



UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Performance Analysis of a PAC System with Card Capacity of c

Andi Zuhra Wardiyah

M.Sc. Thesis

May 2017

Supervisor:

dr. J.C.W van Ommeren.

Stochastic Operation Research Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
The Netherlands

Abstract

Production Authorization Card (PAC) systems are often used in analyzing manufacturing systems. Most of the studies about PAC system consider single capacity cards. In this research, we analyze a PAC system with mult capacity cards. We provide both an exact model and an approximation model. The exact model is developed using Markov chain while the approximation model is developed by transformation of the original system into a multi-server queuing system. The approximation model is validated using simulation and comparison with existing approximation model in the literature.

Acknowledgments

I would like to express my deep gratitude to my supervisor Jan Kees van Ommeren for giving me an interesting topic, for his patience and guidance. I would like to thank the assessment committee members, Richard Boucherie and Bodo Manthey for their time reading this master thesis. I would also like to thank LPDP for giving me the chance to study here in the Netherlands. Finally, a special thanks to my parents and friends who always supported me during my ups and downs.

Contents

Contents	1
1 Introduction	2
1.1 Problem Description	2
1.2 Solution Approach	2
1.3 Report Structure	2
2 Literature Review	4
2.1 Matrix Geometric Method	4
2.2 Closed Queuing Network	5
2.3 G/G/m queuing system	9
Delay Probability and Mean Waiting Time	9
3 PAC System Model	11
3.1 Exact Model	11
3.2 Approximation Model	17
Approximation of service time G	17
4 Results and Discussion	23
5 Conclusion and Recommendations	25
5.1 Conclusion	25
5.2 Recommendations	25
References	27
Appendix	29

1 Introduction

1.1 Problem Description

Production Authorization Card (PAC) systems are widely used in analyzing performance of multi-stage manufacturing systems. The cards are used to control and coordinate materials at each stage of a manufacturing process. Recently, PAC system has been adapted to evaluate performance analysis of various systems such as logistics, transportation, warehousing, restaurant, health care, etc. For instance, consider container transportation between terminals in a port. This transportation system is known as *inter-terminal transportation* (ITT) system. Container delivery in an ITT systems is carried out by vehicles and these vehicles have multiple capacity. In practice, a free vehicle is dispatched as soon as a request is received, but it can carry container up to its capacity. The loading/unloading time at terminals might vary, depending on container types and number of carried containers. When a vehicle is done delivering, it goes back to its pool.

The ITT system described above can be seen as a Production Authorization Card (PAC) system. The requests to move containers is a job in the PAC system, terminals are the network and a fixed number of vehicles are the cards. What is interesting is, instead of serving a single job, cards might serve multiple jobs at a time. This then leads to a question "*What is the best card dispatch scenario such that performance of the system is optimal?*".

1.2 Solution Approach

In this report, we give two approaches for modeling PAC systems. We first provide an exact analysis by assuming exponential service times and inter-arrival times. The states of the system constitute a Markov chain and the stationary distributions of the Markov chain is obtained using the matrix-geometric method. We then provide an approximation model for the case of general distributions of service and inter-arrival time. The approximation model is developed by means of decomposition and is validated using simulation.

1.3 Report Structure

The reminder of this report is organized as follows. We provide a short literature review in section 2. We then present the mathematical model in section 3 followed by the numerical

and simulation results in section 4. A conclusion and recommendations are presented in section 5.

2 Literature Review

A Production Authorization Card (PAC) system consists of a network of stations with either single or multiple servers, a synchronization station and a finite number of authorization cards. In order for a job to be processed, it has to be attached to a card. This process takes place at the synchronization station. Once a job is paired with a card it can proceed to the network. After service completion, the card is released to its buffer.

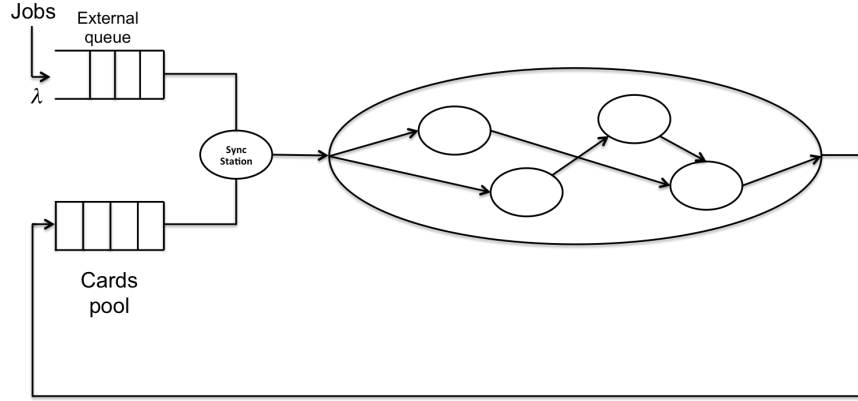


Figure 1: PAC system

In queuing network terminologie, the PAC system described above is known as a semi-open queuing network (SOQN). SOQN is often used in analyzing performance of manufacturing and warehousing system ([3], [5], [10], [14]). It is also used in analyzing dine-in restaurant performance [11]. There are several approaches in analyzing SOQN. In [6], Jian & Heragu used the matrix-geometric method. In [10], [11] and [14], the SOQN problem is solved using a decomposition/aggregation approach.

Studies about multi-capacity cards are very limited. To the best of our knowledge, currently [8] is the only study that considered multi-capacity cards. In this report, both matrix-geometric and decomposition approaches are considered in developing the model. At the rest of this section, we provide basic theory used in developing our model.

2.1 Matrix Geometric Method

Matrix geometric method is a numerical approach to solve a Markov process that has a repetitive structure. This structure is known as matrix-geometric form. According to Neuts [9], having this form allows one to determine the solution of stationary probabilities

recursively. The result of Neuts is described as follows. Consider a Markov process with state space $E = (i, j)$, $i \geq 0$ and j is a vector, having infinitesimal generator \tilde{Q} given by

$$\tilde{Q} = \begin{bmatrix} B_0 & A_0 & & & & \\ B_1 & A_1 & A_0 & & & \\ & A_2 & A_1 & A_0 & & \\ & & A_2 & A_1 & A_0 & \cdots \\ & & & A_2 & A_1 & \cdots \\ & & & & \vdots & \vdots \end{bmatrix}. \quad (1)$$

Let $\pi = [\pi_0, \pi_1, \pi_2, \dots]$ be the vector of stationary distribution satisfying $\pi\tilde{Q} = \mathbf{0}$ and $\pi\mathbf{e} = 1$. Then, π_k is given by

$$\pi_k = \pi_0 R^k, \quad k \geq 0 \quad (2)$$

with R is the minimum non-negative solution of matrix quadratic equation

$$A_0 + A_1 R + A_2 R^2 = 0. \quad (3)$$

R can be obtained by successive substitution of

$$\tilde{R}_{k+1} = -(A_0 + \tilde{R}_k^2 A_2) A_1^{-1} \quad (4)$$

starting with $\tilde{R} = 0$ and \tilde{R} converges to R .

2.2 Closed Queuing Network

Closed queuing networks (CQN) are often used in analyzing flexible manufacturing systems and computer systems. In general, a closed queuing network is as follows:

- The network consists of M stations numbered as $j = 1, 2, \dots, M$.
- A fixed number of N identical jobs circulate around the network.
- Each station is either a single or a multi-server station with expected service time $1/\mu_j$, $j = 1, \dots, M$.
- Upon service completion at station i , job moves with probability p_{ij} to station j for $j = 1, 2, \dots, M$, where $\sum_{j=1}^M p_{ij} = 1$ for all $i = 1, 2, \dots, M$.

In case of exponential service times, we have a closed *Jackson* network. The CQN can be analyzed using exact methods or approximation methods. Exact methods are usually used in analyzing *Jackson* networks and *BCMP* networks [2] while approximation methods are widely used in analyzing queuing network with general service time.

In this report, we use the Approximation Mean Value Analysis (AMVA) method developed in [3]. Similar to the mean value analysis, the AMVA method was developed based on the arrival theorem. Before providing the AMVA algorithm, we introduce the following notations.

- n : number of jobs in the network
- $p_j(k|n)$: marginal probability of k jobs at station j given n jobs in the network
- $TH_j(n)$: throughput rate to station j
- $W_j(n)$: waiting time at station j
- S_j : service time at station j
- S_j^{rem} : remaining service time at station j
- c_j : number of servers at station j
- $L_j^q(n)$: queue length at station j
- $p_j^b(n)$: probability all servers at station j are busy
- V_j : visit ratio at station j
- μ_j : service rate at station j

We now provide AMVA algorithm as follows:

1. (*Initialization*) Set $n = 0$ and $p_j(0|0) = 1$. Set $V_0 = 1$ and determine other visit ratio's V_j , $j = 1, 2, \dots, M$.
2. $n = n + 1$.

