An Improved Approximate Mean Value Analysis Algorithm for Solving Separable Queueing Network Models

Kenneth C. Sevcik and Hai Wang

Department of Computer Science

University of Toronto

Toronto, Ontario, Canada M5S 3G4

E-mail: {kcs,hai}@cs.toronto.edu

Abstract

We present a new approximate Mean Value Analysis (MVA) algorithm, the Queue Line (QL) algorithm, for solving multiple class separable queueing networks. The QL algorithm is always more accurate than, and yet has the same computational complexity as, the Bard-Schweitzer Proportional Estimation (PE) algorithm, the most popular approximate MVA algorithm. The existence and uniqueness of the solutions of the algorithm are proved, and the convergence of the algorithm is established for single class separable class separable queueing networks.

Key Words: queueing network models; mean value analysis; approximate solution techniques

1 Introduction

As computer technology has advanced, computer-communication systems have become increasingly complex. Moreover, coupled with the improvements in the hardware performance, there has been a vast proliferation of workloads and widely varying resource demands, both within each workload type and also between different types. The interactions among numerous system components and workloads make human intuition no longer adequate to predict reliably how changes in the system configuration will impact overall system performance. Hence, it is increasingly important to apply quantitative tools to aid system analysis and design, capacity planning, and resource management. Queueing network models have been widely used as a quantitative tool for performance evaluation. While it is infeasible to compute the exact solution for the general class of queueing networks, solutions can be computed for an important subset, called *separable* (or *product-form*) queueing networks [2]. Many non-separable queueing networks can be accurately approximated by larger and more complex separable queueing networks [19]. Several exact algorithms have been proposed to compute the exact solution for separable queueing networks [3, 6, 7, 8, 14, 16, 17, 22, 23, 24, 31]. Among them, the *Mean Value Analysis (MVA)* algorithm [24] is the most popular and widely used. However, even for separable queueing networks, the computational cost of an exact solution becomes prohibitively expensive as the number of classes, customers, or centers grows.

Numerous *Approximate Mean Value Analysis (AMVA)* algorithms have been proposed for yielding approximate answers with reduced computational cost [1, 4, 5, 9, 11, 13, 15, 25, 26, 28, 29, 33, 34, 35]. Two major algorithms have gained wide acceptance among performance analysts: the Bard-Schweitzer Proportional Estimation (PE) algorithm [25] and the Chandy-Neuse Linearizer algorithm [5]. Both algorithms are iterative algorithms with substantially smaller computation requirements than the exact MVA algorithm. The Linearizer algorithm is more accurate than the PE algorithm, but at higher computational cost. Currently, there is no algorithm that is more accurate than, and yet has the same computational complexity as, the PE algorithm.

In this paper, we present a new AMVA algorithm, called the Queue Line (QL) algorithm, for solving separable queueing networks. The QL algorithm is based on the approximation of the average queue length for each customer class at each center in the network. Because the QL algorithm has the same space and time complexities as the PE algorithm, and always yields more accurate solutions, it dominates the PE algorithm in the spectrum of different algorithms that trade off accuracy and efficiency. Since the QL algorithm is also an iterative algorithm, the existence and uniqueness of its solutions are proved, and the convergence is established for single class separable queueing networks.

The remainder of the paper is organized as follows. Section 2 reviews material relating to AMVA algorithms. Section 3 presents the QL algorithm and examines its computational complexity. Section 4 establishes the existence and uniqueness of the solutions of the QL algorithm. The convergence of the QL algorithm is also shown for the single class separable queueing networks. Section 5 contains a proof that the QL algorithm is more accurate than the PE algorithm. Finally, a summary and conclusions are provided in Section 6.

2 Background

Consider a closed separable queueing network [2, 24] with C customer classes (also termed *routing chains*), and K load independent (also termed *fixed-rate* or *single-server*) service centers. The customer classes are indexed as classes 1 through C, and the centers are indexed as centers 1 through K. The customer population of the queueing network is denoted by the vector $\vec{N} = (N_1, N_2, ..., N_C)$ where N_c denotes the number of customers belonging to class c for $c = 1, 2, ..., C$. Also, the total number of customers in the network is denoted by $N = N_1 + N_2 + \cdots + N_C$. The mean service demand of a class c customer at center k is denoted by $D_{c,k}$ for $c = 1, 2, ..., C$, and $k = 1, 2, ..., K$. (This service demand is defined as the product of two factors: the mean service time of a class c customer during each visit to center k, and the mean number of visits by a class c customer to center k.)

The think time of class c, Z_c, is the sum of the delay (also termed *infinite-server*) center service demands of class c.

Let $D = \sum_{c=1}^{C} \sum_{k=1}^{K} D_{c,k}$, and let $W_c = \max_k\{D_{c,k}\}$ for $c = 1, 2, ..., C$. A service center with service demand W_c for class c is defined as a bottleneck center for class c customers. When $D_{c,i} \neq D_{c,j}$ for some i and j, each service center can be classified as a bottleneck center or a non-bottleneck center for class c customers. Let $\mathbb{B}_c$ denote the set of all bottleneck service centers for class c customers, and let $\overline{\mathbb{B}}_c$ denote the set of all non-bottleneck service centers for class c customers.

For networks with population vector $\vec{N}$, we consider the following performance measures of interest:

$R_{c,k}(\vec{N}) =$ the average residence time of a class c customer at center k.

$R_c(\vec{N}) =$ the average response time of a class c customer in the network.

$X_c(\vec{N}) =$ the throughput of class c.

$Q_{c,k}(\vec{N}) =$ the mean queue length of class c at center k.

$Q_k(\vec{N}) =$ the mean total queue length at center k.

For single class separable queueing networks (i.e., $C = 1$), the subscript c on the network parameters and performance measures is dropped for simplicity, and the network population is denoted by a scalar variable N.

Based on the *Arrival Instant Distribution* theorem [18, 27] and Little's Law [20], the exact MVA algorithm [24] involves repeated applications of the following six equations, in which $\vec{n} = (n_1, n_2, ..., n_C)$ is a population vector ranging from $\vec{0}$ to $\vec{N}$; $A_k^{(c)}(\vec{n})$ is the average number of customers a class c customers finds already at center k when it arrives there, given the network population $\vec{n}$; $\vec{1}_c$ is a C-dimensional vector whose c^{th} element is one and whose other elements are zeros, and $(\vec{n} - \vec{1}_c)$ denotes the population vector $\vec{n}$ with one class c customer removed:

$$A_k^{(c)}(\vec{n}) = Q_k(\vec{n} - \vec{1}_c), \tag{1}$$

$$R_{c,k}(\vec{n}) = D_{c,k} \cdot \left(1 + A_k^{(c)}(\vec{n})\right), \tag{2}$$

$$R_c(\vec{n}) = \sum_{k=1}^{K} R_{c,k}(\vec{n}), \tag{3}$$

$$X_c(\vec{n}) = \frac{n_c}{Z_c + R_c(\vec{n})}, \tag{4}$$

$$Q_{c,k}(\vec{n}) = R_{c,k}(\vec{n}) \cdot X_c(\vec{n}), \tag{5}$$

$$Q_k(\vec{n}) = \sum_{c=1}^{C} Q_{c,k}(\vec{n}), \tag{6}$$

with initial conditions $Q_k(\vec{0}) = 0$ for $k = 1, 2, ..., K$.

For the exact solutions of multiple class separable queueing networks, it has been proved that the following relations hold [10, 11, 13, 30]:

$$X_j(\vec{N}) - X_j(\vec{N} - \vec{1}_c) = \begin{cases} > 0 & \text{if } j = c, \\ < 0 & \text{if } j \neq c, \end{cases} \tag{7}$$

$$Q_{c,k}(\vec{N}) - Q_{c,k}(\vec{N} - \vec{1}_c) > 0, \tag{8}$$

$$\lim_{N_c \to +\infty} [Q_{j,k}(\vec{N}) - Q_{j,k}(\vec{N} - \vec{1}_c)] = 0 \quad \text{for } j \neq c. \tag{9}$$

Hence, the function $Q_{c,k}(\vec{N})$ with respect to N_c is a monotonically increasing function. For single class separable queueing networks, it has been proved that

$$X(N) - X(N-1) \geq X(N+1) - X(N) > 0, \tag{10}$$

and

$$R(N+1) - R(N) \geq R(N) - R(N-1) > 0, \tag{11}$$

which imply that the function $X(N)$ with respect to N is a monotonically increasing concave function, and the function $R(N)$ with respect to N is a monotonically increasing convex function [10, 11, 13, 30].

The key component of the exact MVA algorithm is the recursive expression in equation (1) which relates performance with population vector $\vec{n}$ to that with population vector $(\vec{n} - \vec{1}_c)$. This recursive dependence of the performance measures for one population on lower population levels causes both space and time complexities of the exact MVA algorithm to be $\Theta(KC\prod_{c=1}^{C}(N_c+1))$. Thus, for large number of classes (more than ten) or large populations per class, it is not practical to solve networks with the exact MVA algorithm.

The approximate MVA algorithms for separable queueing networks improve the time and space complexities by substituting approximations for $A_k^{(c)}(\vec{N})$ that are not recursive. Typically, an approximate MVA algorithm approximates $A_k^{(c)}(\vec{N})$ by a function of $Q_k(\vec{N})$ and $Q_{c,k}(\vec{N})$ and other quantities that can be computed without recursion, and the set of equations is solved iteratively. Among all approximate MVA algorithms for separable queueing networks, the Bard-Schweitzer Proportional Estimation (PE) algorithm [25] is a popular algorithm that is currently in wide use [12, 13, 21].

The PE algorithm is based on the approximation

$$Q_{j,k}(\vec{N}-\vec{1}_c) \approx \begin{cases} \frac{N_c-1}{N_c}\cdot Q_{c,k}(\vec{N}) & \text{for } c=j, \\ Q_{j,k}(\vec{N}) & \text{for } c\neq j. \end{cases}$$

Hence, the approximation equation of the PE algorithm is

$$A_k^{(c)}(\vec{N}) = Q_k(\vec{N}-\vec{1}_c) = \sum_{j=1}^{C} Q_{j,k}(\vec{N}-\vec{1}_c) \approx Q_k(\vec{N}) - \frac{1}{N_c}Q_{c,k}(\vec{N}). \tag{12}$$

The system of nonlinear equations (12) and (2) through (6) of the PE algorithm can be solved iteratively by any general purpose numerical techniques, such as successive substitution or Newton's method [4, 21, 25, 35]. Of course, the final solution of any iterative AMVA algorithm should be independent of the numerical technique chosen if the solution exists and is unique.

