# A fluid analysis framework for a Markovian process algebra

Richard A. Hayden[a], Jeremy T. Bradley[a]

[a] *Department of Computing, Imperial College London,*
*Huxley Building, 180 Queen's Gate, London SW7 2AZ, United Kingdom*

**Abstract**

Markovian process algebras, such as PEPA and stochastic $\pi$-calculus, bring a powerful compositional approach to the performance modelling of complex systems. However, the models generated by process algebras, as with other interleaving formalisms, are susceptible to the state space explosion problem. Models with only a modest number of process algebra terms can easily generate so many states that they are all but intractable to traditional solution techniques. Previous work aimed at addressing this problem has presented a *fluid-flow* approximation allowing the analysis of systems which would otherwise be inaccessible. To achieve this, systems of ordinary differential equations describing the fluid flow of the stochastic process algebra model are generated informally.

In this paper, we show formally that for a large class of models, this fluid-flow analysis can be directly derived from the stochastic process algebra model as an approximation to the mean number of component types within the model. The nature of the fluid approximation is derived and characterised by direct comparison with the Chapman–Kolmogorov equations underlying the Markov model. Furthermore, we compare the fluid approximation with the exact solution using stochastic simulation and we are able to demonstrate that it is a very accurate approximation in many cases.

For the first time, we also show how to extend these techniques naturally to generate systems of differential equations approximating higher order moments of model component counts. These are important performance characteristics for estimating, for instance, the variance of the component counts. This is very necessary if we are to understand how precise the fluid-flow calculation is, in a given modelling situation.

*Key words:* stochastic process algebra, fluid approximation, differential equations, higher moments

## 1. Introduction

Continuous-time Markov chains (CTMCs) are often used to capture performance models, biochemical models, or other state-transition systems that involve uncertain knowledge or random behaviour. Instead of directly constructing a CTMC to model a system, it is often preferable to use a higher-level compositional approach provided by, for instance, a *stochastic process algebra* (SPA). In particular, our work is presented in the context of the popular stochastic process algebra, PEPA [1], but could be extended to other Markovian modelling formalisms, such as stochastic $\pi$-calculus [2], EMPA [3], IMC [4] or sCCP [5].

Continuous-time Markov chains are relatively tractable to analyse due, in large part, to the *memoryless* property of the exponential distribution. Despite this, stochastic process algebra models of realistic complexity can easily result in underlying state spaces of computationally intractable size. Specifically, the generation of the state space and subsequent computation of a set of transient or steady state probabilities of such CTMCs can easily become computationally infeasible. This phenomenon is known as *the state space explosion problem* and is a current bottleneck in the field of performance analysis, limiting the size of models and thus the complexity of systems that can be analysed efficiently. This is especially true when attempts are made at modelling massively parallel systems, such as peer-to-peer networks [6], publish–subscribe networks, and other massive client–server architectures.

There has been much research aimed at reducing this problem for stochastic process algebras. To some extent, it is possible to draw on the compositional structure of the model to simplify the problem. Such techniques include state space aggregation [7], component substitution for approximate analysis [8] and for some models, it is possible to find product form solutions [9, 10, 11].

Of all these techniques, the use of lumpability to exploit model symmetry is one of the most widely deployed. Examples of this include Buchholz's direct application to the Markov chain, with steady-state and transient results [12]; also the application to a hierarchically structured Markovian state-space [7, 13]. Additionally, lumpability can be mechanically derived from models which display suitable component replication and are specified in higher level formalisms such as stochastic Petri nets [14] and stochastic process algebras [15]. We exploit the last of these results, using Gilmore's numerical vector form [15] to capture an aggregated state space from a stochastic process algebra model. While the lumping aggregation is effective in reducing the state space, we will show in Section 1.2.1, that the aggregated state space is still far too large to enumerate and analyse for any detailed component description. Hence the need for so-called *fluid-flow* techniques.

Fluid-flow approximation [16] constructs a continuous state space representation of the underlying lumped discrete state space and recasts the discrete model into a system of coupled ordinary differential equations (ODEs). This is extremely powerful when the underlying model exhibits massive symmetry, such as is the case with large numbers of homogeneous agents operating in parallel.

Differential equations are easily amenable to numerical analysis (for example, via the Euler method) and this method of attack is very scalable. Indeed, it is much faster to solve the resulting small systems of coupled differential equations than to solve the discrete model for its transient or steady state probabilities. It is also often quicker than performing stochastic simulation [17] of the underlying stochastic process. In the context of PEPA, the existing transformations from process algebra to differential equations [16, 18] are currently informal. Indeed, there is still limited mathematical understanding of how the solutions to the differential equations behind the fluid-flow approximation are related to the underlying CTMC (see related work below). In a lot of cases, however, fluid-flow analysis appears empirically very promising. We show here, that the deterministic solutions to the ODEs give accurate approximations to discrete model statistics, both in steady state and for transient behaviour as well.

We justify fluid-flow analysis as a first-order moment approximation to the Chapman–Kolmogorov equations which govern the evolution of the underlying CTMC as derived from the PEPA model. Considering the fluid analysis in this manner leads us to develop an extension of the existing approximation scheme to higher order moments. This allows the approximation of key, but up to now inaccessible, higher order quantities such as variance. We also obtain theoretical and empirical estimates of the magnitude of the error terms. Further, in the course of this work, we have expanded the subset of the PEPA language that can be analysed using fluid-flow approximation to include: action hiding; active cooperation between components with differing rates; arbitrary cooperation sets between groups of components; and to allow the presence of more than one component enabling the same action type on either side of a cooperation.