3. For $j = 0, \dots, M$ compute

$$\mathbb{E}[L_j^q(n)] = \sum_{k=c_j+1}^{n-1} (k - c_j) p(k|n-1) \quad (5)$$

$$\mathbb{E}[S_j^{rem}] = \frac{c_j - 1}{c_j + 1} \frac{\mathbb{E}[S_j]}{c_j} + \frac{2}{c_j + 1} \frac{1}{c_j} \frac{\mathbb{E}[S_j^2]}{2\mathbb{E}[S_j]} \quad (6)$$

$$\mathbb{E}[W_j(n)] = p_j^b(n-1) \mathbb{E}[S_j^{rem}] + \mathbb{E}[L_j^q(n-1)] \frac{\mathbb{E}[S_j]}{c_j} + \mathbb{E}[S_j] \quad (7)$$

$$p_j^b(n-1) = \sum_{k=c_j}^{n-1} p_j(k|n-1). \quad (8)$$

4. Compute $TH_0(n)$

$$TH_0(n) = \frac{n}{\sum_{i=0}^M V_i \mathbb{E}[W_i(n)]}$$

and $TH_j(n) = TH_0(n)$, $j = 1, \dots, M$.

5. Compute $p_j(k|n)$, $k = 1, \dots, n$, $j = 0, \dots, M$ from

$$p_j(k|n) = \frac{TH_j(n)}{\mu_j} p_j(k-1|n-1)$$

$$p_j(0|n) = 1 - \sum_{k=0}^n p_j(k|n)$$

6. If $n = N$ then stop, else go to step 2.

In addition to the AMVA analysis, we also provide an approximation model of production-to-order problem developed in [3]. This model will later be adapted in developing our approximation model. The idea of the approximation is to transform the original system into closed queuing network with synchronization as a special server. This approximation is summarized as follows.

1. Evaluation the closed queuing network (CQN) without synchronization station.

The purpose of this analysis is to determine the throughput rate, $TH(N)$, of a card at synchronization station which will later be used in constructing a load-dependent

server. $TH(N)$ can be obtained using an AMVA analysis of closed queuing network as shown in figure (2) with N jobs and zero service rate at the synchronization station.

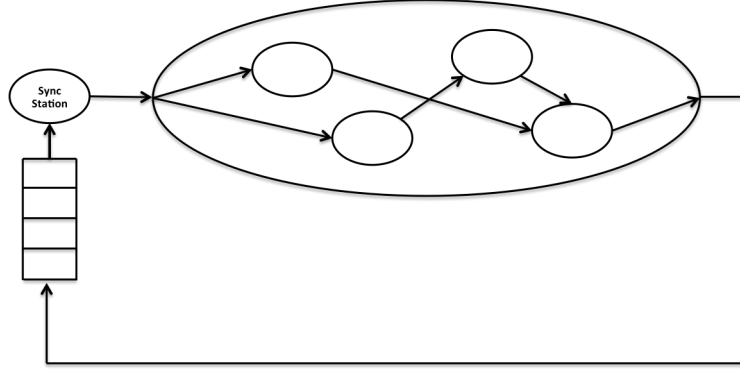


Figure 2: Closed queuing network view of PAC system

2. Construction of a load-dependent server as a substitution of the synchronization station.

To construct a load dependent server, first observe that an arriving card at synchronization station sees there are cards in the queue joins the queue. The waiting time of an arriving card is no other than the interarrival time of a job. Therefore, it is logical to set the service rate of the load dependent server to $\mu_{ld}(n) = \lambda$ for $n = 2, \dots, N$ where λ denotes the arrival rate a job. If at an arrivals no cards are waiting, then the arriving card still has to wait for a job to arrive, this occurs with probability q . The mean waiting time of a card when no other cards waiting is $\frac{q}{\lambda}$ thus the service rate of a card given that no other cards waiting is $\mu_{ld}(1) = \frac{\lambda}{q}$. Probability q is estimated by analyzing synchronization station in isolation (for details, see [3]), which yields

$$q = 1 - \frac{\lambda}{TH(N)}.$$

3. Evaluation CQN together with synchronization station seen as a load-dependent server.

In this step, AMVA analysis is employed with additional equations for evaluating a

load dependent server. These additional equations are the following

$$EW_{ld} = [EL_{ld} + p_{ld}^b(n-1)]\frac{1}{\lambda} + p_{ld}(0|n-1)\frac{q}{\lambda}, \quad (9)$$

$$p_{ld}^b(n) = \frac{TH_{ld}(n)}{\lambda} [p_{ld}^b(n-1) + q p_{ld}(0|n-1)], \quad (10)$$

$$p_{ld}(0|n) = 1 - p_{ld}^b(n), \quad (11)$$

$$EL_{ld}(n) = TH_{ld}(n) EW_{ld}(n). \quad (12)$$

2.3 G/G/m queuing system

Queuing systems with a non Poisson interarrival time and generally distributed service time have been studied for years (see, for instance [7],[12], [15]). In [7], Kimura developed an approximation of the mean waiting time and the queue length distribution of a $G/G/m$ queuing system. His approximation was based on a combination of analytical solutions of simpler systems, for instance a combination of $M/M/m$ and $M/D/m$ systems. In [12], Sadowsky & Szpankowski decomposed the $G/G/m$ queuing system into a busy/idle period and provide a limiting distribution of the maximum queue length and the waiting time distribution of the system. An approximation of the expected waiting time and the delay probability of $G/G/m$ queue are provided in [15]. Whitt's approximation depends on the interarrival time and the service time distribution through their first and second moment. In this report, we mainly use the results from [15].

Delay Probability and Mean Waiting Time

The idea behind the approximation of the delay probability presented in [15] is by approximating the delay probability of $G/M/m$ queue and extending it to $G/G/m$ queue. The final approximation of the delay probability is given by

$$P(W > 0) \approx \min\{\pi, 1\}, \quad (13)$$

with

$$\pi = \begin{cases} \pi_1, & \text{if } m \leq 6 \text{ or } \kappa \leq 0.5 \text{ or } C_a^2 \geq 1 \\ \pi_2, & \text{if } m \geq 7 \text{ and } \kappa \geq 1.0 \text{ and } C_a^2 < 1 \\ \pi_3, & \text{if } m \geq 7 \text{ and } 0.5 < \kappa < 1.0 \text{ and } C_a^2 \leq 1 \end{cases} \quad (14)$$

and

$$\begin{aligned}\pi_1 &= \rho^2 \pi_4 + (1 - \rho^2) \pi_5, \quad \pi_2 = C_a^2 \pi_1 + (1 - C_a^2) \pi_6, \\ \pi_3 &= 2(1 - C_a^2)(\kappa - 0.5) \pi_2 + (1 - [2(1 - C_a^2)(\kappa - 0.5)]) \pi_1, \\ \pi_4 &= \min \left\{ 1, \frac{1 - \Phi(\beta/z)}{1 - \Phi(\beta)} P(W(M/M/m) > 0) \right\}\end{aligned}$$

$$\begin{aligned}\pi_5 &= \min \left\{ 1, \frac{1 - \Phi(a)}{1 - \Phi(\beta)} P(W(M/M/m) > 0) \right\}, \\ \pi_6 &= 1 - \Phi(\kappa), \\ \beta &= (1 - \rho)\sqrt{m}, \quad a = \frac{2\beta}{1 + C_a^2}, \quad \kappa = \frac{m - m\rho - 0.5}{\sqrt{m\rho z}}, \quad z = \frac{C_a^2 + C_s^2}{1 + C_s^2},\end{aligned}$$

where Φ is the cumulative distribution function of standard normal distribution. $P(W(M/M/m) > 0)$ is given by

$$P(W(M/M/m) > 0) = \left[\frac{(m\rho_e)^m}{m!(1 - \rho_e)} \right] \zeta, \quad \zeta = \left[\frac{(m\rho)^m}{m!(1 - \rho)} + \sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} \right]^{-1}. \quad (15)$$

An approximation of the mean waiting time of $G/G/m$ queueing system is also provided in [15]. In this report, we use the refined heavy-traffic approximation of the expected waiting time:

$$EW_q \approx \frac{C_a^2 + C_s^2}{2} EW_q(M/M/m) \quad (16)$$

where the mean waiting time of $M/M/m$ queue is computed using:

$$EW_q(M/M/m) = \frac{\rho^{(-1+\sqrt{2(m+1)})}}{m(1 - \rho)} ES. \quad (17)$$

3 PAC System Model

In this section we provide both an exact and an approximation model of the PAC system with multiple capacity of cards. The exact model is developed using a state space method while the approximation model is developed using a decomposition technique.