Eager and Sevcik [12] proved that the PE algorithm has a unique solution, and showed that the successive substitution implementation converges to that solution for single class separable queueing networks. Pattipati et al. [21] established the existence and uniqueness of the solutions of the PE algorithm for multiple class separable queueing networks when the number of customers in each class approaches infinity, and showed that the Newton's method implementation converges to the solution in that case. Furthermore, it has also been proved that the PE algorithm yields pessimistic bounds on performance measures for single class separable queueing networks [12]. In practice, the PE algorithm requires much less execution time and has lower space requirements than the exact MVA algorithm, and yields relatively accurate solutions for separable queueing networks with either large or small populations.

When the PE algorithm is solved by any numerical technique, if it requires $R_{c,k}$, R_c, X_c, $Q_{c,k}$, and Q_k to be stored simultaneously, then the space requirement of the algorithm is at least $(2KC + 2C + K)$. Hence, the space complexity of the PE algorithm is $O(KC)$. Since the system of nonlinear equations of the PE algorithm can be algebraically reduced to equivalent simpler nonlinear systems containing only R_c or X_c terms [21, 25, 33], the space requirement of the algorithm can be reduced to $O(C)$.

When the PE algorithm is solved by successive substitution, the time complexity is $O(KC)$ per iteration [25]. However, it may converge very slowly in some cases although it converges very quickly for most networks [35]. When the PE algorithm is solved by Newton's method, it involves solving a nonlinear system of at least C equations [4, 21, 25], and the time complexity is $O(C^3)$ per iteration [33], but it may fail to converge [35]. Generally, the successive substitution implementation requires less execution time than the Newton's method implementation, particularly for networks with a large number of classes [33, 35]. Wang et al. [33] found the Linearizer algorithm also has these properties.

3 The Queue Line Approximate MVA Algorithm

3.1 Properties of Separable Queueing Networks

A good approximate MVA algorithm should observe the following fundamental properties of separable queueing networks.

Property 1 *For single class separable queueing networks,* $Q_k(1) = \frac{D_k}{D+Z}$ *for* $k = 1, 2, ..., K$.

Proof. For single class separable queueing networks, $Q_k(0) = 0$ for $k = 1, 2, ..., K$. Hence, by equations (1) through (5),

$$Q_k(1) = X(1) \cdot R_k(1) = \frac{D_k \cdot [1 + Q_k(0)]}{Z + \sum_{k=1}^{K} \{D_k \cdot [1 + Q_k(0)]\}} = \frac{D_k}{Z + D}$$

for $k = 1, 2, ..., K$. ■

Property 2 *For multiple class separable queueing networks,*

$$\sum_{k=1}^{K} Q_{c,k}(\vec{N}) = \frac{N_c \cdot R_c(\vec{N})}{Z_c + R_c(\vec{N})} = N_c - \frac{N_c \cdot Z_c}{Z_c + R_c(\vec{N})} = N_c - X_c(\vec{N}) \cdot Z_c \tag{13}$$

for $c = 1, 2, ..., C$.

Proof. By (3) through (5), equation (13) holds. ■

By Property 2, for multiple class separable queueing networks, when $Z_c = 0$, we have $\sum_{k=1}^{K} Q_{c,k}(\vec{N}) = N_c$. Thus, when $Z_c = 0$ for all $c = 1, 2, ..., C$, we obtain $\sum_{k=1}^{K} Q_k(\vec{N}) = N$.

3.2 The Queue Line Algorithm

The Queue Line (QL) algorithm for multiple class separable queueing networks is based on the following approximations:

Approximation 1 (Approximations of the QL Algorithm)

(1) *When* $N_c = 1$ *and* $c = j$, $Q_{j,k}(\vec{N} - \vec{1}_c) = 0$;

(2) *When* $N_c > 1$ *and* $c = j$, $\dfrac{Q_{j,k}(\vec{N}) - Q_{j,k}(\vec{N} - \vec{1}_c)}{1} = \dfrac{Q_{j,k}(\vec{N}) - Q_{j,k}(\vec{N} - (N_c - 1)\vec{1}_c)}{N_c - 1}$;

(3) *When* $N_c \geq 1$ *and* $c \neq j$, $Q_{j,k}(\vec{N} - \vec{1}_c) = Q_{j,k}(\vec{N})$;

where $\vec{N} - \vec{1}_c$ *denotes the population* $\vec{N}$ *with one class c customer removed, and* $\vec{N} - (N_c - 1)\vec{1}_c$ *denotes the population* $\vec{N}$ *with* $N_c - 1$ *class c customers removed.*

Like the PE algorithm, the QL algorithm assumes that removing a class c customer does not affect the proportion of time spent by customers of any other classes at each service center. However, while the PE algorithm assumes the linear relationship between $Q_{c,k}(\vec{N})$

and N_c, the QL algorithm estimates $Q_{c,k}(\vec{N}-\vec{1}_c)$ by linear interpolation between the points $(1, Q_{c,k}(\vec{N}-(N_c-1)\vec{1}_c))$ and $(N_c, Q_{c,k}(\vec{N}))$. To make it applicable, Approximation 1 can be expressed as follows.

Approximation 2 (Refined Approximations of the QL Algorithm)

$$\textit{If}\quad N_c = 1,\quad Q_{j,k}(\vec{N}-\vec{1}_c) = \begin{cases} 0, & c = j, \\ Q_{j,k}(\vec{N}), & c \neq j. \end{cases}$$

$$\textit{If}\quad N_c > 1,\quad Q_{j,k}(\vec{N}-\vec{1}_c) = \begin{cases} \frac{N_c-2}{N_c-1}\cdot Q_{c,k}(\vec{N}) + \frac{1}{N_c-1}\cdot Q_{c,k}(\vec{N}-(N_c-1)\vec{1}_c), & c = j, \\ Q_{j,k}(\vec{N}), & c \neq j. \end{cases}$$

Based on Approximation 1 or 2, the following theorem establish the approximation equation for the QL algorithm.

Theorem 1 *Under Approximation 1 or 2, the approximation equation that replaces equation (1) in the QL algorithm is*

$$\begin{cases} A_k^{(c)}(\vec{N}) &= Q_k(\vec{N}) - Q_{c,k}(\vec{N}), & \textit{when } N_c = 1, \\ A_k^{(c)}(\vec{N}) &= Q_k(\vec{N}) - \frac{1}{N_c-1}\cdot\left\{Q_{c,k}(\vec{N}) - \frac{D_{c,k}\cdot[1+Q_k(\vec{N})-Q_{c,k}(\vec{N})]}{Z_c+\sum_{l=1}^{K}\{D_{c,l}\cdot[1+Q_l(\vec{N})-Q_{c,l}(\vec{N})]\}}\right\} & \\ &= (1-\beta_{c,k})\cdot Q_k(\vec{N}) - \left(\frac{1}{N_c-1}-\beta_{c,k}\right)\cdot Q_{c,k}(\vec{N}) + \beta_{c,k}, & \textit{when } N_c > 1, \end{cases} \tag{14}$$

where

$$\beta_{c,k} = \frac{1}{N_c-1}\cdot\frac{D_{c,k}}{Z_c+\sum_{l=1}^{K}\left\{D_{c,l}\cdot\left[1+Q_l(\vec{N})-Q_{c,l}(\vec{N})\right]\right\}}.$$

Proof. When $N_c = 1$, by Lemma 1 in the Appendix, equation (14) holds. When $N_c > 1$, by Approximation 1 or 2,

$$A_k^{(c)}(\vec{N}) = Q_k(\vec{N}) - \frac{1}{N_c-1}\cdot\left[Q_{c,k}(\vec{N}) - Q_{c,k}(\vec{N}-(N_c-1)\vec{1}_c)\right]. \tag{15}$$

By equations (3), (4), and (5),

$$Q_{c,k}(\vec{N}-(N_c-1)\vec{1}_c) = \frac{R_{c,k}(\vec{N}-(N_c-1)\vec{1}_c)}{Z_c+\sum_{l=1}^{C}R_{c,l}(\vec{N}-(N_c-1)\vec{1}_c)}.$$

Therefore, equation (15) can be rewritten as

$$A_k^{(c)}(\vec{N}) = Q_k(\vec{N}) - \frac{1}{N_c - 1} \cdot \left[Q_{c,k}(\vec{N}) - \frac{R_{c,k}(\vec{N} - (N_c - 1)\vec{1}_c)}{Z_c + \sum_{l=1}^{C} R_{c,l}(\vec{N} - (N_c - 1)\vec{1}_c)} \right].$$

By equation (2) and Lemma 1 in the Appendix,

$$R_{c,l}(\vec{N} - (N_c - 1)\vec{1}_c) = D_{c,l} \cdot \left[1 + Q_l(\vec{N} - N_c\vec{1}_c)\right] = D_{c,l} \cdot \left[1 + Q_l(\vec{N}) - Q_{c,l}(\vec{N})\right]$$

for any center l. Therefore, equation (14) holds for $N_c > 1$. ■

Note that the approximation equation (14) of the QL algorithm satisfies Properties 1 and 2 of the exact solutions of separable queueing networks. Also, when $N_c = 1$, the approximation equation of the PE algorithm is the same as that of the QL algorithm. When $N_c > 1$, the approximation equation of the PE algorithm is equivalent to assuming each of the β-terms in equation (14) is zero.

3.3 Computational Requirements

As with the PE algorithm, there are several implementations for the QL algorithm. The system of nonlinear equations (2) through (6) and (14) can be solved by any general purpose numerical techniques, such as successive substitution or Newton's method. Since both the PE and QL algorithms have the same structure and involve the same number of variables, when they are solved by the same numerical technique, their space requirements are the same. Hence, the space complexity of the QL algorithm is $O(KC)$ if it requires $R_{c,k}$, R_c, X_c, $Q_{c,k}$, and Q_k to be stored simultaneously. As with the PE algorithm, the system of nonlinear equations of the QL algorithm can be algebraically reduced to equivalent simpler nonlinear systems containing only R_c or X_c terms [32]. Hence, the space requirement of the QL algorithm can be reduced to $O(C)$ [32].

