Robust Queueing Applied to Closed Queueing Networks

Austin Palmer

September 16, 2018

Overview

- Overview of BBY15 Robust Queueing
- Practical Improvement to RQNA
- Application to Closed Queueing Networks

Background and Motivation

- Effective arrival processes of GI/GI/1 queues very frequently do not fit a simple probability distribution.
- Previous approximation techniques (QNA, QNET)
- Robust Queueing treats arrivals and departures as belonging to an uncertainty set, rather than a renewal process.
- RQ then approximates a queue by finding the worst-case system time for the *n*th job in a system.

Lindley Recursion

- A significant motivator for RQ is the Lindley recursion, which provides a basic framework for treating queueing as an optimization problem.
- System time of the nth job in a single-server, FIFO (first-in first-out) queue can be described by the Lindley recursion,

$$S_n = W_n + X_n = \max(W_{n-1} + X_{n-1} - T_n, 0) + X_n \tag{1}$$

$$= \max_{1 \le k \le n} \left(\sum_{i=k}^{n} X_i - \sum_{i=k+1}^{n} T_i \right), \tag{2}$$

Motivation for Uncertainty Sets

- Assume inter-arrival times (T) and service times (X) are i.i.d, with means $\frac{1}{\lambda}$ and $\frac{1}{\mu}$, and finite variances σ_a^2 and σ_s^2.
- By the CLT, we know that

$$\lim_{n\to\infty} \frac{\sum_{i=k+1}^{n} T_i - \frac{n-k}{\lambda}}{\sigma_a\sqrt{n-k}} \sim \mathcal{N}(0,1), \tag{3}$$

and

$$\lim_{n\to\infty} \frac{\sum_{i=k+1}^{n} X_i - \frac{n-k}{\mu}}{\sigma_s\sqrt{n-k}} \sim \mathcal{N}(0,1), \tag{4}$$

- We wil use this fact to bound our arrival and service uncertainty sets.

RQ Assumptions for Single-Server Queues

1. The inter-arrival times $\{T_1, \ldots, T_n\}$ belong to the uncertainty set

$$U^a = \left\{ (T_1, \ldots, T_n) \middle| \frac{\sum_{i=k+1}^{n} T_i - \frac{n-k}{\lambda}}{\sqrt{n-k}} \geq -\Gamma_a, 0 \leq k \leq n-1 \right\},$$

where Γ_a is a variability parameter for interarrival times.

2. The service times $\{X_1, \ldots, X_n\}$ belong to the uncertainty set

$$U^s = \left\{ (X_1, \ldots, X_n) \middle| \frac{\sum_{i=k}^{n} X_i - \frac{n-k+1}{\mu}}{\sqrt{n-k+1}} \leq \Gamma_s, 0 \leq k \leq n-1 \right\}.$$

where Γ_s is a variability parameter for service times.

RQ Assumptions for Single-Server Queues

- The variability parameters Γ_a and Γ_s are used to control the robustness of the optimization.
- Although the underlying distributions for T_i and X_i are generally assumed to be i.i.d., the uncertainty sets are not assumed to be independent.
- We model multi-server queues by modifying the service uncertainty set, and to model heavy-tailed distributions (infinite variance), we change the terms $\sqrt{n-k}$ and $\sqrt{n-k+1}$ to $(n-k)^{1/\alpha_a}$ and $(n-k+1)^{1/\alpha_s}$, where α_a and α_s are the tail coefficients of the arrival and service distributions.
- For our pursposes, we will assume $\alpha_a = \alpha_s = 2$.

Performance Analysis under RQ

- We look to approximate the system time of the nth job, S_n, by finding the worst-case system time it could experience.
- We do this by solving the optimization problem

$$\hat{S}_n = \max_{T \in U^a} \left(\max_{X \in U^s} S_n \right) \tag{5}$$

$$= \max_{T \in U^a} \left(\max_{X \in U^s} \max_{1 \leq k \leq n} \left(\sum_{i=k}^{n} X_n - \sum_{i=k+1}^{n} T_n \right) \right) \tag{6}$$

$$\leq \max_{T \in U^a} \left(\max_{1 \leq k \leq n} \max_{X \in U^s} \left(\sum_{i=k}^{n} X_n - \sum_{i=k+1}^{n} T_n \right) \right). \tag{7}$$

Performance Analysis under RQ

- Importantly, the bound in 7 is tight; that is, if we let $\hat{X}_i = \frac{1}{\mu} + \Gamma_s \left[\sqrt{(n-i+1)} - \sqrt{(n-i)}\right], 1 \leq i \leq n$, then

$$\sum_{i=k}^{n} \hat{X}_i = \max_{X \in U^s} \sum_{i=k}^{n} X_i = \frac{n-k+1}{\mu} + \Gamma_s \sqrt{n-k+1}, 1 \leq k \leq n.$$

  Furthermore, the service times $\hat{X}_i$ are non-decreasing; that is, $\hat{X}_1 \leq \hat{X}_2 \leq \ldots \leq \hat{X}_n$.