3.1 Exact Model

Consider a two-stations PAC system as shown in figure (3). Jobs arrive according to a Poisson process with rate λ . The service time at the first and second station are exponentially distributed with rate μ_1, μ_2 , respectively. There are N available cards with capacity of $c > 1$. Let k be the total number of jobs in the external queue and the first station and

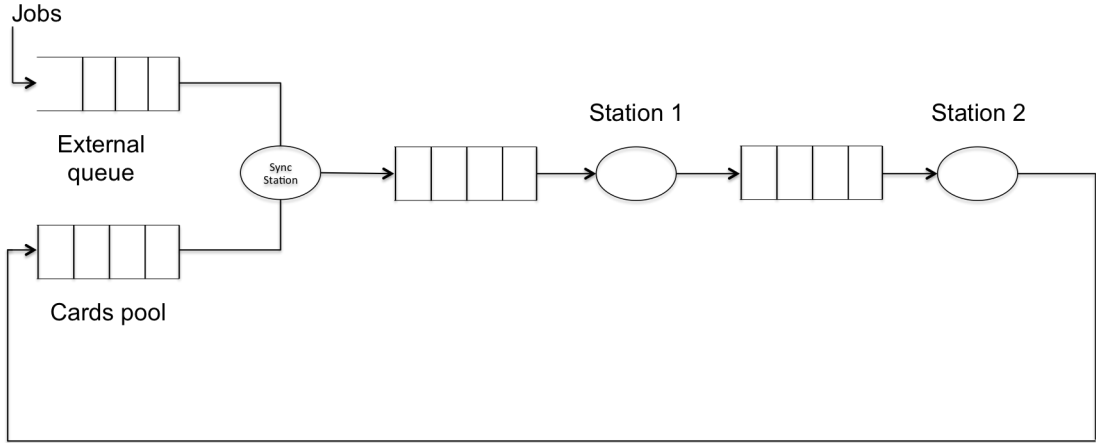


Figure 3: Two-stations PAC system

let l be the total number of jobs at the second station. The state of the system is given by a two dimensional vector (k, l) . Note that for PAC system, the total number of jobs circulating inside the networks are finite and at most N for the case of single capacity or at most Nc jobs for the case of multiple capacity. Given N, k and l , the number of jobs at the first station can be obtained. The transition rate q from state (k, l) to (k', l') is given by

$$q((k, l), (k', l')) = \begin{cases} \lambda, & \text{if } (k', l') = (k + 1, l) \\ \mu_1, & \text{if } (k', l') = (k - c, l + c) \\ \mu_2, & \text{if } (k', l') = (k, l - c) \\ 0, & \text{otherwise} \end{cases},$$

with $k \in \{0, 1, 2, \dots\}$ and $l \in \{0, c, 2c, \dots, Nc\}$. Organizing the state in a lexicographical order yields a generator matrix Q given by

$$Q = \begin{bmatrix} B & A_0 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & B & A_0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & B & \dots & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & \dots & B & A_0 & 0 & \dots \\ A_3 & 0 & 0 & \dots & 0 & A_1 & A_0 & \dots \\ 0 & A_3 & 0 & \dots & 0 & 0 & A_1 & \dots \\ 0 & 0 & A_3 & \dots & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & \dots & A_3 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots & 0 & A_3 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (18)$$

where

$$B = \begin{bmatrix} -\lambda & 0 & 0 & \dots & 0 & 0 \\ \mu_2 & -(\lambda + \mu_2) & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -(\lambda + \mu_2) & 0 \\ 0 & 0 & 0 & \dots & \mu_2 & -(\lambda + \mu_2) \end{bmatrix}, \quad A_0 = \begin{bmatrix} \lambda & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda & 0 \\ 0 & 0 & 0 & \dots & 0 & \lambda \end{bmatrix},$$

$$A_1 = \begin{bmatrix} -(\lambda + \mu_1) & 0 & 0 & \dots & 0 & 0 \\ \mu_2 & -(\lambda + \mu_1 + \mu_2) & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -(\lambda + \mu_1 + \mu_2) & 0 \\ 0 & 0 & 0 & \dots & \mu_2 & -(\lambda + \mu_2) \end{bmatrix},$$

and

$$A_3 = \begin{bmatrix} 0 & \mu_1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \mu_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \mu_1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

The stationary equations of this system are given by

$$\pi_0 B + \pi_c A_3 = 0, \quad (19)$$

$$\pi_{k-1} A_0 + \pi_k B + \pi_{k+c} A_3 = 0, \quad \text{for } k \in \{1, \dots, c-1\} \quad (20)$$

$$\pi_{k-1} A_0 + \pi_k A_1 + \pi_{k+c} A_3 = 0, \quad \text{for } k \geq c \quad (21)$$

$$\sum_{k=0}^{\infty} \pi_k e = 1. \quad (22)$$

It can be seen that Q in the equation (18) has a repeating structure and thus by matrix-geometric theory, we have $(\pi_k)'s$ satisfy equation (2) with R is the minimal non-negative solution of the matrix equation given by

$$A_0 + R A_1 + R^{c+1} A_3 = 0. \quad (23)$$

Rewriting equation (23) into a fixed point equation form, yields

$$R = -(A_0 + R^{c+1} A_3) A_1^{-1}. \quad (24)$$

Matrix R can be obtained by a successive substitution of

$$R_{k+1} = -(A_0 + R_k^{c+1} A_3) A_1^{-1} \quad (25)$$

starting with $R_0 = 0$. The convergence of R_k is stated in the following result.

Proposition 1. *Let R be the minimal non-negative solution of the matrix equation (23). Then, the sequence $\{R_k\}_{k \geq 1}$ obtained from (25) starting with $R_0 = 0$ is a monotone increasing sequence and the sequence converges to R .*

Proof. First note that matrix A_0 and A_3 are non-negative matrices and by the structure of matrix A_1 we have $-A_1^{-1}$ is also a non negative matrix. Now, we prove (by induction)

the sequence $\{R_k\}_{k \geq 1}$ is a monotone increasing sequence as follows. For $n = 0$, we get

$$\begin{aligned} R_1 &= -(A_0 + R_0^{c+1} A_3) A_1^{-1} \\ &= A_0 (-A_1^{-1}) \\ &> 0 \\ &= R_0 \end{aligned}$$

Suppose it is true for $n = k - 1$ that is $R_{k-1} > R_{k-2}$, $R_k > 0$. Then for $n = k$, we have

$$\begin{aligned} R_k &= -(A_0 + R_{k-1}^{c+1} A_3) A_1^{-1} \\ &= A_0 (-A_1^{-1}) + R_{k-1}^{c+1} A_3 (-A_1^{-1}) \\ &> A_0 (-A_1^{-1}) + R_{k-2}^{c+1} A_3 (-A_1^{-1}) \\ &= R_{k-1}. \end{aligned}$$

The strict inequality follows by the fact that $R_{k-1} > 0$ and $R_{k-1} > R_{k-2}$, thus the sequence is monotone increasing. The remain is to show that this sequence converges to R by showing $R_k \leq R$ for all $k \geq 1$. Again, using induction, the proof goes as follows. For $n = 0$, we have

$$\begin{aligned} R_1 &= -(A_0 + R_0^{c+1} A_3) A_1^{-1} \\ &= A_0 (-A_1^{-1}) \\ &\leq A_0 (-A_1^{-1}) + R^{c+1} A_3 (-A_1^{-1}) \\ &= -(A_0 + R^{c+1} A_3) A_1^{-1} \\ &= R. \end{aligned}$$

Suppose it is true for $n = k - 1$, so for $n = k$, we have

$$\begin{aligned} R_k &= -(A_0 + R_{k-1}^{c+1} A_3) A_1^{-1} \\ &\leq -(A_0 + R^{c+1} A_3) A_1^{-1} \quad (\text{by induction hypothesis}) \\ &= R. \end{aligned}$$

Let $\lim_{k \rightarrow \infty} R_k = \tilde{R}$, then we get $\tilde{R} = -(A_0 + \tilde{R}^{c+1} A_3) A_1^{-1}$, which means \tilde{R} is the non-negative solution of (23) and since $\tilde{R} \leq R$, we get $\tilde{R} = R$. \square

Given the matrix R , the stationary probability at the state level $k \geq c$ is determined by

$$\begin{aligned}\pi_k &= \pi_{k-1}R \\ &= \pi_{c-1}R^{k-c+1}.\end{aligned}\tag{26}$$

The stationary probability at the state level $k \leq c-1$ can be obtained by solving the following equations

$$\mathbf{0} = [\pi_0 \ \pi_1 \ \pi_2 \ \cdots \ \pi_{c-1}] \begin{bmatrix} B & A_0 & 0 & \cdots & 0 \\ 0 & B & A_0 & \cdots & 0 \\ 0 & 0 & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ RA_3 & R^2A_3 & R^3A_3 & \cdots & B + R^4A_3 \end{bmatrix}, \tag{27}$$

$$1 = \sum_{k=0}^{\infty} \pi_k e. \tag{28}$$

The expected number of jobs at the external queue (EL_e), the expected number of jobs at station 1 including in service (EL_1), the expected number of jobs at station 2 including in service (EL_2), and the expected throughput time of a job (ETT), are calculated by (29), (30) and (31), respectively. Vector \vec{n} in (30) denotes the vector column $[0, c, 2c, \cdots, Nc]^T$.

$$\begin{aligned}EL_e + EL_1 &= \sum_{k=1}^{\infty} k \pi_k e \\ &= \sum_{k=1}^{c-1} k \pi_k e + \pi_{c-1} \sum_{k=c}^{\infty} k R^{k-c+1} e \\ &= \sum_{k=1}^{c-1} k \pi_k e + \pi_{c-1} [(I - R)^{-2} + (c-2)(I - R)^{-1}] e,\end{aligned}\tag{29}$$

$$\begin{aligned}
EL_2 &= \sum_{k=1}^{\infty} \pi_k \vec{n} \\
&= \sum_{k=1}^{c-1} \pi_k \vec{n} + \pi_{c-1} \sum_{k=c}^{\infty} R^{k-c+1} \vec{n} \\
&= \sum_{k=1}^{c-2} \pi_k \vec{n} + \pi_{c-1} (I - R)^{-1} \vec{n}, \tag{30}
\end{aligned}$$

$$ETT = \frac{EL_e + EL_1 + EL_2}{\lambda}. \tag{31}$$

It can be seen that adding more stations will increase the size of generator matrix Q and the size of rate matrix R might be too large to be found in reasonable time. Furthermore, the exact model can only be used in analyzing a Markovian PAC system. Therefore, we develop an approximation model.