In terms of related work, Bortolussi [19] presents a formulation for the stochastic constraint programming language, sCCP [5], a relative of stochastic $\pi$-calculus with asynchronous communication of variables via a central constraint store. In his paper [20], he presents a first and second-order fluid approximation to internal program variables within the sCCP model. The sCCP fluid approach differs from the fluid approximation of Hillston [16, 18] and the one presented here quite markedly since PEPA is based on CSP [21] in its style of communication and maintains local synchronous communication between components. Also, in this paper, we present a structural transformation from a class of PEPA models to the fluid approximation. This is in contrast to Bortolussi's work which encodes the system parallelism in the program variables and performs fluid approximation on the dynamics of the variables. Finally and distinctly, we present a higher moment formulation (above and beyond the second moment) directly from the PEPA model structure.

Cardelli has previously presented various translations from stochastic process algebras [22, 23], including stochastic $\pi$-calculus and *stochastic interacting processes*, to systems of chemical reactions, and back. Furthermore, these translations are shown to preserve discrete (CTMC-based) semantics and continuous

(ODE-based) semantics. In [24], Cardelli shows that under a natural translation from a subset of CCS to a system of chemical reactions, the dynamics of the underlying CTMCs are preserved by showing that the respective Chapman–Kolmogorov equations are equivalent. However, the relationship between the discrete semantics of the CTMC and continuous semantics of the differential equations is not considered explicitly. In fact, the ODEs generated [24] are an approximation to mean counts of stochastic $\pi$-processes, analogous to the original work for PEPA [16] and no theoretical account of variance or accuracy of resulting analysis is made.

Other recent related work by Geisweiller *et al.* [25] shows that a sequence of discretised CTMCs converges to a first-order fluid-model. They consider a distinct extension of PEPA in the context of biochemistry to include a *mass-action semantics*. Such scenarios consist of modelling the interactions between molecules and these authors define a sequence of CTMCs modelling the biochemical scenario at increasing levels of granularity (with respect to the concentration level of each molecule). They then proceed to show using a theorem of Kurtz [26], that in the limit of increasing granularity, the CTMC solution corresponds with that of a set of ODEs. This is clearly a directly relevant approach for biochemical scenarios. However, the result of Kurtz is a limiting result, and as also noted in [25], no explicit relationship between the solution to the ODE system and a *particular* CTMC is yet known in general. Our work is best viewed as approaching the problem from this perspective: we aim to establish a direct link between the system of ODEs (the fluid model) and the particular underlying CTMC; we also seek to quantify the absolute errors in the fluid model as compared to the transient analysis of the underlying CTMC.

Recently, there has been related work on mean-field approximation of communication systems by Le Boudec *et al.* [27, 28]. The mean-field analysis (MFA) approach is similar to the first moment approximation in this paper. However, in [27] and subsequent publications, the mean-field approach is applied to a discrete-time process in contrast to the continuous-time approach of PEPA. The approach is not perhaps as systematic as one derived from a process algebra, tackling as it does a fixed structure of interaction between groups of cooperating components. Additionally, the MFA approach does not provide access to higher moments of performance measures, as derived here.

Formalisms such as fluid stochastic Petri nets [29] and fluid queueing models [30, 31] incorporate continuous modelling elements explicitly in the initial model. This is commonly used to capture actual continuous components. Alternatively, as in the case of fluid queueing nodes, continuous parameters are often used to approximate large discrete buffer sizes. In such cases, no direct relationship is generally established between the fluid queue model and a purely discrete-state model. However most closely related to our work are heavy traffic analysis techniques, in which sequences of discrete queueing models are shown to converge to a continuous model [32]. By contrast, in our framework we derive approximations to derived moments from process models as opposed to constructing a limiting stochastic process.

The paper is structured as follows. In Section 1.1, we introduce the stochastic process algebra PEPA formally and show how it can be used to model a simple system. In Section 1.2, we present the existing work concerned with fluid analysis of PEPA models, upon which we build later. Then in Section 2, we define a simple modification of the PEPA syntax (a *grouped PEPA model*), which facilitates a far clearer presentation of both the existing work on fluid analysis and our contributions. It also serves to identify structurally the class of PEPA models to which this kind of analysis can naturally be applied. Section 3 presents the existing first order fluid analysis and our extensions to more general PEPA models in the new framework of the grouped PEPA model formalism. Section 4 details the first of our key contributions, where we show how the fluid analysis is an approximation to the first-order moments of the process algebra component counts. We also identify explicitly the simple nature of this approximation for a very large class of models. Following on from this, Section 5 defines a natural extension of the fluid analysis to higher order moments. Finally, Section 6 shows how these techniques can be applied to a more realistic worked example.

### *1.1. Introduction to PEPA*

We begin by introducing PEPA [1], which is a simple stochastic process algebra, but one which has sufficient expressiveness to model a wide variety of systems, including multimedia applications [33], mobile phone

usage [34], GRID scheduling [35], production cell efficiency [36] and web-server clusters [37] amongst others.

As in all process algebras, systems are represented in PEPA as the composition of *components* which undertake *actions*. In PEPA the actions are assumed to have a duration, or delay. Thus the expression $(\alpha, r).P$ denotes a component which can undertake an $\alpha$-action, at rate $r$ to evolve into a component $P$. Here $\alpha \in \mathcal{A}$ where $\mathcal{A}$ is the set of action types and $P \in \mathcal{C}$ where $\mathcal{C}$ is the set of component types. The rate $r$ is interpreted as a random delay sampled from an exponential distribution with parameter $r$. This means that the stochastic behaviour of the model is governed by an underlying continuous-time Markov chain, the explicit definition of which will be given later in this section.