When the PE and QL algorithms are both solved by successive substitution, for the network in which $N_c > 1$ for all c, the QL algorithm requires $(10KC - K - C)$ additions/subtractions and $(6KC + C)$ multiplications/divisions during each iteration, while the PE algorithm requires $(4KC - K)$ additions/subtractions and $(3KC + C)$ multiplications/divisions per iteration. For the network in which $N_c = 1$ for all c, the QL algorithm requires the same number of operations as the PE algorithm. Hence, the time complexity of the QL algorithm is $O(KC)$ per iteration and is identical to that of the PE algorithm when both algorithms are solved by successive substitution.

When both algorithms are solved by Newton's method, they involve solving the same number of nonlinear equations. Since the computational cost of Newton's method is proportional to the number of equations to be solved, as with the PE algorithm, the system of nonlinear equations of the QL algorithm should be algebraically reduced to equivalent simpler nonlinear systems containing only R_c or X_c terms before it is solved by Newton's method [32]. Hence, when both the PE and QL algorithms are solved by Newton's method, the time complexity of the QL algorithm is $O(C^3)$ per iteration, same as that of the PE algorithm [32].

Therefore, when both the PE and QL algorithms are solved by the same numerical technique, their computational complexities are the same.

4 Solution Properties and Convergence

4.1 Solution Properties

The existence and uniqueness properties of a solution of an iterative AMVA algorithm are independent of the numerical technique used to solve the algorithm.

Theorem 2 *There exists a unique solution to the QL algorithm for single class separable queueing networks.*

Proof. By equations (2) through (6), all performance measures of interest are uniquely determined by the R_k-terms. If the R_k-terms exist and are unique, then the theorem is established. When $N = 1$, the QL algorithm sets $R_k(N) = D_k$ for all k, and the theorem holds. When $N > 1$, by equation (14), any solution to the QL algorithm must satisfy

$$R_k(N) = D_k \cdot \left[1 + \frac{(N-2) \cdot N \cdot R_k(N)}{(N-1) \cdot [Z + R(N)]} + \frac{D_k}{(N-1) \cdot (D+Z)}\right] \tag{16}$$

for $k = 1, 2, ..., K$. Hence, if $D_k = 0$, then $R_k(N) = 0$. Now consider the case when $D_k > 0$. By equations (2) through (6) and (14), we have $R(N) \geq R_k(N) > D_k > 0$ and $R(N) > D \geq W \geq D_k > 0$. Moreover, equation (16) can be rewritten as

$$R_k(N) \cdot \left\{1 - \frac{N \cdot (N-2) \cdot D_k}{(N-1) \cdot [Z + R(N)]}\right\} = D_k \cdot \left[1 + \frac{D_k}{(N-1) \cdot (D+Z)}\right]. \tag{17}$$

Since both sides of equation (17) are positive, any solution of $R(N)$ must satisfy

$$1 - \frac{N \cdot (N-2) \cdot D_k}{(N-1) \cdot [Z + R(N)]} > 0.$$

Therefore,

$$R(N) > \max\left\{D, \frac{N \cdot (N-2) \cdot W}{N-1} - Z\right\}. \tag{18}$$

Isolating $R_k(N)$ on the left-hand side in (17), summing over k, and applying (3) yields

$$R(N) = \sum_{m=1}^{K} \frac{R(N) \cdot D_m \cdot [(N-1) \cdot (D+Z) + D_m] + Z \cdot D_m[(N-1) \cdot (D+Z) + D_m]}{[Z + R(N)] \cdot (N-1) \cdot (D+Z) - N \cdot (N-2) \cdot D_m \cdot (D+Z)}. \tag{19}$$

Since the only unknown variable in equation (19) is $R(N)$, we define functions g, f_m and h_m with respect to $R(N)$ for $1 \leq m \leq K$ where

$$f_m(R(N)) = \frac{D_m \cdot [(N-1) \cdot (D+Z) + D_m]}{[Z + R(N)] \cdot (N-1) \cdot (D+Z) - N \cdot (N-2) \cdot D_m \cdot (D+Z)},$$

and

$$h_m(R(N)) = \frac{Z \cdot D_m \cdot [(N-1) \cdot (D+Z) + D_m]}{R(N) \cdot \{[Z + R(N)] \cdot (N-1) \cdot (D+Z) - N \cdot (N-2) \cdot D_m \cdot (D+Z)\}},$$

and

$$g(R(N)) = \sum_{m=1}^{K} [f_m(R(N)) + h_m(R(N))].$$

Hence, equation (19) can be rewritten as

$$1 = \sum_{m=1}^{K} [f_m(R(N)) + h_m(R(N))] = g(R(N)). \tag{20}$$

Note that in the region defined by inequality (18), all of g, f_m, and h_m are continuous functions with respect to $R(N)$. It can be verified that $f'_m(R(N)) < 0$ and $h'_m(R(N)) < 0$ for $1 \leq m \leq K$ in the region defined by inequality (18). Therefore, as shown in Figure 1, g is a monotone decreasing function with respect to $R(N)$ in the region defined by inequality (18). Let ω denote the right-hand side of inequality (18), i.e.,

$$\omega = \max\left\{D, \frac{N \cdot (N-2) \cdot W}{N-1} - Z\right\}.$$

It can be verified that

$$\lim_{R(N) \to \omega} g(R(N)) = +\infty,$$

and

$$\lim_{R(N) \to +\infty} g(R(N)) = 0.$$

Since $g(R(N))$ decreases monotonically from $+\infty$ to 0 as $R(N)$ increases from ω to $+\infty$, it must take on the value 1 when $R(N)$ is the unique solution. According to (17), all R_k-terms are uniquely determined by $R(N)$. Therefore, when $N > 1$, all R_k-terms exist and are unique, and hence the theorem holds. ■

The above result for the QL algorithm is similar to the theorem proved by Eager and Sevcik [12] for the PE algorithm because the two algorithms have similar structures. Just as for the PE algorithm, we lack formal proof of the existence and uniqueness of the solutions of the QL algorithm for multiple class separable queueing networks. Empirically the algorithm can always converge to the same solution despite varying the initial values and numerical techniques used to solve them. Hence, the solutions of the QL algorithm appear to always exist and be unique.

4.2 Convergence

The system of nonlinear equations of the QL algorithm can be solved using any numerical techniques such as successive substitution or Newton's method. Let the superscript $[i]$ on

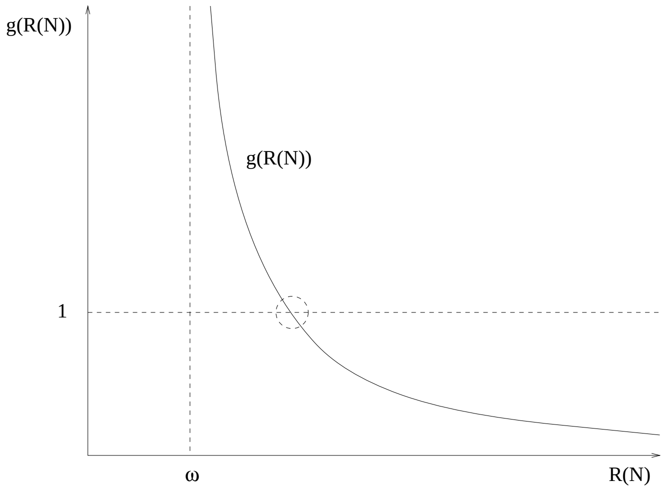

Figure 1: Plot of $g(R(N))$ vs. $R(N)$ for the QL algorithm when $N > 1$ and $D > 0$

the performance estimates denote the quantities resulting from the i^{th} iteration using a particular numerical method, starting from the initial values superscripted by [0].

Theorem 3 *For single class separable queueing networks, suppose that the QL algorithm is solved by successive substitution. The algorithm monotonically converges to the unique solution if initial estimates are such that* $R_b^{[0]}(N) = \frac{N \cdot D}{|\mathbb{B}|}$ *and* $R_p^{[0]}(N) = 0$ *for all* $b \in \mathbb{B}$ *and all* $p \in \overline{\mathbb{B}}$*, where* $|\mathbb{B}|$ *is the number of bottleneck service centers in the network. Particularly, the above initial estimates guarantee that the response time* $R(N)$ *estimated by the QL algorithm monotonically decreases to the unique solution, and the throughput* $X(N)$ *estimated by the QL algorithm monotonically increases to the unique solution.*

Proof. When the QL algorithm is solved by successive substitution with the initial estimates $R_b^{[0]}(N) = \frac{N \cdot D}{|\mathbb{B}|}$ and $R_p^{[0]}(N) = 0$ for all $b \in \mathbb{B}$ and all $p \in \overline{\mathbb{B}}$, it can be verified that the algorithm always converges to the exact solutions for single class separable queueing networks with $N = 1$. For single class separable queueing networks with $N > 1$, the above initial estimates yield $R^{[0]}(N) = N \cdot D$, $Q_b^{[0]}(N) = \frac{N^2 \cdot D}{|\mathbb{B}| \cdot (Z + N \cdot D)}$, and $Q_p^{[0]}(N) = 0$ for all $b \in \mathbb{B}$ and all $p \in \overline{\mathbb{B}}$. By equations (2) through (6) and (14), it can be verified that $R^{[i]}(N) \geq D$ for all $i \geq 0$. Hence, if it can be shown that $R^{[i+1]}(N) \leq R^{[i]}(N)$ for all $i \geq 0$, then by equation (4), $\frac{N}{Z+D} \geq X^{[i+1]}(N) \geq X^{[i]}(N)$ for all $i \geq 0$, and the monotonic convergence will be established.

When $N > 1$, it can be verified that

$$R^{[1]}(N) = D + \frac{D^2}{(N-1)\cdot(D+Z)} + \frac{N^2\cdot(N-2)\cdot D\cdot D_b}{(Z+N\cdot D)\cdot(N-1)} \leq R^{[0]}(N),$$

$$Q_b^{[1]}(N) = \left[D_b + \frac{D_b^2}{(N-1)\cdot(D+Z)} + \frac{N^2\cdot(N-2)\cdot D\cdot D_b}{|\mathbb{B}|\cdot(Z+N\cdot D)\cdot(N-1)}\right]\cdot\left[\frac{N}{Z+R^{[1]}(N)}\right] \leq Q_b^{[0]}(N),$$

and

$$Q_p^{[1]}(N) = \left[D_p + \frac{D_p^2}{(N-1)\cdot(D+Z)}\right]\cdot\left[\frac{N}{Z+R^{[1]}(N)}\right] \geq Q_p^{[0]}(N).$$

Since the estimates resulting from one iteration can be treated as alternative initial estimates superscripted by $[0]'$, Lemma 2 in the Appendix implies that $R^{[i+1]}(N) = R^{[i]'}(N) \leq R^{[i]}(N)$ for all $i \geq 0$. Hence, the theorem is established. ■

For single class separable queueing networks, when the QL algorithm is solved by Newton's method, it involves solving equation $G(R(N)) = 0$ using Newton's method, where $G(R(N)) = g(R(N)) - 1$ and the function $g(R(N))$ has been defined in the proof of Theorem 2. By Theorem 2, there exists a unique solution $R^*(N)$ to $G(R(N)) = 0$. As shown in the proof of Theorem 2, other performance measures of interest are uniquely determined by $R^*(N)$. It can been verified that $G'(R(N)) < 0$ and $G''(R(N)) > 0$. Hence, $G(R(N))$ is a monotonically decreasing and convex function with respect to $R(N)$. Therefore, as shown in Figure 2, Newton's method guarantees the monotonic convergence to the unique solution $R^*(N)$ if the initial estimate $R^{[0]}(N) \in [D, R^*(N)]$.