3.2 Approximation Model

As discussed in the previous section, analyzing a PAC system using an exact method can be impractical since adding more stations in the network will result in a larger state space and matrix R in the equations (23) might be too large to be found in a reasonable time. Therefore, we develop an approximation model. The idea of the approximation model is by transforming the original PAC system (Figure 4a) into a multi-servers queuing system as shown in Figure 4b.

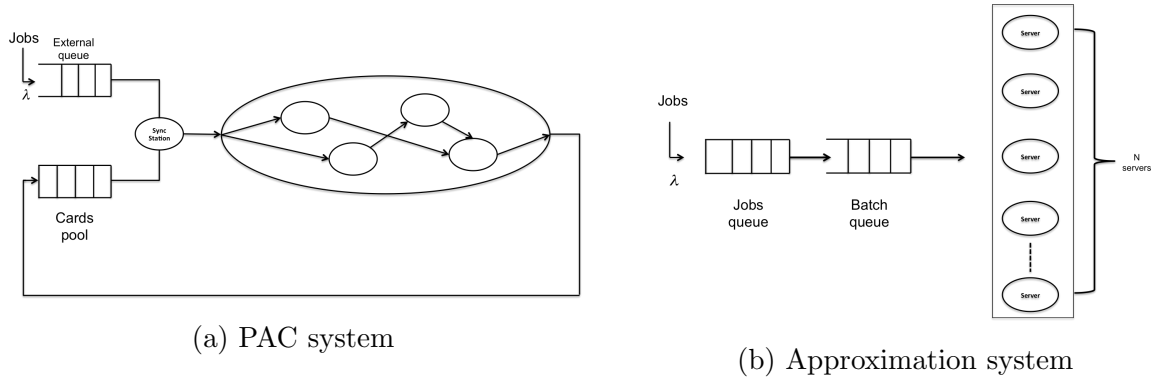


Figure 4: PAC system and Approximation system

Consider a $G/G/N$ queuing system. Customers arrive and are served in batches with arrival rate λ_B . We assume that the size of batches is fixed size c . A job in this $G/G/N$ queuing system is a batch of customers. Let $E[TT]$ denotes the mean throughput time of the approximation system and let $E[W_q]$ be the mean waiting in queue of the $G/G/N$ queuing system. Then, we have

$$E[TT] = E[W_B] + E[W_q] + E[G] \quad (32)$$

where $E[W_B]$ is the mean time of forming a batch. To complete the model, we need the mean and variance of service time, $E[G]$, $\text{Var}(G)$, respectively. This is done by computing the cycle time of a card in the network of the underlying system.

Approximation of service time G

Consider the closed queuing network view (CQN-view) of PAC system as shown in figure (2). The CQN-view of PAC system is evaluated the same way as in the approximation of the production-to-order model presented in section 2.2. Note that the arrivals of batches

at the synchronization station of the underlying system do not follow a Poisson process, but we pretend they do and choose the service rate at the load-dependent server using the same analysis as in the approximation of production-to-stock problem. In evaluating a CQN-view of PAC system with load-dependent server, we consider the fact that an arrival process is not a Poisson process. Thus, instead of using equation (9) and equation (10), we use

$$EW_{ld}(n) = [EL_{ld}(n-1) + p_{ld}^b(n-1)]\frac{1}{\lambda} + p_{ld}(0|n-1)qEA_{res}, \quad (33)$$

$$p_{ld}^b(n) = TH_{ld}(n) [p_{ld}^b(n-1)\frac{1}{\lambda} + p_{ld}(0|n-1)qEA_{res}], \quad (34)$$

where EA_{res} denotes the residual inter-arrival time of a batch. The first term of equation (33) denotes the mean waiting time of an arriving card given that queue is not empty during arrival while the second term denotes the mean waiting time of an arriving card given that queue is empty during arrival.

In the original AMVA algorithm, there is no computation for variance of waiting time at each station. Therefore, in step 3 of the AMVA algorithm we add an additional equation for computing the variance. The variance of waiting time at station j is obtained as follows. Let $W_j(k, n)$ denotes the waiting time of arriving job sees there are $(k-1)$ jobs at station j with n jobs in the network. If during arrival, there are free servers, then waiting time of a job at the station is just its service time. If arriving job sees all servers are busy, then it has to wait for free server and service time of jobs in front of it. Thus we have,

$$W_j(k, n) = \begin{cases} S_j, & \text{if } 1 \leq k \leq c_j \\ S_j^{rem} + \sum_{i=1}^{k-c_j+1} S_j, & \text{if } k > c_j \end{cases} \quad (35)$$

which yields

$$\begin{aligned} \text{Var}(W_j(n)) &= (1 - p_j^b(n-1))^2 \text{Var}(S_j) + (p_j^b(n-1))^2 \text{Var}(S_j^{rem}) + \\ &\quad \left(\sum_{k=c_j}^{n-1} p_j(k|n-1)(k - c_j) \right)^2 \text{Var}(S_j) + \text{Var}(S_j). \end{aligned} \quad (36)$$

Given all necessary variables and using the AMVA algorithm, the CQN-view of PAC system is evaluated. The performance measures of this system are used to approximate the mean and variance of service time of $G/G/N$ queuing system. We assume that the stations

in the network are independent and provide two approximations. The first approximation is by assuming that the number of cards in the system is uniformly distributed and this approximation is denoted by approximation a_1 . This gives:

$$\mathbb{E}[G] = \frac{1}{N} \sum_{n=1}^N \sum_{j=1}^M V_j \mathbb{E}[W_j(n)], \quad (37)$$

$$\text{Var}(G) = \frac{1}{N^2} \sum_{n=1}^N \sum_{j=1}^M V_j^2 \text{Var}(W_j(n)). \quad (38)$$

The second approximation, denoted by a_2 , is by considering a non uniform distribution of number of cards in the network. Let Q_n be the probability having n cards in the network, then we have

$$\mathbb{E}[G] = \sum_{n=1}^N Q_n \sum_{j=1}^M V_j \mathbb{E}[W_j(n)], \quad (39)$$

$$\text{Var}(G) = \sum_{n=1}^N Q_n^2 \sum_{j=1}^M V_j^2 \text{Var}(W_j(n)). \quad (40)$$

Q_n is computed by analyzing the synchronization station in isolation. We assume that arrivals for both batches and cards at the synchronization station follow a state-dependent Poisson process. Let i be the number of batch at external queue and j be the number of cards, then state $(0, j)$ and $(i, 0)$ denote the possible state at the synchronization with $i = 0, 1, 2, \dots$ and $j = 0, 1, \dots, N$. The stationary distribution of $P_{i,j}$ of the Markov process (shown in figure (5)) at the synchronization station satisfies

$$P_{0,j} = \prod_{l=0}^{j-1} \frac{TH(N-l)}{\lambda_B} P_{0,0}, \quad j = 1, \dots, N \quad (41)$$

$$P_{i,0} = \left(\frac{\lambda_B}{TH(N)} \right)^i P_{0,0}, \quad i \geq 1 \quad (42)$$

$$1 = \sum_{i=0}^{\infty} P_{i,0} + \sum_{j=1}^N P_{0,j}, \quad (43)$$

Thus, we have

$$P_n = \begin{cases} \sum_{i=0}^{\infty} P_{i,n} & n = 0, \\ P_{0,n} & n = 1, 2, \dots, N \end{cases}$$

Note that having n cards at the synchronization station means there are $N - n$ cards in the networks. Therefore we have $Pn = Q_{N-n}$.

Up to now, we have considered the case that a card is dispatched only if there are c jobs waiting at the external queue. In other words, the card is always fully loaded. At the rest of this section, we develop an approximation model for the following case. During a free period, that is the period with at least one free card, a card is dispatched as soon as there are $d \leq c$ jobs at the external queue. Meanwhile during a busy period, the period that all cards are busy serving, once a card is available and there are at least $d > 1$ jobs at the external queue, then it immediately serves the jobs up to its capacity. By conditioning on server availability during arrival, $E[TT]$, the mean throughput time of a job is approximated by

$$E[TT] = E[TT|\text{busy period}] P(\text{busy period}) + E[TT|\text{free period}] (1 - P(\text{busy period})). \quad (44)$$

The mean throughput time during a free period is the mean service time $E[G]$ plus the mean waiting to form a batch of size d , that is,

$$E[TT|\text{free period}] = EW_B^{fp} + EG. \quad (45)$$

During a busy period, jobs have to wait for any of the N servers of $G/G/N$ queue to be free. The arrival rate of batches during busy period has random size k , $k = \{d, d+1, \dots, c\}$. Let EW_q^{bp} denotes the mean waiting time during a busy period and EW_q denotes the mean waiting time in queue of $G/G/N$ queuing system, we have

$$\begin{aligned} EW_q^{bp} &= E[W_q | N \text{ busy server}], \\ &= \frac{E[W_q(G/G/N)]}{P(\text{busy period of } G/G/N)}, \end{aligned} \quad (46)$$

where the nominator and denominator of equation (46) are computed using equation (16) and equation (13), respectively. The arrival rate of $G/G/N$ queue during busy period is

approximated by

$$\lambda_B^{bp} = \frac{1}{c-d+1} \sum_{j=d}^c \lambda_{B_j}. \quad (47)$$

where λ_{B_j} denotes the arrival rate of a batch size j and is computed by

$$\lambda_{B_j} = \frac{\lambda}{j}. \quad (48)$$

The mean waiting time for forming a batch during busy period EW_B^{bp} is given by

$$EW_B^{bp} = \frac{1}{c-d+1} \sum_{j=d}^c EW_{B_j}, \quad (49)$$

where EW_{B_j} denotes the mean waiting time for forming a batch size j and it is given by

$$EW_{B_j} = \frac{j-1}{2\lambda}. \quad (50)$$

Finally, we have the mean throughput time of a job that arrives during a busy period,

$$E[TT|\text{busy period}] = EW_B^{bp} + EW_q^{bp} + EG. \quad (51)$$

To complete the model, we need to compute the probability of the system is in a busy