PEPA has a small set of combinators, allowing system descriptions to be built up as the concurrent execution and interaction of simple sequential components. The syntax of the type of PEPA model considered in this paper may be specified formally using the grammar:

$$S ::= (\alpha, r).S \;\mid\; S + S \;\mid\; C_S$$
$$P ::= P \underset{L}{\bowtie} P \;\mid\; P/L \;\mid\; C$$

where $S$ denotes a *sequential component* and $P$ denotes a *model component* which executes in parallel. $C$ stands for a constant which denotes either a sequential component or a model component as introduced by a definition. $C_S$ stands for constants which denote only sequential components. The effect of this syntactic separation between these types of constants is to constrain legal PEPA components to be cooperations of only sequential processes.

The structured operational semantics are shown in Figure 1. A brief discussion of the basic PEPA operators is given below:

**Prefix** The basic mechanism for describing the behaviour of a system with a PEPA model is to give a component a designated first action using the prefix combinator, denoted by a full stop, which was introduced above. As explained, $(\alpha, r).P$ carries out an $\alpha$-action with rate $r$, and it subsequently behaves thereafter as $P$.

**Choice** The component $P + Q$ represents a system which may behave either as $P$ or as $Q$. The activities of both $P$ and $Q$ are enabled. The first activity to complete distinguishes one of them: the other is discarded. The system will behave as the derivative resulting from the evolution of the chosen component.

**Constant** It is convenient to be able to assign names to patterns of behaviour associated with components. Constants are components whose meaning is given by a defining equation. The notation for this is $X \stackrel{def}{=} E$. This also allows the recursive definition of components, for example, $X \stackrel{def}{=} (\alpha, r).X$ performs $\alpha$ at rate $r$ forever.

**Hiding** The possibility to abstract away some aspects of the behaviour of a component is provided by the hiding operator, denoted $P/L$. Here, the set $L$ identifies those activities which are to be considered internal or private to the component and which will appear as the hidden action type $\tau$ in the transition system of the model.

**Cooperation** We write $P \underset{L}{\bowtie} Q$ to denote cooperation between $P$ and $Q$ over $L$. The set which is used as the subscript to the cooperation symbol, the *cooperation set* $L$, determines those activities on which the components are forced to synchronise. For action types not in $L$, the components proceed independently and concurrently with their enabled activities. We write $P \parallel Q$ as an abbreviation for $P \underset{L}{\bowtie} Q$ when $L$ is empty.

If a component enables an activity whose action type is in the cooperation set it will not be able to proceed with that activity until the other component also enables an activity of that type. The two components

**Prefix**

$$\frac{}{(\alpha, r).E \xrightarrow{(\alpha, r)} E}$$

**Competitive Choice**

$$\frac{E \xrightarrow{(\alpha, r)} E'}{E + F \xrightarrow{(\alpha, r)} E'} \qquad \frac{F \xrightarrow{(\alpha, r)} F'}{E + F \xrightarrow{(\alpha, r)} F'}$$

**Cooperation**

$$\frac{E \xrightarrow{(\alpha, r)} E'}{E \underset{S}{\bowtie} F \xrightarrow{(\alpha, r)} E' \underset{S}{\bowtie} F} \quad (\alpha \notin S) \qquad \frac{F \xrightarrow{(\alpha, r)} F'}{E \underset{S}{\bowtie} F \xrightarrow{(\alpha, r)} E \underset{S}{\bowtie} F'} \quad (\alpha \notin S)$$

$$\frac{E \xrightarrow{(\alpha, r_1)} E' \quad F \xrightarrow{(\alpha, r_2)} F'}{E \underset{S}{\bowtie} F \xrightarrow{(\alpha, R)} E' \underset{S}{\bowtie} F'} \quad (\alpha \in S)$$

where $R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E),\, r_\alpha(F))$

**Hiding**

$$\frac{E \xrightarrow{(\alpha, r)} E'}{E/H \xrightarrow{(\alpha, r)} E'/H} \quad (\alpha \notin H) \qquad \frac{E \xrightarrow{(\alpha, r)} E'}{E/H \xrightarrow{(\tau, r)} E'/H} \quad (\alpha \in H)$$

**Constant**

$$\frac{E \xrightarrow{(\alpha, r)} E'}{A \xrightarrow{(\alpha, r)} E'} \quad (A \stackrel{def}{=} E)$$

Fig. 1: PEPA structured operational semantics

then proceed together to complete the *shared activity*. Once enabled, the rate of a shared activity has to be altered to reflect the slower component in a cooperation.

In some cases, when the rate of a shared activity is determined by only one component in the cooperation, then the other component is defined as *passive* with respect to that activity. This means that the rate of the activity is left unspecified (denoted $\top$) and is determined upon cooperation, by the rate of the activity in the other component. All passive actions must be synchronised in the final model.

Within the cooperation framework, PEPA assumes *bounded capacity*: that is, a component cannot be made to perform an activity faster by cooperation, so the rate of a shared activity is the minimum of the apparent rates of the activity in the cooperating components. This is discussed in more detail in [1].