- Similarly, we can find a sequence of $\hat{T}_i$ such that

$$\sum_{i=k+1}^{n} \hat{T}_i = \max_{T \in U^a} \sum_{i=k+1}^{n} T_i = \frac{n-k}{\lambda} + \Gamma_s \sqrt{n-k}, 1 \leq k \leq n-1.$$

System Time Bound for Single-Server Queues

- Since X and T are independent of each other, we can rewrite the system time optimization as

$$\hat{S}_n = \max_{1\leq k\leq n}\left\{\frac{n-k+1}{\mu} + \Gamma_s\sqrt{n-k+1} - \frac{n-k}{\lambda} + \Gamma_a\sqrt{n-k}\right\} \quad (8)$$

$$\leq \max_{1\leq k\leq n}\left\{(\Gamma_a+\Gamma_s)\sqrt{n-k+1} - \frac{n-k}{\lambda} + \frac{n-k+1}{\mu}\right\} \quad (9)$$

$$= \max_{1\leq k\leq n}\left\{(\Gamma_a+\Gamma_s)\sqrt{n-k+1} - \frac{n-k}{\lambda} + \frac{n-k+1}{\mu} + \frac{1}{\lambda} - \frac{1}{\lambda}\right\} \quad (10)$$

$$= \max_{1\leq k\leq n}\left\{(\Gamma_a+\Gamma_s)\sqrt{n-k+1} + (n-k+1)\frac{\lambda-\mu}{\lambda\mu} + \frac{1}{\lambda}\right\} \quad (11)$$

$$= \max_{1\leq k\leq n}\left\{(\Gamma_a+\Gamma_s)\sqrt{n-k+1} + (n-k+1)\frac{1-\rho}{\lambda}\right\} + \frac{1}{\lambda}. \quad (12)$$

System Time Bound for Single-Server Queues

- Letting $x = (n - k + 1)$, $\beta = (\Gamma_a + \Gamma_s)$, and $\delta = \frac{1-\rho}{\lambda}$, the above optimization becomes

$$\hat{S}_n = \max_{1 \leq x \leq n} \beta\sqrt{x} - \delta x + \frac{1}{\lambda} \leq \max_{x \in \mathbb{R}} \beta\sqrt{x} - \delta x + \frac{1}{\lambda} = \frac{1}{4}\frac{\beta^2}{\delta} + \frac{1}{\lambda}. \quad (13)$$

- Subsituting our original values in for x, β, and δ, we have the following:

Worst-Case System Time in a Single-Server FIFO Queue

In a single-server FIFO queue with $T \in U^a, X \in U^s$, and $\rho < 1$,

$$\hat{S}_n \leq \frac{\lambda}{4}\frac{(\Gamma_a + \Gamma_s)^2}{1 - \rho} + \frac{1}{\lambda}.$$

Extension to Networks of Queues

- In order to extend our analysis to networks of queues, we need to derive three important characteristics:
  - The characterization of departures from a queue
  - The characterization of a superposition of arrival processes
  - The characterization of a splitting of a departure process

Departure Characterization

Departure Characterization in RQ

Consider a single-server queue with inter-arrival times $T \in U^a$, and service times $X \in U^s$, with $\rho < 1$. Under adversarial service times, the inter-departure times $D = \{D_1, \ldots, D_n\}$ belong to the uncertainty set

$$U^d \subseteq U^a = \left\{ (D_1, \ldots, D_n) \middle| \frac{\sum_{i=k+1}^{n} D_i - \frac{n-k}{\lambda}}{\sqrt{n-k}} \geq -\Gamma_a, 0 \leq k \leq n-1 \right\}.$$

- This is analagous to Burke's theorem for M/M/1 queues.

