ASIP performance measure approximation

Geelen, Wesley L.C.

Award date:
2023

Technische Universiteit
Eindhoven
University of Technology

July 24, 2023

ASIP performance measure approximation

Final Project - MSc Thesis
SPOR

W.L.C. (Wesley) Geelen (1029454)
w.l.c.geelen@student.tue.nl

Supervisors:

Maria Vlasiou
Yaron Yeger

Contents

Paper: ASIP sites occupancy probabilities approximation using the PSA

Paper: ASIP site occupancy PGF using two moment approximation

ASIP sites occupancy probabilities approximation using the PSA

Wesley Geelen

June 2023

Abstract

The asymmetric simple inclusion process (ASIP) is an $n-$site tandem stochastic network, used to model unidirectional clustered particle flow. At the end of each site in the system, there is a gate that regulates the flow of particles by stochastically opening and closing. When particles are present in a given site, they aggregate together, forming a cluster. Upon gate opening, this cluster moves as a single unit to the subsequent site, combining with the cluster in that site (if any). The existing methods for determining closed-form expressions of the steady-state site occupancy probabilities are computationally intractable for systems with a large number of sites. As a result, *performance measures*, which can be expressed as the expectation of a function (e.g., $g : \mathbb{N}^n \to \mathbb{R}$) applied to the stochastic process $\vec{X}$ representing the site occupancies of the system at steady-state, are generally similarly intractable. The power-series algorithm provides an alternative method of approximating these *performance measures*($\mathbb{E}[g(\vec{X})]$) compared to Monte Carlo simulation. The method utilizes a polynomial approximation expressed in terms of the system's traffic intensity. Here, we show that approximating these *performance measures* by means of the power-series algorithm is preferable over Monte Carlo simulation, when the number of sites is kept small. We found that the power-series algorithm yields approximations that are several orders of magnitudes more accurate than those obtained from Monte Carlo simulation for small systems, despite the simulation being computationally more expensive. Furthermore, we found that this advantage in performance of the power-series algorithm over simulation decreases as the traffic intensity of the system increases. Additionally, we found that the number of polynomial terms required by the power-series algorithm to achieve a given a order of accuracy advantage over simulation varies depending on the specific *performance measure*. Lastly, we found that a linear interpolation between the power-series algorithm approximation and a heavy-traffic limit is not more accurate than the power-series algorithm alone. Our results demonstrate that the power-series algorithm is preferable over Monte Carlo simulation for approximating *performance measures* of system with small number of sites. Additionally, our results demonstrate that using the same predetermined number of polynomial terms for each *performance measure* is non-robust and sub-optimal in terms of accuracy. We anticipate that our results will encourage the use of more sophisticated implementations of the power-series algorithm for approximation of ASIP performance measures. For example, the $\varepsilon -$ *algorithm* could be used to decrease the computational effort required to compute these approximations, thereby potentially increasing the range of system size for which the power-series algorithm is preferable over Monte Carlo simulation.

1 Introduction

The $n-$site Asymmetric Inclusion Process (ASIP) is a specialized case of a tandem stochastic system with n queues (sites). Here 'Asymmetric' refers to the property that customers (particles) move uni-directionally along the system. While 'Inclusion' refers to the inclusion principle. Which is the property that both site capacity and service (gate) capacity are unlimited. Stated differently, a given site allows any number of particles to be present which together form a cluster, upon gate opening all particles present move simultaneously (as a cluster) and instantaneously to the next site or out of the system. In the subsequent site the arriving cluster of particles combines with the particles already present (if any) to form a new cluster. If both the inter-arrival time distribution and inter-gate opening time distributions are exponential, then the system is referred to as the Classical ASIP. If the inter-gate opening time distributions is shared by all gates then the system is referred to as an homogeneous ASIP, otherwise it is referred to as an heterogeneous ASIP.

The service paradigm employed by the ASIP makes it particularly suitable for studying particle systems. It has been investigated for various applications in the literature. One of these applications is using the ASIP as an Aggregation-Fragmentation model for transport in biological systems (see [1], [2]). Another application is found in the study of the thermal dependence of chemical reaction rates. In [3], an Arrhenius law for the ASIP is derived, which captures the activation time of a system from a meta-stable state. Moreover, in [4], the ASIP is used as a thermal engine, for which a bound for the entropy production rate is derived. The ASIP has also been used to model the transport of bosonic particles in heat transfer [5]. Furthermore, the ASIP has been studied in quantum physics as a dynamical stochastic higher spin vector model (see [6]). Such models are at the center-point of the study of quantum gravity [7].

The ASIP has been studied extensively in the past (see [8]–[12]), resulting in closed-form expression for several

steady-state *performance measures*. In particular, closed-form expressions for the steady-state *load* (total number of particles in the system) are readily available for various variations of the ASIP. These the classical heterogeneous ASIP of arbitrary size (see [8]), a generalized ASIP of aritrary (see [12]), the PGF of the normalized *load* of a classical n-site ASIP under heavy-traffic (see [11]). However, it was shown in [8] that for the classical ASIP it is the case that closed-form expressions for the steady-state site occupancy probability generating function (PGF) are difficult to derive for systems of arbitrary size, closed-form expressions are only available for n up to 3, while for arbitrary sizes only recursive expressions were derived. In a follow up paper [12], a compact matrix approach was introduced to derive closed-form expressions of the steady-state site occupancy PGF of an $n-$site generalized ASIP. This novel method greatly reduced the computational effort required to derive the PGF to solving $\mathcal{O}(2^n)$ linear equations. Although this method provides a systematic manner in which to compute the steady-state site occupancy probabilities PGF it is considered computationally intractable for large size systems. Therefore, a systematic manner of computing closed-form expressions of *performance measures* ($\mathbb{E}[g(\vec{X})]$, that can be expressed as the expectation of a function g applied to the stochastic process $\vec{X}$ representing the steady-state site occupancies) which utilizes the aforementioned compact matrix approach can be consider computationally intractable. Approximation provides a computational tractable alternative to closed-form expressions, a systematic manner of approximating steady-state performance measures is by using Monte Carlo simulation. In [8], Monte Carlo simulation was used to approximate how the mean site occupancy and the standard deviation of the site occupancy depend on the site number k. In [10], Monte Carlo simulation was to used to approximate how the site occupancy probability, mean site occupancy conditioned on occupation of the site, and the coefficient of variation of site occupancy depend on the site number k. Additionally, Monte Carlo simulation was used to fit coefficients for affine asymptotic stochastic approximations of the steady-state *load, draining time, inter-exit time, and coalescence time*. In [11], Monte Carlo simulation was used to compare various *performance measures* to exact asymptotic results under heavy-traffic, large-system, and balanced systems regimes. We see that despite the extensive use of Monte Carlo simulation in the study of the ASIP, it has not yet compared to other approximation methods.

In this paper, we compare the power-series algorithm (PSA) to Monte Carlo simulation for approximating the steady-state performance measures of the classical n-site ASIP, this comparison is in terms of accuracy and computational effort. We derive a numerical scheme using the PSA to systemically approximate steady-state performance measures by a polynomial. Additionally, we derive a polynomial approximation for the performance measure with an arbitrary function g using the PSA symbolically. Moreover, we investigate whether the approximation of the *load* obtained by the PSA is more accurate than a linear interpolation between the PSA and the closed-form heavy-traffic expression of the *load*. Our findings indicate that the PSA is orders of magnitude more accurate and computationally efficient than Monte Carlo simulation when the ASIP has a small number of sites. We also find that the PSA is more accurate than the linear interpolation, making the PSA standalone preferable over the linear interpolation. However, it provides an impetus to consider more elaborate interpolation schemes. Furthermore, we find that when it comes to accuracy the optimal number of polynomial terms for approximating a performance measure varies between performance measures. Consequently, when using the PSA, in the approximation of various arbitrary performance measures using the same (apriori choice of) number of polynomial terms is sub-optimal. This suggests that dynamic approaches to select the number of polynomial terms should be investigated to determine if they are robust against this effect.

The power-series algorithm is an approximation method for the steady-state probabilities of multi-site quasi-birth-death processes pioneered by Hooghiemstra et al [13], and extensively studied by Le Blanc (see [14]–[18]). The PSA approximates steady-state performance measures by employing a truncated power-series (i.e, polynomial) expansion in terms of the traffic intensity ρ of the system. The coefficients of this power-series are recursively determined from the global-balance equations of the system. To systematically compute the terms of the resulting polynomial, an iterative numerical scheme is utilized. Each iteration of the scheme results in the addition of a new polynomial term to the approximation. In the current paper, we use the PSA to derive such a numerical scheme for approximating steady-steady *performance measures* of the ASIP. This includes the derivation of global-balance equations and recursive equations for the coefficients of the power-series expansion.

The derived numerical scheme was used symbolically, to determine the coefficients of a 2^{nd} order polynomial approximation of the general performance measure $E[g(\vec{X})]$ of a classical $2-$site ASIP. Symbolically, in this context, refers to the use of symbols that represent the parameters of the system, particularly λ (the inter-arrival rate) and μ (inter-gate opening rate). The term *general performance measure* denotes a steady-state performance measure where the function g applied to the steady-state site occupancy process remains unspecified.

The accuracy of the approximations produced by the PSA and Monte Carlo simulation were compared in terms of their respective relative error to exact values of the performance measures. This comparison was made for the classical three site homogeneous ASIP and various heterogeneous ASIP models. The specific performance measures used in this analysis were the *load* and *busy probability*, for which closed-form expressions are known (see [8] and [11]). We find that the number of polynomial terms associated with the most accurate approximation was significantly larger for the *busy probability* than the *load*. Suggesting, that the optimal number of polynomial terms to be included in the approximation is performance-measure-dependent. Furthermore, we find that the accuracy of the PSA decreases

as the traffic intensity ρ of the system decreases.

The computational effort is defined as the time spent by either the PSA or Monte Carlo simulation to approximate a single steady-state performance measure of an n-site ASIP. The comparison of computational effort between the two methods is conducted by considering the ratio of the Monte Carlo simulation computation time to the PSA computation time on a $\log_{10}$ scale, effectively comparing the computational effort in terms of orders of magnitude. We find that the PSA is significantly less computationally demanding than Monte Carlo simulation by several orders of magnitude for 2 and 3 site systems. In these cases, the PSA approximation included up-to 50 polynomial terms, these PSA approximations are at the same time more accurate. Additionally, we find that as the number of sites n in the system increases, the computational effort of the PSA becomes relatively larger compared to Monte Carlo simulation. Consequently, we conclude that the PSA is only more beneficial than Monte Carlo simulation when the number of sites in the system is kept small.

In literature, it has been established that the PSA is inaccurate for heavy-traffic systems (see [18]). In this paper, we aim to ameliorate this limitation by interpolating between the PSA and a heavy-traffic fluid limit. Traditionally, traffic is indicated by ρ, defined as inflow rate over the outflow rate (i.e., $\rho = \frac{\lambda}{\mu}$). In a heavy-traffic setting, the behavior of a stochastic process can be approximated using a fluid-limit of the process. The fluid limit arises by scaling the original process, the scaling is chosen in such a manner such that the scaled-process tends to a non-degenerate limit. Specifically, both the arrival rate λ and the process X are scaled to $\lambda f(m)$, and $X/f(m)$, respectively, with f increasing in m so that the arrival rate tends to infinity. Meaning, that the non-degenerate limit of $X/f(m)$ if it exists, is used as a proxy for the heavy-traffic behavior of the original process. In [11], such a heavy-traffic limit was derived for the normalized *load* of the $n-$site ASIP, where the appropriate scaling was determined to be linear, i.e., $f(m) = m$. The resulting limiting process was shown to be equal in law to a sum of n independent exponential random variables. In the current paper we interpolate linearly between the PSA approximation and this heavy-traffic fluid limit, aiming to improve the quality of our approximation as $\rho \to 1$. In this paper the performance of our interpolation is compared to both the PSA and exact values for the *performance measures load* and *busy probability*. We found that the interpolation overestimates the exact values, while the PSA only does so, and to a lesser extent, for the *busy probability* as $\rho \to 1$. As a result, we conclude that a linear interpolation between the PSA and heavy-traffic limit is less accurate than the PSA alone and, therefore, not worthwhile using.

Our results show that the PSA is preferable over Monte Carlo simulation for the approximation of steady-state performance measures of the classical ASIP, when the number of sites in the system is kept small. We view the PSA to be especially worthwhile using when the traffic intensity of the system is low (i.e., $\rho << 1$), but consider its use to be less straight-forward when the traffic intensity is high (i.e, ρ close to 1). Additionally, we find that approximating the *load* by linearly interpolating between the PSA and the heavy-traffic value of the *load* is inferior to the approximating the *load* by the PSA alone.

The decrease in accuracy observed when ρ increases highlights further studies using the method of *conformal mapping* or the $\varepsilon -$ *algorithm* to increase the convergence radius of the power-series, and thereby potentially increasing the traffic intensity range for which the PSA remains accurate. We found that the preference of PSA over Monte Carlo simulation is limited by increasingly prohibitive increase of computational effort that results from increasing number of sites in the system. This suggests that further study were more sophisticated implementations of the PSA that can reduce the computational effort required is warranted. For instance, the $\varepsilon -$ *algorithm* could be considered as it increases the convergence speed and thus reduces the computational burden. We anticipate that such an implementation may expand the range of system sizes for which the PSA is a preferable choice over Monte Carlo simulation. Contrary to previous interpolation results using more elaborate schemes in [19]–[21], our current study finds that the linear interpolation is not worthwhile compared to the PSA alone. However, this outcome prompts the consideration of more elaborate interpolation schemes for future research, as they have shown success in improving the quality of approximations using heavy-traffic fluid limits.

This paper is organized as follows. In the two subsections that follow we discuss some of the literature for both the ASIP and the PSA. In Section 2, we provide the model description for the ASIP. In Section 3, we first provide some preliminary results. Subsequently, we derive recursive equations for the coefficients of the power-series. Lastly, we provide the numerical scheme for the PSA. In Section 4, we use the numerical scheme derived in the previous section in a symbolic manner, which yields a symbolic representation of the truncated power-series. In Section 5, we compare the performance of the PSA with the MC simulation. In Section 6, we interpolate between the PSA and a Heavy-traffic result. This interpolation is compared in performance to the PSA. In Section 7, we address the limitations of the current study. In Section 8, we provide the main conclusion of our study. Lastly, in Appendix A we provide results that are analogous to results in the main text, but where omitted for brevity's sake.

1.1 Literature review ASIP

In this section we address some of the previous results on the ASIP. Various methods were used to analyse the ASIP [8]–[12] these include, analysis, steady-state analysis of the state probability generating function (PGF) and the *load* PGF. These efforts resulted in some interesting results including explicit solutions of various steady-state performance measures, and law-like phenomena. We will discuss these results in the following.

First, let's consider the traversal time T, which represents the random time it takes for a particle to traverse the system. In the Markovian setting under consideration, characterized by Poissonian arrivals with parameter λ and exponential gate opening times with parameter μ_k for gate k where $k = 1, \ldots, n$, the traversal time distribution can be explicitly determined. Due to the memoryless property of the exponential distribution, particles spend, on average, an exponential amount of time in each site. Consequently, the traversal time T is simply the sum of these exponential times. Under this assumption, the mean and variance of the traversal time are given by $\mathbb{E}T = \frac{1}{\mu_1} + \cdots + \frac{1}{\mu_n}$ and $\mathbb{V}T = \frac{1}{\mu_1^2} + \cdots + \frac{1}{\mu_n^2}$, respectively.

Next, let's consider performance measures such as the *load*, which refers to the number of particles present at a site or in the entire system. Through analysis, it was shown [8] that the steady-state expected site occupancy is given by $\mathbb{E}X_k = \frac{\lambda}{\mu_k}$. Furthermore, it was demonstrated that the steady-state probability generating function (PGF) of the combined occupancy of the first m sites (as presented in Equation (50) in [8]) is equivalent in law to the PGF of a sum of m independent geometric random variables. From this, we can derive the explicit expressions for the mean and variance of the total occupancy of the first m sites, which are given by $\mathbb{E}[\sum_{k=1}^{m} X_k] = \lambda \left(\frac{1}{\mu_1} + \cdots + \frac{1}{\mu_m} \right)$ and $\mathbb{V}[\sum_{k=1}^{m} X_k] = \lambda \left(\frac{\mu_1+\lambda}{\mu_1^2} + \cdots + \frac{\mu_m+\lambda}{\mu_m^2} \right)$, respectively.

Now, let's direct our attention to the aforementioned law-like phenomena, some of which are well-known in Queuing theory. One such phenomenon is Little's law [22], which was previously shown [8] to hold for the ASIP. In the context of the ASIP, the version of Little's law asserts that the mean number of particles in the system, $\mathbb{E}[\sum_{k=1}^{n} X_k]$, is given by the product $\lambda \mathbb{E}T$. Here, λ represents the flow rate of particles into the system, and $\mathbb{E}T$ represents the mean traversal time. Another well-known result in Queuing theory, the PASTA phenomenon, has also been shown to hold for the ASIP [8]. PASTA states that the fraction of particles that find the system in a particular state S is equivalent to the fraction of time the system spends in state S. This finding is consistent with the general result in [23], which asserts that PASTA holds for systems with Poissonian arrivals.

Another set of noteworthy results for the ASIP are the asymptotic power laws, which exemplify its complex behavior. In [10], it was demonstrated that the ASIP exhibits several asymptotic power laws at steady-state. Specifically, the site occupancy probability, mean number of particles occupying a site, and the standard deviation of the number of particles occupying a site can all be expressed as some power of k, where k denotes the site number. Notably, the former decreases while the latter two increase as k increases. In other words, sites that are located farther along a very large ASIP (i.e., when site number k is close to n) become increasingly volatile. These sites are less likely to be occupied by any particles at all, and when they are occupied, the number of particles present tends to be very large. Consequently, the aforementioned clustering effect results in increasing clustering sizes as the site number k increases.

From the above, it is evident that the previous study of the ASIP has been fruitful, resulting in several explicit results. However, further investigation is warranted since explicit results regarding the occupancy (state of the system) remain elusive. Unlike the PGF for the *load*, which has an explicit formula, the occupancy PGF is not analytically tractable. In [8], explicit solutions for the occupancy PGF are provided for systems up to size $n = 3$. Additionally, the complex nature of the iterative solution branching tree is addressed. In a more recent paper [12], the PGF is tackled using a compact matrix approach instead of an iterative substitution method. This approach successfully reduces the complexity of computing the PGF to solving a set of $\mathcal{O}(2^n)$ linear equations.

1.2 Literature review PSA

The Power-Series Algorithm (PSA) is an efficient method for computing performance measures of queuing systems that are quasi birth-and-death processes. Apart from computing performance measures, the PSA can also be used to compute derivatives of such performance measures. This capability allows the PSA to optimize a system under consideration in terms of a specific performance measure [18].

In the past, PSA has been applied to various types of queuing systems, including polling, parallel server, parallel processor, and layered queue systems [15], [18], [24]. Moreover, it has been demonstrated in [18] that the PSA can also be applied to tandem queue systems and QBDPs with migration.

The PSA is based on expanding the steady-state probabilities into a power-series, often taken in terms of the traffic intensity ρ of the system. The coefficients of the resulting power-series can be computed recursively by substituting

the power-series into the global balance equations. Additionally, the power-series is substituted into the normalization equation, leading to a recursive set that can be solved. When the power-series is expanded in terms of the load ρ, some researchers prefer [18] to normalize the arrival rates so that the system is stable for $0 \leq \rho \leq 1$. Under such assumptions, it was shown in [18] that the power-series exists in terms of the traffic intensity ρ at $\rho = 0$. Once this recursive set for the steady-state probabilities is determined, it can be easily extended to a recursive set for performance measures. However, these performance measures are limited to being the expectation of some function of the system state. As a result, a numerical scheme is devised to compute the terms of the power-series for such a performance measure.

Note that the PSA typically expands in terms of the traffic intensity ρ, and the definition of ρ corresponds to the stability criterion of the specific process under consideration. For the $M/M/1$ queue, the stability criterion is $\frac{\lambda}{\mu} < 1$, leading to $\rho := \frac{\lambda}{\mu} = \lambda \mathbb{E}T$. However, the ASIP has an unlimited service capacity, and as a result, lacks a stability criterion. Therefore, we maintain $\rho = \lambda \mathbb{E}T$ in line with the definition of ρ for the $M/M/1$ queue. As mentioned earlier, the PSA results in a numerical scheme for computing the terms that constitute the power-series of the performance measure. In practice, one truncates this power-series, leading to an approximation. It is evident that the number of terms truncated influences the quality of the approximation.

An intriguing approach is to employ the PSA not only for numerical computations of performance measures but also for symbolic derivations. For instance, in [14], the waiting times of a single server polling system were symbolically derived, and in [18], the symbolic derivations of joint queue lengths in a layered machine repair system was demonstrated. Symbolic derivations allows for the expression of performance measures in terms of system parameters, such as arrival rates and service rates. However, it is essential to truncate the power-series at a low power of ρ and limit the number of parameters in the system. Otherwise, the derivations become computationally intractable.

Previous studies have demonstrated that the PSA approximation performs well in low-traffic settings when ρ is small [15], [25]. However, its performance declines as ρ approaches one, i.e., in high-traffic settings. In an attempt to improve the performance at higher values of ρ, techniques to increase the rate of convergence of the power-series have been explored [17]. Another approach involves interpolating with heavy-traffic results, which is the method of choice in the current paper.

2 Model description

The model considered in this paper is an $n-$site ASIP. It consists of n unlimited capacity sites and n gates. Each site is followed by a gate, the resulting site-gate pairs are arranged in tandem fashion. Meaning that gate k is located at the end of site k, and before site $k+1$ (if $k < n$). The flow of particles through the system is illustrated in Figure 17. The particles arrive only at the first site, all particles present at a given site form a cluster. Upon opening of gate k any particles present in site k move instantaneously and simultaneously as a cluster to site $k+1$, where it forms a new cluster with the particles present (if any). If gate n opens the particles present in site n leave the system. We restrict ourselves to the Markovian setting. That is, particles arrive according to a Poisson process with rate λ. The inter-gate opening times are independent of other gates. The inter-gate opening time of gate j is exponentially distributed with parameter μ_j. Under these assumptions the ASIP is referred to as a classical ASIP.

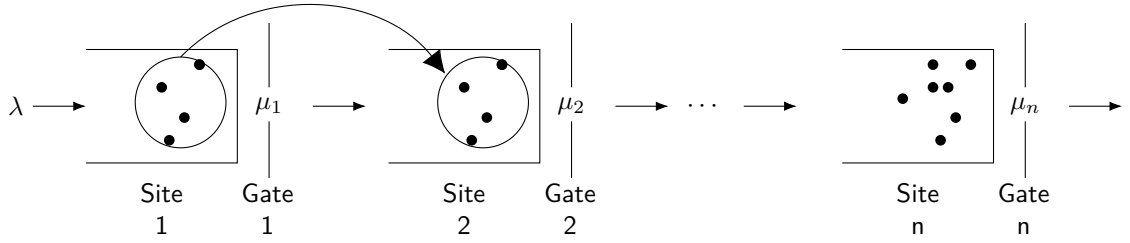


Figure 1: The heterogeneous classical n-site ASIP system

3 PSA

In this section we use the PSA derive numerical schemes for the computation of steady-state probabilities and *performance measures* of the heterogeneous classical $n-$site ASIP. Firstly, we introduce some preliminary results necessary for the use of the PSA. Secondly, we derive recursive equations for the coefficients of the power-series expansions of the steady-state probabilities and *performance measures*. Lastly, we derive numerical schemes for the polynomial approximation of both the steady-state probabilities and *performance measures*.

3.1 Preliminaries

Before we can apply the PSA, we must study the stochastic process vector $\{\vec{X}(t) \in \mathbb{N}^n : t \geq 0\}$ that counts the number of particles in the system at time t. Given this notation the stochastic process $\{X_j(t) \in \mathbb{N} : t \geq 0\}$ counts the number of particles in site $j, j = 1, \ldots, n$ at time t.