*1.1.1. Apparent Rate*

We define the notion of *apparent rate* as the externally observed rate of activities of a particular type. For a given action type $\alpha \in \mathcal{A}$, it is thus calculated by summing the rates of all enabled activities of this type:

$$r_\alpha(P) := \sum_{P \xrightarrow{(\alpha,\, \lambda_i)}} \lambda_i$$

where $\lambda_i \in \mathbb{R}^+ \cup \{n\top \mid n \in \mathbb{Q}, n > 0\}$, $n\top$ is shorthand for $n \times \top$ and $\top$ represents the passive action rate that inherits the rate of the coaction from the cooperating component. If there are many passive activities of a certain action type enabled within the cooperation, each activity can be individually assigned a weight (defined in the operational semantics of Figure 1) to determine the relative probabilities of the possible outcomes for the various activities of that particular action type. The following algebraic definitions and relations support this weighting and the use of $\top$ in the apparent rate function:

$$\begin{aligned} m\top < n\top &: \text{ for } m < n \text{ and } m, n \in \mathbb{Q} \\ r < n\top &: \text{ for all } r \in \mathbb{R}, n \in \mathbb{Q} \\ m\top + n\top = (m+n)\top &: m, n \in \mathbb{Q} \\ \frac{m\top}{n\top} = \frac{m}{n} &: m, n \in \mathbb{Q} \end{aligned}$$

Note that these algebraic rules leave $(r + w\top)$ undefined for all $r \in \mathbb{R}^+$ and $w \in \mathbb{Q}$, $w \neq 0$. Such components which enable both active and passive actions of the same action type at the same time are therefore disallowed in PEPA. An example might be $(a,\, \lambda).P + (a,\, \top).P'$, where $\lambda > 0$.

Apparent rate can also be defined equivalently in a recursive manner over the PEPA grammar as follows:

$$\begin{aligned} r_\alpha((\beta,\, \lambda).P) &:= \begin{cases} \lambda & \text{if } \beta = \alpha \\ 0 & \text{if } \beta \neq \alpha \end{cases} \\ r_\alpha(P + Q) &:= r_\alpha(P) + r_\alpha(Q) \\ r_\alpha(P/L) &:= \begin{cases} r_\alpha(P) & \text{if } \alpha \notin L \\ 0 & \text{if } \alpha \in L \end{cases} \\ r_\alpha(P \underset{L}{\bowtie} Q) &:= \begin{cases} \min(r_\alpha(P),\, r_\alpha(Q)) & \text{if } \alpha \in L \\ r_\alpha(P) + r_\alpha(Q) & \text{if } \alpha \notin L \end{cases} \end{aligned} \tag{1.1}$$

*1.1.2. Execution strategy*

For a given PEPA component $C$, we define its *derivative set* $ds(C)$ as the set of components reachable from $C$ by evolution according to the operational semantics (Figure 1). That is, $ds(C)$ is the smallest set of components such that $C \in ds(C)$ and if for any $C_1 \in ds(C)$, $C_1 \xrightarrow{(\alpha, r)} C_2$ then $C_2 \in ds(C)$.

For a given PEPA component $C$, we may then naturally construct its *derivation graph*, a labelled and directed multigraph. The nodes of this multigraph are the derivative states of $C$, that is, the set of nodes is $ds(C)$. Two nodes in the multigraph, say $C_1$ and $C_2 \in ds(C)$, have a directed arc between them for every transition $C_1 \xrightarrow{(\alpha, \lambda_i)} C_2$. The label of this arc is then the activity corresponding to the transition, that is, $(\alpha,\, \lambda_i)$.

The derivation graph can then be interpreted naturally as a CTMC, whose states are given by the nodes (i.e. derivative states) and each arc represents a transition at the rate of the activity labelling the arc. We term this the *underlying CTMC* of the model.

This is described in more detail in [1, Chapter 3].

*1.1.3. A simple example*

We consider the ubiquitous situation of many processors running in parallel, but each in regular need of some resource (perhaps for example, communications channels or storage mediums). We model each processor as a $Processor_0$ component and each resource as a $Resource_0$ component. Each processor operates forever in a simple loop, completing two tasks in sequence, $task_1$ and then $task_2$:

$$Processor_0 \stackrel{def}{=} (task_1,\, r_1).Processor_1$$
$$Processor_1 \stackrel{def}{=} (task_2,\, q).Processor_0$$

The resources on the other hand first complete a $task_1$ action also, but then complete a *reset* action:

$$Resource_0 \stackrel{def}{=} (task_1,\, r_2).Resource_1$$
$$Resource_1 \stackrel{def}{=} (reset,\, s).Resource_0$$

The $task_1$ action is a shared action between the processors and resources to model the situation of a processor having to acquire a resource which it needs to complete its first task. The actions $task_2$ and *reset* on the other hand will not be shared, meaning they are completed independently and without synchronisation by the processors and resources respectively. In its simplest instance, with just one processor and resource, this system is defined in PEPA by:

$$System \stackrel{def}{=} Processor_0 \underset{\{task_1\}}{\bowtie} Resource_0$$

Furthermore, we may easily exhibit models with larger numbers of processors and resources, such as:

$$System(3,\, 2) \stackrel{def}{=} (Processor_0 \parallel Processor_0 \parallel Processor_0) \underset{\{task_1\}}{\bowtie} (Resource_0 \parallel Resource_0)$$

That is, three processors running in parallel, competing for resources, of which there are only two available. More generally we might define:

$$System(N_p,\, N_r) \stackrel{def}{=} \underbrace{(Processor_0 \parallel \ldots \parallel Processor_0)}_{N_p} \underset{\{task_1\}}{\bowtie} \underbrace{(Resource_0 \parallel \ldots \parallel Resource_0)}_{N_r}$$

We take this opportunity to introduce a simple syntactic shorthand, rewriting the above as:

$$System(N_p,\, N_r) \stackrel{def}{=} Processor_0[N_p] \underset{\{task_1\}}{\bowtie} Resource_0[N_r]$$

Formally, we define the syntactic equivalence:

$$C[n] := \underbrace{(C \parallel \ldots \parallel C)}_{n}$$

The motivating problem of intractable state space sizes is immediately evident even when considering the very simple model $System(N_p,\, N_r)$. Since there are $N_p$ processor components and $N_r$ resource components, each of which can be in one of two states, the underlying CTMC of even this simple model has $2^{N_p+N_r}$ states, that is, exponential growth in the number of processors and resources. This problem would of course be even more pronounced for models of distributed systems with more realistic levels of detail.