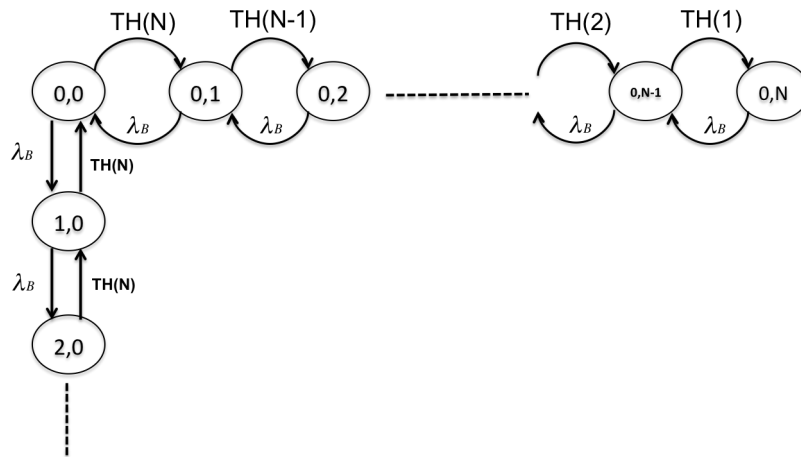


Figure 5: The birth and death process at synchronization station

period. A busy period starts at the moment that there is no card at the synchronization station. Under the assumptions that the arrival process for both cards and batches at the synchronization station follow a state-dependent Poisson process, then the probability of the system is in busy period is the probability of the system is in state $(i, 0)$ of the process shown in figure (5). Note that in this case, the arrival rate of a batch during a busy period differs from the arrival rate of a batch that arrives during a free period. Therefore, the birth and death process as shown in figure (5) is slightly different. For this case, the stationary distribution $P_{i,j}$ satisfies equation (41) with $\lambda_B = \lambda_{B_d}$, equation (42) with $\lambda_B = \lambda_B^{bp}$ and equation (43).

4 Results and Discussion

In this report, the performance measure of interest is the mean throughput time of a job, $E[TT]$. We validate the approximation model using simulation. The simulation is developed using the Python programming language. We run the simulation for a five years simulation period with six months of warm-up period. The purpose of the simulation is to obtain the steady state behavior of the system. Interarrival and service times in the system are in minutes. The proposed approximation model is validated for variety of cases as shown in Table 1. The number of replications for each case is 25.

Table 1: Numerical Experiments

Card Capacity	Card Dispatch Scenario	Number of Station in the Network	Type of Stations	Variation of Arrival Rate
Single capacity	-	2	Exponential	-
Multiple capacity	Fully loaded	2	Exponential	✓
			Erlang-2	
		4	Exponential	✓
	Dispatch as soon as there is a job	2	Exponential	
			Erlang-2	
		4	Exponential	✓
	Dispatch as soon as there are 2 jobs	2	Exponential	
			Erlang-2	
		4	Exponential	

The results are summarized in Table 10 - 14 where $E[G]$ denotes the mean service time of a batch, $E[W_e]$ denotes the mean waiting time of a job at the external queue, and N denotes the number of cards. The percentage of error is calculated as follows:

$$\% \text{ error} = \frac{|\text{Approximation} - \text{Simulation}|}{\text{Simulation}} \times 100\%.$$

Column a_1 denotes the approximation of the mean and variance of service time G using equation (37) and equation (38), respectively. Column a_2 denotes the approximation of the mean and variance of service time $E[G]$ using equation (39) and equation (40), respectively. Column Literature denotes the results of the existing approximation.

Table 3 - 5 show the results for a fully loaded dispatch scenario with card capacity $c = 2$ for different values of arrival rate of a job ($\lambda = 0.3, 0.4, 0.5$) and different types of network. It can be seen that the percentage of error of the mean throughput time of

the approximation a_1 is small compared to the approximation a_2 . It is also shown that the mean throughput time of a job is increasing as the number of cards increases. This contradicts with common case, that the throughput time of a job decreases as the number of cards increases. The reason of the contradiction is because the increment of the mean service time ($E[G]$), is higher than the decrement of the mean waiting time at the external queue which yields an increasing value of mean throughput time.

Besides comparison with simulation, we also compare the proposed approximation with the existing approximation provided in [3] for the case of single capacity card. It can be seen from Table 10 that our approximation model underestimate the mean throughput time of the system with the highest percentage of 16% for approximation a_1 and up to 52% for approximation a_2 . Meanwhile, the existing approximation overestimate the mean throughput time of the system with maximum error of 5%. We also conduct a numerical experiments for larger network as shown in Table 8 and 9.

The results for the case of a non fully loaded scenario are summarized in Table 11 - 14. In this case, we only consider approximation a_1 since even for a fully loaded case, the approximation a_2 already gives a high percentage of error. There is a large error in the mean waiting time at the external queue of the proposed approximation model a_1 . This is because when arrival rate of a job is low, the waiting probability is much lower than the mean waiting time and this yields a large value of the first term of equation (46) and eventually overestimate the mean waiting time during busy period. Specifically, the main error is due to the assumption of the independence of stations in the networks which leads to an underestimation of the variance of service time, $\text{Var}(G)$.

In the following table, we present the mean throughput time of a job with two different dispatch scenario. D_1 and D_2 denote the case of dispatch scenario with $d = 1$, $d = 2$, respectively and card capacity of $c = 5$. The service time is computed using approximation a_1 .

Table 2: Throughput time of a job with different dispatch scenario

N	$\lambda = \mathbf{0.2}$		$\lambda = \mathbf{0.3}$	
	D_1	D_2	D_1	D_2
4	5.893	6.995	7.075	7.023
5	5.912	7.042	7.325	7.063
6	5.965	7.084	7.551	7.104
7	6.027	7.120	7.754	7.142
8	6.087	7.150	7.936	7.180
9	6.140	7.176	8.097	7.223
10	6.186	7.201	8.240	7.284

5 Conclusion and Recommendations

5.1 Conclusion

In drawing the conclusion for this research, we consider the research question:

What is the best card dispatch scenario such that performance of the system is optimal?

We develop an approximation model based on a decomposition technique. From the results, it can be concluded that for a low arrival rate, dispatching a card only if there are two jobs increases the mean throughput time. Meanwhile, for a relatively high arrival rate, dispatching the card only if there are two jobs decreases the mean throughput time compared to dispatching the card as soon as there is job at the external queue. However, it is worth to note that having a good approximation model plays important role. In comparison with simulation, the proposed approximation model generally is a poor approximation.

5.2 Recommendations

Future research can be done to obtain a better approximation of PAC systems. The idea of improvements are the following.

1. Relaxation of the assumption of an exponential arrival rate of cards in analyzing synchronization station.
2. Relaxation of the assumption of independence of stations in the network.

3. Analyzing the system by decomposition of its busy and free period. During a free period, a PAC system can be modeled as a state-dependent queuing system while during a busy period, the PAC system can be modeled as a multi-server queuing system during its busy period.

Due to time constraint for conducting this research, we did not extend our research for the case of capacity dependent service time and for the case of multi-class jobs. Therefore, both those cases are also an interesting area of future research.

References

- [1] Buzacott, J.A., Shanthikumar, J.G., 1993, *Stochastic Models of Manufacturing Systems*, Prentice Hall. ISBN 0-13-847567-9.
- [2] Baldwin, R.O, Davis, N.J., Midkiff, S.F., Kozva, J.E., 2003, *Queuing network analysis: concepts, terminology and methods*. The Journal of System and Software 66:99-117.
- [3] Buithenhek, R., 1998, *Performance Evaluation of Dual Resource Manufacturing Systems*. PhD thesis, University of Twente.
- [4] Briksorn, D., Drexl, A., Hartmann, S., 2006, *Inventory-based dispatching of automated guided vehicles on container terminal*, OR Spectrum 28:611-630.
- [5] Ekren, B.A, Heragu, S.S, Krishnamurthy,A., Malmborg, C.J, 2013, *Matrix-geometric solution for SOQN model of autonomous vehicle storage and retrieval system*, Computers & Industrial Engineering 68:78-86.
- [6] Jian, J., Heragu,S.S., 2009, *Solving semi-open queueing networks*, Operation Research 57(2):391-401.
- [7] Kimura, T., 1994, *Approximation for Multi-server Queues: system interpolations*, Queuing System 17:347-382.
- [8] Mishra, N. and Roy, D., van Ommeren, J.C.W., 2013, *A stochastic model for inter-terminal container transportation*, Memorandum 2032, Department of Applied Mathematics, University of Twente, Enschede. ISSN 1874-4850.
- [9] Neuts, M., 1980, *Matrix-geometric solution in stochastic modeling*, The Johns Hopkins Univeristy Press.