For $\vec{X} \in \mathbb{N}^n$, $j = 1, \ldots, n$, we define the one-step transition rates as follows:

$$\begin{aligned} \rho a_j(\vec{X}) &: \quad \text{the arrival rate to site } j \text{ when system is in state } \vec{X}, \\ d_j(\vec{X}) &: \quad \text{the departure rate from site } j \text{ when system is in state } \vec{X}. \end{aligned}$$

Particles can only depart from site j if the number of particles at site j is nonzero, hence $d_j(\vec{X}) = 0$ when $X_j = 0$. Within the system particles only move to the subsequent sites, thus the departure rate at a given site must match the arrival rate in the next site. Hence $\rho a_j(\vec{X}) = d_{j-1}(\vec{X})$ for $j = 2, \ldots, n$. In general we find that

$$\begin{cases} \rho a_1(\vec{X}) = \lambda & j = 1 \\ \rho a_j(\vec{X}) = d_{j-1}(\vec{X}) = \mu_{j-1}\mathbb{1}_{\{X_{j-1}>0\}} & j = 2, \ldots, n-1 \\ d_n(\vec{X}) = \mu_n \mathbb{1}_{\{X_n>0\}} & j = n. \end{cases}$$

Using the expressions above we find that the global balance equations of the steady-state probabilities $p(\vec{X})$ are as follows:

$$\begin{aligned} &\left[\rho a_1(\vec{X}) + \sum_{j=1}^{n} d_j(\vec{X})\right] p(\vec{X}) = \sum_{i=2}^{n}\sum_{m=1}^{X_i} \mathbb{1}_{\{X_{i-1}=0\}}\mathbb{1}_{\{X_i>0\}} d_{i-1}(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i)) p(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i)) \\ &+ \mathbb{1}_{\{X_1>0\}}\rho a_1(\vec{X} - \hat{e}_1)p(\vec{X} - \hat{e}_1) + \sum_{m=1}^{\infty} \mathbb{1}_{\{X_n=0\}} d_n(\vec{X} + m\hat{e}_n) p(\vec{X} + m\hat{e}_n). \end{aligned} \tag{1}$$

Furthermore, the normalisation equation is given by

$$\sum_{\vec{X}\in\mathbb{N}^n} p(\vec{X}) = 1. \tag{2}$$

Before we proceed with the PSA, we introduce the following preliminary results.

Proposition 3.1. *For each state $\vec{X} \in \mathbb{N}^n$ it holds that $p(\vec{X}) = \mathcal{O}(\rho^{|\vec{X}|})$.*

Proof. For the proof we refer to Theorem 3.2 in [26] with the choice $\ell(\vec{X}) = |\vec{X}|$. This theorem applies to ASIP since Assumption 3.1 holds for tandem queues where transitions only occur from one queue to the next, which is the case for the ASIP. □

Proposition 3.2. *Let $\varepsilon > 0$. Suppose it holds for all $\rho \in [0, \varepsilon)$ that*

$$\sum_{k=0}^{\infty} c_k \rho^k = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} c_{ij}\rho^{i+j}.$$

Then it holds for all $k \in \mathbb{N}$ that

$$c_k = \sum_{i,j\in\mathbb{N}^2} \mathbb{1}_{\{i+j=k\}} c_{ij}.$$

Proof. First, notice that the equality holds for all $\rho \in [0, \varepsilon)$, so we may evaluate in $\rho = 0$. Now differentiating the equality m times with respect to ρ, followed by evaluating in $\rho = 0$ yields the result for c_m. Now simple induction on m gives the result for all $m \in \mathbb{N}$. □

3.2 Recursive equations

In the following we derive recursive equations which form the key ingredients to the PSA.

Recall that ρ represents the load of the system, then we define the power series

$$p(\vec{X}) = \sum_{k=0}^{\infty} b_k(\vec{X})\rho^k.$$

It follows from proposition (3.1) that $b_k(\vec{X}) = 0$ when $k < |\vec{X}|$. Therefore, we can write

$$p(\vec{X}) = \sum_{k=0}^{\infty} b_k(\vec{X})\rho^{|\vec{X}|+k}, \tag{3}$$

where we reuse dummy variable b_k. Now, we can proceed to derive recursive equations for these coefficients $b_k(\vec{X})$. To this end we substitute the above power series (3) in to the balance equations (1). This yields, the following

$$\begin{aligned}\left[\rho a_1(\vec{X}) + \sum_{j=1}^{n} d_j(\vec{X})\right] \sum_{k=0}^{\infty} b_k(\vec{X})\rho^{|\vec{X}|+k} &= \rho a_1(\vec{X} - \hat{e}_1)\mathbb{1}_{\{X_1>0\}} \sum_{k=0}^{\infty} b_k(\vec{X} - \hat{e}_1)\rho^{|\vec{X}|+k-1} \\ + \sum_{i=2}^{n}\sum_{m=1}^{X_i} d_{i-1}(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i))\mathbb{1}_{\{X_{i-1}=0\}}\mathbb{1}_{\{X_i>0\}} &\sum_{k=0}^{\infty} b_k(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i))\rho^{|\vec{X}|+k} \\ &+ \sum_{m=1}^{\infty} d_n(\vec{X} + m\hat{e}_n)\mathbb{1}_{\{X_n=0\}} \sum_{k=0}^{\infty} b_k(\vec{X} + m\hat{e}_n)\rho^{|\vec{X}|+k+m}. \end{aligned} \tag{4}$$

Upon reordering and bringing all ρ within their corresponding sums, we find

$$\begin{aligned}\sum_{j=1}^{n} d_j(\vec{X}) \sum_{k=0}^{\infty} b_k(\vec{X})\rho^{|\vec{X}|+k} &= -a_1(\vec{X}) \sum_{k=0}^{\infty} b_k(\vec{X})\rho^{|\vec{X}|+k+1} + a_1(\vec{X} - \hat{e}_1)\mathbb{1}_{\{X_1>0\}} \sum_{k=0}^{\infty} b_k(\vec{X} - \hat{e}_1)\rho^{|\vec{X}|+k} \\ + \sum_{i=2}^{n}\sum_{m=1}^{X_i} d_{i-1}(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i))\mathbb{1}_{\{X_{i-1}=0\}}\mathbb{1}_{\{X_i>0\}} &\sum_{k=0}^{\infty} b_k(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i))\rho^{|\vec{X}|+k} \\ &+ \sum_{m=1}^{\infty} d_n(\vec{X} + m\hat{e}_n)\mathbb{1}_{\{X_n=0\}} \sum_{k=0}^{\infty} b_k(\vec{X} + m\hat{e}_n)\rho^{|\vec{X}|+k+m}. \end{aligned} \tag{5}$$

Next we eliminate factor $\rho^{|\vec{X}|}$ on both sides, which yields

$$\begin{aligned}\sum_{j=1}^{n} d_j(\vec{X}) \sum_{k=0}^{\infty} b_k(\vec{X})\rho^{k} &= -a_1(\vec{X}) \sum_{k=0}^{\infty} b_k(\vec{X})\rho^{k+1} + a_1(\vec{X} - \hat{e}_i)\mathbb{1}_{\{X_1>0\}} \sum_{k=0}^{\infty} b_k(\vec{X} - \hat{e}_1)\rho^{k} \\ + \sum_{i=2}^{n}\sum_{m=1}^{X_i} d_{i-1}(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i))\mathbb{1}_{\{X_{i-1}=0\}}\mathbb{1}_{\{X_i>0\}} &\sum_{k=0}^{\infty} b_k(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i))\rho^{k} \\ &+ \sum_{m=1}^{\infty} d_n(\vec{X} + m\hat{e}_n)\mathbb{1}_{\{X_n=0\}} \sum_{k=0}^{\infty} b_k(\vec{X} + m\hat{e}_n)\rho^{k+m}. \end{aligned} \tag{6}$$

Now this equation holds for all $\rho \in [0, \epsilon)$ and the equation is of the form $\sum_k c_k\rho^k = \sum_{i,j} c_{ij}\rho^{i+j}$, where all but the last line on the RHS has $i \in \{0\}$. Meaning that condition for proposition (3.2) is satisfied. Using this we find the following recursive equation for the coefficients $b_k(\vec{X})$,

$$\begin{aligned}\sum_{j=1}^{n} d_j(\vec{X})b_k(\vec{X}) &= -a_1(\vec{X})b_{k-1}\mathbb{1}_{\{k>0\}} + a_1(\vec{X} - \hat{e}_1)\mathbb{1}_{\{X_1>0\}}b_k(\vec{X} - \hat{e}_1) \\ + \sum_{i=2}^{n}\sum_{m=1}^{X_i} d_{i-1}(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i))\mathbb{1}_{\{X_{i-1}=0\}}\mathbb{1}_{\{X_i>0\}}&b_k(\vec{X} + m(\hat{e}_{i-1} - \hat{e}_i)) \\ &+ \mathbb{1}_{\{X_n=0\}} \sum_{i=0}^{k-1}\sum_{m=1}^{k} d_n(\vec{X} + m\hat{e}_n)b_i(\vec{X} + m\hat{e}_n)\mathbb{1}_{\{k=i+m\}}. \end{aligned} \tag{7}$$

Next, we set out to find a recursive equation for the coefficients $b_k(\vec{0})$. To this end we substitute the power series form of $p(\vec{X})$ as given in Equation (3) into the normalization equation (2), yielding

$$1 = \sum_{\vec{X}\in\mathbb{N}^n} p(\vec{X}) = \sum_{\vec{X}\in\mathbb{N}^n} \sum_{k=0}^{\infty} b_k(\vec{X})\rho^{|\vec{X}|+k} \tag{8}$$

By considering that $\rho = 0$ gives $b_0(\vec{0}) = 1$, we can rewrite the above as follows

$$b_0(\vec{0}) = \sum_{\vec{X}\in\mathbb{N}^n:|\vec{X}|>0} \sum_{k=0}^{\infty} b_k(\vec{X})\rho^{|\vec{X}|+k} + \sum_{k=1}^{\infty} b_k(\vec{0})\rho^k + b_0(\vec{0}) \tag{9}$$

Upon elimination of $b_0(\vec{0})$ on both sides, we can see that the equation satisfies property (3.2). Using this we find the following recursive equation for $b_k(\vec{0})$

$$b_k(\vec{0}) = -\sum_{0<|\vec{X}|\leq k} b_{k-|\vec{X}|}(\vec{X}). \tag{10}$$

Together with $b_0(\vec{0}) = 1$, Equations (7) and (10) form the basis of the recursive scheme which we will use to determine the coefficients $b_k(\vec{0})$. However, we must first establish the order in which these equations must be used. We recognize that this set of equations admits a total ordering $\prec$ of the vectors $(k, \vec{X})$, where $(k', \vec{X}') \prec (k, \vec{X})$ if

$$\begin{aligned}
&\left[k' + |\vec{X}'| < k + |\vec{X}|\right] \text{ or} \\
&\left[k' + |\vec{X}'| = k + |\vec{X}| \wedge k' < k\right] \text{ or} \\
&\left[k' = k \wedge |\vec{X}'| = |\vec{X}| \wedge X_i' > X_i \wedge X_{i+1}' < X_{i+1} \wedge |X_i - X_i'| = |X_{i+1} - X_{i+1}'| \leq X_{i+1}\right].
\end{aligned} \tag{11}$$

3.3 Numerical scheme

In the following we use the total ordering (see Equation 11) and the recursive set for coefficients b_k (see Equations 7 and 10) to construct numerical schemes to determine the occupancy probabilities and mean performance measures. The iterative numerical scheme is in practice terminated after say $N \in \mathbb{N}$ steps, since it is unrealistic to compute all terms in the power series. Meaning that by terminating after N steps we approximate the power series of $p(\vec{X})$ by

$$p(\vec{X}) \approx \sum_{k=1}^{N} b_k(\vec{X})\rho^k.$$

Similarly, for some performance measure $\mathbb{E}g(\vec{X})$ we approximate by

$$\mathbb{E}g(\vec{X}) \approx \sum_{\vec{X}\in\mathbb{N}^n} \sum_{k=1}^{N} g(\vec{X})b_k(\vec{X})\rho^k.$$

Note that by truncating at step N we consider only the information provided by the states $\vec{X}$ in the following set $\{\vec{X} \in \mathbb{N}^n : |\vec{X}| + l = k, l \in \mathbb{N}, 0 \leq k \leq N\}$. Regardless the approximation holds for all $\vec{X} \in \mathbb{N}^n$.

Notice that the terms which are discarded are of order $o(\rho^N)$. Meaning that the accuracy of the aforementioned approximations increase as $\rho \to 0$. Recall that ρ represents the load of the system, meaning that the approximations are particularly well suited to capture low traffic behavior.

Next, we will give the numerical scheme to approximate the site occupancy probabilities. The numerical scheme computes the power series term by term. Where each term requires the usage of the recursive set of equations to be determined. The scheme is then given by the following,

1. Choose $N \in \mathbb{N}$ and set $l := 1$.
2. For all $(k, \vec{X}) \in \mathbb{N}^{n+1}$ with $\vec{X} \neq \vec{0}$ and with $k + |\vec{X}| = l$, compute $b_k(\vec{X})$ recursively using Equation (7) according to the ordering (11).
3. Compute $b_l(\vec{0})$ using Equation (10).
4. Set $l := l + 1$, if $l \leq N$ return to step (2), otherwise stop.

As noted before not only does the set of balance equations allow for approximation of the steady-state site occupancy probabilities, it also allows us to determine functions of the steady-state occupancy. Consider for example $g : \vec{X} \mapsto \mathbb{R}$, various performance measures can be expressed in the form $\mathbb{E}g(\vec{X})$. Next, we seek to extend the above numerical scheme to allow for the computation of such performance measures. It can be shown that $\mathbb{E}g(\vec{X})$ can be rewritten as a power series in terms of $g(\vec{X})$ and $b_k(\vec{X})$, see for example [24]. The resulting power series is given by

$$\mathbb{E}g(\vec{X}) = \sum_{k=0}^{\infty} \rho^k f(k),$$

where the coefficients $f(k)$ are given by

$$f(k) := \sum_{l=0}^{k} \sum_{|\vec{X}|=l} g(\vec{X}) b_{k-l}(\vec{X}). \tag{12}$$

In a similar manner as the previous scheme, the scheme to compute performance measures proceeds term by term. In particular, we are required to determine the coefficients $f(k)$. These terms consist of sums of the b_k terms determined in the previous scheme, this allows for a simple extension of the previous scheme to compute performance measures. By extending the previous scheme we find the following numerical scheme for the computation of performance measures $\mathbb{E}g(\vec{X})$.

1. Choose $N \in \mathbb{N}$ and set $m := 1$,
2. Compute $f(0) = g(\vec{0})b_0(\vec{0}) = g(\vec{0})$
3. For all $(k, \vec{X}) \in \mathbb{N}^{n+1}$ with $\vec{X} \neq \vec{0}$ and with $k + |\vec{X}| = m$, compute $b_k(\vec{X})$ recursively using Equation (7) according to the ordering (11).
4. Compute $b_m(\vec{0})$ using Equation (10).
5. Compute $f(m)$ according to Equation (12) using the $b_k(\vec{X})$ determined in the previous steps
6. Set $m := m + 1$, if $m \leq N$ return to step (3), otherwise stop.

Upon completion of this scheme we have an approximation of $\mathbb{E}g(\vec{X})$, namely $\sum_{m=0}^{N} f(m)\rho^m$.

4 Symbolic PSA

In this section we will apply the PSA symbolically to compute the performance measure according to the scheme presented in the previous section. For this we will consider a 2-site heterogeneous ASIP, with arrival parameter λ, and inter-gate opening time rates μ_1, μ_2 for gates $1, 2$, respectively. For the performance measure, we consider the generic function $g : \mathbb{N}^2 \mapsto \mathbb{R}$. To keep the computations manageable we run the scheme with $N = 2$. In doing so the resulting power-series approximation of $\mathbb{E}g(\vec{X})$ is given in term of the coefficients $f(k)$ with $k = 1, \ldots 2$. We present these coefficients $f(k)$ symbolically in terms of the parameters, in table (1).

Table 1: Symbolic expression(s) of the performance measure power-series coefficients for the 2-site heterogeneous ASIP .

Coefficient Order k	Coefficient $f(k)$
0	$g(0,0)$
1	$\frac{a_1}{\mu_1\mu_2}\left[g(0,1) + \mu_2 g(1,0) - [1+\mu_2]g(0,0)\right]$
2	$\frac{a_1^2}{\mu_1^2\mu_2^2[\mu_1+\mu_2]}\Big[[2\mu_1 + \mu_1\mu_2 + 2\mu_1\mu_2^2 + \mu_2^3]g(0,0) - \mu_2[\mu_1 + 2\mu_1\mu_2 + 2\mu_2^2]g(1,0)$ $-[2\mu_1^2(1+\mu_2) + \mu_1\mu_2 + 2\mu_1\mu_2^2]g(0,1) + [\mu_2(\mu_1+\mu_2)]g(2,0) + \mu_1\mu_2 g(1,1)$ $+[\mu_1\mu_2(2\mu_1+\mu_2)]g(0,2)\Big]$

Note that a_1 is defined as $a_1 = \frac{\lambda}{\rho}$.

Remark 4.1. *Manual computation of the Symbolic expression is laborious. Therefore, restricting both the number of sites and terms should be considered to keep the computational effort reasonable. Those that desire more terms and or sites could instead consider Symbolic computation packages, to implement the PSA Scheme. Examples of such packages are SymPy, Mathematica, and Matlab's Symbolic Math Toolbox.*

5 PSA performance

In the following sections we compare the performance of the PSA to Monte Carlo simulation. Firstly, we introduce the experimental settings used for this comparison. Secondly, we compare them in terms of accuracy. Thirdly, we compare them in terms of computational effort. Lastly, we investigate the numerical breakdown of the PSA.

5.1 Experimental settings

In this section, we conduct a numerical evaluation of the PSA by comparing its results to those obtained through Monte Carlo simulation. To enable a comprehensive comparison, we consider various ASIP models, including homogeneous systems with up to four sites and different three-site heterogeneous systems.

A homogeneous ASIP implies that all gates have exponential distributions with the same parameter, denoted by μ. To facilitate a meaningful comparison among the heterogeneous systems mentioned earlier, we impose the following condition on the gate parameters

$$C \equiv \sum_{i=1}^{n} \frac{1}{\mu_i}, \tag{13}$$

where we choose $C = 3$, and $n = 3$, and where μ_i refers to the gate parameter of gate i.

Table 2: Experimental ASIP model specification

Name	μ_1	μ_2	μ_3
Homogeneous	1	1	1
Incline	0.34	34	34
Decline	34	34	0.34
Convex	34	0.34	34
Concave	0.67	67	0.67

Remark 5.1. *The gate parameter values shown in Table 2 were determined as follows. Firstly, for the homogeneous model, we set each gate parameter to the value of one, serving as our reference. Next, to observe and highlight different effects of traffic slowdown and speedup, we scale up certain gate parameters by a factor of 100. The choice of the factor 100 is somewhat arbitrary, but it is intended to exaggerate the effects and make observable differences apparent. Finally, we rescale these values to satisfy the constraint given by Equation 13.*

5.2 Accuracy

In this section, we compare the accuracy of the PSA and Monte Carlo simulation. We evaluate their respective relative errors with respect to exact values for both the *load* and *busy probability*. The exact value of the *load* was determined in [8] and takes the form

$$\mathbb{E}|\vec{X}| = \lambda \sum_{i=1}^{n} \frac{1}{\mu_i}. \tag{14}$$

Similarly, the exact value of the Empty probability was determined in [9]. From this the equation for *busy probability* is easily determined to be

$$\mathbb{P}(|\vec{X}| > 0) = 1 - \prod_{j=1}^{n} \frac{\mu_j}{\lambda + \mu_j}. \tag{15}$$

Using these exact values, we can calculate the relative error for both the PSA and simulation, denoted by $\mathrm{RE}_{\mathrm{PSA}}$ and $\mathrm{RE}_{\mathrm{SIM}}$, respectively. To compare their performance, we consider the ratio of these relative errors on the $\log_{10}$ scale, which helps capture the magnitude difference in performance. Thus, we evaluate the following performance measure

$$\log 10(\mathrm{RE}_{\mathrm{SIM}} \ / \ \mathrm{RE}_{\mathrm{PSA}}).$$

In this performance measure, negative values indicate that the simulation outperforms the PSA, zero indicates performance parity, while positive values indicate that the PSA outperforms the simulation. For the Monte Carlo simulation, we use 1280000 realizations, and for the PSA, we consider between 1 and 50 iterations.

Remark 5.2. *In the following, we omit the Incline and Convex models since their results are essentially the same as those for the Decline model, thus omitting them for brevity. Their similarity can be seen in Appendix A.2.*

In Figure 2, we consider the $\log_{10}$ relative error ratio for the *busy probability*. For the Homogeneous and Concave models, we observe that ten and twenty PSA iterations, respectively, are sufficient to outperform simulation for all values of ρ considered. Specifically, for the Homogeneous model, the PSA can achieve an 11.5-order of magnitude performance advantage for all values of ρ considered, in fifty or fewer PSA iterations. For the Concave model, this advantage is 5 orders of magnitude. However, the Decline model can only match the performance of simulation for values of ρ at or below 0.85. Nevertheless, for values of ρ below 0.6, this model can achieve an 8-order magnitude advantage over simulation.

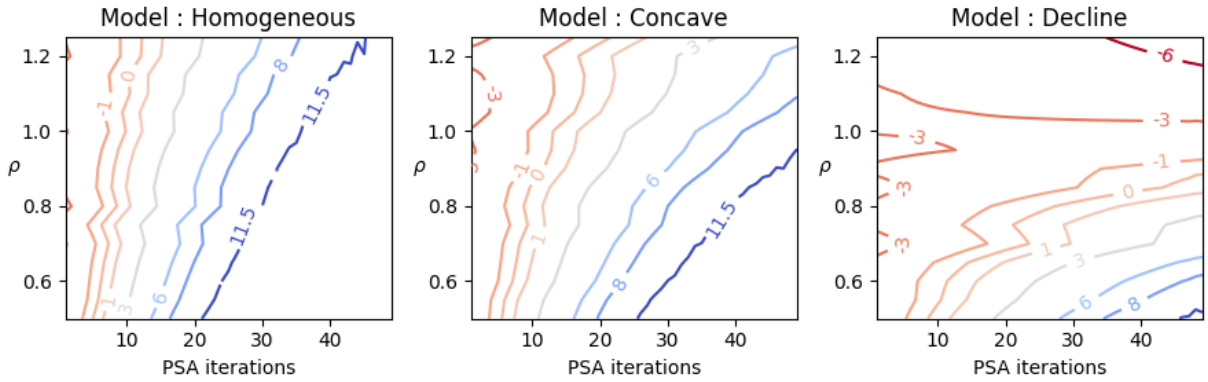


Figure 2: $\log_{10}$ Relative Error ratio (RE_{sim} / RE_{psa}) - contour plot - *busy probability*

In Figure 3, we consider the $\log_{10}$ relative error ratio for the *load*. For all models, we observe that at five PSA iterations, the PSA achieves a 12-order of magnitude advantage over simulation for all values of ρ considered. Once again, larger values of ρ lead to worsened PSA performance when PSA iterations are kept fixed. It can be seen that for the *load*, increasing the number of PSA iterations at a fixed ρ results in worse PSA performance. However, only the Decline model is outperformed by simulation when large ρ is combined with more than thirty-five PSA iterations. On the other hand, the other models maintain at least a 6-order magnitude advantage across the board.