*1.2. Existing work*

This section provides an introduction to the existing work in the field [16, 18]. We also identify the current shortcomings which we go some way towards addressing in this work.

The fluid-flow approach to analysis is not applicable for modelling all types of systems. In the case of stochastic process algebra models, it is restricted to highly symmetric models, in the sense that we have

groups of many homogeneous components acting in parallel. Then instead of tracking the state of each individually, we simply count how many are in each possible state for a given group and construct a continuous approximation to these counters.

In terms of the underlying CTMC, a desire to count the number of components in each state as opposed to tracking the state of every component can be expressed through a natural aggregation of the state space.

### *1.2.1. State space aggregation*

One possible approach to coping with massive state spaces is through state space aggregation. In certain situations, many Markov chain states can be merged into one state to create an *aggregated state space*, with the same stochastic behaviour as the original model. The reduced size of the state space is paid for of course in that the modeller has less information, since it is no longer possible to tell which of the original unaggregated states the model is in, knowing only that it is in a given aggregated state. This state space transformation does not solve the problem of state space explosion in general as we will see, but it exposes clearly the sense in which we will define the fluid analysis in the next section. This is the reason we present it here.

In the case of CTMCs, it is well-known that the stochastic behaviour of the model is preserved if one takes a *lumpable* partition as the aggregated state space with the aggregated rates being the sum of the instantaneous transition rates between members of the partition [38].

**Definition 1.1** (Lumpable partition of a CTMC) *A CTMC with finite state space $\{X_i\}_{i\in I}$ and instantaneous transition rate between state $X_i$ and $X_j$, $q(X_i, X_j)$ is lumpable with respect to the partition $\mathcal{X} = \{X_{[j]}\}_{j\in J}$ if and only if for any $X_{[k]}$, $X_{[l]} \in \mathcal{X}$ and $X_i$, $X_j \in X_{[k]}$ we have:*

$$q(X_i, X_{[l]}) = q(X_j, X_{[l]})$$

*where $q(X_i, X_{[l]})$ is the aggregated transition rate from $X_i$ to all states in $X_{[l]}$, i.e. $q(X_i, X_{[l]}) := \sum_{X_m \in X_{[l]}} q(X_i, X_m)$.*

In the context of PEPA, the *strong equivalence* relation defined on the process algebra naturally produces a lumpable partition on the underlying CTMC of the model [1, Chapter 8]. We repeat this result here in the following definition and proposition.

**Definition 1.2** (Strong equivalence) *For a PEPA component $P$, we denote the conditional transition rate from derivative state $P_i \in ds(P)$ to $P_j \in ds(P)$ via an action type $\alpha$ as $q(P_i, P_j, \alpha)$. This is the sum of the rates of all $\alpha$ transitions between $P_i$ and $P_j$, i.e. $q(P_i, P_j, \alpha) := \sum_{P_i \xrightarrow{(\alpha, \lambda_k)} P_j} \lambda_k$. We then define the total conditional transition rate from $P_i$ to $S \subseteq ds(P)$ a set of derivative states, denoted $q[C_i, S, \alpha] := \sum_{C_j \in S} q(C_i, C_j, \alpha)$.*

*An equivalence relation over derivative states $\mathcal{R} \subseteq ds(P) \times ds(P)$ is a strong equivalence if whenever $(P, Q) \in \mathcal{R}$ then for all $\alpha \in \mathcal{A}$ and for all $S \in ds(P)/\mathcal{R}$:*

$$q[P, S, \alpha] = q[Q, S, \alpha]$$

**Theorem 1.3** *For any PEPA component $P$ and strong equivalence $\mathcal{R} \subseteq ds(P) \times ds(P)$, $ds(P)/\mathcal{R}$ induces a lumpable partition on the state space of the underlying CTMC.*

*Proof.* See [1, Proposition 8.5.1]. □

We now illustrate how we might use strong equivalence to aggregate the state space of a simple PEPA model.

Example 1.4 Aggregation of a simple PEPA model
Consider again the processor/resource model $System(N_p, N_r)$ of Section 1.1.3, for $N_p = N_r = 2$. The first action to be performed, for example, must be a $task_1$, but it can happen in one of four distinct ways (writing $P_0$ for $Processor_0$, $R_0$ for $Resource_0$ and so on):