- [10] Roy, D., Krishnamurthy, A., Heragu, S.S, Malmborg, C.J, 2015, *Queuing models to analyze dwell-point and cross aisle location in autonomous vehicle-based warehouse systems*, European Journal of Operational Research 242:72-87.
- [11] Roy, D., Bandyopadhyay, A., Banerjee, P., 2016, *A nested semi-open queuing network model for analyzing dine-in restaurant performance*, Computers & Operations Research 65:29-41.
- [12] Sadowsky, J.S., Szpankowski, W.,1989, *On the Analysis of the Tail Queue Length and Waiting Time Distribution of a GI/GI/c queue*, Computer Science Technical Reports. Retrieved from <http://docs.lib.purdue.edu/cstech/797>.
- [13] Tierney, K., Voss, S., Stahlbock, R., 2014, *A mathematical model of inter-terminal transportation*, European Journal of Operational Research 235:448-460.
- [14] Xiaobo, Z., Zhou, Z., 1999, *A semi-open decomposition approach for an open queuing network in a general configuration with kanban blocking mechanism*, Int. J. Production Economics 60-61:375-380.
- [15] Whitt, W., 1993. *Approximation for GI/GI/c queue*, Production and Operations Management 2(2):114-161.

Appendix

In this appendix, we provide tables that show the results of the numerical experiments for various cases as mention in section 4. Column a_1 denotes the approximation of the mean and variance of service time G using equation (37) and equation (38), respectively. Column a_2 denotes the approximation of the mean and variance of service time $E[G]$ using equation (39) and equation (40), respectively. Table 3 - 5 show the results for a fully loaded dispatch scenario for different types of network.

Table 3: Card capacity $c = 2$ with two-stations exponential network, $\lambda = 0.3$ jobs per min, $\mu_1 = \mu_2 = 0.5$ batch per min

N	E[TT] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	6.933	3.972	$6.656 \pm 2.378\text{E-}05$	4%	40%
5	7.016	4.010	$6.651 \pm 4.467\text{E-}05$	5%	40%
6	7.072	4.021	$6.648 \pm 3.934\text{E-}05$	6%	40%
7	7.112	4.024	$6.653 \pm 4.615\text{E-}05$	7%	40%
8	7.143	4.024	$6.652 \pm 3.615\text{E-}05$	7%	40%
9	7.167	4.023	$6.654 \pm 2.442\text{E-}05$	8%	40%
10	7.187	4.022	$6.650 \pm 2.763\text{E-}05$	8%	40%

N	E[G] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	5.251	2.304	$4.973 \pm 2.605\text{E-}05$	6%	54%
5	5.346	2.343	$4.980 \pm 4.694\text{E-}05$	7%	53%
6	5.404	2.354	$4.982 \pm 3.452\text{E-}05$	8%	53%
7	5.445	2.357	$4.986 \pm 4.209\text{E-}05$	9%	53%
8	5.476	2.357	$4.986 \pm 4.055\text{E-}05$	10%	53%
9	5.500	2.357	$4.988 \pm 2.304\text{E-}05$	10%	53%
10	5.520	2.356	$4.984 \pm 3.093\text{E-}05$	11%	53%

N	E[W _e] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	1.682	1.668	$1.684 \pm 3.653\text{E-}06$	0%	1%
5	1.671	1.667	$1.671 \pm 2.156\text{E-}06$	0%	0%
6	1.668	1.667	$1.667 \pm 1.670\text{E-}06$	0%	0%
7	1.667	1.667	$1.667 \pm 2.919\text{E-}06$	0%	0%
8	1.667	1.667	$1.666 \pm 1.999\text{E-}06$	0%	0%
9	1.667	1.667	$1.666 \pm 3.558\text{E-}06$	0%	0%
10	1.667	1.667	$1.667 \pm 3.050\text{E-}06$	0%	0%

Table 4: Card capacity $c = 2$ with two-stations exponential network, $\lambda = 0.4$ jobs per min, $\mu_1 = \mu_2 = 0.5$ jobs per min

N	E[TT] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	7.021	4.275	$6.972 \pm 3.774\text{E-}05$	1%	39%
5	7.186	4.421	$6.962 \pm 6.825\text{E-}05$	3%	36%
6	7.301	4.486	$6.958 \pm 6.282\text{E-}05$	5%	36%
7	7.384	4.513	$6.959 \pm 3.498\text{E-}05$	6%	35%
8	7.446	4.523	$6.960 \pm 7.462\text{E-}05$	7%	35%
9	7.494	4.526	$6.958 \pm 4.349\text{E-}05$	8%	35%
10	7.533	4.527	$6.962 \pm 4.716\text{E-}05$	8%	35%

N	E[G] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	5.729	3.019	$5.619 \pm 2.921\text{E-}05$	2%	46%
5	5.923	3.170	$5.680 \pm 6.386\text{E-}05$	4%	44%
6	6.047	3.235	$5.698 \pm 5.849\text{E-}05$	6%	43%
7	6.132	3.263	$5.706 \pm 3.838\text{E-}05$	7%	43%
8	6.195	3.273	$5.709 \pm 8.118\text{E-}05$	9%	43%
9	6.244	3.276	$5.708 \pm 4.514\text{E-}05$	9%	43%
10	6.283	3.277	$5.712 \pm 4.864\text{E-}05$	10%	43%

N	E[W_e] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	1.314	1.261	$1.354 \pm 4.978\text{E-}06$	3%	7%
5	1.271	1.253	$1.282 \pm 9.251\text{E-}07$	1%	2%
6	1.257	1.251	$1.260 \pm 1.654\text{E-}06$	0%	1%
7	1.253	1.250	$1.253 \pm 1.282\text{E-}06$	0%	0%
8	1.251	1.250	$1.251 \pm 8.226\text{E-}07$	0%	0%
9	1.250	1.250	$1.250 \pm 1.248\text{E-}06$	0%	0%
10	1.250	1.250	$1.250 \pm 1.649\text{E-}06$	0%	0%

Table 5: Card capacity $c = 2$ with two-stations exponential network, $\lambda = 0.5$ jobs per min, $\mu_1 = \mu_2 = 0.5$ jobs per min

N	E[TT] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	7.317	4.536	$7.868 \pm 1.348\text{E-}04$	7%	42%
5	7.568	4.921	$7.788 \pm 6.115\text{E-}05$	3%	37%
6	7.777	5.139	$7.761 \pm 1.284\text{E-}04$	0%	34%
7	7.939	5.260	$7.756 \pm 8.853\text{E-}05$	2%	32%
8	8.064	5.325	$7.757 \pm 1.012\text{E-}04$	4%	31%
9	8.162	5.358	$7.762 \pm 7.073\text{E-}05$	5%	31%
10	8.240	5.375	$7.757 \pm 1.062\text{E-}04$	6%	31%

N	E[G] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	6.170	3.509	$6.442 \pm 6.894\text{E-}05$	4%	46%
5	6.515	3.911	$6.611 \pm 4.001\text{E-}05$	1%	41%
6	6.757	4.136	$6.687 \pm 9.264\text{E-}05$	1%	38%
7	6.931	4.258	$6.724 \pm 7.931\text{E-}05$	3%	37%
8	7.061	4.324	$6.743 \pm 9.379\text{E-}05$	5%	36%
9	7.160	4.358	$6.756 \pm 6.891\text{E-}05$	6%	35%
10	7.240	4.375	$6.755 \pm 1.064\text{E-}05$	7%	35%

N	E[W_e] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	1.147	1.027	$1.426 \pm 1.348\text{E-}04$	20%	28%
5	1.053	1.010	$1.177 \pm 6.115\text{E-}05$	11%	14%
6	1.020	1.004	$1.074 \pm 1.284\text{E-}04$	5%	7%
7	1.008	1.001	$1.032 \pm 8.853\text{E-}05$	2%	3%
8	1.003	1.001	$1.013 \pm 1.013\text{E-}04$	1%	1%
9	1.001	1.000	$1.006 \pm 7.073\text{E-}05$	0%	1%
10	1.001	1.000	$1.002 \pm 1.062\text{E-}04$	0%	0%

Table 6: Card capacity $c = 2$ with two-stations Erlang-2 network, $\lambda = 0.7$ jobs per min,
 $\mu_1 = 2$ jobs per min, $\mu_2 = 1$ jobs per min

N	E[TT] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	5.650	3.265	$6.034 \pm 7.267\text{E-}05$	6%	46%
5	6.052	3.703	$6.031 \pm 5.086\text{E-}05$	0%	39%
6	6.391	4.022	$6.0301 \pm 7.376\text{E-}05$	6%	33%
7	6.671	4.249	$6.032 \pm 4.842\text{E-}05$	11%	30%
8	6.898	4.411	$6.033 \pm 6.184\text{E-}05$	14%	27%
9	7.081	4.524	$6.033 \pm 1.119\text{E-}04$	17%	25%
10	7.230	4.604	$6.029 \pm 9.746\text{E-}05$	20%	24%