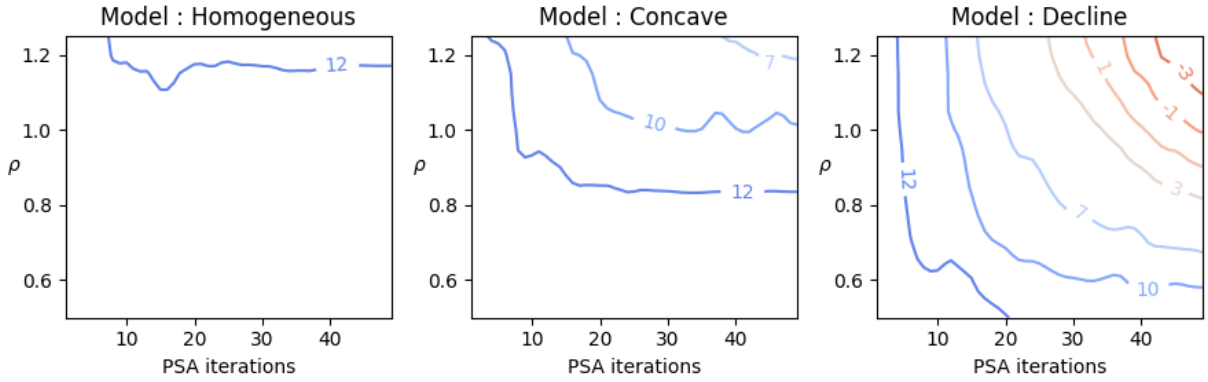


Figure 3: $\log_{10}$ Relative Error ratio (RE_{sim} / RE_{psa}) - contour plot - *load*

The above results can be summarized as follows: (1) The PSA outperforms simulation by several orders of magnitude at smaller values of ρ for both the *load* and *busy probability*. (2) At larger values of ρ, the PSA maintains this several-order magnitude advantage for both the Homogeneous and Concave models. (3) However, for the Decline model, the advantage over simulation is present only at small values of ρ for the *busy probability*. For values of ρ exceeding 0.85, the Decline model is unable to match simulation in performance.

From these findings, we can conclude that the PSA outperforms simulation by several orders of magnitude, provided that ρ is kept small and a sufficient number of PSA iterations are considered.

The above results demonstrate that the PSA performs well in terms of accuracy compared to simulation. However, to determine how good of an approximation the PSA provides, we need to consider only the relative error of the PSA with respect to the exact value.

In Figure 4, we observe the relative error of the *busy probability*. The Homogeneous and Concave models exhibit excellent performance, with ten and twenty iterations respectively being sufficient to achieve a relative error at or below 10^{-3} for all values of ρ. Moreover, these models can achieve a relative error at or below 10^{-14} up to a value of ρ equal to one with fifty PSA iterations. In contrast, the Decline model performs significantly worse, requiring

fifty iterations at a value of ρ equal to 0.85 to achieve a relative error of 10^{-3} or below. Furthermore, it is the only model that has relative errors exceeding one, occurring for values of ρ exceeding one, regardless of the number of iterations.

Overall, we observe the following: (1) PSA performance follows the decreasing order of Homogeneous, Concave, and Decline models in terms of accuracy. (2) Larger values of ρ necessitate more PSA iterations to achieve the same relative error. (3) Increasing the number of PSA iterations at a fixed value of ρ reduces the relative error.

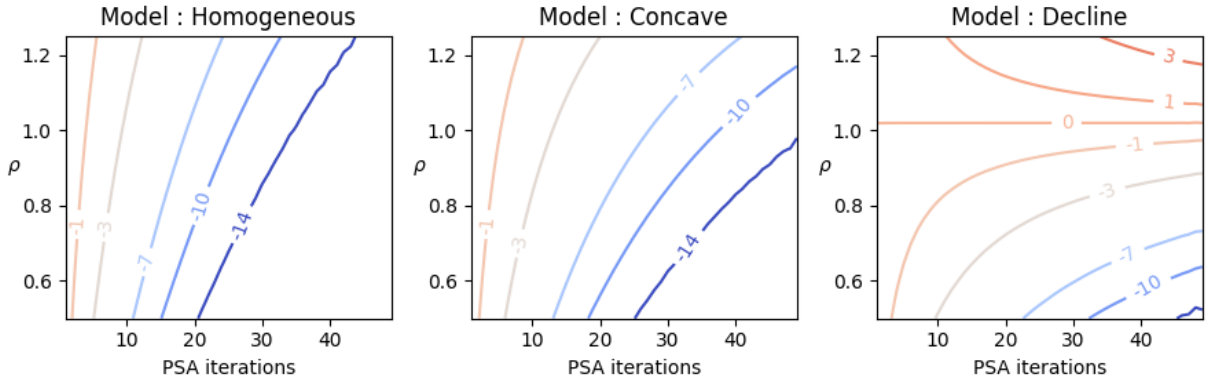


Figure 4: Log_{10} relative error (exact) - contour plot - *busy probability*

In Figure 4, we examine the relative error of the *load*. For all models, we observe that ten PSA iterations are sufficient to achieve a relative error at or below 10^{-15} for all values of ρ considered. Similar to the *busy probability*, increasing ρ results in worse performance when the number of PSA iterations is fixed. Moreover, we see the same performance ordering among the models as seen in the *busy probability* analysis. However, unlike the *busy probability*, increasing the number of PSA iterations results in worsened performance for the *load*.

This discrepancy between performance measures is consistent with the performance measure-dependent numerical breakdown addressed in section 5.4.1.

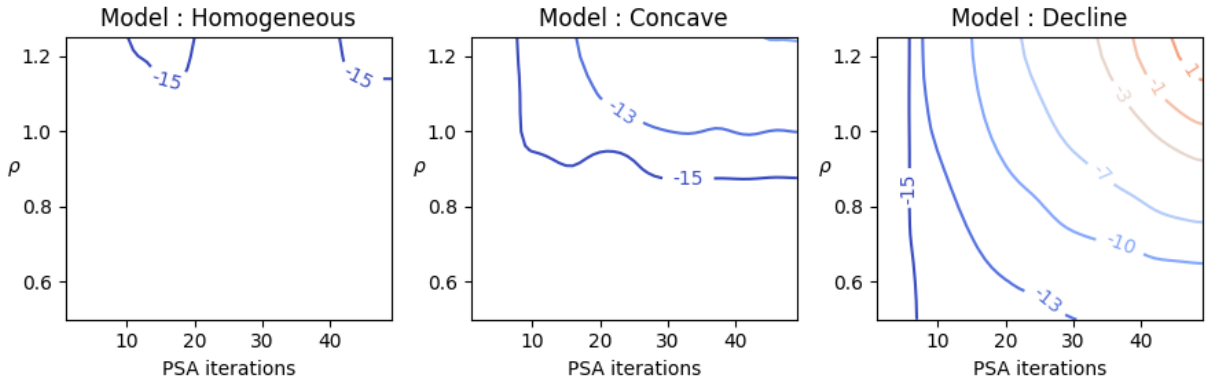


Figure 5: Log_{10} relative error (exact) - contour plot - *load*

5.3 Computational complexity

To justify the practical use of the PSA over simulation, there should be benefits of using the former over the latter. The question we investigate next is whether the (possible) sacrifice made in terms of accuracy is met with significant advantages in terms of computational complexity. To analyze this, we consider the ratio of simulation time to the PSA computation time in seconds on a base-ten logarithmic scale, as shown in Figure 6.

We observe that compared to a simulation with over one million iterations, the computation time needed for the PSA with one to twelve iterations is six to four magnitudes less than the simulation. Surprisingly, the computation time required for this number of PSA iterations is even one order of magnitude less than that of a simulation with only a thousand iterations. Such a simulation can be considered significantly inaccurate. This suggests that in terms of computation time, the PSA is undoubtedly worth using.

One might wonder how this computation time ratio is affected by the number of sites in the ASIP. We find that the number of sites in the ASIP indeed strongly influences the computation time ratio, as shown in Figure 7. We observe

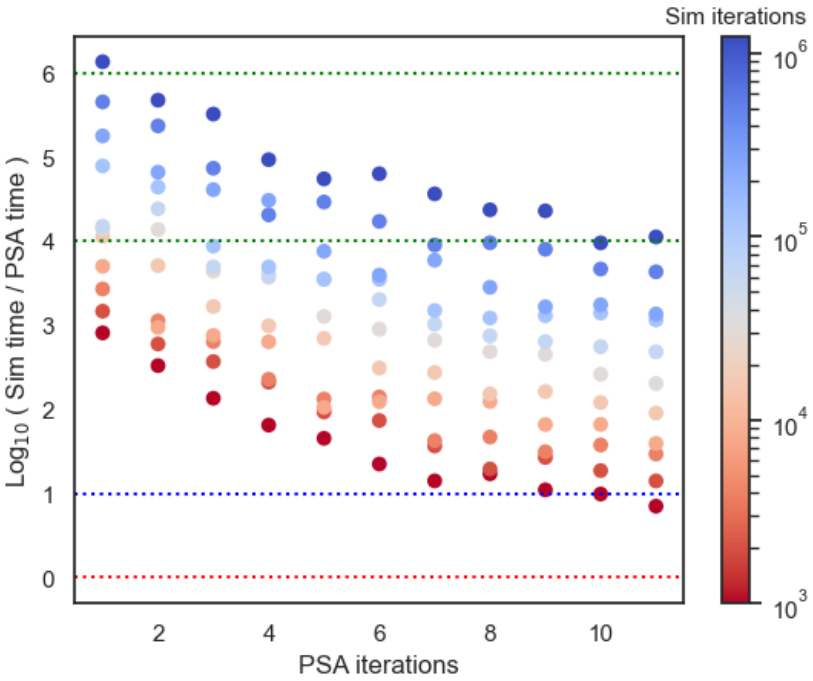


Figure 6: Magnitude computation time difference of simulation to the PSA

that at fifty PSA iterations, the advantage of the PSA decreases by one order of magnitude per site added to the ASIP.

Another property of note is that each subsequent PSA iteration is computationally more expensive than the previous one. This is due to the fact that the k-th iteration considers all n-vectors up to *load* k, resulting in both memory and processing time expenses.

This observation suggests that when dealing with systems with a large number of sites, we should limit the number of PSA iterations to make it worthwhile. Furthermore, there is a limit to how many sites can be considered beyond which the PSA becomes less practical than simulation. Consequently, in practical use of the PSA, a trade-off must be made between the number of sites and the number of iterations to be considered.

Remark 5.3. *Unlike the number of sites, neither the model nor the performance measure elicit a strong effect on the computation time ratio. For brevity, we choose to depict this in Appendix A.3.*

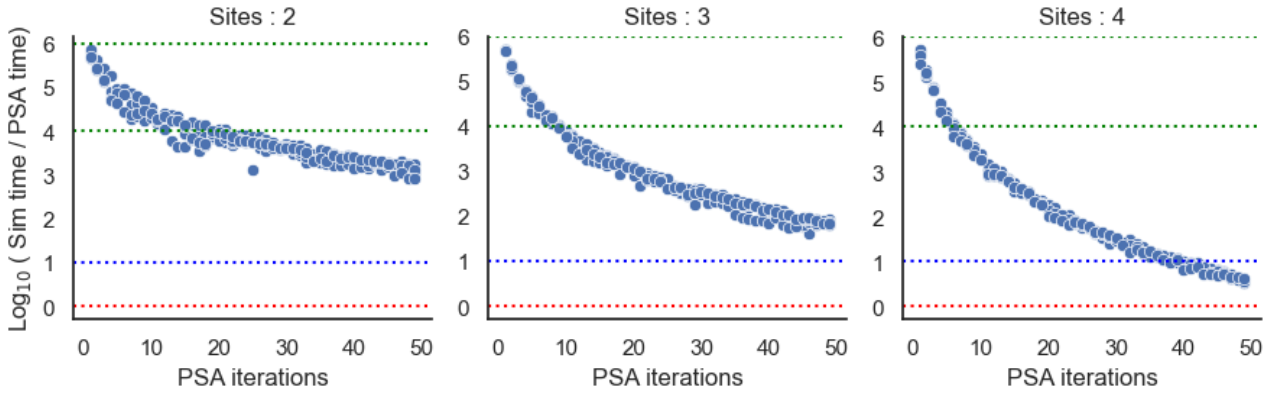


Figure 7: Magnitude computation time difference of simulation to the PSA - Effect of the number of sites

Based on the above findings, we can expect that in terms of computational effort, the PSA may not be worthwhile compared to simulation for an ASIP with more than fifteen sites. However, for systems with a small number of sites, the PSA can outperform simulation in terms of computation time.

5.4 Numerical breakdown

In this section, we investigate the numerical breakdown of the PSA and examine the effects of various parameters, such as ρ, the number of PSA iterations, the type of model, and the number of sites.

First, we address a constrained optimization problem for which results from the literature are available. Given a fixed arrival rate λ, the objective is to find the gate parameter allocation μ_i that minimizes the *load*, while subject to the constraint $\sum_{i=1}^{n} \mu_i = C$, where C is a positive constant. This constraint limits the total amount of resources available to operate the gates. Previous studies in [8] and [9] have shown that for this constrained optimization problem, the homogeneous allocation of gate parameters (i.e., a homogeneous model) minimizes the *load*. The homogeneous model was also found to be optimal for various similar constrained optimization problems.

To explore the effects of λ, we consider various values for this parameter and a constraint of $\sum_{i=1}^{3} \mu_i = 3$, where the homogeneous model has all gate parameters equal to 1.

Running the PSA for nine iterations, we obtained the results shown in Table 2. As expected, for values of λ from 10^{-5} to 0.98, the homogeneous model achieves the smallest *load*, consistent with previous findings. However, for λ values of 0.99 and 1, the Incline model produces a negative *load*, which is invalid since the actual *load* must be non-negative. This indicates that the PSA is unable to produce valid results for these values of λ after nine iterations.

Similarly, when considering the decline model for λ values of 0.98 and 0.999, a seemingly small change in λ results in a large change in the approximated *load*. This suggests that the PSA may suffer from inaccuracy in these cases.

Table 3: *load* optimization

	PSA approximated *load*				
(μ_1, μ_2, μ_3) / λ	Incline (0.01, 1.49, 1.49)	Concave (0.03, 2.94, 0.03)	Convex (1.49, 0.01, 1.49)	Decline (1.49, 1.49, 0.01)	Homogeneous (1.0, 1.0, 1.0)
.00001	0.0007	0.0007	0.0007	0.0007	.00003
0.0010	0.0683	0.0683	0.0683	0.0683	0.0030
0.1000	6.8340	6.8340	6.8340	6.8340	0.3000
0.5000	29.8069	34.1490	34.5339	39.7696	1.5000
0.9000	57.9162	87.9207	3078.6535	34.3065	2.7000
0.9800	57.4380	123.7861	6559.5431	9.2717	2.9400
0.9990	-2169.5272	57.5770	253.3570	2934.1528	2.9970
1.0000	-2189.7241	57.5486	255.1002	2960.1684	3.0000

In the following, we investigate if the observed inaccuracy in the incline and decline models at $\lambda = 0.999$ can be mitigated by adjusting the number of PSA iterations. Specifically, we consider the incline and decline models with $\lambda = 0.999$ and iterate from 0 to 14 times, as shown in Table 4.

For both models, we find that iterations from one to seven yield relatively consistent values. However, starting from eight iterations, clear signs of numerical breakdown become evident. Each subsequent iteration results in a two-fold change in magnitude, and the value changes sign multiple times during this iteration range.

Limiting the number of iterations to values between one and seven seems to avoid numerical breakdown and can improve accuracy. Reducing the number of PSA iterations below seven does not lead to significant changes in the results. On the other hand, increasing iterations further exacerbates the effects of numerical breakdown.

Interestingly, we observe that the sign of the approximated *load* changes several times as the number of iterations increases. This suggests that the previously observed negative value of the incline model at nine iterations is also present in the decline model but at a different iteration number. This implies that the previously observed erroneous negative value of the incline model's *load* is likely a result of numerical breakdown, rather than an independent effect.

Overall, these experiments indicate that the aberrant results observed were the consequence of numerical breakdown. This numerical breakdown occurs when the number of PSA iterations becomes sufficiently large. In Table 3, no signs of numerical breakdown were observed for various combinations of model and λ. Therefore, the iteration value at which numerical breakdown occurs is likely dependent on traffic (λ) and the model, with larger traffic volume and heterogeneity being more prone to numerical breakdown.

Based on the above analysis, we can conclude that the PSA is indeed susceptible to numerical breakdown. The factors that affect this susceptibility are the traffic intensity λ, the model's heterogeneity, and the number of PSA iterations.

Table 4: PSA *load* approximation - extreme cases

PSA iterations	Incline	Decline
0	0.0000	0.0000
1	68.2717	68.2717
2	68.2717	68.2717
3	68.2717	68.2717
4	68.2717	68.2717
5	68.2717	68.2716
6	68.2716	68.2699
7	68.4431	68.3006
8	91.8726	43.5677
9	-2169.5272	2934.1528
10	-370084.9183	-391755.9291
11	-235406852.2741	-62287856.1330
12	32717498957.2634	957720645.9064
13	1854741374801.5564	-1828274742509.6406
14	-538827162873100.3750	-488632544442142.8750

When all these factors are large, the likelihood of numerical breakdown increases. However, we also find that by reducing one or more of these factors, we can avoid numerical breakdown and improve the accuracy of the PSA approximation. Specifically, limiting the number of PSA iterations and avoiding extremely high traffic intensity or excessive heterogeneity can help prevent numerical breakdown and enhance the reliability of the PSA results.

Remark 5.4. *In Table 3, we did not observe numerical breakdown for the homogeneous model. However, this does not imply that this model is immune to numerical breakdown. It simply indicates that the specific combination of λ and PSA iterations we considered did not result in numerical breakdown for the homogeneous model. To further investigate this, we need to examine a wider range of values for the traffic ρ and increase the number of PSA iterations.*

Furthermore, it's essential to note that we only considered the load performance measure and did not explore the busy probability. Hence, in the following analysis, we will investigate how the choice of performance measure influences numerical breakdown in the PSA.

5.4.1 Performance-measure-dependent numerical breakdown

In this section, we explore and compare the behavior of the PSA for different performance measures, specifically focusing on the *load* and *busy probability*. We aim to understand how numerical breakdown varies depending on the chosen performance measure and its implications for the practical application of the PSA to the ASIP.

Remark 5.5. *We focus on two key performance measures: the busy probability and the load. These measures are representative of the behavior observed in several other performance measures, such as the empty probability, squared load, and occupancy of the first site (as detailed in Appendix A.4). Considering these additional performance measures is not expected to provide new insights but would require significant computational resources. As a result, we limit our analysis to the busy probability and the load.*

Remark 5.6. *In the following, we demonstrate that the numerical breakdown behavior of the PSA is dependent on the performance measure. We illustrate this using the decline model as an example, but it is important to note that this behavior is also present in the other models considered, as shown in Appendix A.4.*

Now, we will address the experimental settings used to highlight this difference in behavior. We will consider the relative error of the expected performance measure between the simulated value and the PSA approximation, using the three-site heterogeneous decline model as an example (see Table 2). For the PSA, we consider up to 50 iterations, and for the simulation, we use 1280000 realizations.

Firstly, let's explore the numerical breakdown behavior for the *busy probability*, as shown in Figure 8. For each value of ρ shown (0.95, 1, and 1.05), we can observe that the relative error starts above 0.5 when a single PSA iteration is used. As we increase the number of PSA iterations, the relative error monotonically reduces for ρ equal to 0.95 and 1. However, this is not the case for ρ equal to 1.05, where we see the relative error actually increasing with further iterations.

Upon further investigation, we notice the differences between ρ equal to 0.95 and 1. We observe that the former achieves larger reductions in relative error per iteration compared to the latter. Moreover, the former reaches a

relative error below 0.01 with fifty iterations, while the latter does not manage to drop below 0.15 even with fifty iterations.

Overall, we observe that for ρ equal to 0.95, a small relative error can be achieved in fifty iterations, for ρ equal to 1, although we see a reduction in relative error, fifty iterations are insufficient to achieve an acceptable relative error, and for ρ equal to 1.05, the relative error increases rapidly. This change in behavior captures a transition from convergence to divergence. Hence, we can conclude that in this case, the PSA breaks down numerically for values of ρ larger than one.

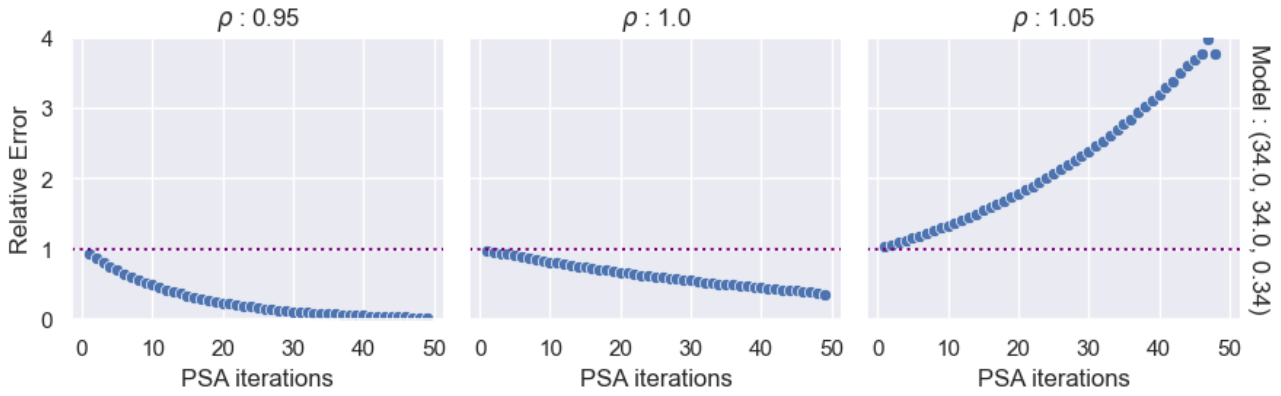


Figure 8: *busy probability* - divergence

Next, we examine the numerical breakdown behavior for the *load*, as shown in Figure 9. For all values of ρ, we observe that the relative error at a single iteration is around 0.005. Unlike what we observed for the *busy probability*, for the *load*, increasing the number of PSA iterations does not result in noticeable changes to the relative error until we reach forty iterations. In fact, any changes in relative error up to that point are below 0.001. Beyond forty iterations, we notice that for ρ equal to 0.95, 1, and 1.05, the relative error experiences significant changes: a strong decrease, a strong increase, and an explosion, respectively.

For the *load*, when numerical breakdown occurs, the relative error remains small and stable initially. However, as we increase the number of iterations sufficiently, the relative error eventually explodes. The number of iterations required before observing this explosion decreases as ρ increases. Once again, the behavior resulting from increasing ρ indicates numerical breakdown. Therefore, we can assert that the PSA diverges for values of ρ larger than one in the case of the *load* performance measure as well.

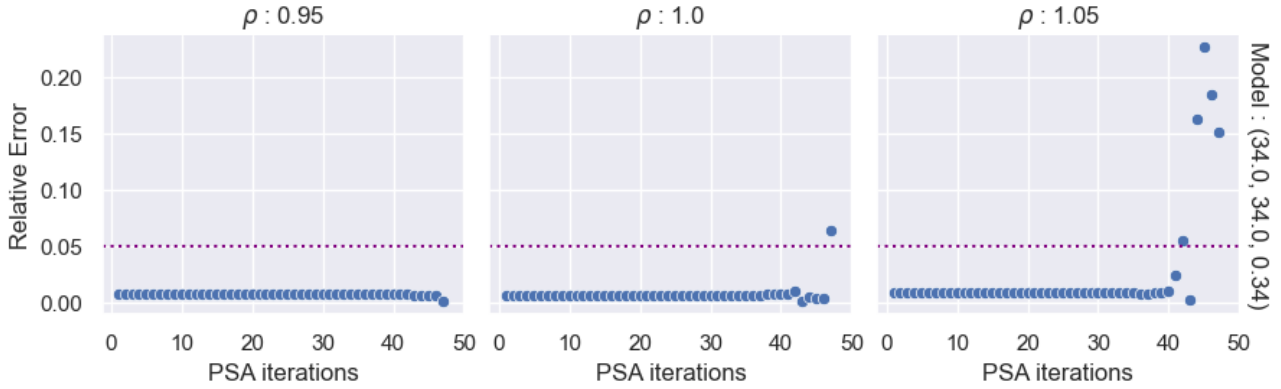


Figure 9: *load* - Numerical Breakdown

Now, let's highlight the important differences in behavior between the *load* and *busy probability*. Firstly, the *load* achieves a relative error below 0.01 even with only a single iteration, whereas the *busy probability* requires several iterations (dependent on ρ) to reach a relative error below 0.01. Secondly, the first several iterations (dependent on ρ) of the PSA for the *busy probability* result in significant changes (above 0.01) to the relative error, in contrast to the *load*. Thirdly, when numerical breakdown takes place, the relative error for the *load* remains small (below 0.01) for the first several iterations before exploding. However, for the *busy probability*, the relative error immediately explodes.

