4,5,4,5,4,5)	17	(1.0705,1.0920)	1.97
(1,1,1,1,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(5,5,5,5,6,5)	19	(1.1116,1.1293)	1.57
(1,1,1,1,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(5,6,7,7,6,6)	22	(1.1610,1.1563)	0.41
(1,1,1,1,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(6,7,7,7,7,7)	24	(1.1878,1.1895)	0.14
$(1,1,\frac{1}{2},\frac{1}{2},1,1)$	(0.75,0.75,1.50,1.50,0.75,0.75)	(4,5,5,5,5,5)	18	(1.1119,1.1189)	0.63
(1,1,2,2,1,1)	(0.75,0.75,1.50,1.50,0.75,0.75)	(5,7,5,5,7,6)	21	(1.1106,1.1277)	1.52

W_α, W_s	%	B&B
(29.383,28.301)	3.82	7269
(31.761,31.133)	2.02	7164
(34.186,33.647)	1.60	7871
(37.898,38.049)	0.40	577
(40.410,40.350)	0.15	463
(32.376,32.171)	0.64	7523
(37.816,37.242)	1.54	9527

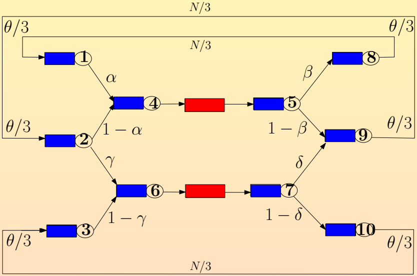
Ten-stage Split and Merge



s^2	$\bar{\mu}^*$	K^*	N^*
(1,...,1)	(0.625,1.25,0.625,1.25,...,1.25,0.625,1.25,0.625)	(7,8,8,8,8,8,8,8,8,8)	45
(1,...,1)	(0.625,1.25,0.625,1.25,...,1.25,0.625,1.25,0.625)	(8,12,12,12,11,12,8,12,12,12)	61
(1,...,1)	(0.625,1.25,0.625,1.25,...,1.25,0.625,1.25,0.625)	(12,14,14,14,14,14,14,14,14,14)	74
(1,...,1)	(0.625,1.25,0.625,1.25,...,1.25,0.625,1.25,0.625)	(16,17,17,17,17,17,17,17,16,17)	89
$(1, \frac{1}{2}, \ldots, \frac{1}{2}, 1)$	(0.625,1.25,0.625,1.25,...,1.25,0.625,1.25,0.625)	(12,13,13,13,13,13,13,13,13,13)	70
(1,2,...,2,1)	(0.625,1.25,0.625,1.25,...,1.25,0.625,1.25,0.625)	(17,18,18,18,19,18,18,18,18,19)	96

θ_α, θ_s	%	W_α, W_s	%	B&B
(0.9416,0.9120)	3.25	(47.791,49.342)	3.14	231
(1.002,0.9981)	0.39	(60.878,61.116)	0.39	2303
(1.0316,1.0347)	0.30	(71.733,71.583)	0.21	1083
(1.0547,1.0526)	0.20	(84.384,84.553)	0.20	3415
(1.0402,1.0436)	0.33	(67.295,67.078)	0.33	1304
(1.0302,1.0116)	1.84	93.186,94.899)	1.81	1123

Ten-stage Split and Merge with Conveyors



s^2	$\bar{\mu}^*$	K*	N^*
(1,...,1)	(0.625,1.25,0.625,1.25,. . . ,1.25,0.625,1.25,0.625)	(16,17,17,17,17,17,17,17,17,17)	90
$(1,\frac{3}{2},\ldots,\frac{3}{2},1)$	(0.625,1.25,0.625,1.25,. . . ,1.25,0.625,1.25,0.625)	(19,22,19,19,19,19,19,19,19,22)	103
(1,2,. . . ,2,1)	(0.625,1.25,0.625,1.25,. . . ,1.25,0.625,1.25,0.625)	(23,24,23,23,23,24,24,23,23,24)	122

θ_α,θ_s	%	W_α,W_s	%	B&B
(1.0504,1.0582)	0.74	(85.682,85.050)	0.74	8348
(1.0499,1.0499)	0.00	(98.105,98.105)	0.00	2074
(1.0497,1.0573)	0.72	(116.224,115.388)	0.72	12727

Resulting Rule Patterns

- For the μ, given the right hand size service rate bound m, the $\mu-$ allocation should follow the topological split and branching probabilities proportionally such that the expected utilization rate

$$\mu \rightarrow \rho_\ell \approx 1 \ \forall \ell \in G(V,E)$$

- For the buffer allocation, a uniform allocation should prevail no matter what the split-merge topology configuration.

$$K_\ell \rightarrow \frac{K}{m}$$

- This latter result is surprising.
- The combined two pattern rules seem to be very robust.