Superposition of Arrivals

Superposition of Arrivals in RQ

Consider a queue fed by p separate arrival processes, each characterized by an arrival uncertainty set with rate λ_j and variability parameter $\Gamma_{a,j}$. Then these arrival processes form a merged arrival process, that is characterized by the uncertainty set

$$U^a_{sup} = \left\{ (T_1^{sup}, \ldots, T_n^{sup}) \middle| \frac{\sum_{i=k+1}^{n} T_i^{sup} - \frac{n-k}{\lambda_{sup}}}{\sqrt{n-k}} \geq -\Gamma_{a,sup}, 0 \leq k \leq n-1 \right\},$$

where $\lambda_{sup} = \sum_{j=1}^{p} \lambda_j$, and $\Gamma_{a,sup} = \frac{1}{\lambda_{sup}} \sqrt{\left(\sum_{j=1}^{p} (\lambda_j \Gamma_{a,j})^2 \right)}$.

Superposition of Arrivals

Splitting of Departure

Splitting of the Departure Process in RQ

Consider an arrival process characterized by the above uncertainty set, with rate λ and variability parameter Γ_a. Then, the "thinned" process, comprised of a fraction f of those arrivals, is described by the uncertainty set

$$U^a_{split} = \left\{ (T_1^{split}, \ldots, T_n^{split}) \left| \frac{\sum_{i=k+1}^{n} T_i^{split} - \frac{n-k}{\lambda_{split}}}{\sqrt{n-k}} \geq -\Gamma_{a,split}, 0 \leq k \leq n-1 \right. \right\}$$

where $\lambda_{split} = \lambda f$, and $\Gamma_{a,split} = \frac{\Gamma_a}{\sqrt{f}}$.

Queueing Network Characterization in RQ

RQNA

Consider a network of single-class, single-server FIFO queues, with arrival and service processes characterized above, and a routing matrix P. Then, the behavior of the network is equivalent to a collection of independent queues, where the arrival process at each node j belongs to the uncertainty set

$$U_j^a = \left\{ (T_1^j, \ldots, T_n^j) \left| \frac{\sum_{i=k+1}^{n} T_i^j - \frac{n-k}{\bar{\lambda}_j}}{\sqrt{n-k}} \geq -\bar{\Gamma}_{a,j}, 0 \leq k \leq n-1 \right. \right\},$$

where $\{\bar{\lambda}_1, \ldots, \bar{\lambda}_M\}$ and $\{\bar{\Gamma}_{a,1}, \ldots, \bar{\Gamma}_{a,M}\}$ satisfy the following set of equations:

$$\bar{\lambda}_j = \lambda_j + \sum_{}^{M} (\bar{\lambda}_i P_{ij}), \bar{\Gamma}_{a,j} \quad = \frac{1}{\bar{\lambda}_j} \sqrt{(\lambda_j \Gamma_{a,j})^2 + \sum_{}^{M} (\bar{\lambda}_i \bar{\Gamma}_{a,i})^2 P_{i,j}}. \quad (14)$$

Problems with RQ Framework

- Setting values for $\Gamma_{a,j}$ and $\Gamma_{s,j}$
- Departure characterization
- Accuracy of the approximation

Determining $\Gamma_{a,j}$ and $\Gamma_{s,j}$

- The original Bandi et al paper determined the variability parameters using linear regression, setting

$$\Gamma_{a,j} = \sigma_j \tag{15}$$
$$\Gamma_{s,j} = f(\rho, \sigma_a, \sigma_s, \alpha), \tag{16}$$

  where

$$f(\rho, \sigma_a, \sigma_s, \alpha) = (\theta_0 + \theta_1 * \sigma_s^2/m + \theta_2 \sigma_a^2 \rho^2 m)^{(\alpha-1)/\alpha} - \sigma_a m^{(\alpha-1)/\alpha},$$

  m is the number of servers in the queue, and $\theta_0, \theta_1, \theta_2$ are parameters to be determined via linear regression.

- However, this approach quickly leads to overfitting, and works poorly in general.

Other Methods of Determining Variability Parameters

- Whitt and You (2018) developed a framework similar to BBY15, only with a single uncertainty set for both arrivals and departures, and a single variabiliy parameter for both.
- Showed that, if chosed, correctly, their RQ bound exactly matches the Kingman bound (1962). In the BBY15 formulation, this is not possible due to the dual variability parameters.

Departure Characterization in BBY

- When expanding analysis to cover a network of queues, BBY also faces a problem with the departure characterization.
- In QNA (AND OTHERS), the departure variance is usually approximated as a combination of the arrival and service variance; however, in BBY, the departures are considered to belong to the exact same uncertainty set as the arrivals.
- in GI/GI/1 queues this is very frequently not the case.

Problems with BBY15

ρ_1	ρ_2	$E[W_1]$	$E[W_2]$	BBY(W_1)	BBY(W_2)
0.3	0.3	.0127	.0301	.1964	.1964
0.3	0.6	.0128	.3442	.1964	
0.3	0.9				
0.6	0.3				
0.6	0.6				
0.6	0.9				
0.9	0.3				
0.9	0.6				
0.9	0.9				

Table: Comparison of original BBY15 bound to simulation in an open series of 2 queues.