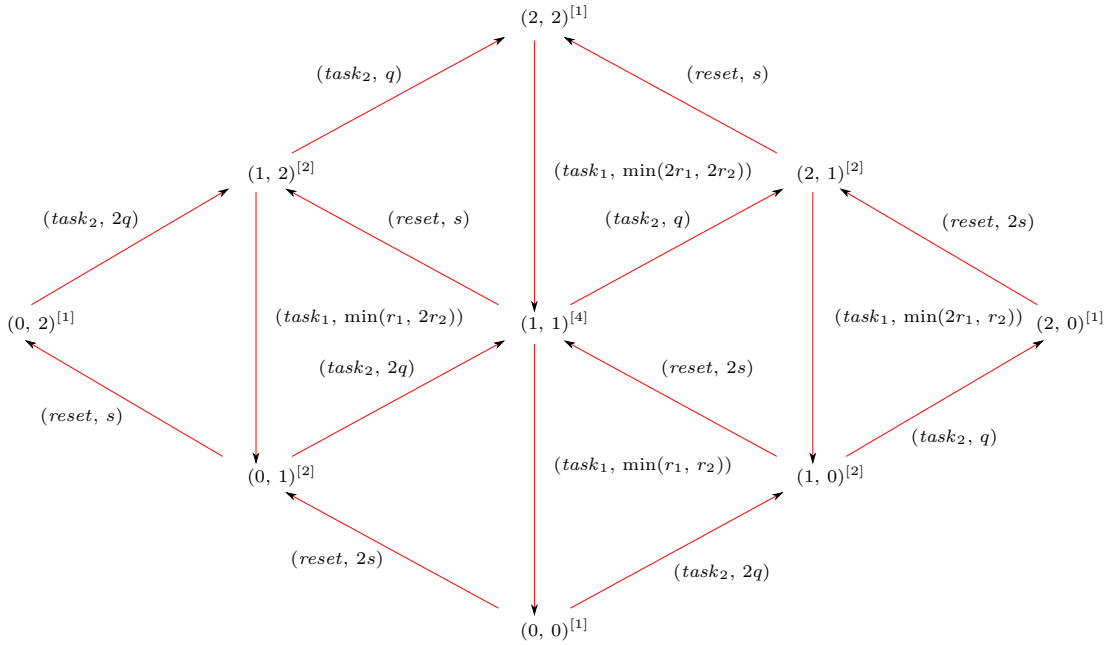



Fig. 2: Aggregated state space of a simple 2-processor/2-resource model.

1. $(P_0 \parallel P_0) \underset{\{task_1\}}{\bowtie} (R_0 \parallel R_0) \xrightarrow{(task_1, \frac{1}{4}\min(2r_1, 2r_2))} (P_1 \parallel P_0) \underset{\{task_1\}}{\bowtie} (R_1 \parallel R_0)$

2. $(P_0 \parallel P_0) \underset{\{task_1\}}{\bowtie} (R_0 \parallel R_0) \xrightarrow{(task_1, \frac{1}{4}\min(2r_1, 2r_2))} (P_1 \parallel P_0) \underset{\{task_1\}}{\bowtie} (R_0 \parallel R_1)$

3. $(P_0 \parallel P_0) \underset{\{task_1\}}{\bowtie} (R_0 \parallel R_0) \xrightarrow{(task_1, \frac{1}{4}\min(2r_1, 2r_2))} (P_0 \parallel P_1) \underset{\{task_1\}}{\bowtie} (R_1 \parallel R_0)$

4. $(P_0 \parallel P_0) \underset{\{task_1\}}{\bowtie} (R_0 \parallel R_0) \xrightarrow{(task_1, \frac{1}{4}\min(2r_1, 2r_2))} (P_0 \parallel P_1) \underset{\{task_1\}}{\bowtie} (R_0 \parallel R_1)$

Note that the PEPA components on the right-hand sides, above, all have the same number of $P_0$ and $R_0$ components, perhaps just occurring in different orders. If we were interested in counting just the number of each component on each side of the cooperation, we might thus be wise to aggregate all of the states appearing on the right-hand side of the above transitions, summing the rates to give an aggregate rate of $\min(2r_1, 2r_2)$ in this particular case.

Indeed, if we define a relation on the derivative states of the original model $System(2, 2)$ by defining two derivative states to be in this relation if and only if they have the same number of $P_0$ and $R_0$ components (or equivalently the same number of $P_1$ and $R_1$ components), this is clearly an equivalence relation. It can also be shown that it is a strong equivalence, which we do in a general setting in Section 2.

A state in the resulting aggregated state space is uniquely identified by the integer count of $P_0$ and of $R_0$ components, that is, by two integer variables each with range from 0 to 2. Figure 2 shows the full aggregated state space where $(X, Y)^{[s]}$ represents the aggregated state of $X \times P_0$ components and $Y \times R_0$ components. The superscript indicates how many states in the original state space have been merged into this particular aggregated state. This style of labelling is termed *numerical vector form* by Gilmore *et al.* [15]. □

The above aggregation reduces a model of $2^4 = 16$ states to one with $3^2 = 9$ states. In exchange for this reduction in complexity however we lose the ability to track the state of individual components, rather we are only able to track the number of individuals in a particular component derivative state. In general, for

large component counts, this style of aggregation results in state spaces that experience exponential growth in the number of derivative states of the individual components rather than the number of components operating together in parallel in each group. It is thus still very easy to construct modest models with massive aggregated state spaces. Consider a more realistic model of processors and resources each with 10 derivative states and 20 copies of each. The original state space would have of the order of $10^{40}$ states, which is massively intractable to traditional techniques. The aggregated state space would be of the order of:

$$\left(\frac{(10+20-1)!}{20!(10-1)!}\right)^2 = \left(\frac{29!}{20!9!}\right)^2 = 10015005^2 \approx 10^{14} \text{ states}$$

a significant improvement, but still out of the reach of traditional analysis techniques.

It is clear that the technique of state space aggregation for groups of identical components cooperating in parallel presented here is not sufficient alone to eliminate the state space explosion problem. It merely exchanges one form of exponential growth for another. However, as we have discussed, when viewed as a transformation of the state space, it is the first step in defining the fluid approximation.