Resulting Rule Patterns

- For the μ, given the right hand size service rate bound m, the $\mu-$ allocation should follow the topological split and branching probabilities proportionally such that the expected utilization rate

$$\mu \to \rho_\ell \approx 1 \; \forall \ell \in G(V,E)$$

- For the buffer allocation, a uniform allocation should prevail no matter what the split-merge topology configuration.

$$K_\ell \to \frac{K}{m}$$

- This latter result is surprising.
- The combined two pattern rules seem to be very robust.

Resulting Rule Patterns

- For the μ, given the right hand size service rate bound m, the $\mu-$ allocation should follow the topological split and branching probabilities proportionally such that the expected utilization rate

$$\mu \rightarrow \rho_\ell \approx 1 \ \forall \ell \in G(V,E)$$

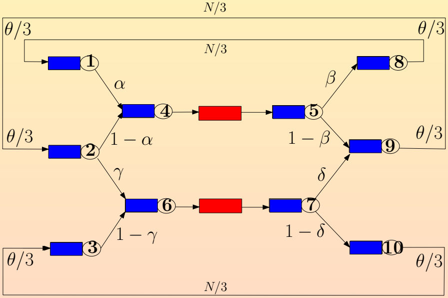
- For the buffer allocation, a uniform allocation should prevail no matter what the split-merge topology configuration.

$$K_\ell \rightarrow \frac{K}{m}$$

- This latter result is surprising.
- The combined two pattern rules seem to be very robust.

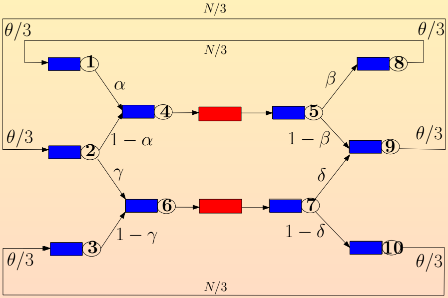
Summary & Conclusions and Open Questions

- Service Rate & Buffer Allocation Problem
  - Simultaneous $x = (\mu, K)$ Optimization
  - Uniform Pattern verification
    - $\mu(m)$ should follow the topology and be proportional to $\rho_\ell \equiv 1$
    - K should be uniform.
  - Includes general service and the material handling system.
  - Performs pretty well.
- Open Questions
  - Larger Networks → Patterns
  - Patterns for $\{\lambda, \mu, c, K, N\}$ simultaneously
  - Mixed Network Topologies



Complex Networks

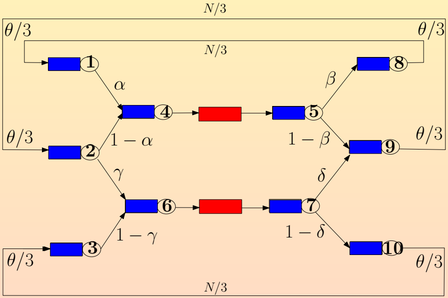
- Service Rate & Buffer Allocation Problem
  - Simultaneous $x = (\mu, K)$ Optimization
  - Uniform Pattern verification
    - $\mu(m)$ should follow the topology and be proportional to $\rho_\ell \equiv 1$
    - K should be uniform.
  - Includes general service and the material handling system.
  - Performs pretty well.
- Open Questions
  - Larger Networks → Patterns
  - Patterns for $\{\lambda, \mu, c, K, N\}$ simultaneously
  - Mixed Network Topologies