Improving on BBY Framework

- Instead of determing $\Gamma_{a,j}$ and $\Gamma_{s,j}$ using linear regression, we set $\Gamma_{a,j} = k\sigma_a^j, \Gamma_{s,j} = k\sigma_s^j$. This provides a reasonable amount of robustness to the uncertainty sets, and provides reasonably accurate approximations in practice.
- This is the same approach used in Whitt and You (2018); however, whereas they chose to set $k = \sqrt{2}$ to match up with the Kingman bound, we have chosen to let $k = 1$, which performs better in practice.
- Rather than treating the departures the same as arrivals, we will instead assume the departure uncertainty set has the following form:

$$U^d \subseteq U^a = \left\{ (D_1, \ldots, D_n) \middle| \frac{\sum_{i=k+1}^{n} D_i - \frac{n-k}{\lambda}}{\sqrt{n-k}} \geq -\Gamma_d, 0 \leq k \leq n-1 \right\}$$

where $\Gamma_d = \sqrt{\rho^2 \Gamma_s^2 + (1-\rho^2)\Gamma_a^2}$.

Improving on BBY Framework

- To reflect this change, we will modify our set of linear equations to be

$$\bar{\lambda}_j = \lambda_j + \sum_{i=1}^{M}(\bar{\lambda}_i P_{ij}), x_j = \frac{1}{\bar{\lambda}_j^2}(\lambda_j \Gamma_{a,j})^2 + \sum_{i=1}^{M}(\bar{\lambda}_i y_i P_{i,j}, y_j = \Gamma_{s,j}^2 + \left(1 - \left(\frac{\bar{\lambda}_j}{\mu_j}\right.\right.$$

  where $x_j = \bar{\Gamma}_{a,j}^2$, and $y_j = \bar{\Gamma}_{d,j}^2$.

- Lastly, we modfiy the original $\hat{S}_n$ from BBY to be

$$\hat{S}_n = \frac{\lambda}{4}\frac{(\Gamma_a + \Gamma_s)^2}{1-\rho} + \frac{1}{\mu}.$$

- This change makes more intuitive sense (now the system time is split up into wait time and service time). It also vastly improves the approximation in most cases, and makes it feasible to approximate CQNS with few jobs in the system.

Comparing Original RQ to Modified RQ

ρ_1	ρ_2	BBY(W_1)	BBY(W_2)	MRQ(W_1)	MRQ(W_2)
0.3	0.3	.0127	.0301	.1964	.1964
0.3	0.6	.0128	.3442	.1964	
0.3	0.9				
0.6	0.3				
0.6	0.6				
0.6	0.9				
0.9	0.3				
0.9	0.6				
0.9	0.9				

Table: Comparison of original BBY15 bound to simulation in an open series of 2 queues.

Applying RQNA to CQNs

- To use RQNA for CQNs, we will use the fixed-population-mean method (Whitt).
- However, rather than the original FPM, which only required one parameter (λ), our framework requires two parameters (λ and Γ_a).

RQFPM Process

- As before, we choose one node to be the "cut" node; however, instead of choosing any arbitrary node, we instead choose a bottleneck node.
- For our purposes, a bottleneck node is defined as a node $i*$, such that

$$i* = \max_{1 \leq i \leq n} \frac{\bar{\lambda}_i}{\mu_i}.$$

- It is computationally very easy to find this node, since all we have to do is solve equation, for which we don't need to know either of our unknown parameters. This also tends to maximize the accuracy of the approximation.

Applying RQNA to CQNs

- Once we have our cut node $i*$, all flows heading into node $i*$ are replaced with a single arrival uncertainty set, with parameters $\bar{\lambda}_{i*}$ and $\bar{\Gamma}_{a,i*}$.
- Instead of keeping $\bar{\Gamma}_{a,i*}$ a free variable, we instead set its initial value to

$$\bar{\Gamma}_{a,i*} = \sqrt{\frac{1}{\bar{\lambda}_{i*}} \sum_{i=1}^{J} \bar{\lambda}_i P_{i,i*} \Gamma_{s,i}^2}.$$