For multiple class separable queueing networks, we found empirically that the successive substitution implementation of the QL algorithm always converges. However, the Newton's method implementation may fail to converge for some initial estimates because Newton's method is generally sensitive to the initial estimates. In general, the successive substitution implementation requires less execution time than the Newton's method implementation [32].

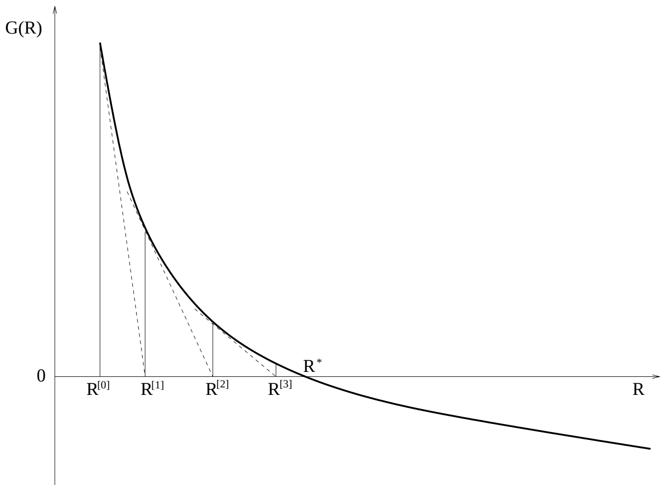

Figure 2: The convergence of Newton's method for $G(R(N)) = 0$

5 Comparison of the Accuracy of the Solutions of the PE and QL Algorithms

The accuracy of the solutions of the PE and QL algorithms is proportional to the accuracy of the approximations of these algorithms to $Q_k(\vec{N} - \vec{1}_c)$. Let the superscript $[ALG]$ on the performance measures at population $\vec{N}$ denote the solutions yielded by an AMVA algorithm ALG, and let the superscript $[ALG]$ on the performance measures at population $\vec{N} - \vec{1}_c$ denote the quantities estimated by the approximations of ALG, where ALG can be the PE or the QL algorithm.

5.1 The Accuracy of the Approximations

The errors in the approximations of the PE or the QL algorithm are defined by

$$E_{j,k}^{(c)[ALG]} = Q_{j,k}^{[ALG]}(\vec{N} - \vec{1}_c) - Q_{j,k}(\vec{N} - \vec{1}_c), \tag{21}$$

and

$$E_{k}^{(c)[ALG]} = Q_{k}^{[ALG]}(\vec{N} - \vec{1}_c) - Q_{k}(\vec{N} - \vec{1}_c). \tag{22}$$

By equation (6),

$$E_k^{(c)[ALG]} = \sum_{j=1}^{C} E_{j,k}^{(c)[ALG]}. \tag{23}$$

Both the PE and QL algorithms assume that removing a class c customer does not affect the proportion of time spent by customers of any other class at each service center. Hence, for $j \neq c$,

$$Q_{j,k}^{[PE]}(\vec{N} - \vec{1}_c) = Q_{j,k}^{[QL]}(\vec{N} - \vec{1}_c) = Q_{j,k}(\vec{N}),$$

and

$$E_{j,k}^{(c)[PE]} = E_{j,k}^{(c)[QL]}. \tag{24}$$

However, the PE and QL algorithms use different approximations for $Q_{c,k}(\vec{N} - \vec{1}_c)$. When $N_c = 1$,

$$Q_{c,k}^{[PE]}(\vec{N} - \vec{1}_c) = Q_{c,k}^{[QL]}(\vec{N} - \vec{1}_c) = Q_{c,k}(\vec{N} - \vec{1}_c) = 0,$$

and

$$E_{c,k}^{(c)[PE]} = E_{c,k}^{(c)[QL]} = 0. \tag{25}$$

When $N_c > 1$, the PE algorithm estimates $Q_{c,k}(\vec{N} - \vec{1}_c)$ by linear interpolation between the points $(0, 0)$ and $(N_c, Q_{c,k}(\vec{N}))$, while the QL algorithm estimates $Q_{c,k}(\vec{N} - \vec{1}_c)$ by linear interpolation between the points $(1, Q_{c,k}(\vec{N} - (N_c - 1)\vec{1}_c))$ and $(N_c, Q_{c,k}(\vec{N}))$. Hence, the relationship between $E_{c,k}^{(c)[PE]}$ and $E_{c,k}^{(c)[QL]}$ depends on the behavior of the function $Q_{c,k}(\vec{n})$ with respect to n_c, where $\vec{n} = (n_1, n_2, ..., n_C)$ is a population vector ranging from $\vec{0}$ to $\vec{N}$.

When $N_c > 1$, $D_{c,1} = D_{c,2} = \cdots = D_{c,K}$, and $Z_c = 0$, by Property 2, $\sum_{i=1}^{K} Q_{c,i}(\vec{n}) = n_c$, and $Q_{c,k}(\vec{n}) = \frac{n_c}{K}$. Therefore, as shown in Figure 3, the function $Q_{c,k}(\vec{n})$ is a linear function with respect to n_c. Because both the PE and QL algorithms assume the same linear function,

$$Q_{c,k}^{[PE]}(\vec{N} - \vec{1}_c) = Q_{c,k}^{[QL]}(\vec{N} - \vec{1}_c) = Q_{c,k}(\vec{N} - \vec{1}_c),$$

and

$$E_{c,k}^{(c)[PE]} = E_{c,k}^{(c)[QL]} = 0. \tag{26}$$

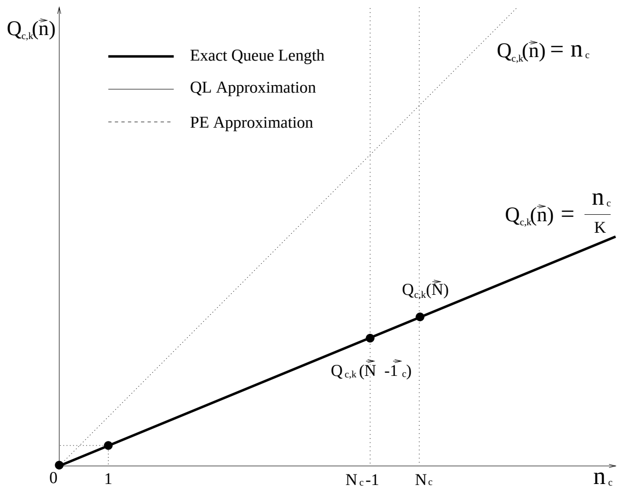

Figure 3: Plot of the exact $Q_{c,k}(\vec{n})$ with respect n_c, and the approximations of the PE and QL algorithms when $D_{c,1} = D_{c,2} = \cdots = D_{c,K}$ and $Z_c = 0$

When $N_c > 1$, $D_{c,1} = D_{c,2} = \cdots = D_{c,K}$, and $Z_c > 0$, $Q_{c,k}(\vec{n}) = \frac{\sum_{i=1}^{K} Q_{c,i}(\vec{n})}{K}$. By Property 2 and equation (7),

$$\sum_{i=1}^{K} Q_{c,i}(\vec{n}) - \sum_{i=1}^{K} Q_{c,i}(\vec{n} - \vec{1}_c) = 1 + Z_c \cdot [X_c(\vec{n} - \vec{1}_c) - X_c(\vec{n})] < 1. \tag{27}$$

Moreover, by equation (8),

$$\sum_{i=1}^{K} Q_{c,i}(\vec{n}) - \sum_{i=1}^{K} Q_{c,i}(\vec{n} - \vec{1}_c) > 0.$$

Furthermore, when $C = 1$ or $N_c \to +\infty$, by equations (9), (10), and (27),

$$\sum_{i=1}^{K} Q_{c,i}(\vec{n} + \vec{1}_c) - \sum_{i=1}^{K} Q_{c,i}(\vec{n}) \geq \sum_{i=1}^{K} Q_{c,i}(\vec{n}) - \sum_{i=1}^{K} Q_{c,i}(\vec{n} - \vec{1}_c).$$

Therefore,

$$0 < Q_{c,k}(\vec{n}) - Q_{c,k}(\vec{n} - \vec{1}_c) < \frac{1}{K},$$

and

$$Q_{c,k}(\vec{n} + \vec{1}_c) - Q_{c,k}(\vec{n}) \geq Q_{c,k}(\vec{n}) - Q_{c,k}(\vec{n} - \vec{1}_c).$$

Hence, as shown in Figure 4, the function $Q_{c,k}(\vec{n})$ with respect to n_c is a monotonically increasing convex function. Therefore,

$$Q_{c,k}^{[PE]}(\vec{N} - \vec{1}_c) \geq Q_{c,k}^{[QL]}(\vec{N} - \vec{1}_c) \geq Q_{c,k}(\vec{N} - \vec{1}_c),$$

and

$$E_{c,k}^{(c)[PE]} \geq E_{c,k}^{(c)[QL]} \geq 0. \tag{28}$$

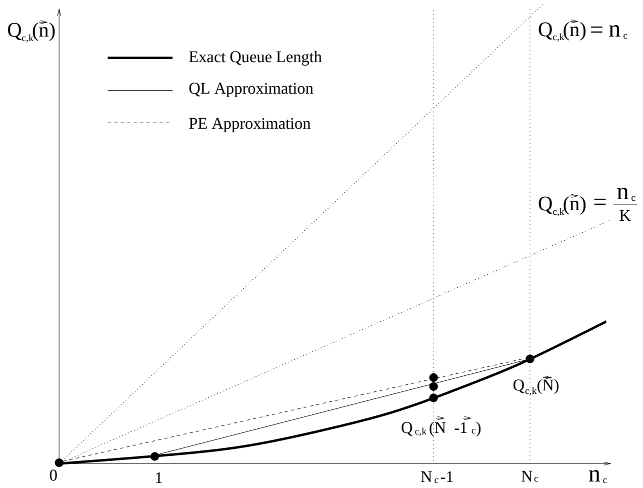

Figure 4: Plot of the exact $Q_{c,k}(\vec{n})$ with respect n_c, and the approximations of the PE and QL algorithms when $D_{c,1} = D_{c,2} = \cdots = D_{c,K}$ and $Z_c > 0$