It is worth considering by how much this aggregation alone actually reduces the size of the state space. Indeed, consider a single component group consisting of $N$ identical components in parallel, each with $D$ derivative states. This is the best-case scenario for this type of lumping aggregation, where there is going to be the potential for the greatest reduction in state-space. The size of its unaggregated state space is trivially $D^N$ states. The size of the aggregated numerical vector form state space can be calculated by considering how many ways there are of choosing $N$ elements from $D$ possible types with replacement where order is unimportant, that is $\frac{(D+N-1)!}{N!(D-1)!}$ states.

Clearly this is a substantial improvement on the explicit state space, and goes some distance towards addressing the state space explosion problem for such models. Having said this, such a state space can still be far too large for standard CTMC analysis techniques. Indeed, assuming $D \ll N$ then:

$$\frac{(D+N-1)!}{N!(D-1)!} = \frac{(D+N-1)(D+N-2)\ldots(N+1)}{(D-1)!} > \frac{N^{D-1}}{(D-1)!} \sim N^{D-1}$$

so it is clear this aggregation of the state space is insufficient alone if we wish to be able to analyse models with reasonably large component groups.

### *1.2.2. Fluid-flow analysis with ODEs*

The style of aggregation introduced in the previous section introduces explicit integer counters into the state space, which can be subject to a fluid-flow approximation. Indeed, it has been shown how systems of coupled first-order ODEs can be derived in terms of these counters (approximated by real variables) intuitively from such a PEPA model [16, 18]. We illustrate this by means of a simple example.

Example 1.5 Fluid analysis of a simple PEPA model
Consider again the simple model $System(N_p, N_r)$ introduced in Section 1.1.3 and its aggregated state space (as shown in Figure 2). Define integer-valued stochastic processes to count the number of each of the $Processor_0$, $Processor_1$, $Resource_0$ and $Resource_1$ components present at a time $t$, say $N_{P_0}(t)$, $N_{P_1}(t)$, $N_{R_0}(t)$ and $N_{R_1}(t)$ respectively. Note that their values fully determine the state at time $t$ of the aggregated CTMC. For each component, consider the rate of transitions which create new copies of that component and the rate of transitions which lose copies of that component. For example, if we consider $Processor_0$ components at time $t$, we see that the sum of all of the transitions contributing $Processor_0$ components has rate $r_2 N_{P_1}(t)$, but on the other hand, we lose $Processor_0$ components at rate $\min(r_1 N_{P_0}(t), r_2 N_{R_0}(t))$. Very similar considerations can be made for the other components. If we treat these integer-valued stochastic processes instead as deterministic, real-valued functions, say $v_{P_0}(t)$, $v_{P_1}(t)$, $v_{R_0}(t)$ and $v_{R_1}(t)$ respectively, we can intuitively construct a system of first-order ODEs. We will do this by considering for each component,

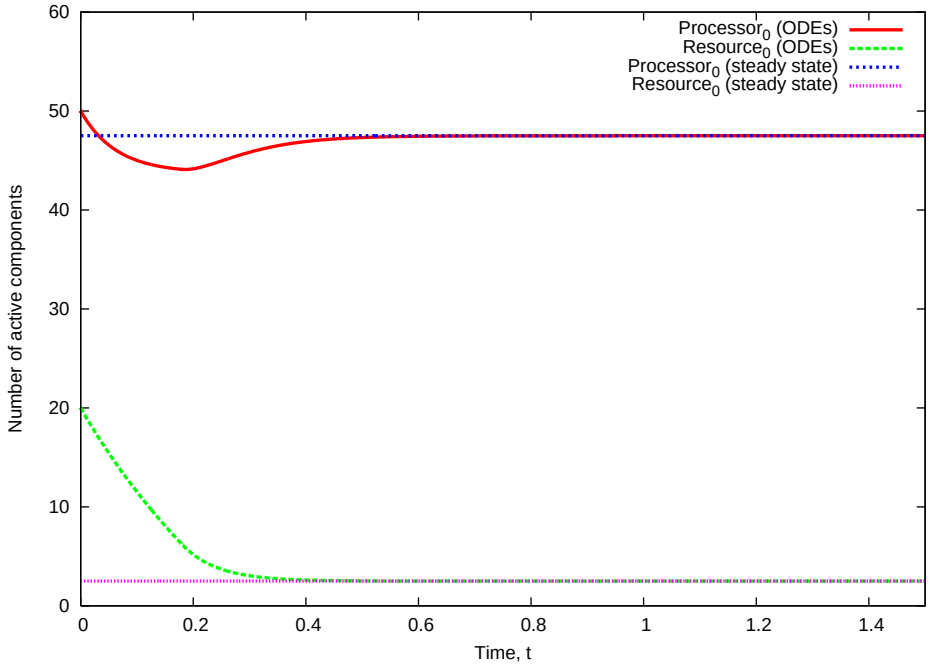



Fig. 3: Comparison of ODE solutions with steady state solution of the underlying CTMC for simple processor/resource model. Rates used are $r_1 = 2.0$, $r_2 = 14.0$, $q = 14.0$ and $s = 2.0$. Initial conditions are 50 $Processor_0$ and 20 $Resource_0$ components.

the rate at which new components "enter the model" balanced against that at which they "leave the model":

$$\begin{aligned}
\dot{v}_{P_0}(t) &= -\min(r_1 v_{P_0}(t),\, r_2 v_{R_0}(t)) + q v_{P_1}(t) \\
\dot{v}_{P_1}(t) &= \min(r_1 v_{P_0}(t),\, r_2 v_{R_0}(t)) - q v_{P_1}(t) \\
\dot{v}_{R_0}(t) &= -\min(r_1 v_{P_0}(t),\, r_2 v_{R_0}(t)) + s v_{R_1}(t) \\
\dot{v}_{R_1}(t) &= \min(r_1 v_{P_0}(t),\, r_2 v_{R_0}(t)) - s v_{R_1}(t)
\end{aligned}$$

Figure 3 compares the ODE solution with the steady state solution of the underlying aggregated CTMC obtained through the usual methods.