Complex Networks

Summary & Conclusions and Open Questions

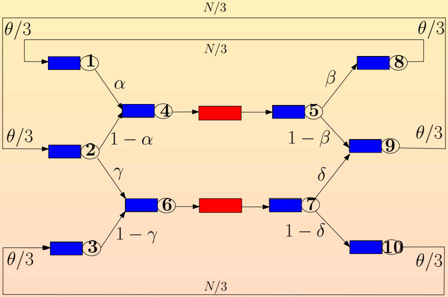
- Service Rate & Buffer Allocation Problem
  - Simultaneous $x = (\mu, K)$ Optimization
  - Uniform Pattern verification
    - $\mu(m)$ should follow the topology and be proportional to $\rho_\ell \equiv 1$
    - K should be uniform.
  - Includes general service and the material handling system.
  - Performs pretty well.
- Open Questions
  - Larger Networks → Patterns
  - Patterns for $\{\lambda, \mu, c, K, N\}$ simultaneously
  - Mixed Network Topologies



Complex Networks

Summary & Conclusions and Open Questions

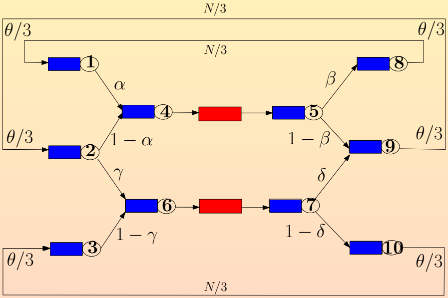
- Service Rate & Buffer Allocation Problem
  - Simultaneous $x = (\mu, K)$ Optimization
  - Uniform Pattern verification
    - $\mu(m)$ should follow the topology and be proportional to $\rho_\ell \equiv 1$
    - K should be uniform.
  - Includes general service and the material handling system.
  - Performs pretty well.



- Open Questions
  - Larger Networks → Patterns
  - Patterns for $\{\lambda, \mu, c, K, N\}$ simultaneously
  - Mixed Network Topologies

Complex Networks

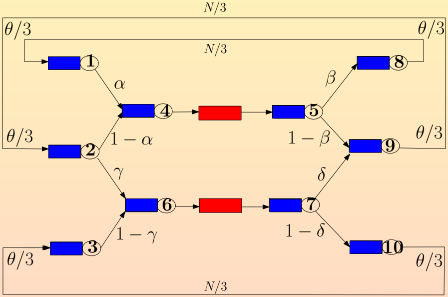
- Service Rate & Buffer Allocation Problem
  - Simultaneous $x = (\mu, K)$ Optimization
  - Uniform Pattern verification
    - $\mu(m)$ should follow the topology and be proportional to $\rho_\ell \equiv 1$
    - K should be uniform.
  - Includes general service and the material handling system.
  - Performs pretty well.
- Open Questions
  - Larger Networks → Patterns
  - Patterns for $\{\lambda, \mu, c, K, N\}$ simultaneously
  - Mixed Network Topologies



Complex Networks

Summary & Conclusions and Open Questions

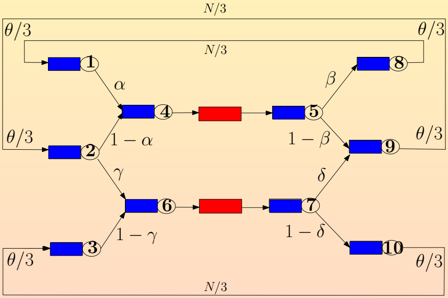
- Service Rate & Buffer Allocation Problem
  - Simultaneous $x = (\mu, K)$ Optimization
  - Uniform Pattern verification
    - $\mu(m)$ should follow the topology and be proportional to $\rho_\ell \equiv 1$
    - K should be uniform.
  - Includes general service and the material handling system.
  - Performs pretty well.
- Open Questions
  - Larger Networks $\rightarrow$ Patterns
  - Patterns for $\{\lambda, \mu, c, K, N\}$ simultaneously
  - Mixed Network Topologies



Complex Networks

Summary & Conclusions and Open Questions

- Service Rate & Buffer Allocation Problem
  - Simultaneous $x = (\mu, K)$ Optimization
  - Uniform Pattern verification
    - $\mu(m)$ should follow the topology and be proportional to $\rho_\ell \equiv 1$
    - K should be uniform.
  - Includes general service and the material handling system.
  - Performs pretty well.
- Open Questions
  - Larger Networks $\rightarrow$ Patterns
  - Patterns for $\{\lambda, \mu, c, K, N\}$ simultaneously
  - Mixed Network Topologies



Complex Networks