Now consider the case when $N_c > 1$ and $D_{c,i} \neq D_{c,j}$ for some i and j. Let $b \in \mathbb{B}_c$ and $p \in \overline{\mathbb{B}}_c$. Then $D_{c,b} > D_{c,p}$. By equations (1) through (8),

$$Q_{c,b}(\vec{n}) - Q_{c,b}(\vec{n} - \vec{1}_c) > Q_{c,p}(\vec{n}) - Q_{c,p}(\vec{n} - \vec{1}_c) > 0.$$

Moreover, by (7), (8), and Property 2,

$$0 < \sum_{i=1}^{K} Q_{c,i}(\vec{n}) - \sum_{i=1}^{K} Q_{c,i}(\vec{n} - \vec{1}_c) < 1. \tag{29}$$

Furthermore, by equations (9), (10), and (29), when $C = 1$ or $N_c \to +\infty$,

$$\sum_{i=1}^{K} Q_{c,i}(\vec{n} + \vec{1}_c) - \sum_{i=1}^{K} Q_{c,i}(\vec{n}) \geq \sum_{i=1}^{K} Q_{c,i}(\vec{n}) - \sum_{i=1}^{K} Q_{c,i}(\vec{n} - \vec{1}_c).$$

Since $Q_{c,b}(\vec{n}) = Q_{c,p}(\vec{n}) = 0$ when $n_c = 0$, as shown in Figure 5, it can be verified that

$$0 < Q_{c,b}(\vec{n}) - Q_{c,b}(\vec{n} - \vec{1}_c) \leq Q_{c,b}(\vec{n} + \vec{1}_c) - Q_{c,b}(\vec{n}) < 1,$$

and

$$1 > Q_{c,p}(\vec{n}) - Q_{c,p}(\vec{n} - \vec{1}_c) \geq Q_{c,p}(\vec{n} + \vec{1}_c) - Q_{c,p}(\vec{n}) > 0,$$

which means the function $Q_{c,b}(\vec{n})$ with respect to n_c is a monotonically increasing convex function and the function $Q_{c,p}(\vec{n})$ with respect to n_c is a monotonically increasing concave function when $C = 1$ or $N_c \to +\infty$. When $k \in \mathbb{B}_c$, as shown in Figure 6, it can be verified that

$$Q_{c,k}^{[PE]}(\vec{N} - \vec{1}_c) \geq Q_{c,k}^{[QL]}(\vec{N} - \vec{1}_c) \geq Q_{c,k}(\vec{N} - \vec{1}_c),$$

and

$$E_{c,k}^{(c)[PE]} \geq E_{c,k}^{(c)[QL]} \geq 0, \tag{30}$$

When $k \in \overline{\mathbb{B}}_c$, as shown in Figure 7, it can be verified that

$$Q_{c,k}^{[PE]}(\vec{N} - \vec{1}_c) \leq Q_{c,k}^{[QL]}(\vec{N} - \vec{1}_c) \leq Q_{c,k}(\vec{N} - \vec{1}_c),$$

and

$$E_{c,k}^{(c)[PE]} \leq E_{c,k}^{(c)[QL]} \leq 0. \tag{31}$$

Therefore, when $C = 1$ or $N_c \to +\infty$,

$$\left|E_{c,k}^{(c)[PE]}\right| \geq \left|E_{c,k}^{(c)[QL]}\right| \tag{32}$$

and

$$E_{c,k}^{(c)[PE]} E_{c,k}^{(c)[QL]} \geq 0. \tag{33}$$

For single class separable queueing networks, equations (32) and (33) also establish that the error associated with the approximation of the QL algorithm to $Q_k(\vec{N} - \vec{1}_c)$ never exceeds that of the PE algorithm. The following two theorems further establish that the same relationship still holds between the error associated with the approximation of the QL algorithm to $Q_k(\vec{N} - \vec{1}_c)$ and that of the PE algorithm for multiple class separable queueing networks.

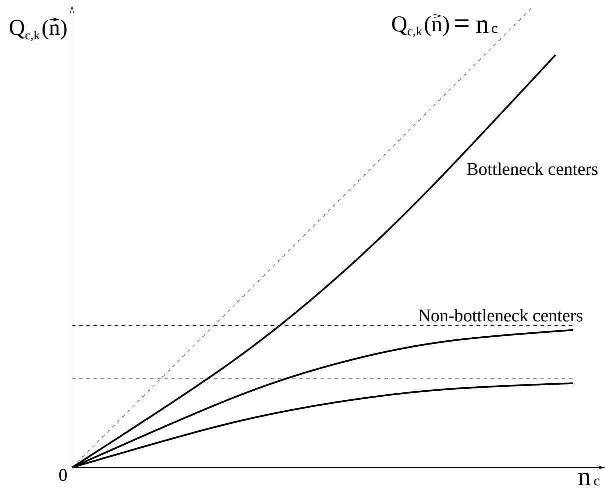

Figure 5: Plot of the exact $Q_{c,k}(\vec{n})$ with respect to n_c when all service centers can be classified as bottleneck centers and non-bottleneck centers for class c customers

Theorem 4 *For multiple class separable queueing networks, when* $N_c = 1$*,*

$$E_k^{(c)[PE]} = E_k^{(c)[QL]}.$$

Proof. The proof follows from equations (23) through (25). ■

Theorem 5 *For multiple class separable queueing networks, there exists an integer* N^* *such that if* $N_c > N^*$ *then*

$$\left| E_k^{(c)[PE]} \right| \geq \left| E_k^{(c)[QL]} \right|$$

and

$$E_k^{(c)[PE]} E_k^{(c)[QL]} \geq 0.$$

Proof. We will show that the theorem holds for each of the following cases: **(1)** $E_{c,k}^{(c)[PE]} = E_{c,k}^{(c)[QL]} = 0$; **(2)** $E_{c,k}^{(c)[PE]} \geq E_{c,k}^{(c)[QL]} \geq 0$; **(3)** $E_{c,k}^{(c)[PE]} \leq E_{c,k}^{(c)[QL]} \leq 0$. By (32) and (33), these cases include all possible relationships between $E_{c,k}^{(c)[PE]}$ and $E_{c,k}^{(c)[QL]}$.

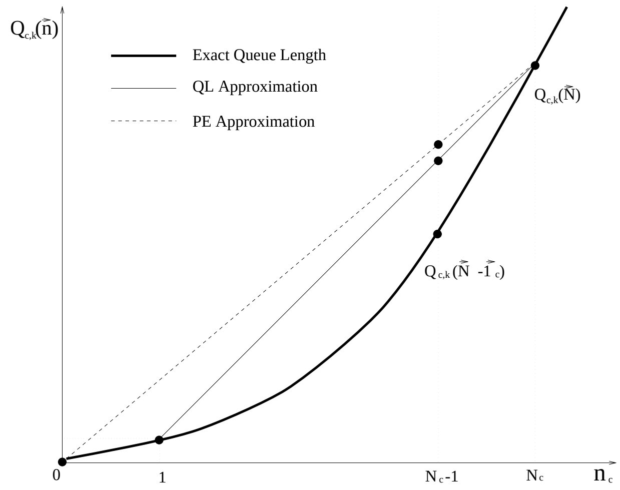

Figure 6: Plot of the exact $Q_{c,k}(\vec{n})$ with respect to n_c and the approximations of the PE and QL algorithms when $k \in \mathbb{B}_c$

When $E_{c,k}^{(c)[PE]} = E_{c,k}^{(c)[QL]} = 0$, by equations (23) and (24), $E_k^{(c)[PE]} = E_k^{(c)[QL]}$, and the theorem is established in this case.

Now consider the cases when $E_{c,k}^{(c)[PE]} \geq E_{c,k}^{(c)[QL]} \geq 0$ or $E_{c,k}^{(c)[PE]} \leq E_{c,k}^{(c)[QL]} \leq 0$. By equations (9) and (24),

$$\lim_{N_c \to +\infty} E_{j,k}^{(c)[PE]} = \lim_{N_c \to +\infty} E_{j,k}^{(c)[QL]} = \lim_{N_c \to +\infty} [Q_{j,k}(\vec{N}) - Q_{j,k}(\vec{N} - \vec{1}_c)] = 0$$

for $j \neq c$. Hence,

$$\lim_{N_c \to +\infty} \sum_{j \neq c} E_{j,k}^{(c)[PE]} = \lim_{N_c \to +\infty} \sum_{j \neq c} E_{j,k}^{(c)[QL]} = 0.$$

Therefore, by equation (23),

$$\lim_{N_c \to +\infty} E_k^{(c)[PE]} = \lim_{N_c \to +\infty} E_{c,k}^{(c)[PE]}$$

and

$$\lim_{N_c \to +\infty} E_k^{(c)[QL]} = \lim_{N_c \to +\infty} E_{c,k}^{(c)[QL]}.$$

By the definition of limit, there exists an integer N^* such that if $N_c > N^*$ then

$$E_k^{(c)[PE]} = E_{c,k}^{(c)[PE]}$$

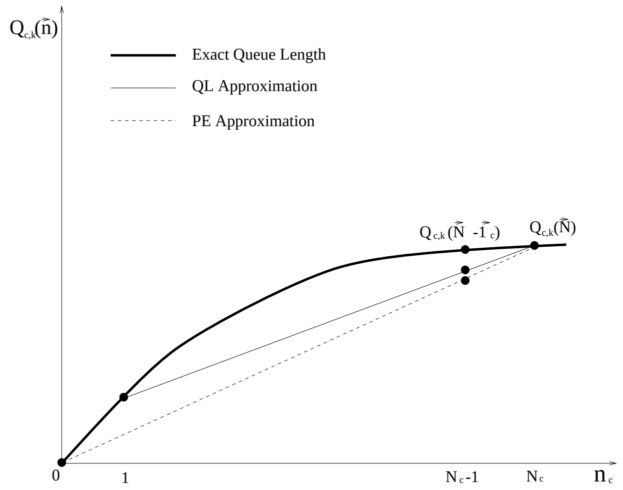

Figure 7: Plot of the exact $Q_{c,k}(\vec{n})$ with respect to n_c and the approximations of the PE and QL algorithms when $k \in \overline{\mathbb{B}}_c$

and

$$E_k^{(c)[QL]} = E_{c,k}^{(c)[QL]}.$$

Therefore, by equations (32) and (33), this theorem is established. ■

5.2 The Accuracy of the Solutions

Since the error associated with the approximation of the QL algorithm to $Q_k(\vec{N} - \vec{1}_c)$ never exceeds that of the PE algorithm, we expect that the solution of the QL algorithm is more accurate than that of the PE algorithm. In fact, both the QL and PE algorithm yield the exact solution for single class separable queueing network with population one. For single class separable queueing networks with population two, the QL algorithm yields the exact solution, while the PE algorithm yields a higher mean response time and a lower throughput [12]. Next, we investigate the relationship between the errors of the solutions of the PE and QL algorithms for general separable queueing networks.