N	E[G] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	4.843	2.542	$4.947 \pm 2.383\text{E-}05$	2%	49%
5	5.297	2.984	$5.132 \pm 2.966\text{E-}05$	3%	42%
6	5.659	3.305	$5.228 \pm 5.941\text{E-}05$	7%	37%
7	5.948	3.534	$5.273 \pm 4.518\text{E-}05$	12%	33%
8	6.179	3.696	$5.297 \pm 5.288\text{E-}05$	16%	30%
9	6.365	3.810	$5.308 \pm 1.156\text{E-}04$	20%	28%
10	6.515	3.890	$5.309 \pm 9.487\text{E-}05$	23%	27%

N	E[W_e] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	0.808	0.723	$1.087 \pm 2.285\text{E-}05$	26%	33%
5	0.754	0.719	$0.898 \pm 6.960\text{E-}06$	16%	20%
6	0.732	0.716	$0.802 \pm 2.353\text{E-}06$	9%	11%
7	0.723	0.715	$0.758 \pm 2.253\text{E-}06$	5%	6%
8	0.718	0.715	$0.736 \pm 2.016\text{E-}06$	2%	3%
9	0.716	0.714	$0.724 \pm 4.255\text{E-}07$	1%	1%
10	0.715	0.714	$0.719 \pm 9.746\text{E-}05$	1%	1%

Table 7: Card capacity $c = 2$ with two-stations Erlang-2 network, $\lambda = 0.8$ jobs per min,
 $\mu_1 = 2$ jobs per min $\mu_2 = 1$ jobs per min

N	E[TT] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	5.885	2.913	$7.686 \pm 4.562\text{E-}04$	23%	62%
5	6.424	3.557	$7.666 \pm 5.891\text{E-}04$	16%	54%
6	6.933	4.111	$7.664 \pm 3.037\text{E-}04$	10%	46%
7	7.396	4.575	$7.667 \pm 3.657\text{E-}04$	4%	40%
8	7.808	4.963	$7.668 \pm 3.009\text{E-}04$	2%	35%
9	8.170	5.283	$7.667 \pm 2.604\text{E-}04$	7%	31%
10	8.488	5.544	$7.677 \pm 1.408\text{E-}04$	11%	28%

N	E[G] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	5.088	2.279	$5.747 \pm 5.812\text{E-}05$	11%	60%
5	5.714	2.927	$6.208 \pm 8.381\text{E-}05$	8%	53%
6	6.264	3.482	$6.504 \pm 1.016\text{E-}04$	4%	46%
7	6.748	3.948	$6.699 \pm 1.610\text{E-}04$	1%	41%
8	7.170	4.337	$6.823 \pm 1.379\text{E-}04$	5%	36%
9	7.538	4.657	$6.901 \pm 1.813\text{E-}04$	9%	33%
10	7.858	4.919	$6.959 \pm 9.916\text{E-}05$	13%	29%

N	E[W_e] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	0.796	0.634	$1.939 \pm 2.068\text{E-}04$	59%	67%
5	0.709	0.631	$1.458 \pm 2.586\text{E-}04$	51%	57%
6	0.669	0.629	$1.160 \pm 9.622\text{E-}05$	42%	46%
7	0.649	0.627	$0.967 \pm 7.797\text{E-}05$	33%	35%
8	0.638	0.626	$0.845 \pm 5.249\text{E-}05$	24%	26%
9	0.632	0.626	$0.765 \pm 2.197\text{E-}05$	17%	18%
10	0.629	0.625	$0.718 \pm 1.732\text{E-}05$	12%	13%

Table 8: Card capacity $c = 2$ with four-stations exponential network, $\lambda = 0.2$ jobs per min, $\mu_1 = \mu_2 = 0.5, \mu_3 = 1.0, \mu_4 = 1.5$ batch per min

N	E[TT] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	9.040	5.265	$8.743 \pm 4.066\text{E-}05$	3%	40%
5	9.086	5.279	$8.744 \pm 9.065\text{E-}06$	4%	40%
6	9.121	5.281	$8.743 \pm 4.508\text{E-}05$	4%	40%
7	9.147	5.280	$8.740 \pm 2.647\text{E-}05$	5%	40%
8	9.167	5.280	$8.744 \pm 3.601\text{E-}05$	5%	40%
9	9.183	5.279	$8.742 \pm 3.718\text{E-}05$	5%	40%
10	9.197	5.279	$8.739 \pm 4.996\text{E-}05$	5%	40%

N	E[G] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	6.529	2.788	$6.239 \pm 2.037\text{E-}05$	5 %	55 %
5	6.583	2.784	$6.244 \pm 4.515\text{E-}05$	5 %	55 %
6	6.620	2.782	$6.243 \pm 3.397\text{E-}05$	6 %	55 %
7	6.647	2.781	$6.242 \pm 2.634\text{E-}05$	6 %	55 %
8	6.667	2.780	$6.244 \pm 3.036\text{E-}05$	7 %	55 %
9	6.683	2.779	$6.242 \pm 2.586\text{E-}05$	7 %	55 %
10	6.697	2.779	$6.239 \pm 5.175\text{E-}05$	7 %	55 %

N	E[W_e] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	2.512	2.501	$2.503 \pm 1.075\text{E-}05$	0%	0%
5	2.503	2.500	$2.500 \pm 9.065\text{E-}06$	0%	0%
6	2.501	2.500	$2.500 \pm 9.440\text{E-}06$	0%	0%
7	2.500	2.500	$2.498 \pm 5.276\text{E-}06$	0%	0%
8	2.500	2.500	$2.500 \pm 7.190\text{E-}06$	0%	0%
9	2.500	2.500	$2.500 \pm 1.577\text{E-}05$	0%	0%
10	2.500	2.500	$2.500 \pm 1.322\text{E-}05$	0%	0%

Table 9: Card capacity $c = 2$ with four-stations exponential network, $\lambda = 0.3$, $\mu_1 = \mu_2 = 0.5$, $\mu_3 = 1.0$, $\mu_4 = 1.5$

N	E[TT] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	8.687	5.536	$8.510 \pm 4.692\text{E-}05$	2%	35%
5	8.785	5.627	$8.501 \pm 6.173\text{E-}05$	3%	34%
6	8.859	5.657	$8.505 \pm 6.762\text{E-}05$	4%	33%
7	8.914	5.665	$8.501 \pm 3.348\text{E-}05$	5%	33%
8	8.958	5.666	$8.500 \pm 4.658\text{E-}05$	5%	33%
9	8.992	5.666	$8.496 \pm 4.037\text{E-}05$	6%	33%
10	9.021	5.665	$8.501 \pm 2.644\text{E-}05$	6%	33%

N	E[G] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	6.982	4.036	$6.807 \pm 3.482\text{E-}05$	3 %	41 %
5	7.107	4.018	$6.826 \pm 6.168\text{E-}05$	4 %	41 %
6	7.189	4.009	$6.837 \pm 7.245\text{E-}05$	5 %	41 %
7	7.246	4.005	$6.835 \pm 2.992\text{E-}05$	6 %	41 %
8	7.291	4.002	$6.834 \pm 4.012\text{E-}05$	7 %	41 %
9	7.325	4.000	$6.829 \pm 4.398\text{E-}05$	7 %	41 %
10	7.354	3.998	$6.834 \pm 2.249\text{E-}05$	8 %	41 %

N	E[W_e] (min)			% Error	
	a_1	a_2	Simulation	a_1	a_2
4	1.705	1.672	$1.704 \pm 3.604\text{E-}06$	0%	2%
5	1.678	1.668	$1.675 \pm 2.270\text{E-}06$	0%	0%
6	1.670	1.667	$1.668 \pm 2.071\text{E-}06$	0%	0%
7	1.668	1.667	$1.667 \pm 2.605\text{E-}06$	0%	0%
8	1.667	1.667	$1.666 \pm 3.349\text{E-}06$	0%	0%
9	1.667	1.667	$1.666 \pm 2.479\text{E-}06$	0%	0%
10	1.667	1.667	$1.667 \pm 1.924\text{E-}06$	0%	0%

This table shows the comparison between our approximation model with an existing approximation (Column Literature).

Table 10: Single capacity with two-stations exponential network, $\lambda = 0.2$ jobs per min,
 $\mu_1 = \mu_2 = 0.5$ jobs per min.