From the above, we can see that numerical breakdown behavior occurs for values of ρ larger than one, for both the

relative error of the *load* and *busy probability*. Interestingly, when we compare the numerical breakdown behavior seen for these two performance measures, we find a clear qualitative difference between them. This difference in behavior has practical consequences. For the *load*, the PSA can still provide accurate results beyond the value of ρ at which numerical breakdown occurs, provided that a small number of iterations are considered. However, for the *busy probability*, the PSA does not yield acceptable results beyond the value of ρ at which numerical breakdown takes place. Another consequence is that it is not worthwhile to consider more than a few PSA iterations when dealing with the *load*.

This performance measure-dependent behavior holds true for all of the different ASIP models given in Table 2. Although the same qualitative behavior is present, there are quantitative differences. The main quantitative difference lies in the value of ρ at which numerical breakdown occurs. Refer to Table 5 for both the *busy probability* and the *load*. We can see that for every model, numerical breakdown takes place at the same value of ρ or a smaller value for the *load* compared to the *busy probability*. Moreover, the heterogeneous models experience numerical breakdown at smaller values of ρ than the homogeneous model. Interestingly, the Concave model distinguishes itself from the other heterogeneous models, with numerical breakdown taking place at a larger value of ρ than the other three. Furthermore, for the homogeneous models, increasing the number of sites results in numerical breakdown occurring at larger values of ρ. Therefore, we can conclude that the PSA is more stable for larger and homogeneous models.

Table 5: factor dependence of Numerical breakdown

Name	Sites	Breakdown ρ - *busy probability*	Breakdown ρ - *load*
Incline	3	1	1
Decline	3	1	1
Convex	3	1	1
Concave	3	1.35	1
Homogeneous	1	1	1
Homogeneous	2	1.9	1.5
Homogeneous	3	2.5	2.1
Homogeneous	4	2.5	2.5

From the above, we can conclude that numerical breakdown is dependent on the performance measure being considered. Specifically, the performance measure *busy probability* is less susceptible to numerical breakdown than the *load*. Additionally, we find that the Homogeneous model is indeed susceptible to numerical breakdown, but it occurs at larger values of ρ compared to Heterogeneous models. Furthermore, Homogeneous models with more sites experience numerical breakdown at larger ρ values than those with fewer sites.

6 Heavy traffic interpolation

In this section, we investigate if we can improve the performance of the PSA in heavy traffic situations through interpolation. Firstly, we explain the motivation behind such interpolation. Secondly, we provide details of the interpolation and its components. Lastly, we compare its performance to the PSA and known exact values.

Existing literature [15], [25] has established that the PSA performs well for small values of ρ but deteriorates as ρ approaches one. This observation aligns with the findings in Section 5.2. In general, approximation methods without limitations are preferred. Therefore, a method that accurately approximates the ASIP across the entire traffic spectrum (small and large ρ) is more desirable than the PSA. Now, consider the possibility of a technique that accurately determines $E[g(\vec{X})]$ as ρ tends to one. In such a scenario, it would be advantageous to combine this technique with the PSA to obtain approximations that are accurate for values of $\rho \in (0, 1]$. This suggests using a linear interpolation that favors the PSA at low values of ρ and the other technique at higher values of ρ.

Fortunately, techniques that fulfill this requirement already exist and are well-established in the literature. These techniques include Fluid and Heavy traffic limits. The basic idea behind these limits is to appropriately scale the arrival rate λ and the *load* of the process using an increasing function of N denoted as $f(N)$. The resulting limit process as N tends to infinity is known as the Fluid limit. The scaling is considered appropriate when the resulting limit process is non-degenerate.

Remark 6.1. *In the following, we demonstrate that scaling the parameters of the n-site heterogeneous ASIP by $f(N)$ results in an equivalent system. Let φ_N be the probability generating function (PGF) of the load of the ASIP with arrival rate λ_N and gate parameters $\mu_{(k,N)}$ for gates $k = 1, \ldots, n$. Due to the scaling, we have $\lambda_N = \lambda \cdot f(N)$ and $\mu_{(k,N)} = \mu_k \cdot f(N)$.*

Next, we substitute these parameters into the PGF of the ASIP as determined in [11]. We then factor out $f(N)$ from both the numerator and denominator, yielding:

$$\varphi_N(z) = \prod_{k=1}^{n} \frac{\mu_{(k,N)}}{\mu_{k,N} + \lambda_N(1-z)} = \prod_{k=1}^{n} \frac{\mu_k \cdot f(N)}{\mu_k \cdot f(N) + f(N) \cdot \lambda(1-z)} = \prod_{k=1}^{n} \underbrace{\frac{f(N)}{f(N)}}_{1} \frac{\mu_k}{\mu_k + \lambda(1-z)} = \varphi(z).$$

Since $\varphi(s)$ is the PGF of the non-scaled ASIP, we have shown that the scaled and non-scaled systems are equivalent.

In [11], a Heavy traffic limit was determined for the n-site heterogeneous ASIP using a linear scaling in N for the normalized *load*. Applying this scaling to the known Laplace transform of the normalized *load* and taking N to infinity results in the following limit

$$\lim_{N\to\infty} \mathbb{E}\left[\exp\left(-s\frac{|\vec{X}|}{\langle m\rangle \lambda N}\right)\right] = \prod_{k=1}^{n} \frac{1}{1 + \frac{m_k}{\langle m\rangle} s},$$

where $m_k = \frac{1}{\mu_k}$ and $\langle m\rangle = \frac{1}{n}\sum_{k=1}^{n} m_k$. This limiting process is equal in distribution to the sum of n independent exponential random variables with means $\frac{m_k}{\langle m\rangle}$, where $k = 1, \ldots, n$.

From this, we can easily determine the expected *load* and *busy probability* for the limiting system. Let $|\vec{X}|^*$ be the Heavy traffic limit of the *load*, then the expected *load* and the *busy probability* are respectively given by

$$\mathbb{E}\left[|\vec{X}|^*\right] = n, \qquad \text{and} \qquad \mathbb{P}\left(|\vec{X}|^* > 0\right) = 1.$$

Now that we have determined the Heavy traffic limits for our performance measures of interest, we have the necessary ingredients to build our interpolation. Let $\hat{\mathbb{E}}\left[|\vec{X}|\right]$ PSA and $\hat{\mathbb{P}}\left(|\vec{X}| > 0\right)$ PSA be the performance measures as approximated by the PSA. For $\rho \in (0, 1]$, our Heavy Traffic interpolation for the expected *load* and the *busy probability* are respectively given by

$$(1-\rho)\hat{\mathbb{E}}\left[|\vec{X}|\right]_{\text{PSA}} + \rho\mathbb{E}\left[|\vec{X}|^*\right], \qquad \text{and} \qquad (1-\rho)\hat{\mathbb{P}}\left(|\vec{X}| > 0\right)_{\text{PSA}} + \rho\mathbb{P}\left(|\vec{X}|^* > 0\right).$$

Next, we compare the performance of these interpolations to both the PSA approximation and the known exact values as given in Equations 14 and 15. For the sake of convenience, we consider fifteen PSA iterations. We use the three-site decline model (see Table 2) since we know that the PSA performance for this model declines as ρ tends to one. The results are depicted in Figure 10.

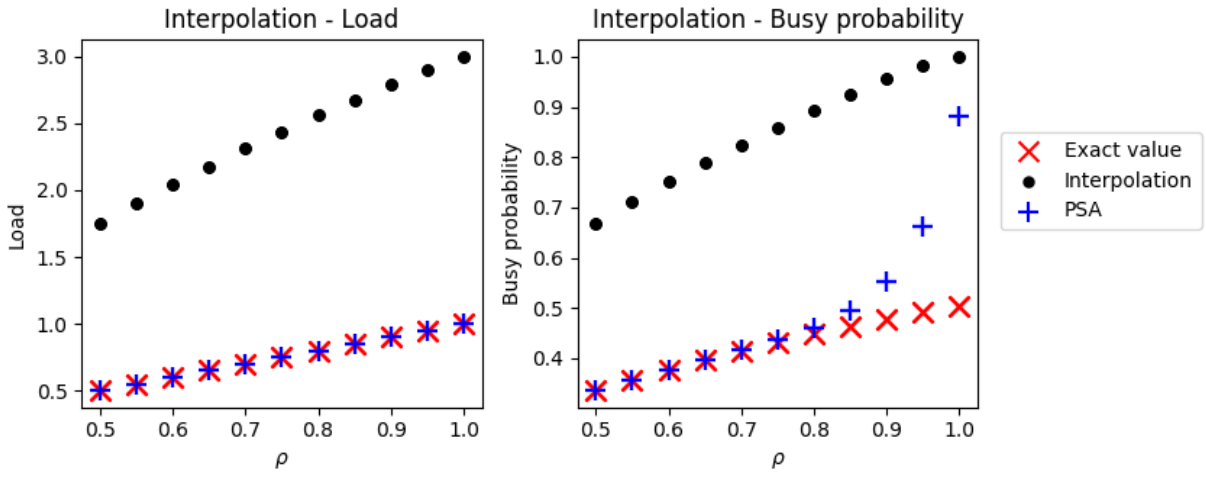


Figure 10: Heavy traffic interpolation

As we can see, the interpolation for the *load* overestimates the exact value, increasingly so as ρ increases, and it never matches the exact value. This is unlike the PSA approximation, which roughly matches the exact value throughout the range of ρ values. Turning our attention to the *busy probability*, we see yet again that the interpolation overestimates the exact value. Similar to the *load*, this overestimation increases as ρ increases. As for the PSA, we see that it roughly matches the exact value for ρ up to 0.8, after which it also starts to overestimate the exact value. However, the overestimation in PSA becomes significant as ρ tends to one, although it still outperforms the interpolation in accuracy.

From the above, we find that linearly interpolating between the PSA and the Heavy traffic limit in [11] does not result in improved performance compared to the PSA alone. In fact, the interpolation performs considerably worse for all values of ρ considered. This is surprising since linear interpolations have shown excellent results in the past, as demonstrated in [21] and [24]. The current study reveals that the simplest version of linear interpolation does not always lead to improved performance and can even result in worsened accuracy.

In the literature, more elaborate interpolation schemes have been used to derive better approximation results, as shown in [19], [20], and [21]. Therefore, it might still be worthwhile to investigate more sophisticated interpolation techniques to address the low-traffic limitation of the PSA and improve its accuracy for heavy traffic situations.

7 Limitations

In this section, we address some of the limitations of this study and propose potential ways to address them.

One limitation is inherent in numerically studying a parameterized stochastic process like the ASIP. The parameter space includes the arrival rate λ, the gate parameters μ_i (for gates $i = 1, \ldots, n$), and the performance measure $g(\vec{X})$, resulting in an infinite-dimensional space. Conducting exhaustive measurements in such a space is not feasible due to finite resources. Therefore, we can only make a limited number of measurements, and careful interpolation and extrapolation are needed to understand the behavior in regions not directly measured.

In this study, we focused on the performance measures *load* and *busy probability*. While Appendix A.4 showed that other performance measures behave similarly to either the *load* or the *busy probability*, we cannot generalize this behavior to all possible performance measures. Hence, the study's findings are specific to the *load* and *busy probability*, and we cannot make definitive claims about the PSA's behavior with arbitrary performance measures.

Another limitation is the computational effort of the PSA and simulation, which can be influenced by their implementation. The current study used Python as the programming language, and certain implementation improvements were applied to enhance computational efficiency. However, implementation improvements can significantly impact their relative computational effort, and further optimization could influence the comparison results. To better capture the difference in relative computational effort, both methods should be implemented with any available improvements, without favoring one method over the other.

Additionally, we did not compare the memory usage between the PSA and simulation. As discussed in Section 5.3, the PSA's memory requirement increases with the number of iterations and the number of sites. For ASIPs with a large number of sites, the PSA's memory usage may exceed available resources, rendering it impractical to use. However, we concluded that the PSA is not worthwhile for ASIPs with fifteen sites or more due to computation time constraints, making further limitations related to memory usage less practically relevant.

Another limitation of the current study pertains to the Heavy-traffic interpolation. We considered only linear interpolation between the PSA and the Heavy-traffic result. However, in other settings, more complicated interpolation schemes have shown to yield better results than linear interpolation, as seen in [19], [21]. Thus, our conclusion is limited to linear interpolation specifically, and other types of interpolation schemes might yield performance results that exceed those of the PSA alone. Further investigation into more elaborate interpolation methods could provide more accurate approximations in the low-traffic regime.

In summary, while the study provides valuable insights into the PSA's behavior for specific performance measures and ASIP configurations, its findings may not generalize to all scenarios. To address the limitations, future research should explore other performance measures, consider further implementation improvements for both methods, and conduct more comprehensive comparisons of computational effort between the PSA and simulation.

8 Conclusion

In this paper, we demonstrated that the PSA is applicable to the $n-$ASIP and can compute the truncated power series of $\mathbb{E}g(\vec{X})$ symbolically. However, computational complexity imposes limitations on the number of sites and terms that can be considered.

Regarding accuracy, we showed that the PSA outperforms a one million iteration simulation for the performance measures *load* and *busy probability* in low-traffic settings, with a several magnitude advantage for the PSA. In high-traffic scenarios, simulation performed better for the *busy probability* in heterogeneous ASIP models but not for the *load*. The PSA consistently outperformed simulation for homogeneous ASIPs.

In terms of computational effort, the PSA outperformed simulation with fifteen or fewer iterations for ASIP models with four or fewer sites. However, simulation became more efficient as we increased the number of PSA iterations

and the number of sites. For ASIPs with more than fifteen sites, simulation was computationally superior to the PSA.

Numerical breakdown was observed in the PSA for values of ρ close to or exceeding one, and the behavior depended on the performance measure and ASIP model. A linear interpolation between the PSA and the ASIP heavy-traffic limit did not improve accuracy compared to the PSA alone.

In conclusion, the PSA is worthwhile using over simulation for ASIP models with a small number of sites in low-traffic settings. The optimal number of PSA iterations and values of ρ for optimal results depend on the performance measure and ASIP model, making general prescriptions challenging.

We recognize several avenues for further research, which we will address hereafter. In the current study, we observed that the optimal number of PSA iterations for achieving the best accuracy depends on the performance measure. Using a fixed number of PSA iterations may not yield optimal results. Therefore, it would be worthwhile to investigate the $\delta - \textit{algorithm}$, which is a modification of the PSA numerical scheme. The $\delta - \textit{algorithm}$ terminates the PSA iterations as soon as the newest term added to the polynomial results in a change less than a tolerance parameter $\delta > 0$, instead of using a fixed number of terms. Understanding how the choice of δ influences accuracy and whether it is influenced by the performance measure could provide insights into achieving better results.

Another avenue for further research is addressing the decline in accuracy observed for larger values of ρ in the PSA, despite the system being stable for such values. One approach to improve this situation is to use a *conformal mapping*, a method previously utilized by Le Blanc for this purpose [18]. By replacing ρ with a new variable through a parameterized bilinear mapping of the interval $[0, 1]$ onto itself, singularities may be removed from the unit disk, potentially increasing the radius of convergence of the power series. In our case, the radius of convergence is not known, making the choice of the parameter heuristic and requiring further research. Implementing this technique could help ameliorate both the model and performance-dependent discrepancies in accuracy.

Additionally, investigating the $\varepsilon - \textit{algorithm}$, another method used by Le Blanc [18], could be beneficial for two reasons. Firstly, like the *conformal mapping*, it aims to increase the radius of convergence of the power series. Secondly, it enhances the rate of convergence. The $\varepsilon - \textit{algorithm}$ is a recursive scheme applied to the truncated power series resulting from the PSA numerical scheme. It converts the initial sequence of polynomials into a sequence of quotients of two polynomials. Given that the PSA becomes less efficient than simulation as the number of sites in the system increases, exploring the $\varepsilon - \textit{algorithm}$ may lead to an increased range of system sizes for which the PSA is more efficient than simulation.

Finally, it would be worthwhile to investigate non-linear interpolation methods between the PSA and the heavy-traffic limit. As linear interpolation may not fully capture the optimal combination of the two, exploring more sophisticated interpolation schemes could provide better approximations and extend the PSA's applicability to a wider range of ASIP models.

References

[1] H. Sachdeva, M. Barma, and M. Rao, "Condensation and intermittency in an open-boundary aggregation-fragmentation model," en, *Phys Rev Lett*, vol. 110, no. 15, p. 150 601, Apr. 2013.

[2] H. Sachdeva, "Aggregation-Fragmentation Models for Transport in a Biological System," PhD thesis, Tata Institute of Fundamental Research, Mumbai, Aug. 2014.

[3] V. Kumar, A. Pal, and O. Shpielberg, *Arrhenius law for interacting diffusive systems*, 2023. arXiv: 2306.06879 [cond-mat.stat-mech].

[4] O. Shpielberg and A. Pal, "Thermodynamic uncertainty relations for many-body systems with fast jump rates and large occupancies," *Physical Review E*, vol. 104, no. 6, Dec. 2021. DOI: 10.1103/physreve.104.064141. [Online]. Available: https://doi.org/10.1103%2Fphysreve.104.064141.

[5] L. Garbe, Y. Minoguchi, J. Huber, and P. Rabl, *The bosonic skin effect: Boundary condensation in asymmetric transport*, 2023. arXiv: 2301.11339 [quant-ph].

[6] A. Aggarwal, *Dynamical stochastic higher spin vertex models*, 2017. arXiv: 1704.02499 [math-ph].

[7] H. C. Steinacker, "On the quantum structure of space-time, gravity, and higher spin in matrix models," *Classical and Quantum Gravity*, vol. 37, no. 11, p. 113 001, May 2020. DOI: 10.1088/1361-6382/ab857f. [Online]. Available: https://dx.doi.org/10.1088/1361-6382/ab857f.

[8] S. Reuveni, I. Eliazar, and U. Yechiali, "Asymmetric inclusion process," eng, *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 84, no. 4 Pt 1, p. 041 101, Oct. 2011, ISSN: 1550-2376. DOI: 10.1103/PhysRevE.84.041101.

[9] Y. Yeger and U. Yechiali, "Performance measures in a generalized asymmetric simple inclusion process," *Mathematics*, vol. 10, no. 4, 2022, ISSN: 2227-7390. DOI: 10.3390/math10040594. [Online]. Available: https://www.mdpi.com/2227-7390/10/4/594.

[10] S. Reuveni, I. Eliazar, and U. Yechiali, "Asymmetric inclusion process as a showcase of complexity," *Phys. Rev. Lett.*, vol. 109, p. 020 603, 2 Jul. 2012. DOI: 10.1103/PhysRevLett.109.020603. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.109.020603.

[11] S. Reuveni, I. Eliazar, and U. Yechiali, "Limit laws for the asymmetric inclusion process," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 86, p. 061 133, Dec. 2012. DOI: 10.1103/PhysRevE.86.061133.

[12] U. Yechiali and Y. Yeger, "Matrix approach for analyzing n-site generalized asip systems: Pgf and site occupancy probabilities," *Mathematics*, vol. 10, no. 23, 2022, ISSN: 2227-7390. DOI: 10.3390/math10234624. [Online]. Available: https://www.mdpi.com/2227-7390/10/23/4624.

[13] G. Hooghiemstra, M. Keane, and S. Van De Ree, "Power series for stationary distributions of coupled processor models," *SIAM Journal on Applied Mathematics*, vol. 48, no. 5, pp. 1159–1166, 1988. DOI: 10.1137/0148069. eprint: https://doi.org/10.1137/0148069. [Online]. Available: https://doi.org/10.1137/0148069.

[14] J. Blanc, *Cyclic polling systems: Limited service versus Bernoulli schedules* (Research memorandum / Tilburg University, Department of Economics), English. Unknown Publisher, 1990, vol. FEW 422, Pagination: 28, v.

[15] J. Blanc, "An algorithmic solution of polling models with limited service disciplines," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1152–1155, 1992. DOI: 10.1109/26.153357.

[16] J. Blanc, "On a numerical method for calculating state probabilities for queueing systems with more than one waiting line," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 119–125, 1987, ISSN: 0377-0427. DOI: https://doi.org/10.1016/0377-0427(87)90129-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042787901294.

[17] J. Blanc, "A numerical approach to cyclic-service queueing models," English, *Queueing Systems: Theory and applications*, vol. 6, no. 1, pp. 173–188, 1990, ISSN: 0257-0130.

[18] J. Blanc, "Performance analysis and optimization with the power-series algorithm," English, in *Performance evaluation of computer and communication systems*, L. Donatiello and R. Nelson, Eds., ser. Lecture notes in computer science, Pagination: 28, Springer, 1993, pp. 53–80, ISBN: 354057297X.

[19] T. Kimura, "Approximations for multi-server queues: System interpolations," *Queueing Systems*, vol. 17, no. 3, pp. 347–382, Sep. 1994, ISSN: 1572-9443. DOI: 10.1007/BF01158699. [Online]. Available: https://doi.org/10.1007/BF01158699.

[20] M. I. Reiman and B. Simon, "An interpolation approximation for queueing systems with poisson input," *Operations Research*, vol. 36, no. 3, pp. 454–469, 1988, ISSN: 0030364X, 15265463. [Online]. Available: http://www.jstor.org/stable/170988 (visited on 05/01/2023).

[21] W. Whitt, "An interpolation approximation for the mean workload in a gi/g/1 queue," *Operations Research*, vol. 37, no. 6, pp. 936–952, 1989, ISSN: 0030364X, 15265463. [Online]. Available: http://www.jstor.org/stable/171475 (visited on 05/01/2023).

[22] J. D. C. Little, "A proof for the queuing formula: $L = \lambda W$," *Operations Research*, vol. 9, no. 3, pp. 383–387, Jun. 1961. DOI: 10.1287/opre.9.3.383. [Online]. Available: https://doi.org/10.1287/opre.9.3.383.

[23] R. W. Wolff, "Poisson arrivals see time averages," *Oper. Res.*, vol. 30, pp. 223–231, 1982.

[24] J. L. Dorsman, R. D. v. d. Mei, and M. Vlasiou, "Analysis of a two-layered network by means of the power-series algorithm," *Performance Evaluation*, vol. 70, no. 12, pp. 1072–1089, 2013, ISSN: 0166-5316. DOI: https://doi.org/10.1016/j.peva.2013.09.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166531613001120.

[25] D. Fiems and T. Phung-Duc, "Light-traffic analysis of random access systems without collisions," *Annals of Operations Research*, vol. 277, no. 2, pp. 311–327, Jun. 2019, ISSN: 1572-9338. DOI: 10.1007/s10479-017-2636-7. [Online]. Available: https://doi.org/10.1007/s10479-017-2636-7.

[26] G. Koole, "On the power series algorithm," *Report Department of Operations Research Statistics and System Theory*, pp. 1–16, 1994.