By equations (2) and (22),

$$R_{c,k}^{[PE]}(\vec{N}) = D_{c,k} \cdot \left[1 + Q_k(\vec{N} - \vec{1}_c) + E_k^{(c)[PE]}\right], \tag{34}$$

$$R_{c,k}^{[QL]}(\vec{N}) = D_{c,k} \cdot \left[1 + Q_k(\vec{N} - \vec{1}_c) + E_k^{(c)[QL]}\right]. \tag{35}$$

Hence,

$$\left|R_{c,k}^{[PE]}(\vec{N}) - R_{c,k}(\vec{N})\right| = D_{c,k} \cdot \left|E_k^{(c)[PE]}\right|, \tag{36}$$

and

$$\left|R_{c,k}^{[QL]}(\vec{N}) - R_{c,k}(\vec{N})\right| = D_{c,k} \cdot \left|E_k^{(c)[QL]}\right|. \tag{37}$$

Therefore, for single class separable queueing networks, by equations (32) and (33),

$$\left|R_{c,k}^{[PE]}(\vec{N}) - R_{c,k}(\vec{N})\right| \geq \left|R_{c,k}^{[QL]}(\vec{N}) - R_{c,k}(\vec{N})\right|.$$

Also, by Theorem 4, for multiple class separable queueing networks, when $N_c = 1$,

$$\left|R_{c,k}^{[PE]}(\vec{N}) - R_{c,k}(\vec{N})\right| = \left|R_{c,k}^{[QL]}(\vec{N}) - R_{c,k}(\vec{N})\right|. \tag{38}$$

By Theorem 5, for multiple class separable queueing networks, there exists an integer N^* such that if $N_c > N^*$ then

$$\left|R_{c,k}^{[PE]}(\vec{N}) - R_{c,k}(\vec{N})\right| \geq \left|R_{c,k}^{[QL]}(\vec{N}) - R_{c,k}(\vec{N})\right|.$$

Theorem 6 *For single class separable queueing networks,*

$$R_c^{[PE]}(\vec{N}) \geq R_c^{[QL]}(\vec{N}) \geq R_c(\vec{N}),$$

$$\left|R_c^{[PE]}(\vec{N}) - R_c(\vec{N})\right| \geq \left|R_c^{[QL]}(\vec{N}) - R_c(\vec{N})\right|,$$

$$X_c^{[PE]}(\vec{N}) \leq X_c^{[QL]}(\vec{N}) \leq X_c(\vec{N}),$$

and

$$\left|X_c^{[PE]}(\vec{N}) - X_c(\vec{N})\right| \geq \left|X_c^{[QL]}(\vec{N}) - X_c(\vec{N})\right|.$$

Proof. When $C = 1$, $E_k^{(c)[PE]} = E_{c,k}^{(c)[PE]}$ and $E_k^{(c)[QL]} = E_{c,k}^{(c)[QL]}$. Hence, by equation (4) and Lemma 3 in the Appendix, this theorem holds. ■

Theorem 7 *For multiple class separable queueing networks, there exist integers N_c^* for $c = 1, 2, ..., C$, such that if $N_c > N_c^*$ for all $c = 1, 2, ..., C$, then*

$$R_c^{[PE]}(\vec{N}) \geq R_c^{[QL]}(\vec{N}) \geq R_c(\vec{N}),$$

$$\left| R_c^{[PE]}(\vec{N}) - R_c(\vec{N}) \right| \geq \left| R_c^{[QL]}(\vec{N}) - R_c(\vec{N}) \right|,$$

$$X_c^{[PE]}(\vec{N}) \leq X_c^{[QL]}(\vec{N}) \leq X_c(\vec{N}),$$

and

$$\left| X_c^{[PE]}(\vec{N}) - X_c(\vec{N}) \right| \geq \left| X_c^{[QL]}(\vec{N}) - X_c(\vec{N}) \right|.$$

Proof. By equations (3), (34), and (35),

$$R_c^{[PE]}(\vec{N}) = R_c(\vec{N}) + \sum_{k=1}^{K} D_{c,k} E_k^{(c)[PE]},$$

and

$$R_c^{[QL]}(\vec{N}) = R_c(\vec{N}) + \sum_{k=1}^{K} D_{c,k} E_k^{(c)[QL]}.$$

Because by equations (24) and (9),

$$\lim_{N_c \to +\infty} E_{j,k}^{(c)[PE]} = \lim_{N_c \to +\infty} E_{j,k}^{(c)[QL]} = \lim_{N_c \to +\infty} [Q_{j,k}(\vec{N}) - Q_{j,k}(\vec{N} - \vec{1}_c)] = 0$$

for $j \neq c$, we have

$$\lim_{N_c \to +\infty} E_k^{(c)[PE]} = \lim_{N_c \to +\infty} \left[E_{c,k}^{(c)[PE]} + \sum_{j \neq c} E_{j,k}^{(c)[PE]} \right] = \lim_{N_c \to +\infty} E_{c,k}^{(c)[PE]}$$

and

$$\lim_{N_c \to +\infty} E_k^{(c)[QL]} = \lim_{N_c \to +\infty} \left[E_{c,k}^{(c)[QL]} + \sum_{j \neq c} E_{j,k}^{(c)[QL]} \right] = \lim_{N_c \to +\infty} E_{c,k}^{(c)[QL]}$$

for all $c = 1, 2, ..., C$. Hence, by Lemma 3 in the Appendix and the definition of limit, this theorem holds. ■

By Theorems 6 and 7, for single class separable queueing networks or multiple class separable queueing networks with large populations, the solutions of both the QL and PE algorithms are pessimistic relative to the exact solution. In particular, both the QL and PE algorithm yield a higher mean system response time and a lower throughput, and the solution of the QL algorithm is more accurate than that of the PE algorithm. These results are consistent with the analytic results obtained by Eager and Sevcik for the PE algorithm [12].

Although Theorem 7 is an asymptotic result applying only to networks with sufficiently large populations, our experimentation supports that the QL algorithm is consistently more accurate than the PE algorithm even for multiple class networks with small populations [34]. However, although the QL algorithm never yields larger errors than the PE algorithm, the two algorithms converge as population increases.

6 Summary and Conclusions

This paper has presented a new AMVA algorithm, the Queue Line (QL) algorithm. The QL algorithm is more accurate than, yet has the same computational complexity as the PE algorithm, the most popular and most widely used AMVA algorithm.

We proved that the solution of the QL algorithm exists and is unique. For single class separable queueing networks and multiple class separable queueing networks with large population, we showed that the solutions of both the PE and QL algorithms are pessimistic relative to the exact solution. In particular, both algorithms yield a higher mean system response time and a lower throughput. The solution of the QL algorithm is always more accurate than that of the PE algorithm.

As with the PE algorithm, the system of nonlinear equations of the QL algorithm can be solved by any general purpose numerical techniques such as successive substitution or Newton's method. We established the convergence of different implementations of the QL algorithm, and showed that the QL algorithm has the same computational complexity as the PE algorithm when they are solved by the same numerical technique.

Therefore, the QL algorithm dominates the PE algorithm in the spectrum of AMVA algorithms that trade off accuracy and efficiency.

References

[1] Y. Bard, "Some extensions to multiclass queueing network analysis," M. Arato, A. Butrimenko and E. Gelenbe, eds., *Performance of Computer Systems*, pp. 51-62, Amsterdam, Netherlands: North-Holland, 1979.

[2] F. Baskett, K. M. Chandy, R. R. Muntz and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *Journal of the ACM*, vol. 22, no. 2, pp. 248-260, April 1975.

[3] J. P. Buzen, "Computational algorithms for closed queueing networks with exponential servers," *Communications of the ACM*, vol. 16, no. 9, pp. 527-531, September 1973.

[4] W.-M. Chow, "Approximations for large scale closed queueing networks," *Performance Evaluation*, vol. 3, no. 1, pp. 1-12, 1983.

[5] K. M. Chandy and D. Neuse, "Linearizer: A heuristic algorithm for queueing network models of computing systems," *Communications of the ACM*, vol. 25, no. 2, pp. 126-134, February 1982.

[6] K. M. Chandy and C. H. Sauer, "Computational algorithms for product form queueing networks," *Communications of the ACM*, vol. 23, no. 10, pp. 573-583, October 1980.

[7] A. E. Conway and N. D. Georganas, "RECAL – A new efficient algorithm for the exact analysis of multiple-chain closed queueing networks," *Journal of the ACM*, vol. 33, no. 4, pp. 768-791, October 1986.

[8] A. E. Conway, E. de Souza e Silva and S. S. Lavenberg, "Mean value analysis by chain product form queueing networks," *IEEE Transactions on Computers*, vol. C-38, no. 3, pp. 432-442, March 1989.

[9] P. Cremonesi, P. J. Schweitzer, G. Serazzi and F. Schiavoni, "An interpolation-matching framework for the approximate solution of closed multiclass queueing networks," Technical Report, 1999.

[10] L. W. Dowdy, D. L. Eager, K. D. Gordon and L. V. Saxton, "Throughput concavity and response time convexity," *Information Processing Letters*, vol. 19, no. 4, pp. 209-212, November 1984.

[11] D. L. Eager, *Bounding Algorithms for Queueing Network Models of Computer Systems*, Ph.D. Thesis, Tech. Rept. CSRG-156, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 1984.