- Use a linear-algebra software package to solve the system of equations, and use the equation

$$E[S] = \sum_{i=1}^{J} \frac{\bar{\lambda}_i}{\bar{\lambda}_{i*}} E[S_i] \approx \sum_{i=1}^{J} \frac{\bar{\lambda}_i}{\bar{\lambda}_{i*}} \left(\frac{\bar{\lambda}_i (\bar{\Gamma}_{a,i} + \Gamma_{s,i})^2}{4(1 - \frac{\bar{\lambda}_i}{\mu_i})} + \frac{1}{\mu_i} \right).$$

Applying RQNA to CQNs

- Once we have our $E[S]$, we can apply Little's Law, and find the root of the equation
$$f(\bar{\lambda}_{i*}) = K - \bar{\lambda}_{i*} E[S].$$
- $f(\bar{\lambda}_{i*})$ usually has multiple roots; however, if we restrict our domain to $[0, \mu_{i*})$, we will be able to find our throughput approximation $\bar{\lambda}_{i*}$.

Examples

- We will look at two examples to showcase our results.
- First, we consider a network of 9 queues in series; the first 8 queues have service times distributed as rate-.5 Erlang distributions, and the last queue has a rate-1 hyperexponential distribution.
- Second, we will look at a closed-queueing version of Kuehn's nine-node network [KEUHN]; in this network, departures from node 7 are instead routed to node 1, departures from node 9 are routed to node 2, and departures from node 6 are routed to node 3. We will look at the case when nodes 1,2, and 3 have the same distribution, with $\mu = 0.4$, and nodes 4-9 have the same distribution, with $\mu = 1.0$. We will consider a variety of different distributions and scvs.

Numerical Results

$(c_{s,1}^2, c_{s,2}^2)$	K	Simulated Throughput	Approximate Throughput	Relat
(1, 4)	18	.3513	.4017	0.
(1, 8)	18	.3444	.3943	0.
(1, 4)	36	.4135	.4386	0.
(1, 8)	36	.4083	.4339	0.
(.5, 4)	18	.395	.4339	0.
(.5, 8)	18	.3834	.4255	0.
(.5, 4)	36	.4469	.4618	0.
(.5, 8)	36	.4408	.457	0.
(.25,4)	18	.4279	.457	0.
(.25,8)	18	.4145	.4482	0.
(.25,4)	36	.4688	.4766	0.
(.25,8)	36	.4619	.4719	0.

Table: RQFPM results for a series of 9 queues.

Numerical Results

$(c_{s,1}^2, c_{s,2}^2)$	K	Simulated Throughput	Approximate Throughput	Relati

Table: RQFPM results for Keuhn's 9-node network.

Alternative Methods

- The results from RQ-FPM are mostly very accurate, but in small networks the error is much larger than larger networks.
- In large networks, however, the algorithm takes significantly more time.
- Problem is exacerbated by the messy nature of the system of equations, which quickly becomes difficult to analyze numerically by some software packages.
- Lastly, many of the problems present in BBY15 are not dealt with entirely in RQ-FPM; how to properly set the variability parameters is still an issue, and probably the most significant to the accuracy of the algorithm, but departure characterization is also problematic.

Iterated FPM

- The results from RQ-FPM are very promising, but there are some casees where the error exceeds 10 percent.
- If we run RQ-FPM multiple times, updating our guess of $\bar{\Gamma}_{a,i*}$ each time, we will converge to a steady value of $\bar{\Gamma}_{a,i*}$.
- Calculating our approximate throughput this way also allows us to consider the entire network at once.
- However, the convergent value of $\bar{\Gamma}_{a,i*}$ generally leads to less accurate approximations as of now.

Example

- Consider our first example, with $c_{s,1}^2 = 1, c_{s,2}^2 = 8$, and $K = 18$. Our initial guess for $\bar{\Gamma}_{a,i*}$ is

$$\bar{\Gamma}_{a,i*} = \sqrt{\frac{1}{\bar{\lambda}_{i*}} \sum_{i=1}^{J} \bar{\lambda}_i P_{i,i*} \Gamma_{s,i}^2} \tag{17}$$

$$= \sqrt{\sum_{i=1}^{J} P_{i,i*} \Gamma_{s,i}^2} = \Gamma_{s,9} = \sqrt{8} \approx 2.828. \tag{18}$$

- After running RQ-FPM, our approximate throughput would be $\bar{\lambda}_1 \approx .3943$. If we treat the departure variance from node 9 as our new guess for $\bar{\Gamma}_{a,i*}$, we have

$$\bar{\Gamma}_{a,i*} \approx \sqrt{4.5437} = 2.1316. \tag{19}$$

- After 2 more iterations, our $\bar{\Gamma}_{a,i*}$ converges to roughly 2.1316, with an approximate throughput of $\bar{\lambda}_{i*} = .3684$.