A Appendix

A.1 Notation

In this paper we define to the following notations

ε : a small positive real number close to zero, less than one

ρ : the traffic intensity of the system

$\vec{X}$: an n-dimensional vector of the site occupancies, that is each component represents the number of particles present at that particular site

X_j : the j-th component of $\vec{X}$, representing the number of particles present at the j-th site

$|\vec{X}|$: the total occupancy, where $|\vec{X}| = \sum_{j=1}^{n} X_j$,i.e. the total number of particles present in the system

(x, y) : a 2-vector with components x and y

$\rho a_j(\vec{X})$: the arrival rate of particles at site j

$d_j(\vec{X})$: the departure rate of particles at site j

$\mathbb{1}_{\{A\}}$: indicator function, which evaluates to 1 if Boolean statement A holds true, otherwise evaluates to 0

$\hat{e}_j$: the j-th unit vector

$\mathbb{N}$: the natural numbers including zero

$x \prec y$: indicates that x precedes y, in our case this indicates that y depends on x in a recursive manner

$\wedge$: logical 'and'

$o(\rho^N)$: we consider $f(\rho) = o(\rho^N)$ to be equivalent to $\lim_{\rho \to 0} \frac{f(\rho)}{\rho^N} = 0$, this can be intuitively understood as $f(\rho)$ tends to zero more quickly than ρ^N as ρ tends to zero

$\mathcal{O}(\rho^N)$: we consider$f(\rho) = \mathcal{O}(\rho^N)$ to be equivalent to $\limsup_{\rho \to \infty} \frac{f(\rho)}{\rho^N} < \infty$, this can be intuitively understood as $|f(\rho)|$ is bounded above asymptotically by ρ^N up to some constant factor

A.2 Accuracy

In Section 5.2, we made the following Remark 5.2. In the following we see that the accuracy for the Decline, Incline, and Convex models are qualitatively similar for both the busy probability and load, see Figure 11 and Figure 12, respectively.

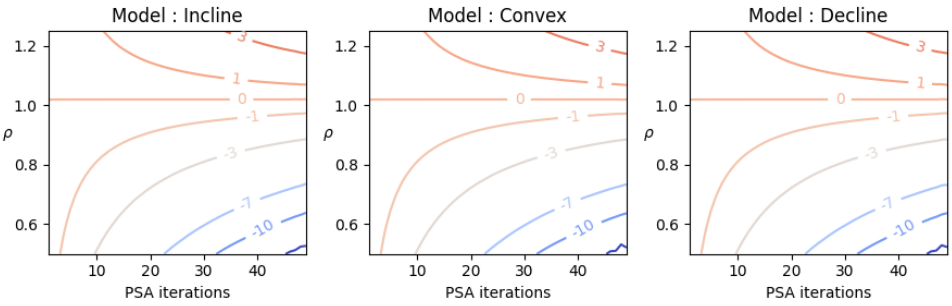


Figure 11: Busy probability - accuracy - Heterogeneous models

A.3 Computation Complexity

In Section 5.3, we made the following Remark 5.3. In the following we show that neither the performance measure nor the model significantly influence the $\log_{10}$ computation time ratio of the simulation time to the PSA computation time. In Figure 13 it can be seen that for this ratio there is little to no discernible difference between the busy probability and the load. Similarly, it can be seen in Figure 14 that the various models yields essentially the same result.

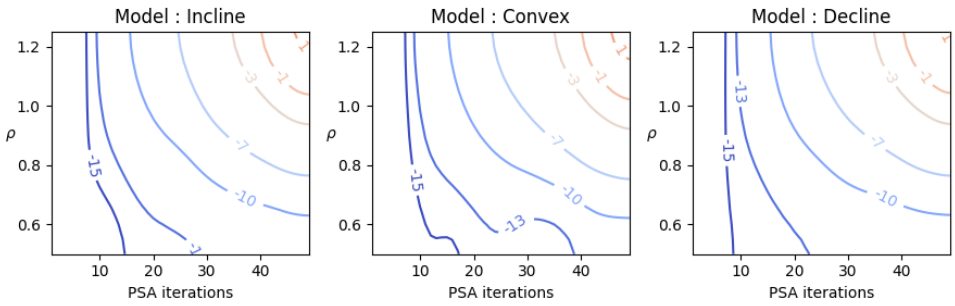


Figure 12: Load - accuracy - Heterogeneous models

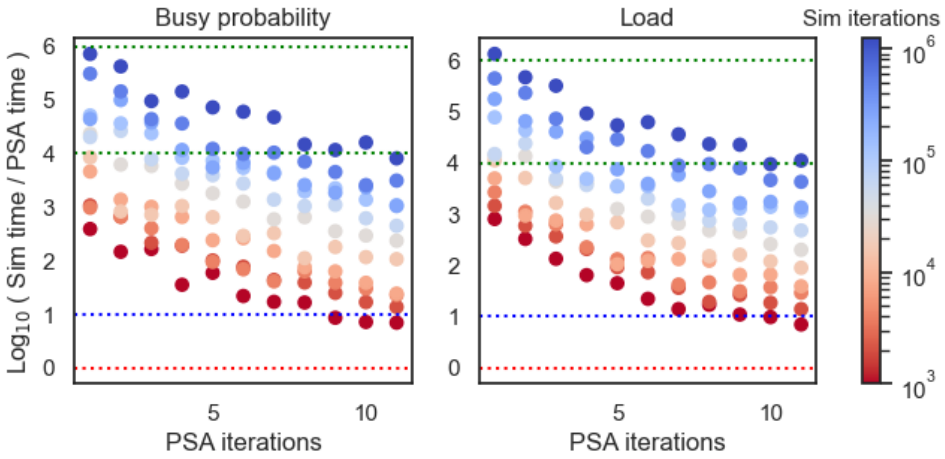


Figure 13: The $\log_{10}$ computation time ratio of Sim to PSA for both the busy probability and load

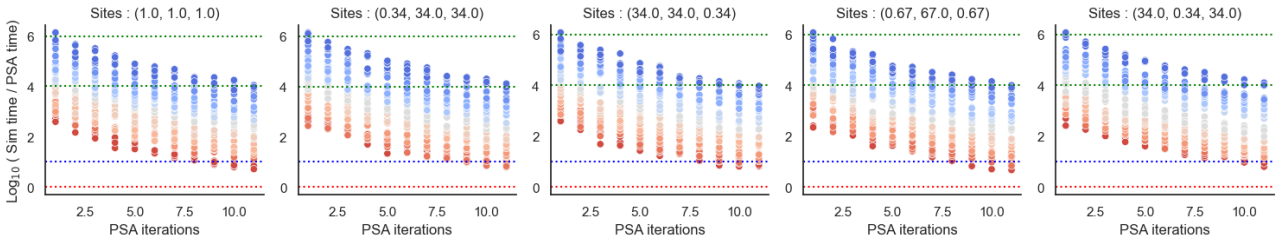


Figure 14: The $\log_{10}$ computation time ratio of Sim to PSA for the various models

A.4 Performance-measure-dependent Numerical breakdown

In Section 5.4.1, we made the following Remark 5.5. In the following we show that indeed several other performance measures behave similar to either the busy probability or the load. To this end we consider the PSA approximation of the busy probability, empty probability, load, load^2 (second moment of the load), and X_1 (load of the first site) for the Decline model at ten PSA iterations, see Table 6. It can be seen that indeed the numerical breakdown for the busy probability and the empty probability is very similar. Moreover, the load^2 and X_1 show behavior similar to the load. There is a clear distinction between these groups in the λ that results in observable numerical breakdown, and the rate at which the approximation explodes.

A.5 Numerical breakdown - Heterogeneous

In the following see that the divergence behavior for the Decline, Incline, and Convex models are qualitatively similar. Meaning that the same performance-measure-dependent dichotomy in divergence behavior is present, see Figure 15 and Figure 16.

Table 6: Numerical breakdown - similarities between performance measure

λ	busy probability	empty probability	load	load^2	X_1
.00001	0	1	0	0	0
0.0010	0	1	0	0	0
0.1000	-161830998	161830999	7	98	7
0.5000	-1763613412932104	1763613412932104	-333	1155	-1326
0.9000	-637910234339705600	637910234339705600	170361	577700	-264479
0.9800	-1496835117981491456	1496835117981491456	399136	1346901	-620057
0.9990	-1814228633580618240	1814228633580618240	-370085	-1149870	-1380908
1.0000	-1832498078001921280	1832498078001921280	-373805	-1161479	-1394796

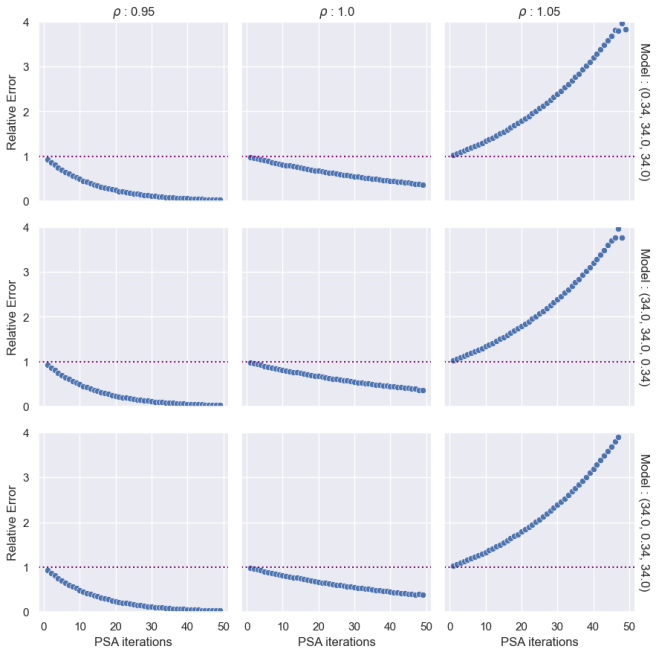


Figure 15: Busy probability - divergence - Heterogeneous models

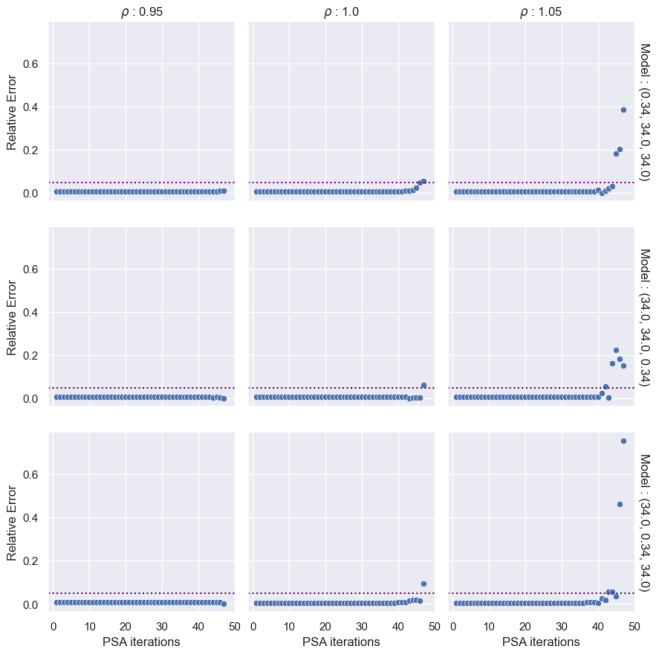


Figure 16: Load - divergence - Heterogeneous models

ASIP site occupancy PGF using two moment approximation

Wesley Geelen

June 2023

Abstract

The asymmetric simple inclusion process (ASIP) is an $n-$site tandem stochastic network, used to model unidirectional clustered particle flow. At the end of each site in the system, there is a gate that regulates the flow of particles by stochastically opening and closing. When particles are present in a given site, they aggregate together, forming a cluster. Upon gate opening, this cluster moves as a single unit to the subsequent site, combining with the cluster in that site (if any). For the general ASIP a closed-form expression for the *load* in terms of the first two moments of the inter-arrival time and inter-gate opening time distributions is not known. Here, we show that approximating the steady-state *load* of a general ASIP by fitting a phase-type ASIP using 2-moment approximations yields closed-form approximations. We derive a recursive equation for the steady-state site occupancy PGF of the phase-type ASIP. Furthermore, we derive a closed-form expression for the steady-state *load* PGF of the phase-type ASIP. A 2-moment approximation is used to derive a closed-form approximation of the expected *load* by fitting the phase-type ASIP. The dependence of the expected *load* on the first two moments of the inter-arrival and inter-gate opening time distribution is investigated. We find that a reduced mean and increased variance of the inter-arrival time distribution increase the *load*, and a reduced mean and increased variance of the inter-gate opening time distribution reduce the *load*. Our result show that the inter-arrival and inter-gate opening distributions have opposite effects on the expected *load*. Additionally, our results indicate that systems with more unreliable servers (increased inter-gate opening time variance) exhibit superior expected performance, assuming that the servers' expected performance is equivalent. We anticipate that our results will encourage the use of more sophisticated fitting methods to fit the phase-type ASIP to the general ASIP. Additionally, we encourage the investigation of higher moments of the expected *load* to better understand the general ASIP's performance characteristics.

1 Introduction

The $n-$site Asymmetric Inclusion Process (ASIP) is a specialized case of a tandem stochastic system with n queues (sites). Here 'Asymmetric' refers to the property that customers (particles) move uni-directionally along the system. While 'Inclusion' refers to the inclusion principle. Which is the property that both site capacity and service (gate) capacity are unlimited. Stated differently, a given site allows any number of particles to be present which together form a cluster, upon gate opening all particles present move simultaneously (as a cluster) and instantaneously to the next site or out of the system. In the subsequent site the arriving cluster of particles combines with the particles already present (if any) to form a new cluster. If both the inter-arrival time distribution and inter-gate opening time distributions are exponential, then the system is referred to as the Classical ASIP. If the inter-gate opening time distributions is shared by all gates then the system is referred to as an homogeneous ASIP, otherwise it is referred to as an heterogeneous ASIP.

The service paradigm employed by the ASIP makes it particularly suitable for studying particle systems. It has been investigated for various applications in the literature. One of these applications is using the ASIP as an Aggregation-Fragmentation model for transport in biological systems (see [1], [2]). Another application is found in the study of the thermal dependence of chemical reaction rates. In [3], an Arrhenius law for the ASIP is derived, which captures the activation time of a system from a meta-stable state. Moreover, in [4], the ASIP is used as a thermal engine, for which a bound for the entropy production rate is derived. The ASIP has also been used to model the transport of bosonic particles in heat transfer [5]. Furthermore, the ASIP has been studied in quantum physics as a dynamical stochastic higher spin vector model (see [6]). Such models are at the center-point of the study of quantum gravity [7].

The ASIP has been extensively studied in the past (see [8]–[12]), resulting in closed-form expressions for several steady-state performance measures. Specifically, closed-form expressions for the steady-state load (total number of particles in the system) are readily available for various variations of the ASIP. These include the classical heterogeneous ASIP of arbitrary size (see [8]), a generalized renewal ASIP of arbitrary size (see [12]), and the PGF of the normalized load of a classical n-site ASIP under heavy-traffic (see [11]). However, for the general ASIP with real positive distributed inter-arrival times and inter-gate opening times, no such closed-form expression has been derived. Existing techniques used to derive such expressions rely on exploiting the Markovian dynamics of the process, which the general ASIP

does not admit. Therefore, these methods are not directly applicable to the general ASIP, leaving the dependence of the steady-state load on the moments of the inter-arrival and inter-gate opening time distributions under-explored. Specifically, it remains unexplored how the steady-state load of this general ASIP changes as the coefficient of variation of the inter-arrival distribution and/or inter-gate opening time distribution changes. Additionally, the effect of changing the first two moments of these distributions on the load, both in terms of expectation and distribution, has not been studied yet.

In this paper, we utilize a phase-type ASIP as an approximation to the general ASIP. By fitting its phase-type distributions to the distributions of the general ASIP using a 2-moment approximation, we aim to derive a closed-form expression for the steady-state *load* of the phase-type ASIP. To achieve this, we first derive the Markovian representation for the phase-type ASIP. The Markovian representation is then utilized to derive a recursive equation for the steady-state site occupancies PGF of the n-site phase-type ASIP. This result is subsequently used to obtain a closed-form approximation of the PGF of the steady-state *load* of the first k sites of the n-site phase-type ASIP. Using the 2-moment approximation, we fit the parameters of the phase-type ASIP to approximate the general ASIP, and then derive a closed-form approximation for the steady-state *load* in terms of its distributional parameters. We perform numerical studies by varying the first two moments of both the inter-arrival time distribution and the inter-gate opening time distribution. Our findings reveal that decreasing the first moment of the inter-arrival time distribution increases the *load*, while increasing its second moment leads to an increase in the *load*. Conversely, decreasing the first moment of the inter-gate opening time distribution decreases the *load*, and increasing its second moment decreases the *load*. The first moments of both distributions have a secondary effect, influencing the *load*'s sensitivity to changes in the other distribution, for the inter-arrival time distribution and inter-gate opening time distribution, respectively. Additionally, we examine the distribution of the expected *load* of the general ASIP using Monte Carlo simulation for Uniform, Exponential, and Pareto distributions with coefficients of variation less than one, equal to one, and larger than one, respectively. Our findings show that the Pareto distribution ASIP exhibits extreme right-tailed behavior, while the other distributions have a kurtosis less than a normal distribution (i.e., less than 3). Consequently, we conclude that the inter-arrival time distribution and inter-gate opening time distribution have somewhat opposite effects on the *load*. Furthermore, we observe that the right-tailed behavior of these distributions is also present in the distribution of the *load*.

A phase-type distribution is a probability distribution that can be represented as a convolution or mixture of exponential distributions [13]. This distribution can be associated with the time until absorption of a corresponding Markov chain into its singular absorbing state. The states in the Markov chain represent different phases of the distribution, and the completion of each phase corresponds to the passage of an exponentially distributed period of time. In this paper, we focus on two specific phase-type distributions: the hyper-exponential and mixed-Erlang distributions. These distributions can be used to approximate the distribution of a non-negative random variable for which the first two moments are known, using a 2-moment approximation [13]. Utilizing the Markov chain representation of these distributions, we can analyze the Markovian dynamics of a phase-type ASIP. Here, a phase-type ASIP refers to an ASIP in which the inter-arrival time distribution and inter-gate opening time distribution are hyper-exponential or mixed-Erlang distributed. By exploiting the Markovian dynamics, we derive recursive equations for the steady-state site occupancy PGF.

Moment approximations are techniques used to approximate real positive distributions with known first m moments using phase-type distributions fitted based on the first $1 \leq k \leq m$ moments. These methods have been applied to approximate inter-arrival time distributions and service-time distributions of queuing models, such as GI/M/1 (see [14] and references) and GI/G/1 (see [15] and references) models, using PH/M/1 and PH/G/1 models. These approximations enable the estimation of steady-state queue length, waiting time, and other performance measures. In this paper, we employ 2-moment approximations (see [13]) to approximate real positive inter-arrival time distributions and inter-gate opening time distributions with known first two moments. We fit phase-type distributions based on these first two moments. Specifically, we utilize this approach to approximate the steady-state *load* of the general n-site homogeneous ASIP, which features real positive inter-arrival time distributions and inter-gate opening time distributions. We derive a closed-form approximation for the steady-state *load* PGF and obtain the expected steady-state *load* from the PGF. Additionally, we investigate the expected *load*'s dependence on the first two moments of both the inter-arrival and inter-gate opening time distributions.

Our results demonstrate that the phase-type ASIP admits a closed-form expression for the steady-state *load* PGF. This discovery opens up new possibilities for approximating the general ASIP by utilizing this phase-type ASIP through various approximation methods. In particular, we demonstrate that a 2-moment approximation allows for such an approximation. Additionally, our analysis reveals that the *load* can be influenced by changes in the distributions. Specifically, we find that the inter-arrival time and inter-gate opening time distributions have somewhat opposite effects on the *load*. Furthermore, when comparing two ASIPs with the same first moment for the inter-gate opening time distribution, we observe that the one with the larger second moment results in a smaller *load*. In other words, a system with more inconsistent or unreliable (higher variance) servers (gates) performs better, on average. This suggests that systems adequately modeled using the general ASIP should consider using unreliable servers instead of

reliable ones, if expected performance is of paramount importance.

We expect that our findings will stimulate the use of approximating the general ASIP using the phase-type ASIP. While our current paper focuses on the 2-moment approximation, it serves as a stepping stone for future studies comparing various approximation methods. Methods that warrant further exploration include different moment approximations, Kullback-Leibler Divergence minimization using maximum likelihood estimation, and hybrid methods. Moreover, our results highlight the importance of investigating the dependence of higher moments of the *load* on the first two moments of both the inter-arrival time distribution and the inter-gate opening time distribution. Although we observed that systems with more unreliable servers show improved expected performance, it doesn't necessarily mean that such a system is superior overall. The variance of the *load* for the more unreliable server system could potentially offset its superior mean performance, making a comprehensive analysis of higher moments crucial in understanding the general ASIP's performance characteristics.

This paper is organized as follows. In Section 2, we provide the model description for the ASIP. In Section 3, we first provide some preliminary results pertaining to the two moment fitting method. In section 4 we provide the main results of the paper. In Section 5, we use numerics and simulation to study the dependence of the approximated-*load* and the *load* CDF on the coefficient of variation. In Section 6, we provide the main conclusion of our study. In Appendix A we provide the proofs for the results stated in the main text, but where omitted for brevity's sake. Lastly, in Appendix B we provide the parameter values considered for each of the cases investigated by numerics.

2 Model description

The model considered in this paper is an $n-$site ASIP. It consist of n unlimited capacity sites and n gates. Each site is followed by a gate, the resulting site-gate pairs are arranged in tandem fashion. Meaning that gate k is located at the end of site k, and before site $k+1$ (if $k<n$). The flow of particles through the system is illustrated in Figure 17. The particles arrive only at the first site, all particles present at a given site form a cluster. Upon opening of gate k any particles present in site k move instantaneously and simultaneously as a cluster to site $k+1$, where it forms a new cluster with the particles present (if any). If gate n opens the particles present in site n leave the system. We consider the general setting. That is, particles arrive according to a general positive real-valued distribution F_A. The inter-gate opening times are independent of other gates. The inter-gate opening time of gate j is distributed according a general positive real-valued distribution F_{B_j}.

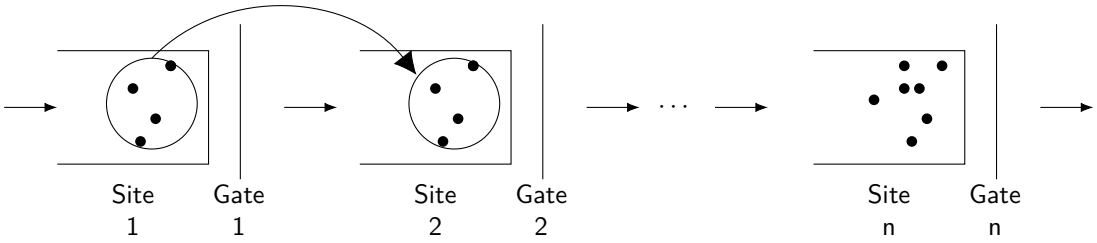


Figure 17: The heterogeneous n-site ASIP system

3 Preliminaries

In this paper we the approximate inter-arrival time and inter-gate opening time distributions by either hyper-exponential or mixed-Erlang distributions. Therefore, we first define these distributions. Subsequently, we state the result (2-moment approximation) that approximates a given distribution by one of these two distributions, based on its fist two moments.

Definition 3.1 (Hyper-exponential random variable)**.** *A random variable H is said to be Hyper-exponentially distributed if with probability p_k it is exponentially distributed with mean $\frac{1}{\mu_k}$, where $k=1,\ldots,m$ and $\sum_{k=1}^{m} p_k = 1$. In particular, we define the random variable $H_m(p_1,\ldots,p_m,\mu_1,\ldots,\mu_m)$ to be the random variable with the following probability distribution*

$$H(x) = \sum_{k=1}^{m} p_k(1-e^{-\mu_k x}), \qquad x \geq 0, \quad 1 \geq p_k \geq 0, \quad \sum_{k=1}^{m} p_k = 1.$$

Definition 3.2 (mixed-Erlang random variable)**.** *A random variable E is said to be mixed-Erlang distributed with parameters p,μ,m, if with probability p it is Erlang distributed with parameters $\mu, m-1$ and with probability $1-p$ it*

is Erlang distributed with parameters μ, m. In particular, we define the random variable $E_m(p,\mu)$ to be the random variable with the following probability distribution

$$E(x) = p\left(1 - e^{-\mu x}\sum_{k=0}^{m-2}\frac{(\mu x)^j}{j!}\right) + (1-p)\left(1 - e^{-\mu x}\sum_{k=0}^{m-1}\frac{(\mu x)^j}{j!}\right), \qquad x \geq 0, \quad 1 \geq p \geq 0.$$

Next, we present the two-moment approximation result. Suppose X is a positive random variable with distribution F_X, for which only the first two moments are known. In such cases, we can approximate F_X by an phase-type distribution, which is fitted based on these first two moments.