N	E[TT] (min)				% Error		
	Literature	a_1	a_2	Simulation	Literature	a_1	a_2
4	7.115	5.651	3.284	$6.732 \pm 2.18\text{E-}04$	6 %	16 %	51 %
5	6.853	5.789	3.255	$6.691 \pm 2.62\text{E-}04$	2 %	13 %	51 %
6	6.745	5.906	3.240	$6.673 \pm 2.51\text{E-}04$	1 %	12 %	51 %
7	6.699	6.000	3.233	$6.663 \pm 3.02\text{E-}04$	0 %	10 %	52 %
8	6.681	6.074	3.228	$6.663 \pm 2.19\text{E-}04$	0 %	9 %	52 %
9	6.673	6.134	3.225	$6.663 \pm 1.38\text{E-}04$	0 %	8 %	52 %
10	6.669	6.184	3.222	$6.663 \pm 1.51\text{E-}04$	1 %	7 %	51 %

N	E[G] (min)				% Error		
	Literature	a_1	a_2	Simulation	Literature	a_1	a_2
4	6.634	5.582	3.270	$6.337 \pm 2.886\text{E-}05$	5 %	12 %	48 %
5	6.652	5.767	3.250	$6.507 \pm 3.913\text{E-}05$	5 %	9 %	49 %
6	6.659	5.898	3.239	$6.595 \pm 6.843\text{E-}05$	5 %	7 %	49 %
7	6.663	5.997	3.233	$6.639 \pm 5.639\text{E-}05$	5 %	5 %	49 %
8	6.665	6.073	3.228	$6.657 \pm 1.943\text{E-}04$	5 %	4 %	49 %
9	6.666	6.134	3.225	$6.654 \pm 9.100\text{E-}05$	5 %	3 %	49 %
10	6.666	6.183	3.222	$6.664 \pm 1.053\text{E-}04$	5 %	2 %	49 %

N	E[W_e] (min)				% Error		
	Literature	a_1	a_2	Simulation	Literature	a_1	a_2
4	0.480	0.068	0.018	$0.394 \pm 1.619\text{E-}05$	22%	83%	96%
5	0.200	0.022	0.005	$0.174 \pm 6.519\text{E-}06$	15%	87%	98%
6	0.086	0.008	0.001	$0.077 \pm 3.275\text{E-}06$	11%	90%	99%
7	0.037	0.003	0.000	$0.035 \pm 1.934\text{E-}06$	6%	92%	99%
8	0.016	0.001	0.000	$0.015 \pm 7.119\text{E-}07$	9%	93%	99%
9	0.007	0.000	0.000	$0.006 \pm 3.089\text{E-}07$	9%	93%	100%
10	0.003	0.000	0.000	$0.003 \pm 1.001\text{E-}07$	6%	94%	100%

The following tables show the result for a non fully loaded dispatch scenario for a two stations network.

Table 11: Minimum card capacity $d = 1$, maximum card capacity $c = 5$, $\lambda = 0.2$, $\mu_1 = \mu_2 = 0.5$

N	E[TT] (min)		% Error
	a_1	Simulation	
4	5.893	$6.604 \pm 6.969\text{E-}05$	11%
5	5.912	$6.628 \pm 8.861\text{E-}05$	11%
6	5.965	$6.640 \pm 1.209\text{E-}04$	10%
7	6.027	$6.656 \pm 2.463\text{E-}04$	9%
8	6.087	$6.664 \pm 7.422\text{E-}05$	9%
9	6.140	$6.665 \pm 8.618\text{E-}05$	8%
10	6.186	$6.668 \pm 5.618\text{E-}05$	7%

N	E[G] (min)		% Error
	a_1	Simulation	
4	5.582	$6.295 \pm 2.966\text{E-}05$	11%
5	5.767	$6.487 \pm 4.777\text{E-}05$	11%
6	5.898	$6.578 \pm 7.921\text{E-}05$	10%
7	5.997	$6.629 \pm 1.593\text{E-}04$	10%
8	6.073	$6.651 \pm 6.969\text{E-}05$	9%
9	6.134	$6.660 \pm 8.410\text{E-}05$	8%
10	6.183	$6.666 \pm 5.579\text{E-}05$	7%

N	E[W_e] (min)		% Error
	a_1	Simulation	
4	0.311	$0.310 \pm 1.349\text{E-}05$	0%
5	0.145	$0.140 \pm 1.106\text{E-}05$	3%
6	0.067	$0.063 \pm 7.475\text{E-}06$	7%
7	0.030	$0.028 \pm 1.325\text{E-}05$	11%
8	0.014	$0.013 \pm 3.136\text{E-}07$	8%
9	0.006	$0.005 \pm 1.306\text{E-}07$	21%
10	0.003	$0.002 \pm 6.461\text{E-}08$	26%

Table 12: Minimum card capacity $d = 1$, maximum card capacity $c = 5$, $\lambda = 0.3$, $\mu_1 = \mu_2 = 0.5$

N	E[TT] (min)		% Error
	a_1	Simulation	
4	7.075	$8.801 \pm 6.969\text{E-}05$	20%
5	7.325	$9.066 \pm 8.861\text{E-}05$	19%
6	7.551	$9.287 \pm 1.208\text{E-}04$	19%
7	7.754	$9.481 \pm 2.463\text{E-}04$	18%
8	7.936	$9.608 \pm 2.317\text{E-}04$	17%
9	8.097	$9.721 \pm 2.139\text{E-}04$	17%
10	8.240	$9.804 \pm 1.473\text{E-}04$	16%

N	E[G] (min)		% Error
	a_1	Simulation	
4	6.426	$7.636 \pm 2.966\text{E-}05$	16%
5	6.899	$8.304 \pm 4.777\text{E-}05$	17%
6	7.270	$8.788 \pm 7.921\text{E-}05$	17%
7	7.569	$9.153 \pm 1.593\text{E-}04$	17%
8	7.814	$9.397 \pm 1.699\text{E-}04$	17%
9	8.017	$9.584 \pm 1.704\text{E-}04$	16%
10	8.188	$9.715 \pm 1.305\text{E-}04$	16%

N	E[W_e] (min)		% Error
	a_1	Simulation	
4	0.649	$1.165 \pm 1.349\text{E-}05$	44%
5	0.426	$0.763 \pm 1.106\text{E-}05$	44%
6	0.281	$0.499 \pm 7.475\text{E-}06$	44%
7	0.185	$0.328 \pm 1.325\text{E-}05$	44%
8	0.122	$0.211 \pm 6.317\text{E-}06$	42%
9	0.080	$0.138 \pm 4.709\text{E-}06$	42%
10	0.052	$0.089 \pm 2.258\text{E-}06$	42%

Table 13: Minimum card capacity $d = 2$, maximum card capacity $c = 5$, $\lambda = 0.2$, $\mu_1 = \mu_2 = 0.5$

N	E[TT] (min)		% Error
	a_1	Simulation	
4	7.313	$6.980 \pm 2.255\text{E-}05$	5%
5	7.334	$6.980 \pm 4.042\text{E-}05$	5%
6	7.356	$6.978 \pm 2.407\text{E-}05$	5%
7	7.374	$6.977 \pm 3.309\text{E-}05$	6%
8	7.390	$6.977 \pm 2.279\text{E-}05$	6%
9	7.403	$6.977 \pm 3.309\text{E-}05$	6%
10	7.415	$6.978 \pm 4.477\text{E-}05$	6%

N	E[G] (min)		% Error
	a_1	Simulation	
4	4.784	$4.478 \pm 1.212\text{E-}05$	7%
5	4.825	$4.481 \pm 3.118\text{E-}05$	8%
6	4.852	$4.475 \pm 2.407\text{E-}05$	8%
7	4.871	$4.478 \pm 3.309\text{E-}05$	9%
8	4.886	$4.476 \pm 2.278\text{E-}05$	9%
9	4.898	$4.478 \pm 3.309\text{E-}05$	9%
10	4.908	$4.478 \pm 4.477\text{E-}05$	10%

N	E[W_e] (min)		% Error
	a_1	Simulation	
4	2.529	$2.502 \pm 9.427\text{E-}06$	1%
5	2.509	$2.499 \pm 1.051\text{E-}05$	0%
6	2.504	$2.502 \pm 6.405\text{E-}06$	0%
7	2.503	$2.498 \pm 9.979\text{E-}06$	0%
8	2.503	$2.501 \pm 5.026\text{E-}06$	0%
9	2.504	$2.498 \pm 9.979\text{E-}06$	0%
10	2.507	$2.500 \pm 1.835\text{E-}05$	0%

Table 14: Minimum card capacity $d = 2$, maximum card capacity $c = 5$, $\lambda = 0.3$, $\mu_1 = \mu_2 = 0.5$

N	E[TT] (min)		% Error
	a_1	Simulation	
4	6.995	$6.647 \pm 3.346\text{E-}05$	5%
5	7.042	$6.649 \pm 3.198\text{E-}05$	6%
6	7.084	$6.650 \pm 4.603\text{E-}05$	7%
7	7.120	$6.649 \pm 2.968\text{E-}05$	7%
8	7.150	$6.652 \pm 5.579\text{E-}05$	7%
9	7.176	$6.652 \pm 4.298\text{E-}05$	8%
10	7.201	$6.649 \pm 3.034\text{E-}05$	8%

N	E[G] (min)		% Error
	a_1	Simulation	
4	5.251	$4.964 \pm 3.410\text{E-}05$	6%
5	5.346	$4.980 \pm 3.400\text{E-}05$	7%
6	5.404	$4.983 \pm 4.056\text{E-}05$	8%
7	5.445	$4.984 \pm 4.579\text{E-}05$	9%
8	5.476	$4.986 \pm 4.783\text{E-}05$	10%
9	5.500	$4.986 \pm 2.769\text{E-}05$	10%
10	5.520	$4.982 \pm 3.193\text{E-}05$	11%

N	E[W_e] (min)		% Error
	a_1	Simulation	
4	1.744	$1.682 \pm 1.800\text{E-}06$	5%
5	1.697	$1.669 \pm 6.585\text{E-}05$	2%
6	1.680	$1.667 \pm 9.962\text{E-}05$	1%
7	1.675	$1.665 \pm 5.150\text{E-}06$	1%
8	1.674	$1.667 \pm 9.561\text{E-}05$	0%
9	1.676	$1.666 \pm 2.486\text{E-}06$	0%
10	1.681	$1.668 \pm 3.130\text{E-}06$	0%