[12] D. L. Eager and K. C. Sevcik, "Analysis of an approximation algorithm for queueing networks," *Performance Evaluation*, vol. 4, no. 4, pp. 275-284, 1984.

[13] D. L. Eager and K. C. Sevcik, "Bound hierarchies for multiple-class queueing networks," *Journal of the ACM*, vol. 33, no. 1, pp. 179-206, January 1986.

[14] K. P. Hoyme, S. C. Bruell, P. V. Afshari and R. Y. Kain, "A tree-structured mean value analysis algorithm," *ACM Transactions on Computer Systems*, vol. 4, no. 2, pp. 178-185, May 1986.

[15] C. T. Hsieh and S. S. Lam, "PAM – A noniterative approximate solution method for closed multichain queueing networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 16, no. 1, pp. 261-269, May 1988.

[16] S. S. Lam, "A simple derivation of the MVA and LBANC algorithms from the convolution algorithm," *IEEE Transactions on Computers*, vol. C-32, no. 11, pp. 1062-1064, November 1983.

[17] S. S. Lam and Y. L. Lien, "A tree convolution algorithm for the solution of queueing networks," *Communications of the ACM*, vol. 26, no. 3, pp. 203-215, March 1983.

[18] S. S. Lavenberg and M. Reiser, "Stationary state probabilities of arrival instants for closed queueing networks with multiple types of customers," *Journal of Applied Probability*, vol. 17, no. 4, pp. 1048-1061, December 1980.

[19] E. D. Lazowska, G. S. Graham, J. Zahorjan, and K. C. Sevcik, *Quantitative System Performance: Computer System Analysis Using Queueing Network Models*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[20] J. D. C. Little, "A proof of the queueing formula $L = \lambda W$," *Operations Research*, vol. 9, no. 3, pp. 383-387, May/June 1961.

[21] K. R. Pattipati, M. M. Kostreva and J. L. Teele, "Approximate mean value analysis algorithms for queueing networks: existence, uniqueness, and convergence results," *Journal of the ACM*, vol. 37, no. 3, pp. 643-673, July 1990.

[22] M. Reiser and H. Kobayashi, "Queueing networks with multiple closed chains: Theory and computational algorithms," *IBM Journal of Research and Development*, vol. 19, no. 3, pp. 283-294, May 1975.

[23] M. Reiser and H. Kobayashi, "On the convolution algorithm for separable queueing networks," *Proceedings of International Symposium on Computer Performance Measurement, Modeling, and Evaluation*, pp. 109-117, Cambridge, Great Britain, 1976.

[24] M. Reiser and S. S. Lavenberg, "Mean value analysis of closed multichain queueing networks," *Journal of the ACM*, vol. 27, no. 2, pp. 313-322, April 1980.

[25] P. J. Schweitzer, "Approximate analysis of multiclass closed networks of queues," *Proceedings of International Conference on Stochastic Control and Optimization*, pp. 25-29, Amsterdam, Netherlands, 1979.

[26] P. J. Schweitzer, G. Serazzi and M. Broglia, "A queue-shift approximation technique for product-form queueing networks," *Proceedings of the International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, R. Puigjaner,

N. N. Savino and B. Serra, eds., *Lecture Notes in Computer Science*, vol. 1469, pp. 267-279, New York, NY: Springer, 1998.

[27] K. C. Sevcik and I. Mitrani, "The distribution of queueing network states at input and output instants," *Journal of the ACM*, vol. 28, no. 2, pp. 358-371, April 1981.

[28] E. de Souza e Silva, S. S. Lavenberg and R. R. Muntz, "A clustering approximation technique for queueing network models with a large number of chains," *IEEE Transactions on Computers*, vol. C-35, no. 5, pp. 419-430, May 1986.

[29] E. de Souza e Silva and R. R. Muntz, "A note on the computational cost of the linearizer algorithm for queueing networks," *IEEE Transactions on Computers*, vol. 39, no. 6, pp. 840-842, June 1990.

[30] R. Suri, "A concept of monotonicity and its characterization for closed queueing networks," *Operations Research*, vol. 33, no. 3, pp. 606-624, May-June 1985.

[31] S. Tucci and C. H. Sauer, "The tree MVA algorithm," *Performance Evaluation*, vol. 5, no. 3, pp. 187-196, 1985.

[32] H. Wang, *Approximate MVA Algorithms for Solving Queueing Network Models*, M.Sc. Thesis, Tech. Rept. CSRG-360, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 1997.

[33] H. Wang, K. R. Jackson, K. C. Sevcik and G. Serazzi, "Numerical properties and computational costs of approximate mean value analysis algorithms," submitted for publication, 2000.

[34] H. Wang and K. C. Sevcik, "Experiments with improved approximate mean value analysis algorithms," *Performance Evaluation*, vol. 39, no. 1-4, pp. 189-206, 2000.

[35] J. Zahorjan, D. L. Eager and H. M. Sweillam, "Accuracy, speed, and convergence of approximate mean value analysis," *Performance Evaluation*, vol. 8, no. 4, pp. 255-270, 1988.

Appendix

This appendix states and proves required lemmas.

Lemma 1 *Under Approximation 1 or 2,* $Q_k(\vec{N} - N_c\vec{1}_c) = Q_k(\vec{N}) - Q_{c,k}(\vec{N})$.

Proof. By Approximation 1 or 2, when $c \neq j$, $Q_{j,k}(\vec{N} - i \cdot \vec{1}_c) = Q_{j,k}(\vec{N})$ for $1 \leq i \leq N_c$. Moreover, when $c = j$, $Q_{j,k}(\vec{N} - N_c \cdot \vec{1}_c) = 0$. Hence, by equation (6), the lemma holds. ■

Lemma 2 *For single class separable queueing networks with* $N > 1$, *suppose that the QL algorithm is solved by successive substitution. If two sets of initial performance measure estimates, superscripted by* $[0]$ *and* $[0]'$, *are such that*

$$R^{[0]'}(N) \leq R^{[0]}(N),$$

$$\forall b \in \mathbb{B} : Q_b^{[0]'}(N) \leq Q_b^{[0]}(N),$$

$$\forall p \in \overline{\mathbb{B}} : Q_p^{[0]'}(N) \geq Q_p^{[0]}(N),$$

then for all $I \geq 0$,

$$R^{[I]'}(N) \leq R^{[I]}(N), \tag{39}$$

$$\forall b \in \mathbb{B} : Q_b^{[I]'}(N) \leq Q_b^{[I]}(N), \tag{40}$$

$$\forall p \in \overline{\mathbb{B}} : Q_p^{[I]'}(N) \geq Q_p^{[I]}(N). \tag{41}$$

Proof. The proof is by induction on I.

By assumption, the lemma holds for $I = 0$. Assume that the lemma holds for $I = i (i \geq 0)$. Consider the lemma for $I = i + 1$. Since inequality (39) holds for $I = i$,

$$\frac{N \cdot R^{[i]'}(N)}{Z + R^{[i]'}(N)} \leq \frac{N \cdot R^{[i]}(N)}{Z + R^{[i]}(N)}.$$

By equations (2) through (6), it is equivalent to

$$\sum_{k=1}^{K} Q_k^{[i]'}(N) \leq \sum_{k=1}^{K} Q_k^{[i]}(N).$$

Hence,

$$\sum_{\forall b \in \mathbb{B}} \left[Q_b^{[i]}(N) - Q_b^{[i]\prime}(N) \right] - \sum_{\forall p \in \overline{\mathbb{B}}} \left[Q_p^{[i]\prime}(N) - Q_p^{[i]}(N) \right] \geq 0. \tag{42}$$

Since inequalities (40) and (41) hold for $I = i$, each term in (42) is non-negative. Moreover, since $D_b \geq D_p \geq 0$ for all $b \in \mathbb{B}$ and all $p \in \overline{\mathbb{B}}$, (42) implies

$$\sum_{\forall b \in \mathbb{B}} D_b \cdot \left[Q_b^{[i]}(N) - Q_b^{[i]\prime}(N) \right] - \sum_{\forall p \in \overline{\mathbb{B}}} D_p \cdot \left[Q_p^{[i]\prime}(N) - Q_p^{[i]}(N) \right] \geq 0.$$

By equations (2) through (6) and (14), it is equivalent to

$$\begin{array}{rl} & \sum_{k=1}^{K} D_k \cdot \left[1 + \left(\frac{N-2}{N-1} \right) \cdot Q_k^{[i]\prime}(N) + \frac{D_k}{(N-1)(D+Z)} \right] \\ \leq & \sum_{k=1}^{K} D_k \cdot \left[1 + \left(\frac{N-2}{N-1} \right) \cdot Q_k^{[i]}(N) + \frac{D_k}{(N-1)(D+Z)} \right], \end{array}$$

and thus inequality (39) holds for $I = i+1$. Moreover, since (39) holds for $I = i+1$,

$$\sum_{k=1}^{K} \frac{N \cdot R_k^{[i+1]\prime}(N)}{Z + R^{[i+1]\prime}(N)} \leq \sum_{k=1}^{K} \frac{N \cdot R_k^{[i+1]}(N)}{Z + R^{[i+1]}(N)}. \tag{43}$$

From inequality (43), it can be verified that inequalities (40) or (41) could be violated for $I = i+1$ only if

$$\frac{R_p^{[i+1]\prime}(N)}{Z + R^{[i+1]\prime}(N)} < \frac{R_p^{[i+1]}(N)}{Z + R^{[i+1]}(N)}$$

for some $p \in \overline{\mathbb{B}}$. By equations (2) through (6) and (14), it is equivalent to

$$\frac{1 + \left(\frac{N-2}{N-1} \right) \cdot Q_p^{[i]\prime}(N) + \frac{D_p}{(N-1)(D+Z)}}{1 + \left(\frac{N-2}{N-1} \right) Q_p^{[i]}(N) + \frac{D_p}{(N-1)(D+Z)}} < \frac{Z + R^{[i+1]\prime}(N)}{Z + R^{[i+1]}(N)} \tag{44}$$

However, the left-hand side of (44) is at least one, and the right-hand side is at most one. Hence, it is a contradiction, and thus inequalities (40) and (41) hold for $I = i+1$. Therefore, the lemma holds. ■

Lemma 3 *If* $E_k^{(c)[PE]} = E_{c,k}^{(c)[PE]}$ *and* $E_k^{(c)[QL]} = E_{c,k}^{(c)[QL]}$, *then*

$$R_c^{[PE]}(\vec{N}) \geq R_c^{[QL]}(\vec{N}) \geq R_c(\vec{N}).$$

Proof. By equations (3), (34) and (35),

$$R_c^{[PE]}(\vec{N}) = R_c(\vec{N}) + \sum_{k=1}^{K} D_{c,k} E_k^{(c)[PE]}, \tag{45}$$

and

$$R_c^{[QL]}(\vec{N}) = R_c(\vec{N}) + \sum_{k=1}^{K} D_{c,k} E_k^{(c)[QL]}. \tag{46}$$

In the remainder of this proof, we will show that the theorem holds for each of the following cases: **(1)** $N_c = 1$; **(2)** $N_c > 1$, $D_{c,1} = D_{c,2} = \cdots = D_{c,K}$, and $Z_c = 0$; **(3)** $N_c > 1$, $D_{c,1} = D_{c,2} = \cdots = D_{c,K}$, and $Z_c > 0$; **(4)** $N_c > 1$ and $D_{c,i} \neq D_{c,j}$ for some i and j.