Lemma 3.3 (Distribution approximation using two moments)**.** *Let X be a positive random variable with mean $\mathbb{E}[X]$ and coefficient of variation c_X. We approximate X based on c_X as follows [13].*

- $\diamond$ $(c_X \leq 1)$*: In this case we fit the mixed-Erlang random variable $E_m(p,\mu)$ so that the mean and squared coefficient of variation match those of random variable X. To this end, take m such that $1/m \leq c_X^2 \leq 1/(m-1)$ for some $m = 2, 3, \ldots$, furthermore p and μ are chosen as follows*

$$p = \left(mc_X^2 - \sqrt{m(1+c_X^2) - m^2c_X^2}\right)/(1+c_X^2), \qquad \mu = \frac{m-p}{\mathbb{E}[X]}.$$

- $\diamond$ $(c_X > 1)$ *: In this case we fit the Hyper-exponential random variable $H_m(p_1,\ldots,p_m,\mu_1,\ldots,\mu_m)$ so that the mean and squared coefficient of variation match those of random variable X. To this end, we take $m = 2$, and furthermore parameters p_1, p_2, μ_1, μ_2 are chosen follows*

$$p_1 = \frac{1}{2}\left(1 + \sqrt{\frac{c_X^2-1}{c_X^2+1}}\right), \qquad p_2 = 1 - p_1, \qquad \text{and} \qquad \mu_1 = \frac{2p_1}{\mathbb{E}[X]}, \qquad \mu_2 = \frac{2p_2}{\mathbb{E}[X]}.$$

4 Main results

In this section we present the main results of this paper, these being a recursive PGF for the site occupancy and exact PGF for the *load* of the phase-type $n-$site ASIP. Where phase-type $n-$site ASIP is defined as follows.

Definition 4.1 (Phase-type $n-$site ASIP)**.** *Consider the $n-$site general-distribution ASIP but instead limit the inter-arrival time to either the mixed-Erlang $E_{m_0}(p_0,\lambda)$ or the hyper-exponential $H_{m_0}(p_{01},\ldots,p_{0;m_0},\lambda_1,\ldots,\lambda_{m_0})$ random variable. Furthermore, limit the inter-gate opening time of gate i to either the mixed-Erlang $E_{m_i}(p_i,\mu_i)$ or the hyper-exponential $H_{m_i}(p_{i1},\ldots,p_{i;m_i},\mu_i,\ldots,\mu_{m_i})$ random variable, where $i = 1,\ldots,n$. Lastly, we impose the limitation that all gates must be of the same type of distribution, but allow for difference in parameters.*

The above definition gives rise to four cases, namely

- $\diamond$ E_{m_0}/E_{m_i} : The case with mixed-Erlang arrivals and inter-gate opening times.
- $\diamond$ E_{m_0}/H_{m_i} : The case with mixed-Erlang arrivals and hyper-exponential inter-gate opening times.
- $\diamond$ H_{m_0}/H_{m_i}: The case with hyper-exponential arrivals and inter-gate opening times.
- $\diamond$ H_{m_0}/E_{m_i}: The case with hyper-exponential arrivals and mixed-Erlang inter-gate opening times.

In the current paper we are primarily interested in the site occupancy and the *load* of the Phase-type $n-$site ASIP. The continuous time random process which captures its site occupancy can be defined as follows $\{\vec{X}(t) = (X_1(t),\ldots,X_n(t)) : X_i \in \mathbb{N}_{\geq 0},\ t \geq 0\}$, where $X_i(t)$ represents the process that captures the number of particles present in site i at time t.

However, we consider this definition inadequate for the phase-type $n-$site ASIP. This is explained as follows. Notice that the mixed-Erlang random variable consists of multiple exponential phases. Meaning that for a Markovian representation we are required to include the phase. The aforementioned definition does not account for these phases. The state dynamics are dependent on these phases, therefore it is non-Markovian. Since we desire to exploit the Markovian dynamics of the process, we consider the definition inadequate.

The phase-type $n-$site ASIP includes a different number of mixed-Erlang random variables depending on the case. In order, to capture the phases of each of the included mixed-Erlang random variables we must make a case-based definition. To this end, give the following definition for the process that captures the phase-type $n-$site ASIP.

Definition 4.2 (Phase-type $n-$site ASIP Process)**.** *The continuous time vector valued random process is composed of two other continuous time vector valued random processes. Namely, $\vec{X}(t)$ which captures the site occupancy of*

the system, and $\vec{S}(t)$ which captures the phases of the arrival and or inter-gate opening time processes. This leads to the following definition $\vec{Y}(t) = (\vec{S}(t), \vec{X}(t))$. In turn $X(t)$ is defined as $\{\vec{X}(t) = (X_1(t), \ldots, X_n(t)) : X_i \in \mathbb{N}_{\geq 0},\ t \geq 0\}$. The Hyper-exponential distribution does not have multiple exponential phases, while the mixed-Erlang does. Therefore, $Y(t)$ is defined per case

$$\vec{Y}(t) = \begin{cases} (S_0(t), S_1(t), \ldots, S_n(t), \vec{X}(t)) & \text{if } E_{m_0}/E_{m_i}, \\ (S_1(t), \ldots, S_n(t), \vec{X}(t)) & \text{if } H_{m_0}/E_{m_i}, \\ (S_0(t), \vec{X}(t)) & \text{if } E_{m_0}/H_{m_i}, \\ \vec{X}(t) & \text{if } H_{m_0}/H_{m_i}, \end{cases}$$

where $t \geq 0, S_0(t) \in \{1, \ldots, m_A\}$, and $S_i(t) \in \{1, \ldots, m_B\}$, with $i = 1, \ldots, n$.

In the following theorem we give the recursive definition of the phase-type n-site ASIP site occupancy PGF

Theorem 4.3 (Site occupancy PGF of the phase-type ASIP)**.** *Let $G_X(z_1, \ldots, z_n)$ denote the steady-state site occupancy PGF, then by manner of definition $G_X(z_1, \ldots, z_n) = \mathbb{E}\left[z_1^{X_n} \cdots z_n^{X_n}\right]$. Next, we give the recursive definition of G_X on a case by case basis.*

◇ *Case (E_{m_0}/E_{m_i}) :*

$$G_X(z_1, \ldots, z_n)\left[\lambda(1 - \mathbb{P}(S_0 \neq m_0)) + \sum_{k=1}^{n} \mu_k(1 - \mathbb{P}(S_k \neq m_k))\right] = \lambda z_1 \mathbb{P}(S_0 = m_0) G_X(z_1, \ldots, z_n)$$
$$+ \mu_1 \mathbb{P}(S_1 = m_1) G_X(z_2, z_2, z_3, \ldots, z_n) + \cdots + \mu_{n-1} \mathbb{P}(S_{n-1} = m_{n-1}) G_X(z_1, \ldots, z_{n-2}, z_n, z_n)$$
$$+ \mu_n \mathbb{P}(S_n = m_n) G_X(z_1, \ldots, z_{n-1}, 1).$$

◇ *Case (E_{m_0}/H_{m_i}) :*

$$G_X(z_1, \ldots, z_n)\left[\lambda(1 - \mathbb{P}(S_0 \neq m_0)) + \sum_{i=1}^{n}\sum_{j=1}^{m_i} p_{ij}\mu_{ij}\right] = \lambda z_1 \mathbb{P}(S_0 = m_0) G_X(z_1, \ldots, z_n)$$
$$+ \sum_{j=1}^{m_1} p_{1j}\mu_{1j} G_X(z_2, z_2, z_3, \ldots, z_n) + \cdots + \sum_{j=1}^{m_{n-1}} p_{n-1;j}\mu_{n-1;j} G_X(z_1, \ldots, z_{n-2}, z_n, z_n)$$
$$+ \sum_{j=1}^{m_n} p_{nj}\mu_{nj} G_X(z1, \ldots, z_{n-1}, 1).$$

◇ *Case (H_{m_0}/E_{m_i}) :*

$$G_X(z_1, \ldots, z_n)\left[(1 - z_1)\sum_{j=1}^{m_0} p_{0j}\lambda_j + \sum_{i=1}^{n} \mu_i(1 - \mathbb{P}(S_i \neq m_i))\right] = \mu_1 \mathbb{P}(S_1 = m_1) G_X(z_2, z_2, z_3, \ldots, z_n)$$
$$+ \cdots + \mu_{n-1} \mathbb{P}(S_{n-1} = m_{n-1}) G_X(z_1, \ldots, z_{n-2}, z_n, z_n) + \mu_n \mathbb{P}(S_n = m_n) G_X(z_1, \ldots, z_{n-1}, 1).$$

◇ *Case (H_{m_0}/H_{m_i}) :*

$$G_X(z_1, \ldots, z_n)\left[(1 - z_1)\sum_{j=1}^{m_0} p_{0j}\lambda_j + \sum_{i=1}^{n}\sum_{j=1}^{m_i} p_{ij}\mu_{ij}\right] = \sum_{j=1}^{m_1} p_{1j}\mu_{1j} G_X(z_2, z_2, z_3, \ldots, z_n)$$
$$+ \cdots + \sum_{j=1}^{m_{n-1}} p_{n-1;j}\mu_{n-1;j} G_X(z_1, \ldots, z_{n-2}, z_n, z_n) + \sum_{j=1}^{m_n} p_{nj}\mu_{nj} G_X(z_1, \ldots, z_{n-1}, 1).$$

Where $\mathbb{P}(S_i = m_i)$ refers to the steady-state probability that the phase S_i of the Markov chain corresponding to mixed-Erlang $E_{m_i}(p_i, \mu_i)$ is m_i, it is given by

$$\mathbb{P}(S_i = m_i) = \frac{1}{m_i - p_i}.$$

find distribution function from the pgf

Proof in Appendix A.2.

The previous result can be used to establish an explicit form for the PGF of the *load* in the first k sites, which is given in the following theorem.

Theorem 4.4 (PGF for the *load* of the first k sites of the phase-type ASIP)**.** *Let $X_{(k)}$ denote the load of the first k sites in the system, that is $X_{(k)} = \sum_{i=1}^{k} X_i$. This leads to PGF $G_{X_{(k)}}(z)$ for all cases of the phase-type ASIP we have that*

$$G_{X_{(k)}}(z) = \prod_{i=1}^{k} \frac{q_i}{1-(1-q_i)z},$$

where q_i represents a probability which depends on the case of the phase-type ASIP. We enumerate the cases

- ◇ *Case (E_{m_0}/E_{m_i}) :*

$$q_i = \frac{\mu_i \mathbb{P}(S_i = m_i)}{\lambda \mathbb{P}(S_0 = m_0) + \mu_i \mathbb{P}(S_i = m_i)}.$$

- ◇ *Case (E_{m_0}/H_{m_i}) :*

$$q_i = \frac{\sum_{j=1}^{m_i} p_{ij}\mu_{ij}}{\lambda \mathbb{P}(S_0 = m_0) + \sum_{j=1}^{m_i} p_{ij}\mu_{ij}}.$$

- ◇ *Case (H_{m_0}/E_{m_i}) :*

$$q_i = \frac{\mu_i \mathbb{P}(S_i = m_i)}{\sum_{j=1} p_{0j}\lambda_j + \mu_i \mathbb{P}(S_i = m_i)}.$$

- ◇ *Case (H_{m_0}/H_{m_i}) :*

$$q_i = \frac{\sum_{j=1}^{m_i} p_{ij}\mu_{ij}}{\sum_{j=1}^{m_0} p_{oj}\lambda_j + \sum_{j=1}^{m_i} p_{ij}\mu_{ij}}.$$

Proof in Appendix A.2.

Remark 4.5. *The product form of the PGF $G_{X_{(k)}}$ indicates that the load of the first k sites of the phase-type ASIP is equivalent in law to the sum of k independent geometric random variables on the non-negative integers. The can be seen as follows. Firstly, note that the probability mass function of a geometric random variable X_i on the non-negative integers which has probability of success q_i is given by*

$$\mathbb{P}(X_i = x) = (1-q_i)^x q_i.$$

Consequently, its PGF G_{x_i} is given by

$$G_{X_i}(z) = \mathbb{E}[z^{X_i}] = \sum_{x=0}^{\infty} \mathbb{P}(X_i = x) z^x = \sum_{x=0}^{\infty} (1-q_i)^x q_i z^x = q_i \sum_{x=0}^{\infty} ((1-q_i)z)^x = \frac{q_i}{1-(1-q_i)z}.$$

The PGF of $\sum_{i=1}^{k} X_i$ is equal to the product of the PGFs of X_i, due to the independence of the random variables X_i. This result does not come unexpected. The same product-form was found to hold for the classic n-site heterogeneous ASIP [8], but with a different expression for the probability q_i.

Corollary 4.6 (Two moment approximated *load* of the general ASIP)**.** *Consider the phase-type ASIP where we consider homogeneous inter-gate opening times, that is all gates share the same distribution. Furthermore, for the Hyper-exponential and mixed-Erlang random variables consider the parameters imposed by Lemma 3.3. Then the steady-state total load of the resulting approximated $n-site$ ASIP is given by*

$$\mathbb{E}[|\vec{X}|] = \begin{cases} n\lambda(m_1 - p_1)/\mu_1(m_0 - p_0) & \text{if } E_{m_0}/E_{m_i} \\ n(p_{01}\lambda_1 + p_{02}\lambda_2)(m_1 - p_1)/\mu_1 & \text{if } H_{m_0}/E_{m_i} \\ n\lambda/(p_{11}\mu_1 + p_{12}\mu_{12})(m_0 - p_0) & \text{if } E_{m_0}/H_{m_i} \\ n(p_{01}\lambda_1 + p_{02}\lambda_2)/(p_{11}\mu_1 + p_{12}\mu_2) & \text{if } H_{m_0}/H_{m_i} \end{cases}$$

Proof in Appendix A.2.

Remark 4.7. *Interestingly, we see that the load of the $n-$site approximated ASIP scales linearly in n for all cases.*

5 Numerics

In this section we investigate the *load* of the general n-site ASIP numerically. Firstly, we investigate how its 2-moment approximation depends of the first two moments of the inter-arrival time distribution and inter-gate opening time distribution of the system. Secondly, we investigate its CDF depends on the coefficient of variation of the inter-arrival time distribution and inter-gate opening time distribution, when these are identical.

5.1 Load - approximation

In the current section we investigate how the *load* of the two moment approximation of the 10-site ASIP depends on both the first two moments of the arrival distribution ($\mathbb{E}[A]$ and $\mathbb{E}[A^2]$) and the first two moments of the inter-gate opening time distribution ($\mathbb{E}[B]$ and $\mathbb{E}[B^2]$). We consider a range of values for the moments so that all cases in Corollary 4.6 are considered. For each case for a given distribution we consider one moment fixed and the other variable, leading to 4 possible cases. The details for these cases are omitted for brevity's sake, and are given in Appendix B. For each case we plot the *load* against c_A^2 and c_B^2, as shown in the figures below.

In Figures 18a and 18b, we have that $E[A]$ is constant and $E[A^2]$ is variable. Meaning that an increase in c_A^2 represents an increase in $E[A^2]$. It can be seen that as c_A^2 increases the *load* increases. Meaning, that increasing the variance of the arrivals increase the *load*. Similarly, in Figures 19a and 19d we also see that increasing $E[A^2]$ results in an increased *load*.

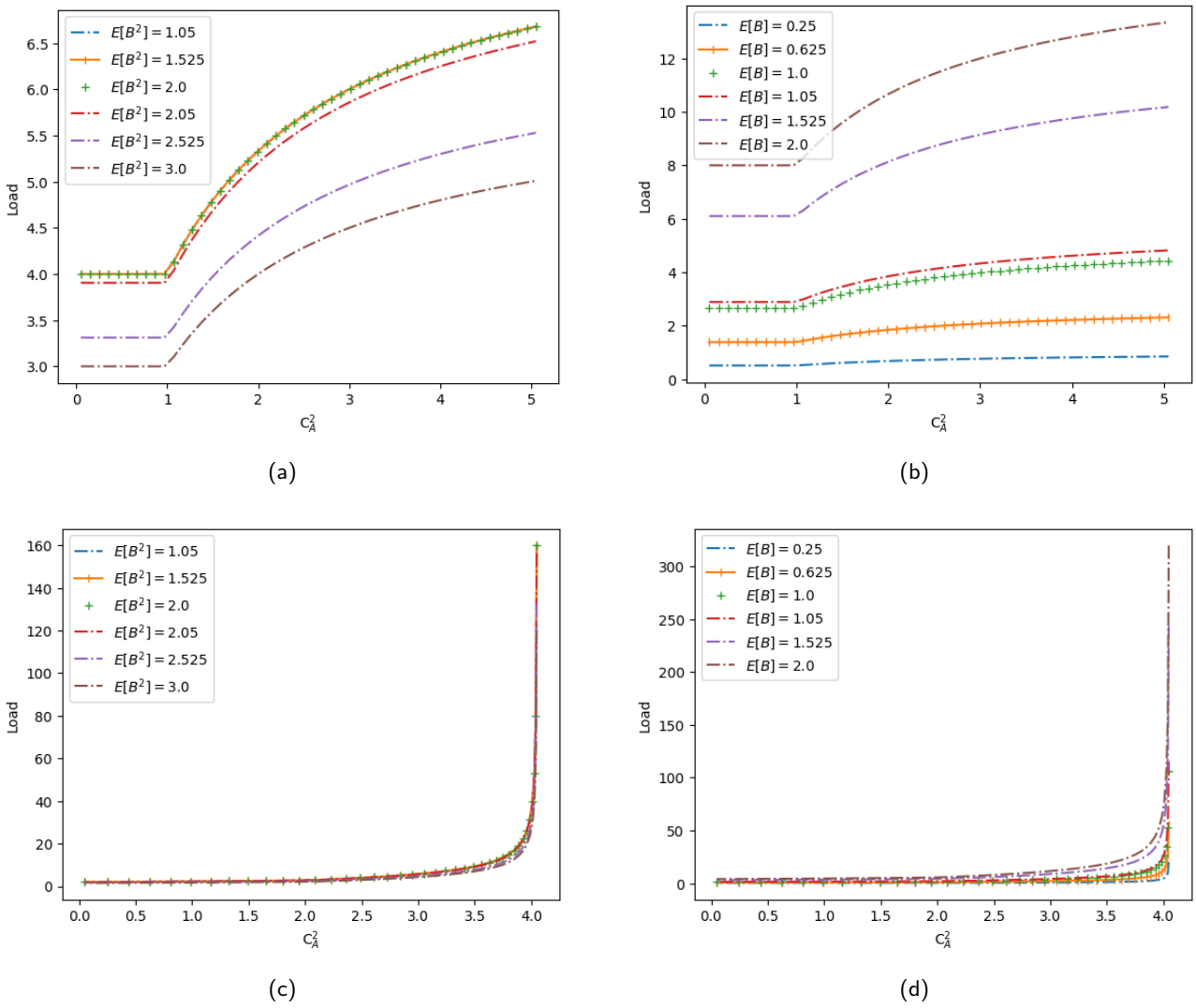


Figure 18: *load* of the approximated system against c_A^2

In Figures 18c and 18d, we have that $E[A]$ is variable and $E[A^2]$ is constant. Meaning that an increase in c_A^2 represents a decrease in $E[A]$. It can be seen that as c_A^2 increases the *load* increases. Meaning, that decreasing the mean inter-arrival time increases the *load*. Similarly, in Figures 19b and 19c we also see that decreasing $E[A]$ increases the *load*. Furthermore, we can see that decreasing $E[A]$ increases the rate at which increases in c_B^2 decrease

the *load*.

In Figures 19a and 19b, we have that $E[B]$ is constant and $E[B^2]$ is variable. Meaning that an increase in c_B^2 represents an increase in $E[B^2]$. It can be seen that as c_B^2 increases the *load* decreases. Meaning, that increasing the variance of the inter-gate opening times decreases the *load*. Similarly, in Figure 18a and 18c we also see that increasing $E[B^2]$ results in an increased *load*.

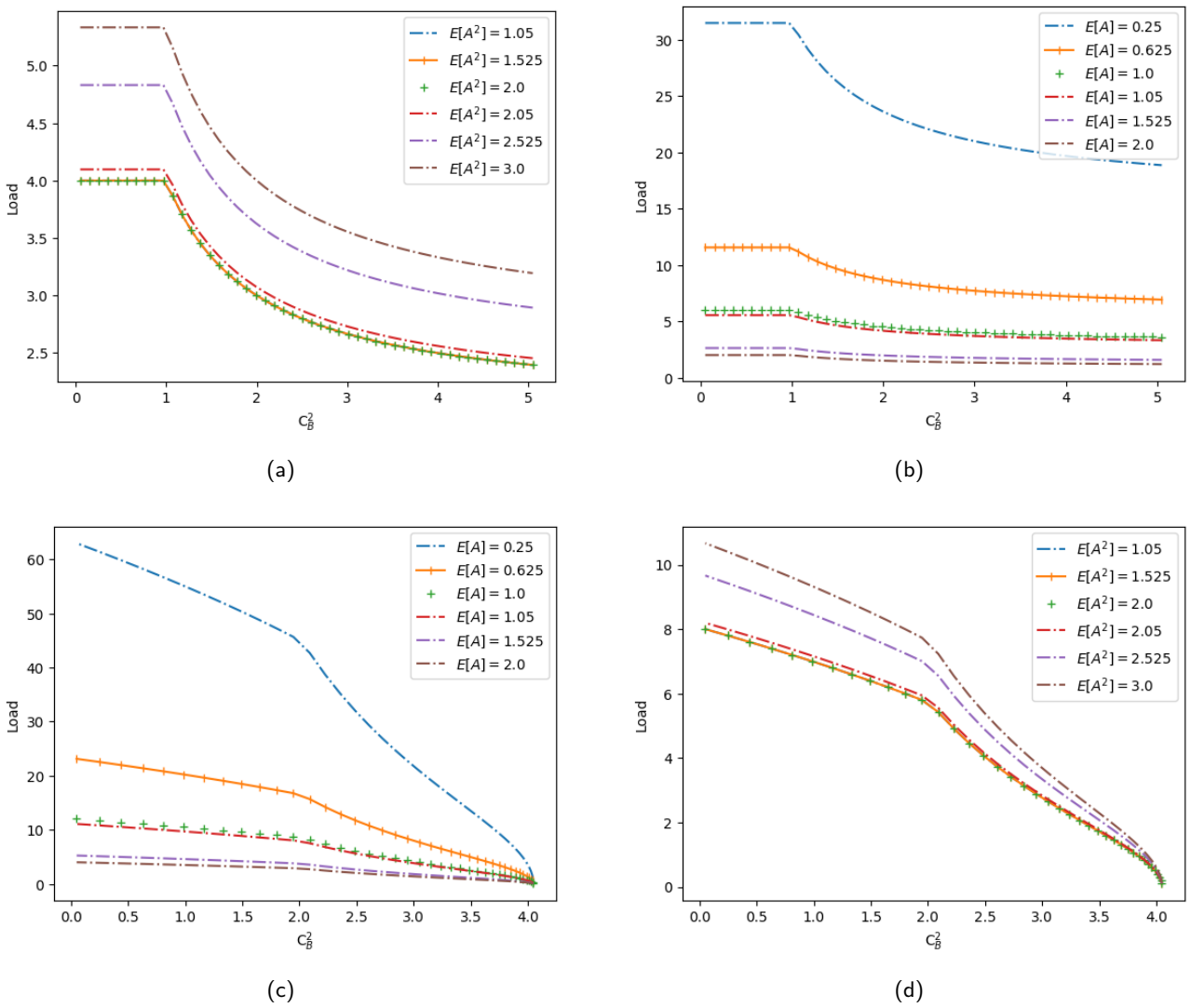


Figure 19: *load* of the approximated system against c_B^2

In Figures 19c and 19d, we have that $E[B]$ is variable and $E[B^2]$ is constant. Meaning that an increase in c_B^2 represents a decrease in $E[B]$. It can be seen that as c_B^2 increases the *load* decreases. Meaning, that decreasing the mean inter-gate opening time decreases the *load*. Similarly, in Figures 18b and 18d we also see that decreasing $E[B]$ decreases the *load*. Furthermore, we see that decreasing $E[B]$ decreases the rate at which increases in c_A^2 increase the *load*.

Overall we observe that: (1) Increasing the variance of the arrivals increases the *load*, (2) Increasing the variance of the inter-gate openings decreases the *load*, (3) Decreasing $E[B]$ decreases the *load*, and reduces the rate at which increases in the squared coefficient of variation of the arrival distribution (c_A^2) increase the *load*, (4) Decreasing $E[A]$ increases the *load*, and increases the rate at which increases the squared coefficient of variation of the inter-gate opening time distribution (c_B^2) decreases the *load*.

5.2 Load CDF - simulation

In this section, we employ Monte Carlo simulation to approximate the cumulative distribution function (CDF) of the *load* of the ASIP when both the inter-arrival times and the inter-gate opening times are distributed according to some non-negative distribution F_X. Our main focus is to understand how this approximate CDF varies with the

squared coefficient of variation. To achieve this, we consider three distinct cases for the distribution F_X, namely:

$$c_X^2 < 1, \qquad c_X^2 = 1, \qquad \text{and} \qquad c_X^2 > 1.$$

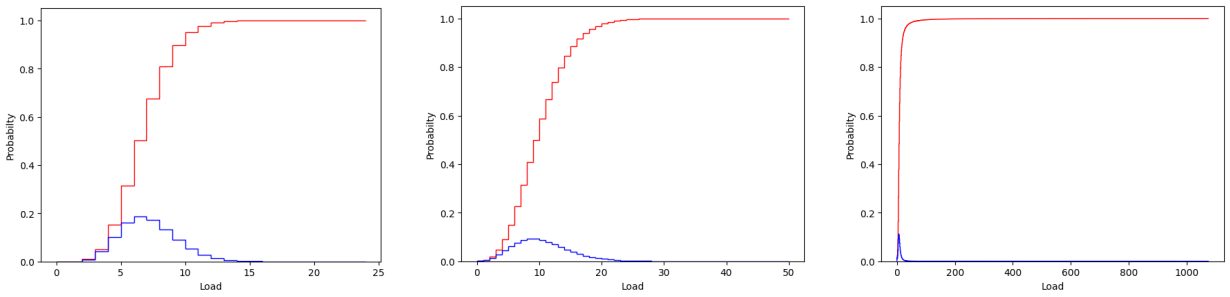
Furthermore, we choose $\mathbb{E}[X]$ equal to $\frac{1}{2}$ in all three cases. Thereby, reducing the degrees of freedom which allows for better comparison between cases. For the case c_X^2 we consider the continuous Uniform random variable $U(0,1)$, which has $c_X^2 = \frac{1}{3}$. For the c_X^2 we consider the Exponential random variable $Exp(2)$, which has $c_X^2 = 1$. For, the case c_X^2 we consider the Pareto random variable $P(x_{min}, \alpha)$ with $x_{min} = \frac{\sqrt{17}-3}{4}$ and $\alpha = 2 + x_{min}$, then $c_X^2 \approx 1.56$.

We simulate the $n-$site ASIP for each of the three aforementioned cases using Monte Carlo simulation. We measure the *load* for each simulated realisation. As a termination condition we use $1.96 \cdot \mathrm{SE} < 0.001$, but terminate early if 12 hours of simulation time have been reached.

In Table 7, we present the descriptive statistics of the simulated *load* for each of the distributions. We observe that all distributions have a minimum value of 0. As for the maximum value, mean, standard deviation, skewness, and kurtosis, the distributions follow the increasing order: Uniform, Exponential, and Pareto. However, this ordering does not apply to the median, where the Exponential distribution has a value of 10 and the Pareto distribution has a value of 8. This difference can be attributed to the extreme skewness and kurtosis of the Pareto distribution. It is noteworthy that all three distributions exhibit positive skewness, which is not surprising since the *load* cannot be negative. The skewness for the Pareto distribution is approximately 28 times larger than that of the Exponential distribution, while the kurtosis is roughly 830 times larger. This demonstrates that the Pareto distribution results in a *load* distribution with a heavy right-tail, as evident in Figure 20. Additionally, we can observe that qualitatively the Uniform and Exponential cases are more similar to each other than to the Pareto case.

Table 7: Descriptive statistics - simulated *load*

	Uniform	Exponential	Pareto
Min	0	0	0
Max	24	50	1076
Median	6	10	8
Mean	6.677601	9.999918	11.109262
Std	2.167824	4.471261	21.642106
Skew	0.501953	0.669483	18.778553
Kurtosis	0.336665	0.645940	535.949086



(a) Uniform (b) Exponential (c) Pareto

Figure 20: Histogram estimate of *load* density (blue) and *load* cdf (red)

In Table 8 the 95% CI for each case is given. It can be seen that these intervals do not overlap, meaning that the means are significantly different.

Table 8: 95% Confidence Interval(s) for the simulated *load*

	95% CI (LB, UB)
Uniform	(6.676847519724739, 6.678354302279025)
Exponential	(9.99824030643231, 10.00159632729567)
Pareto	(11.100548640469045, 11.117975430319737)

We find that despite each case having the same mean of $\frac{1}{2}$, there are significant qualitative and quantitative differences in the distribution of the *load*. This indicates that increasing c_X^2 and the associated changes in distribution have a substantial effect. Overall, increasing c_X^2 leads to higher values for the mean, standard deviation, skewness, and

kurtosis of the *load*. This increase in mean *load* aligns with the findings from the previous section, where larger c_X resulted in a heavier right-tailed behavior. Consequently, we can conclude that increasing c_X induces a greater clustering behavior.

Using Corollary 4.6, we can approximate the expected *load* for each of the aforementioned cases, which happens to be 10. This approximation provides an accurate result for the exponential case since 10 falls within the 95% confidence interval. However, for the Uniform distribution, it overestimates the *load* by about 3.32 (50%), equivalent to 0.332 (5%) per site. Conversely, for the Pareto distribution, it underestimates the *load* by about 1.11 (10%), equivalent to 0.111 (1%) per site. As a result, we can conclude that when the inter-arrival distribution and inter-gate opening distribution are the same, the approximation is most accurate when $c_X^2 \geq 1$.

6 Conclusions

From the approximation of the *load* of general ASIP, which used the two-moment approximation to approximate the inter-arrival and inter-gate opening time distribution, we conclude that: decreasing the first moment of the arrival distribution increases the *load*, and increasing the second moment of the arrival distribution increases the *load*. Furthermore, we conclude that decreasing the first moment of the inter-gate opening time distribution decreases the *load*, and increasing the second moment of the inter-gate opening time distribution decreases the *load*. Moreover, we can conclude the first moments of either distribution also influence the *load* indirectly. Namely, we found that decreasing the first moment of the inter-arrival distribution increases the rate at which increases in the second moment of the inter-gate opening time distribution decrease the *load*. Similarly, we found that decreasing the first moment of the inter-gate opening time distribution decreases the rate at which increases in the second moment of the inter-arrival distribution increase the *load*.

In this paper, we also considered the approximation of the *load* of a general 10 site ASIP using Monte Carlo simulation with the single distribution F_X for both the inter-arrivals and inter-gate opening times comparing three cases for c_X^2. Namely, $c_X^2 < 1, c_X^2 = 1$, and c_X^2 corresponding the Uniform, Exponential, and Pareto distributions, respectively. From the MC simulation results we can conclude that increases in c_X^2 are associated with increases in the mean, standard deviation, skewness, and kurtosis of the *load*. Meaning that as c_X^2 increases the distribution of the *load* becomes increasingly heavily right-tailed. Thereby, increasing clustering behavior.

Furthermore, we found that the two-moment approximation of the *load* falls within the 95% CI produced by the MC simulation for the case $c_X^2 = 1$. Meaning that in this case the approximation is accurate. This was note the case for the remaining cases, for $c_X^2 < 1$ the *load* was overestimated by 50%, while for $c_X^2 > 1$ the *load* was underestimated by 10%.

From the above we recognize three distinct avenues for further research. Namely, (1) use Monte Carlo simulation to verify the effect of the first and second moments of the inter-arrival and inter-gate opening time distribution as established by the two-moment approximation, (2) examine higher moments of the 2-moment approximated *load* to determine their dependence on the moments, in order to gain a better understanding of the performance characteristics of the general ASIP, (3) consider more elaborate fitting and or approximation techniques to fit the general distribution F_X to a phase-type distribution. Methods that warrant further exploration include different moment approximations, Kullback-Leibler Divergence minimization using maximum likelihood estimation, and hybrid methods.

References

[1] H. Sachdeva, M. Barma, and M. Rao, "Condensation and intermittency in an open-boundary aggregation-fragmentation model," en, *Phys Rev Lett*, vol. 110, no. 15, p. 150 601, Apr. 2013.

[2] H. Sachdeva, "Aggregation-Fragmentation Models for Transport in a Biological System," PhD thesis, Tata Institute of Fundamental Research, Mumbai, Aug. 2014.

[3] V. Kumar, A. Pal, and O. Shpielberg, *Arrhenius law for interacting diffusive systems*, 2023. arXiv: 2306.06879 [cond-mat.stat-mech].

[4] O. Shpielberg and A. Pal, "Thermodynamic uncertainty relations for many-body systems with fast jump rates and large occupancies," *Physical Review E*, vol. 104, no. 6, Dec. 2021. DOI: 10.1103/physreve.104.064141. [Online]. Available: https://doi.org/10.1103%2Fphysreve.104.064141.

[5] L. Garbe, Y. Minoguchi, J. Huber, and P. Rabl, *The bosonic skin effect: Boundary condensation in asymmetric transport*, 2023. arXiv: 2301.11339 [quant-ph].

[6] A. Aggarwal, *Dynamical stochastic higher spin vertex models*, 2017. arXiv: 1704.02499 [math-ph].

[7] H. C. Steinacker, "On the quantum structure of space-time, gravity, and higher spin in matrix models," *Classical and Quantum Gravity*, vol. 37, no. 11, p. 113 001, May 2020. DOI: 10.1088/1361-6382/ab857f. [Online]. Available: https://dx.doi.org/10.1088/1361-6382/ab857f.

[8] S. Reuveni, I. Eliazar, and U. Yechiali, "Asymmetric inclusion process," eng, *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 84, no. 4 Pt 1, p. 041 101, Oct. 2011, ISSN: 1550-2376. DOI: 10.1103/PhysRevE.84.041101.

[9] Y. Yeger and U. Yechiali, "Performance measures in a generalized asymmetric simple inclusion process," *Mathematics*, vol. 10, no. 4, 2022, ISSN: 2227-7390. DOI: 10.3390/math10040594. [Online]. Available: https://www.mdpi.com/2227-7390/10/4/594.

[10] S. Reuveni, I. Eliazar, and U. Yechiali, "Asymmetric inclusion process as a showcase of complexity," *Phys. Rev. Lett.*, vol. 109, p. 020 603, 2 Jul. 2012. DOI: 10.1103/PhysRevLett.109.020603. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.109.020603.

[11] S. Reuveni, I. Eliazar, and U. Yechiali, "Limit laws for the asymmetric inclusion process," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 86, p. 061 133, Dec. 2012. DOI: 10.1103/PhysRevE.86.061133.

[12] U. Yechiali and Y. Yeger, "Matrix approach for analyzing n-site generalized asip systems: Pgf and site occupancy probabilities," *Mathematics*, vol. 10, no. 23, 2022, ISSN: 2227-7390. DOI: 10.3390/math10234624. [Online]. Available: https://www.mdpi.com/2227-7390/10/23/4624.

[13] H. C. Tijms, *A first course in stochastic models / Henk C. Tijms*, eng. Chichester, England ; Hoboken, NJ: Wiley, 2003, pp. 444–447, ISBN: 0471498815.

[14] M. A. Johnson, "An empirical study of queueing approximations based on phase-type distributions," *Communications in Statistics. Stochastic Models*, vol. 9, no. 4, pp. 531–561, 1993. DOI: 10.1080/15326349308807280. eprint: https://doi.org/10.1080/15326349308807280. [Online]. Available: https://doi.org/10.1080/15326349308807280.

[15] T. Altiok, "On the phase-type approximations of general distributions," *IIE Transactions*, vol. 17, no. 2, pp. 110–116, 1985. DOI: 10.1080/07408178508975280. eprint: https://doi.org/10.1080/07408178508975280. [Online]. Available: https://doi.org/10.1080/07408178508975280.

A Appendix

A.1 Notation

In this paper we define to the following notations

$\vec{X}$: an n-dimensional vector of the site occupancies, that is each component represents the number of particles present at that particular site

X_j : the j-th component of $\vec{X}$, representing the number of particles present at the j-th site

$|\vec{X}|$: the total occupancy, where $|\vec{X}| = \sum_{j=1}^{n} X_j$,i.e. the total number of particles present in the system

$X_{(k)}$: the combined occupancy of the first k sites , where $X_{(k)} = \sum_{j=1}^{k} X_j$.

(x, y) : a 2-vector with components x and y

$\mathbb{1}_{\{A\}}$: indicator function, which evaluates to 1 if Boolean statement A holds true, otherwise evaluates to 0

$\mathbb{N}$: the natural numbers including zero

c_X : coefficient of variation of r.v. X, $\quad c_X = \dfrac{\sqrt{Var[X]}}{\mathbb{E}[X]}$

A.2 Proofs

Proof. of Theorem 4.3. Consider the case where the inter-arrival times are $E_{m_0}(p_0, \lambda)$ distributed and inter-gate opening times are $E_{m_i}(p_i, \mu_i)$ distributed. Define $\vec{Y}' = \vec{Y}(t + \Delta)$ and $\vec{Y} = \vec{Y}(t)$. Moreover, define $\mu = \sum_{i=1}^{n} \mu_i$. Then the Markovian dynamics of $\vec{Y}'$ are given by

$$
\begin{aligned}
&(S_0', S_1', \ldots, S_n', X_1', \ldots, X_n') = \\
&\begin{cases}
(S_0, S_1, \ldots, S_n, X_1, \ldots, X_n) & \text{w.p. } 1 - (\lambda + \mu)\Delta \\
(S_0(1 - \mathbb{1}_{\{S_0 = m_0\}}) + 1, S_1, \ldots, S_n, X_1 + \mathbb{1}_{\{S_0 = m_0\}}, X_2, \ldots, X_n) & \text{w.p. } \lambda p_0 \Delta \\
(S_0 - (S_0 - 1)\mathbb{1}_{\{S_0 = m_0\}} + 1, S_1, \ldots, S_n, X_1 + \mathbb{1}_{\{S_0 = m_0\}}, X_2, \ldots, X_n) & \text{w.p. } \lambda(1 - p_0)\Delta \\
(S_0, S_1(1 - \mathbb{1}_{\{S_1 = m_1\}}) + 1, S_2, \ldots, S_n, X_1(1 - \mathbb{1}_{\{S_1 = m_1\}}), X_2 + X_1 \mathbb{1}_{\{S_1 = m_1\}}, X_3, \ldots, X_n) & \text{w.p. } \mu_1 p_1 \Delta \\
(S_0, S_1 - (S_1 - 1)\mathbb{1}_{\{S_1 = m_1\}} + 1, S_2, \ldots, S_n, X_1(1 - \mathbb{1}_{\{S_1 = m_1\}}), X_2 + X_1 \mathbb{1}_{\{S_1 = m_1\}}, X_3, \ldots, X_n) & \text{w.p. } \mu_1(1 - p_1)\Delta \\
\vdots & \vdots \\
(S_0, \ldots, S_{n-1}, S_n(1 - \mathbb{1}_{\{S_n = m_n\}}) + 1, X_1, \ldots, X_{n-1}, X_n(1 - \mathbb{1}_{\{S_n = m_n\}})) & \text{w.p. } \mu_n p_n \Delta \\
(S_0, \ldots, S_{n-1}, S_n - (S_n - 1)\mathbb{1}_{\{S_n = m_n\}} + 1, X_1, \ldots, X_{n-1}, X_n(1 - \mathbb{1}_{\{S_n = m_n\}})) & \text{w.p. } \mu_n(1 - p_n)\Delta.
\end{cases}
\end{aligned}
$$

Using the law of total expectation and the Markov property, we can use the above Markovian dynamics to write the PGF of $\vec{Y}'$ as follows

$$
\begin{aligned}
\mathbb{E}\left[\prod_{k=0}^{2n} z_k^{Y_k'}\right] &= \mathbb{E}\left[\mathbb{E}\left[\prod_{k=0}^{2n} z_k^{Y_k'} \mid \vec{Y}\right]\right] \\
&= [1-(\lambda+\mu)\Delta]\,\mathbb{E}\left[\prod_{k=0}^{2n} z_k^{Y_k}\right] \\
&\quad + p_0\lambda z_0\Delta\mathbb{E}\left[z_0^{S_0(1-\mathbb{1}_{\{S_0=m_0\}})} z_{n+1}^{X_1+\mathbb{1}_{\{S_0=m_0\}}} \prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k}\right] \\
&\quad + (1-p_0)\lambda z_0\Delta\mathbb{E}\left[z_0^{S_0-(S_0-1)\mathbb{1}_{\{S_0=m_0\}}} z_{n+1}^{X_1+\mathbb{1}_{\{S_0=m_0\}}} \prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k}\right] \\
&\quad + p_1\mu_1 z_1\Delta\mathbb{E}\left[z_1^{S_1(1-\mathbb{1}_{\{S_1=m_1\}})} z_{n+1}^{X_1(1-\mathbb{1}_{\{S_1=m_1\}})} z_{n+2}^{X_2+X_1\mathbb{1}_{\{S_1=m_1\}}} \prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k}\right] \\
&\quad + (1-p_1)\mu_1 z_1\Delta\mathbb{E}\left[z_1^{S_1-(S_1-1)\mathbb{1}_{\{S_1=m_1\}}} z_{n+1}^{X_1(1-\mathbb{1}_{\{S_1=m_1\}})} z_{n+2}^{X_2+X_1\mathbb{1}_{\{S_1=m_1\}}} \prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k}\right] \\
&\quad + \\
&\quad \vdots \\
&\quad + \\
&\quad + p_n\mu_n z_n\Delta\mathbb{E}\left[z_n^{S_n(1-\mathbb{1}_{\{S_n=m_n\}})} z_{2n}^{X_n(1-\mathbb{1}_{\{S_n=m_n\}})} \prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k}\right] \\
&\quad + (1-p_n)\mu_n z_n\Delta\mathbb{E}\left[z_n^{S_n-(S_n-1)\mathbb{1}_{\{S_n=m_n\}}} z_{2n}^{X_n(1-\mathbb{1}_{\{S_n=m_n\}})} \prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k}\right].
\end{aligned}
$$

To proceed in our derivation we seek to eliminate the indicator functions $\mathbb{1}_{\{S_i=m_i\}}$ to do so, we condition on the value of S_i. Then by application of the law of total expectation we find the following

$$
\begin{aligned}
&\mathbb{E}\left[\prod_{k=0}^{2n} z_k^{Y_k'}\right] = [1-(\lambda+\mu)\Delta]\,\mathbb{E}\left[\prod_{k=0}^{2n} z_k^{Y_k}\right] \\
&+ p_0\lambda z_0\Delta\mathbb{E}\left[\mathbb{E}\left[z_0^{S_0(1-\mathbb{1}_{\{S_0=m_0\}})} z_{n+1}^{X_1+\mathbb{1}_{\{S_0=m_0\}}} \prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k} \mid S_0=m_0\right]\right]\mathbb{P}(S_0=m_0) \\
&+ p_0\lambda z_0\Delta\mathbb{E}\left[\mathbb{E}\left[z_0^{S_0(1-\mathbb{1}_{\{S_0=m_0\}})} z_{n+1}^{X_1+\mathbb{1}_{\{S_0=m_0\}}} \prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k} \mid S_0\neq m_0\right]\right]\mathbb{P}(S_0\neq m_0) \\
&+ (1-p_0)\lambda z_0\Delta\mathbb{E}\left[\mathbb{E}\left[z_0^{S_0-(S_0-1)\mathbb{1}_{\{S_0=m_0\}}} z_{n+1}^{X_1+\mathbb{1}_{\{S_0=m_0\}}} \prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k} \mid S_0=m_0\right]\right]\mathbb{P}(S_0=m_0) \\
&+ (1-p_0)\lambda z_0\Delta\mathbb{E}\left[\mathbb{E}\left[z_0^{S_0-(S_0-1)\mathbb{1}_{\{S_0=m_0\}}} z_{n+1}^{X_1+\mathbb{1}_{\{S_0=m_0\}}} \prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k} \mid S_0\neq m_0\right]\right]\mathbb{P}(S_0\neq m_0) \\
&+p_1\mu_1 z_1\Delta\mathbb{E}\left[\mathbb{E}\left[z_1^{S_1(1-\mathbb{1}_{\{S_1=m_1\}})} z_{n+1}^{X_1(1-\mathbb{1}_{\{S_1=m_1\}})} z_{n+2}^{X_2+X_1\mathbb{1}_{\{S_1=m_1\}}} \prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k} \mid S_1=m_1\right]\right]\mathbb{P}(S_1=m_1) \\
&+p_1\mu_1 z_1\Delta\mathbb{E}\left[\mathbb{E}\left[z_1^{S_1(1-\mathbb{1}_{\{S_1=m_1\}})} z_{n+1}^{X_1(1-\mathbb{1}_{\{S_1=m_1\}})} z_{n+2}^{X_2+X_1\mathbb{1}_{\{S_1=m_1\}}} \prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k} \mid S_1\neq m_1\right]\right]\mathbb{P}(S_1\neq m_1) \\
&+(1-p_1)\mu_1 z_1\Delta\mathbb{E}\left[\mathbb{E}\left[z_1^{S_1-(S_1-1)\mathbb{1}_{\{S_1=m_1\}}} z_{n+1}^{X_1(1-\mathbb{1}_{\{S_1=m_1\}})} z_{n+2}^{X_2+X_1\mathbb{1}_{\{S_1=m_1\}}} \prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k} \mid S_1=m_1\right]\right]\mathbb{P}(S_1=m_1) \\
&+(1-p_1)\mu_1 z_1\Delta\mathbb{E}\left[\mathbb{E}\left[z_1^{S_1-(S_1-1)\mathbb{1}_{\{S_1=m_1\}}} z_{n+1}^{X_1(1-\mathbb{1}_{\{S_1=m_1\}})} z_{n+2}^{X_2+X_1\mathbb{1}_{\{S_1=m_1\}}} \prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k} \mid S_1\neq m_1\right]\right]\mathbb{P}(S_1\neq m_1) \\
&+\cdots+ \\
&+ p_n\mu_n z_n\Delta\mathbb{E}\left[\mathbb{E}\left[z_n^{S_n(1-\mathbb{1}_{\{S_n=m_n\}})} z_{2n}^{X_n(1-\mathbb{1}_{\{S_n=m_n\}})} \prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k} \mid S_n=m_n\right]\right]\mathbb{P}(S_n=m_n) \\
&+ p_n\mu_n z_n\Delta\mathbb{E}\left[\mathbb{E}\left[z_n^{S_n(1-\mathbb{1}_{\{S_n=m_n\}})} z_{2n}^{X_n(1-\mathbb{1}_{\{S_n=m_n\}})} \prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k} \mid S_n\neq m_n\right]\right]\mathbb{P}(S_n\neq m_n) \\
&+ (1-p_n)\mu_n z_n\Delta\mathbb{E}\left[\mathbb{E}\left[z_n^{S_n-(S_n-1)\mathbb{1}_{\{S_n=m_n\}}} z_{2n}^{X_n(1-\mathbb{1}_{\{S_n=m_n\}})} \prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k} \mid S_n=m_n\right]\right]\mathbb{P}(S_n=m_n) \\
&\quad+ (1-p_n)\mu_n z_n\Delta\mathbb{E}\left[\mathbb{E}\left[z_n^{S_n-(S_n-1)\mathbb{1}_{\{S_n=m_n\}}} z_{2n}^{X_n(1-\mathbb{1}_{\{S_n=m_n\}})} \prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k} \mid S_n\neq m_n\right]\right]\mathbb{P}(S_n\neq m_n).
\end{aligned}
$$