Case (1): When $N_c = 1$, if $E_k^{(c)[PE]} = E_{c,k}^{(c)[PE]}$ and $E_k^{(c)[QL]} = E_{c,k}^{(c)[QL]}$, then by equation (25),

$$\sum_{k=1}^{K} D_{c,k} E_k^{(c)[PE]} = \sum_{k=1}^{K} D_{c,k} E_k^{(c)[QL]} = 0.$$

Therefore, by equations (45) and (46), $R_c^{[PE]}(\vec{N}) = R_c^{[QL]}(\vec{N}) = R_c(\vec{N})$, and the theorem holds in this case.

Case (2): When $N_c > 1$, $D_{c,1} = D_{c,2} = \cdots = D_{c,K}$, and $Z_c = 0$, if $E_k^{(c)[PE]} = E_{c,k}^{(c)[PE]}$ and $E_k^{(c)[QL]} = E_{c,k}^{(c)[QL]}$, then by equation (26),

$$\sum_{k=1}^{K} D_{c,k} E_k^{(c)[PE]} = \sum_{k=1}^{K} D_{c,k} E_k^{(c)[QL]} = 0.$$

Therefore, by equations (45) and (46), $R_c^{[PE]}(\vec{N}) = R_c^{[QL]}(\vec{N}) = R_c(\vec{N})$, and the theorem holds in this case.

Case (3): When $N_c > 1$, $D_{c,1} = D_{c,2} = \cdots = D_{c,K}$, and $Z_c > 0$, if $E_k^{(c)[PE]} = E_{c,k}^{(c)[PE]}$ and $E_k^{(c)[QL]} = E_{c,k}^{(c)[QL]}$, then by equation (28),

$$\sum_{k=1}^{K} D_{c,k} E_k^{(c)[PE]} \geq \sum_{k=1}^{K} D_{c,k} E_k^{(c)[QL]} \geq 0.$$

Therefore, by equations (45) and (46), $R_c^{[PE]}(\vec{N}) \geq R_c^{[QL]}(\vec{N}) \geq R_c(\vec{N})$, and the theorem holds in this case.

Case (4): When $N_c > 1$ and $D_{c,i} \neq D_{c,j}$ for some i and j, by (45) and (46),

$$R_c^{[PE]}(\vec{N}) - R_c^{[QL]}(\vec{N}) = \sum_{k=1}^{K} D_{c,k} E_k^{(c)[PE]} - \sum_{k=1}^{K} D_{c,k} E_k^{(c)[QL]}, \tag{47}$$

and

$$R_c^{[QL]}(\vec{N}) - R_c(\vec{N}) = \sum_{k=1}^{K} D_{c,k} E_k^{(c)[QL]}. \tag{48}$$

If $E_k^{(c)[PE]} = E_{c,k}^{(c)[PE]}$ and $E_k^{(c)[QL]} = E_{c,k}^{(c)[QL]}$, then by (30) and (31),

$$\sum_{b \in \mathbb{B}_c} \left\{ E_b^{(c)[PE]} - E_b^{(c)[QL]} \right\} \geq 0, \tag{49}$$

and

$$\sum_{p \in \overline{\mathbb{B}}_c} \left\{ E_p^{(c)[PE]} - E_p^{(c)[QL]} \right\} \leq 0. \tag{50}$$

Suppose that

$$\sum_{k=1}^{K} E_k^{(c)[PE]} \geq \sum_{k=1}^{K} E_k^{(c)[QL]} \geq 0, \tag{51}$$

then

$$\sum_{k=1}^{K} D_{c,k} E_k^{(c)[QL]} \geq 0,$$

and

$$\sum_{k=1}^{K} \left(E_k^{(c)[PE]} - E_k^{(c)[QL]} \right) = \sum_{k=1}^{K} E_k^{(c)[PE]} - \sum_{k=1}^{K} E_k^{(c)[QL]} \geq 0.$$

Hence, if (51) holds, then by equation (48), $R_c^{[QL]} - R_c(\vec{N}) \geq 0$. Moreover, since $W_c = \max_k\{D_{c,k}\}$, if (51) holds, then by (47), (49) and (50),

$$\begin{aligned}
R_c^{[PE]}(\vec{N}) - R_c^{[QL]}(\vec{N}) &= \sum_{k=1}^{K} D_{c,k} \cdot \left\{ E_k^{(c)[PE]} - E_k^{(c)[QL]} \right\} \\
&= \sum_{b \in \mathbb{B}_c} D_{c,b} \cdot \left\{ E_b^{(c)[PE]} - E_b^{(c)[QL]} \right\} + \sum_{p \in \overline{\mathbb{B}}_c} D_{c,p} \cdot \left\{ E_p^{(c)[PE]} - E_p^{(c)[QL]} \right\} \\
&\geq \sum_{b \in \mathbb{B}_c} W_c \cdot \left\{ E_b^{(c)[PE]} - E_b^{(c)[QL]} \right\} + \sum_{p \in \overline{\mathbb{B}}_c} W_c \cdot \left\{ E_p^{(c)[PE]} - E_p^{(c)[QL]} \right\} \\
&= \sum_{k=1}^{K} W_c \cdot \left\{ E_k^{(c)[PE]} - E_k^{(c)[QL]} \right\} \geq 0.
\end{aligned}$$

Therefore, if (51) holds, then $R_c^{[PE]}(\vec{N}) \geq R_c^{[QL]}(\vec{N}) \geq R_c(\vec{N})$, and the theorem is established in this case.

To complete the proof, we only need show that (51) holds in this case. When $N_c > 1$, if $E_k^{(c)[PE]} = E_{c,k}^{(c)[PE]}$ and $E_k^{(c)[QL]} = E_{c,k}^{(c)[QL]}$, then by Property 2,

$$\sum_{k=1}^{K} E_k^{(c)[PE]} - \sum_{k=1}^{K} E_k^{(c)[QL]} = \sum_{k=1}^{K} E_{c,k}^{(c)[PE]} - \sum_{k=1}^{K} E_{c,k}^{(c)[QL]}$$

$$
\begin{aligned}
&= \sum_{k=1}^{K}\left[\left(\frac{N_c-1}{N_c}\right)\cdot Q_{c,k}(\vec{N})\right] - \sum_{k=1}^{K}\left[\left(\frac{N_c-2}{N_c-1}\right)\cdot Q_{c,k}(\vec{N}) + \left(\frac{1}{N_c-1}\right)\cdot Q_{c,k}(\vec{N}-(N_c-1)\vec{1}_c)\right] \\
&= \frac{1}{N_c-1}\cdot\left[\frac{\sum_{k=1}^{K} Q_{c,k}(\vec{N})}{N_c} - \sum_{k=1}^{K} Q_{c,k}(\vec{N}-(N_c-1)\vec{1}_c)\right] \\
&= \frac{Z_c}{N_c-1}\cdot\left[\frac{1}{Z_c+R_c(\vec{N}-(N_c-1)\vec{1}_c)} - \frac{1}{Z_c+R_c(\vec{N})}\right] \geq 0.
\end{aligned}
$$

Moreover, as illustrated in Figure 8, by Property 2,

$$
\begin{aligned}
&\sum_{k=1}^{K} E_k^{(c)[QL]} = \sum_{k=1}^{K} E_{c,k}^{(c)[QL]} \\
&= \sum_{k=1}^{K}\left[\left(\frac{N_c-2}{N_c-1}\right)\cdot Q_{c,k}(\vec{N}) + \left(\frac{1}{N_c-1}\right)\cdot Q_{c,k}(\vec{N}-(N_c-1)\vec{1}_c) - Q_{c,k}(\vec{N}-\vec{1}_c)\right] \\
&= \sum_{k=1}^{K} Q_{c,k}(\vec{N}) - \sum_{k=1}^{K} Q_{c,k}(\vec{N}-\vec{1}_c) - \frac{1}{N_c-1}\cdot\left[\sum_{k=1}^{K} Q_{c,k}(\vec{N}) - \sum_{k=1}^{K} Q_{c,k}(\vec{N}-(N_c-1)\vec{1}_c)\right] \\
&= Z_c\cdot\left\{\frac{X_c(\vec{N})-X_c(\vec{N}-(N_c-1)\vec{1}_c)}{N_c-1} - [X_c(\vec{N})-X_c(\vec{N}-\vec{1}_c)]\right\} \geq 0.
\end{aligned}
$$

Therefore, (51) holds, and the proof is complete. ■

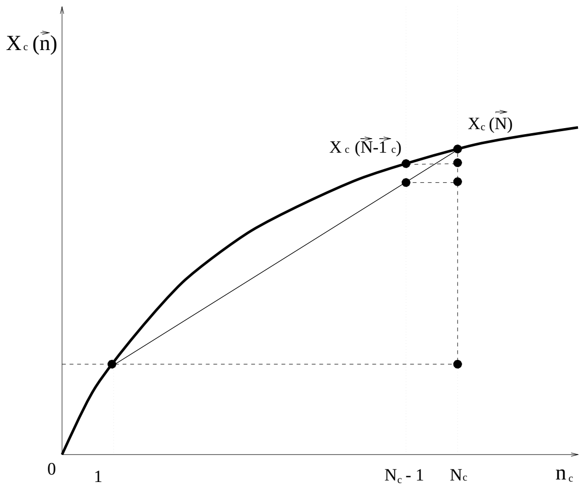

Figure 8: Plot of the function $X_c(\vec{n})$ with respect to n_c