Next, we make use of the conditioning to simplify the expressions within the expectations, this gives use the following

$$
\begin{aligned}
\mathbb{E}\left[\prod_{k=0}^{2n} z_k^{Y_k'}\right] &= [1-(\lambda+\mu)\Delta]\,\mathbb{E}\left[\prod_{k=0}^{2n} z_k^{Y_k}\right] \\
&\quad + p_0\lambda z_0\Delta\mathbb{E}\left[\mathbb{E}\left[z_0^0 z_{n+1}^{X_1+1}\prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k} \mid S_0=m_0\right]\right]\mathbb{P}(S_0=m_0) \\
&\quad + p_0\lambda z_0\Delta\mathbb{E}\left[\mathbb{E}\left[z_0^{S_0} z_{n+1}^{X_1}\prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k} \mid S_0\neq m_0\right]\right]\mathbb{P}(S_0\neq m_0) \\
&\quad + (1-p_0)\lambda z_0\Delta\mathbb{E}\left[\mathbb{E}\left[z_0^1 z_{n+1}^{X_1+1}\prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k} \mid S_0=m_0\right]\right]\mathbb{P}(S_0=m_0) \\
&\quad + (1-p_0)\lambda z_0\Delta\mathbb{E}\left[\mathbb{E}\left[z_0^{S_0} z_{n+1}^{X_1}\prod_{k\neq\{0,n+1\}}^{2n} z_k^{Y_k} \mid S_0\neq m_0\right]\right]\mathbb{P}(S_0\neq m_0) \\
&\quad + p_1\mu_1 z_1\Delta\mathbb{E}\left[\mathbb{E}\left[z_1^0 z_{n+1}^0 z_{n+2}^{X_2+X_1}\prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k} \mid S_1=m_1\right]\right]\mathbb{P}(S_1=m_1) \\
&\quad + p_1\mu_1 z_1\Delta\mathbb{E}\left[\mathbb{E}\left[z_1^{S_1} z_{n+1}^{X_1} z_{n+2}^{X_2}\prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k} \mid S_1\neq m_1\right]\right]\mathbb{P}(S_1\neq m_1) \\
&\quad + (1-p_1)\mu_1 z_1\Delta\mathbb{E}\left[\mathbb{E}\left[z_1^1 z_{n+1}^0 z_{n+2}^{X_2+X_1}\prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k} \mid S_1=m_1\right]\right]\mathbb{P}(S_1=m_1) \\
&\quad + (1-p_1)\mu_1 z_1\Delta\mathbb{E}\left[\mathbb{E}\left[z_1^{S_1} z_{n+1}^{X_1} z_{n+2}^{X_2}\prod_{k\neq\{1,n+1,n+2\}}^{2n} z_k^{Y_k} \mid S_1\neq m_1\right]\right]\mathbb{P}(S_1\neq m_1) \\
&\quad \vdots \\
&\quad + p_n\mu_n z_n\Delta\mathbb{E}\left[\mathbb{E}\left[z_n^0 z_{2n}^0\prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k} \mid S_n=m_n\right]\right]\mathbb{P}(S_n=m_n) \\
&\quad + p_n\mu_n z_n\Delta\mathbb{E}\left[\mathbb{E}\left[z_n^{S_n} z_{2n}^{X_n}\prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k} \mid S_n\neq m_n\right]\right]\mathbb{P}(S_n\neq m_n) \\
&\quad + (1-p_n)\mu_n z_n\Delta\mathbb{E}\left[\mathbb{E}\left[z_n^1 z_{2n}^0\prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k} \mid S_n=m_n\right]\right]\mathbb{P}(S_n=m_n) \\
&\quad + (1-p_n)\mu_n z_n\Delta\mathbb{E}\left[\mathbb{E}\left[z_n^{S_n} z_{2n}^{X_n}\prod_{k\neq\{n,2n\}}^{2n} z_k^{Y_k} \mid S_n\neq m_n\right]\right]\mathbb{P}(S_n\neq m_n).
\end{aligned}
$$

Now use the definition of the PGF G_Y to simplify the expressions further, this yields

$$
\begin{aligned}
&G_Y(t',z_0,\ldots,z_{2n}) = [1-(\lambda+\mu)\Delta]\,G_Y(t,z_0,\ldots,z_{2n}) \\
&+\lambda p_0 z_0 z_{n+1}\mathbb{P}(S_0=m_0)\Delta G_Y(t,1,z_1,\ldots,z_{2n}) + \lambda p_0 z_0\mathbb{P}(S_0\neq m_0)\Delta G_Y(t,z_0,\ldots,z_{2n}) \\
&+\lambda(1-p_0)z_0^2 z_{n+1}\mathbb{P}(S_0=m_0)\Delta G_Y(t,1,z_1,\ldots,z_{2n}) + \lambda(1-p_0)z_0\mathbb{P}(S_0\neq m_0)\Delta G_Y(t,z_0,\ldots,z_{2n}) \\
&+\mu_1 p_1 z_1\mathbb{P}(S_1=m_1)\Delta G_Y(t,z_0,1,z_2,\ldots,z_n,z_{n+2},z_{n+2},z_{n+3},\ldots,z_{2n}) + \mu_1 p_1 z_1\mathbb{P}(S_1\neq m_1)\Delta G_Y(t,z_0,\ldots,z_{2n}) \\
&+\mu_1(1-p_1)z_1^2\mathbb{P}(S_1=m_1)\Delta G_Y(t,z_0,1,z_2,\ldots,z_n,z_{n+2},z_{n+2},z_{n+3},\ldots,z_{2n}) \\
&+\mu_1(1-p_1)z_1\mathbb{P}(S_1\neq m_1)\Delta G_Y(t,z_0,\ldots,z_{2n}) \\
&\vdots \\
&+\mu_n p_n z_n\mathbb{P}(S_n=m_n)\Delta G_Y(t,z_0,\ldots,z_{n-1},1,z_{n+1},\ldots,z_{2n-1},1) + \mu_n p_n z_n\mathbb{P}(S_n\neq m_n)\Delta G_Y(t,z_0,\ldots,z_{2n}) \\
&\quad +(1-p_n)\mu_n z_n^2\mathbb{P}(S_n=m_n)\Delta G_Y(t,z_0,\ldots,z_{2n-1},1) + (1-p_n)\mu_n z_n\mathbb{P}(S_n=m_n)\Delta G_Y(t,z_0,\ldots,z_{2n}).
\end{aligned}
$$

Next, we group all terms that share the same dummy variables for G_Y. In doing so, we get the following

$$
\begin{aligned}
G_Y(t', z_0, \ldots, z_{2n}) = &\left[1 - (\lambda+\mu)\Delta + \lambda z_0 \mathbb{P}(S_0 \neq m_0)\Delta + \sum_{j=1}^{n} \mu_j z_j \mathbb{P}(S_j \neq m_j)\Delta\right] G_Y(t, z_0, \ldots, z_{2n}) \\
&+ \lambda z_{n+1}\left[p_0 z_0 + (1-p_0)z_0^2\right]\mathbb{P}(S_0 = m_0)\Delta G_Y(t, 1, z_1, \ldots, z_{2n}) \\
&+ \mu_1\left[p_1 z_1 + (1-p_1)z_1^2\right]\mathbb{P}(S_1 = m_1)\Delta G_Y(t, z_0, 1, z_2, \ldots, z_n, z_{n+2}, z_{n+2}, z_{n+3}, \ldots, z_{2n}) \\
&\vdots \\
&+ \mu_{n-1}\left[p_{n-1} z_{n-1} + (1-p_{n-1})z_{n-1}^2\right]\mathbb{P}(S_{n-1} = m_{n-1})\Delta G_Y(t, z_0, z_1, \ldots, z_{n-2}, 1, z_n, \ldots, z_{2n-2}, z_{2n}, z_{2n}) \\
&\qquad + \mu_n\left[p_n z_n + (1-p_n)z_n^2\right]\mathbb{P}(S_n = m_n)\Delta G_Y(t, z_0, \ldots, z_{n-1}, 1, z_{n+1}, \ldots, z_{2n-1}, 1).
\end{aligned}
$$

Next, we take the derivative w.r.t time. To this end, first subtract $G_Y(t, z_0, \ldots, z_{2n})$ on both sides of the equation. Subsequently, divide both sides by Δ. Lastly, take the limit $\Delta \to 0$. This yields the following result

$$
\begin{aligned}
\frac{\partial}{\partial t} G_Y(t, z_0, \ldots, z_{2n}) = &\left[\lambda(z_0 \mathbb{P}(S_0 \neq m_0) - 1) + \sum_{j=1}^{n} \mu_j (z_j \mathbb{P}(S_j \neq m_j) - 1)\right] G_Y(t, z_0, \ldots, z_{2n}) \\
&+ \lambda z_{n+1}\left[p_0 z_0 + (1-p_0)z_0^2\right]\mathbb{P}(S_0 = m_0) G_Y(t, 1, z_1, \ldots, z_{2n}) \\
&+ \mu_1\left[p_1 z_1 + (1-p_1)z_1^2\right]\mathbb{P}(S_1 = m_1) G_Y(t, z_0, 1, z_2 \ \ldots, z_n, z_{n+2}, z_{n+2}, z_{n+3}, \ldots, z_{2n}) \\
&\vdots \\
&+ \mu_{n-1}\left[p_{n-1} z_{n-1} + (1-p_{n-1})z_{n-1}^2\right]\mathbb{P}(S_{n-1} = m_{n-1}) G_Y(t, z_0, \ldots, z_{n-2}, 1, z_n, \ldots, z_{2n-2}, z_{2n}, z_{2n}) \\
&\qquad + \mu_n\left[p_n z_n + (1-p_n)z_n^2\right]\mathbb{P}(S_n = m_n) G_Y(t, z_0, \ldots, z_{n-1}, 1, z_{n+1}, \ldots, z_{2n-1}, 1).
\end{aligned}
$$

The above yields the evolution equation of the PGF of the state vector $\vec{Y}(t)$, taking $z_0 = \cdots = z_n = 1$ gives the evolution equation of the PGF of the site occupancy vector $\vec{X}(t)$. Subsequently re-numerating the dummy variables $\{z_i : i \in n+1, \ldots, 2n\}$ to $\{z_i : i \in 1, \ldots, 2\}$ gives the following

$$
\begin{aligned}
\frac{\partial}{\partial t} G_X(t, z_1, \ldots, z_n) = &\left[\lambda(\mathbb{P}(S_0 \neq m_0) - 1) + \sum_{j=1}^{n} \mu_j (\mathbb{P}(S_j \neq m_j) - 1)\right] G_X(t, z_1, \ldots, z_n) \\
&+ \lambda z_1 \mathbb{P}(S_0 = m_0) G_X(t, z_1, \ldots, z_n) + \mu_1 \mathbb{P}(S_1 = m_1) G_X(t, z_2, z_2, z_3, \ldots, z_n) \\
&\qquad + \cdots + \mu_{n-1}\mathbb{P}(S_{n-1} = m_{n-1}) G_X(t, z_1, \ldots, z_{n-2}, z_n, z_n) + \mu_n \mathbb{P}(S_n = m_n) G_X(t, z_1, \ldots, z_{n-1}, 1).
\end{aligned}
$$

From this we derive the steady-state PGF. To this end, take $\frac{\partial}{\partial t} G_X = 0$, take G_X and $\mathbb{P}(S_i = m_i)$ independent of t. Then we find

$$
\begin{aligned}
G_X(z_1, \ldots, z_n)&\left[\lambda(1 - \mathbb{P}(S_0 \neq m_0)) + \sum_{j=1}^{n} \mu_j (1 - \mathbb{P}(S_j \neq m_j))\right] = \lambda z_1 \mathbb{P}(S_0 = m_0) G_X(z_1, \ldots, z_n) \\
&+ \mu_1 \mathbb{P}(S_1 = m_1) G_X(z_2, z_2, z_3, \ldots, z_n) + \cdots + \mu_{n-1}\mathbb{P}(S_{n-1} = m_{n-1}) G_X(z_1, \ldots, z_{n-2}, z_n, z_n) \\
&\qquad + \mu_n \mathbb{P}(S_n = m_n) G_X(z_1, \ldots, z_{n-1}, 1).
\end{aligned}
$$

Now we have shown the result for the case with mixed-Erlang arrivals and mixed-Erlang inter-gate opening times. The other cases can be derived in a similar fashion, and are omitted for brevity's sake. Note that these remaining cases include hyper-exponential distributions, which are not composed of multiple exponential phases. As a consequence the vector $\vec{S}(t)$ is simpler in these cases, and therefore the derivations are simpler than the case treated above.

It still remains to be shown that $\mathbb{P}(S_i = m_i) = \frac{1}{m_i - p_i}$. Recall that S_i is the process corresponding to the Markov chain of the mixed-Erlang random variable $E_{m_i}(p_i, \mu_i)$. Recall, that $\mathbb{P}(S_i = m_i)$ represents the steady-state probability

that S_i is in state m_i. To this end, we compute the stationary distribution of the associated Markov chain. Its $m_i \times m_i$ rate matrix Q_i is given by

$$Q_i = \begin{bmatrix} -\mu_i & \mu_i & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & -\mu_i & \mu_i \\ (1-p_i)\mu_i & p_i\mu_i & 0 & \cdots & 0 & -\mu_i. \end{bmatrix}$$

Now we solve $\pi Q_i = 0$ under the constraint $\sum_{j=1}^{m_i} \pi_j = 1$, this gives the following equations

$$\begin{aligned} -\mu_i\pi_1 + (1-p_i)\mu_i\pi_{m_i} &= 0 \\ \mu_i\pi_1 - \mu_i\pi_2 + p_i\mu\pi_{m_i} &= 0 \\ \mu_i\pi_2 - \mu_i\pi_3 &= 0 \\ &\vdots \\ \mu_i\pi_{m_{i-1}} - \mu_i\pi_{m_i} &= 0. \end{aligned}$$

From this is follows that $\pi_2 = \cdots = \pi_{m_i}$. Substituting this into the constraint yields

$$\pi_1 = 1 - (\pi_2 + \cdots + \pi_{m_i}) = 1 - (m_i - 1)\pi_{m_i}.$$

Now substituting this into the first equation yields,

$$-\mu_i\left[1 - (m_i - 1)\pi_{m_i}\right] + (1-p_i)\mu_i\pi_{m_i} = 0.$$

Upon isolating π_{m_i} we find

$$\pi_{m_i} = \frac{1}{m_i - p_i}.$$

Meaning that

$$\mathbb{P}(S_i = m_i) = \frac{1}{m_i - p_i}.$$

Which is the result that had to be shown, thereby concluding the proof. □

Proof. of Theorem 4.4. We seek to find an expression for $G_{X_{(k)}}(z)$ we do so by using the recursive equation for $G_X(z)$, given in Theorem 4.3. Consider the case where the inter-arrival times are $E_{m_0}(p_0, \lambda)$ distributed and inter-gate opening times are $E_{m_i}(p_i, \mu_i)$ distributed. First, notice that $G_X(z, \dots, z, 1, \dots, 1) = G_{X_{(k)}}(z)$ if we set $z_1 = \cdots = z_k = z$ and $z_{k+1} = z_n = 1$ in the expression for G_X. Upon applying this change to the equation in Theorem 4.3, we find

$$\begin{aligned} &\left[\lambda\mathbb{P}(S_0 = m_0) + \sum_{j=1}^{n} \mu_j\mathbb{P}(S_j = m_j)\right] G_{X_{(k)}}(z) = \lambda z\mathbb{P}(S_0 = m_0)G_{X_{(k)}}(z) \\ &+ \mu_1\mathbb{P}(S_1 = m_1)G_{X_{(k)}}(z) + \cdots + \mu_{k-1}\mathbb{P}(S_{k-1} = m_{k-1})G_{X_{(k)}}(z) + \mu_k\mathbb{P}(S_k = m_k)G_{X_{(k-1)}}(z) \\ &\qquad + \mu_{k+1}\mathbb{P}(S_{k+1} = m_{k+1})G_{X_{(k)}}(z) + \cdots + \mu_n\mathbb{P}(S_n = m_n)G_{X_{(k)}}(z) \end{aligned}$$

Combining like terms, and upon cancellation of various $P(S_i = m_i)$ terms gives the following

$$\left[\lambda(1-z)\mathbb{P}(S_0 = m_0) + \mu_k\mathbb{P}(S_k = m_k)\right] G_{X_{(k)}}(z) = \mu_k\mathbb{P}(S_k = m_k)G_{X_{(k-1)}}(z).$$

Now we isolate $G_{X_{(k)}}(z)$ which yields the recursive equation

$$G_{X_{(k)}}(z) = \frac{\mu_k\mathbb{P}(S_k = m_k)}{\lambda(1-z)\mathbb{P}(S_0 = m_0) + \mu_k\mathbb{P}(S_k = m_k)} G_{X_{(k-1)}}(z).$$

Note that for case $k = 1$ we have

$$G_{X_{(1)}}(z) = \frac{\mu_1 \mathbb{P}(S_1 = m_1)}{\lambda(1-z)\mathbb{P}(S_0 = m_0) + \mu_1 \mathbb{P}(S_1 = m_1)} G_{X_{(0)}}(1),$$

where $G_{X_{(0)}}(1) = 1$. Using this base case, we can iterate the above recursive equation to get the following explicit result

$$G_{X_{(k)}}(z) = \prod_{i=1}^{k} \frac{\mu_i \mathbb{P}(S_i = m_i)}{\lambda(1-z)\mathbb{P}(S_0 = m_0) + \mu_i \mathbb{P}(S_i = m_i)} = \prod_{i=1}^{k} \frac{q_i}{1-(1-q_i)z},$$

where q_i is chosen to be

$$q_i = \frac{\mu_i \mathbb{P}(S_i = m_i)}{\lambda P(S_0 = m_0) + \mu_i \mathbb{P}(S_i = m_i)}.$$

This last step above results from several simple algebraic manipulations. With this we have shown the result for the current case, the other cases follow analogously. □

Proof. of Corollary 4.6. In this proof we give the derivation of the explicit expression for the expected load for each of the four cases considered. For each case we make use of the result found in Theorem 4.4, which shows that the PGF of $G_{X_n}(z)$ is equivalent in law to the PGF of the sum of n independent geometric random variables on the non-negative integers, with probability q_i for $i = 1, \ldots, n$. Such a random variable has mean $\frac{1-q_i}{q_i}$. From this we can easily, deduce the expected value of the load. Before doing so, we furthermore apply the restrictions imposed by Lemma 3.3, and consider homogeneous inter-gate opening times. That is each gate follows the same distribution. Then the expected load for the pure mixed-Erlang case is derived as follows

$$\begin{aligned}
\mathbb{E}\left[|\vec{X}|\right] &= \sum_{i=1}^{n} \frac{1-q_i}{q_i} \\
&= \sum_{i=1}^{n} \frac{\lambda \mathbb{P}(S_i = m_i)/[\lambda \mathbb{P}(S_0 = m_0) + \mu_i \mathbb{P}(S_i = m_i)]}{\mu_i \mathbb{P}(S_i = m_i)/[\lambda \mathbb{P}(S_0 = m_0) + \mu_i \mathbb{P}(S_i = m_i)]} \\
&= \sum_{i=1}^{n} \frac{\lambda \mathbb{P}(S_0 = m_0)}{\mu_i \mathbb{P}(S_i = m_i)} \\
&= \sum_{i=1}^{n} \frac{\lambda / \mathbb{P}(S_i = m_i)}{\mu_i / \mathbb{P}(S_0 = m_0)} \\
&= n\lambda(m_1 - p_1)/\mu_1(m_0 - p_0).
\end{aligned}$$

The last step follows from homogeneity of the inter-gate opening times, i.e. $\mu_1 = \cdots = \mu_n$, $m_1 = \cdots = m_n$, $p_1 = \cdots = p_n$, and the expression for $\mathbb{P}(S_i = m_i)$ which is

$$\mathbb{P}(S_i = m_i) = \frac{1}{m_i - p_i}.$$

The remaining cases follow analogously. □

B Appendix

In this section we provide details for the cases considered in Section 5.1. In particular the are 4 different cases,

1. $E[A]$ and $E[B]$ constant, and $E[A^2]$ and $E[B^2]$ variable, see Figures 18a and 19a.
2. $E[A]$ and $E[B^2]$ constant, and $E[A^2]$ and $E[B]$ variable, see Figures 18b and 19d.
3. $E[A^2]$ and $E[B]$ constant, and $E[A]$ and $E[B^2]$ variable, see Figures 18c and 19b.
4. $E[A^2]$ and $E[B^2]$ constant, and $E[A]$ and $E[B]$ variable. see Figures 18d and 19c.

Each case is associated with two sub-cases depending on whether C_A^2 or C_B^2 is on the x-axis. We enumerate these as follows

1. - In the case where C_A^2 is on the x-axis, as seen in Figure 18a. The following parameter values are considered: $E[A^2] \in \{1.05, 1.15, \ldots, 6.05\}$ and $E[B^2] \in \{1.05, 1.525, 2, 2.05, 2.525, 3\}$ and $E[A] = E[B] = 1$.
   - In the case where C_B^2 is on the x-axis, as seen in Figure 19a. The following parameter values are considered: $E[B^2] \in \{1.05, 1.15, \ldots, 6.05\}$ and $E[A^2] \in \{1.05, 1.525, 2, 2.05, 2.525, 3\}$ and $E[A] = E[B] = 1$.
2. - In the case where C_A^2 is on the x-axis, as seen in Figure 18b. The following parameter values are considered: $E[A^2] \in \{1.05, 1.15, \ldots, 6.05\}$ and $E[B] \in \{.25, .625, 1, 1.05, 1.525, 2\}$ and $E[A] = 1$ and ,$E[B^2] = 4.05$.
   - In the case where C_B^2 is on the x-axis, as seen in Figure 19d. The following parameter values are considered: $E[B] \in \{.05, 0.1, 0.15, \ldots, 2.0\}$ and $E[A^2] \in \{1.05, 1.525, 2, 2.05, 2.525, 3\}$ and $E[A] = 1$, and $E[B^2] = 4.05$.
3. - In the case where C_A^2 is on the x-axis, as seen in Figure 18c. The following parameter values are considered: $E[A] \in \{.05, .1, .15, \ldots, 2\}$ and $E[B^2] \in \{1.05, 1.525, 2, 2.05, 2.525, 3\}$ and $E[A^2] = 4.05$ and ,$E[B] = 1$.
   - In the case where C_B^2 is on the x-axis, as seen in Figure 19b. The following parameter values are considered: $E[A] \in \{.25, 0.625, 1, 1.05, 1.525, 2.0\}$ and $E[B^2] \in \{1.05, 1.15, \ldots, 6.05\}$ and $E[A^2] = 4.05$, and $E[B] = 1$.
4. - In the case where C_A^2 is on the x-axis, as seen in Figure 18d. The following parameter values are considered: $E[A] \in \{.05, .1, .15, \ldots, 2\}$ and $E[B] \in \{.25, .625, 1, 1.05, 1.525, 2\}$ and $E[A^2] = E[B^2] = 4.05$.
   - In the case where C_B^2 is on the x-axis, as seen in Figure 19c. The following parameter values are considered: $E[B] \in \{.05, .1, .15, \ldots, 2\}$ and $E[A] \in \{.25, .625, 1, 1.05, 1.525, 2\}$ and $E[A^2] = E[B^2] = 4.05$.