We see that at least in the (deterministic) limit of the steady state, the results appear to agree (for this set of rates and initial conditions). □

The class of PEPA models considered by [18] is fairly general but includes some types of models which are not obviously "fluid-like", leading to systems of ODEs with discontinuous right-hand sides (and thus no guaranteed global solution). Furthermore, no mathematical relation of the ODEs to the original aggregated CTMC is given, although empirically, it would appear that there is a quantitative relationship between the ODE solutions and the expected value of the corresponding counting stochastic processes, at least for a class of PEPA models.

The first contribution of this paper is to identify syntactically this class of naturally "fluid-like" models, upon which we should focus initially. In particular, we define the notion of a *grouped PEPA model*. Within this new framework we present both the existing fluid analysis defined by [16, 18] and extend it to handle also action hiding, active cooperation between components with differing rates, arbitrary cooperation sets

between groups of components and to allow the presence of more than one component enabling the same action type on either side of a cooperation.

Furthermore, we will see also that the grouped PEPA model abstraction makes the clear presentation of the contributions of the latter part of this paper possible.

## 2. Grouped PEPA models

As discussed, the techniques of fluid analysis introduced are suited naturally to a particular class of PEPA model, consisting of cooperating groups of similar components. In order to allow such groups to be identified in the syntax explicitly, we present a natural augmentation of the PEPA grammar, instances of which we call *grouped PEPA models*. Grouped PEPA models are a conservative extension of standard PEPA models.

We maintain the standard PEPA component definitions from Section 1.1:

$$S ::= (\alpha, r).S \;\mid\; S + S \;\mid\; C_S$$
$$P ::= P \underset{L}{\bowtie} P \;\mid\; P/L \;\mid\; C$$

When we wish to make clear that we are referring to a standard PEPA component as opposed to a grouped PEPA model, we will say explicitly *standard PEPA component*. We wish to define a new type of component that explicitly represents a purely concurrent group of standard PEPA components. To this end we define a *component group*, $D$:

$$D ::= D \parallel D \;\mid\; P \tag{2.1}$$

where $P$ is a standard PEPA component as defined above. The purpose of introducing this extra hierarchy is to specify syntactically the level at which the fluid analysis is to be performed, that is, at the level of component groups. Cooperation within components groups is not considered since such models do not have an obvious fluid interpretation.[1]

The fluid semantics for passive cooperation presented in [18] suffered from a related problem.[2] Therefore, we exclude passive cooperation by restricting the types of standard PEPA components we will allow in component groups to those not enabling any action passively in any of their derivative states. Thus we re-write Equation (2.1):

$$D ::= D \parallel D \;\mid\; P \qquad \forall P' \in ds(P) \quad \nexists(\alpha,\, r) \in \mathcal{A}ct(P') \text{ such that } r \notin \mathbb{R}^+ \tag{2.2}$$

The next definition allows the arbitrary combination of component groups into a *grouped PEPA model*, $M$:

$$M ::= M \underset{L}{\bowtie} M \;\mid\; M/L \;\mid\; Y\{D\} \tag{2.3}$$

where $L$ is a cooperation set. A grouped PEPA model consists of arbitrarily cooperating component groups (instances of $Y\{D\}$, called *labelled component groups*). $Y$ is a component group label drawn from some sufficiently large label set, the purpose of which is to provide a label uniquely identifying component groups. To this end, each labelled component group in a given model is assumed to have a distinct label.

We might therefore represent the model $System(N_p,\, N_r)$ of Section 1.1.3 as the grouped PEPA model $System_G(N_p,\, N_r)$:

$$\mathbf{Processors}\{Processor_0[N_p]\} \underset{\{task_1\}}{\bowtie} \mathbf{Resources}\{Resource_0[N_r]\}$$

[1] Indeed if the programme presented here is naturally extended to such systems, the resulting ODE systems will have jumps in their right-hand sides and generally deliver meaningless quantitative results, if indeed they even have a solution at all.

[2] However, a significant improvement on this was suggested in [39], whereby passive cooperation is replaced by an equivalent active cooperation. This approach could be adopted here, in which case this work applies directly.

In this particular case, the two component groups (identified by the labels **Processors** and **Resources**) specify that the fluid analysis will happen at the level of $Processor_0$, $Processor_1$, $Resource_0$ and $Resource_1$ components. That is, these are the four derivative states we will count copies of; there will be one ordinary differential equation defined for each of these four component states.

We now exhibit the natural standard PEPA model to which a grouped PEPA model corresponds. A grouped PEPA model should be thought of simply as the corresponding standard PEPA model, but with additional annotations (the group labellings), which specify unambiguously the level at which the fluid analysis should be performed. Accordingly, many definitions for grouped PEPA models can be made directly in terms of the corresponding standard PEPA model.

**Definition 2.1** (Model flattening function) *For any grouped PEPA model $G$, the corresponding standard PEPA model, $\mathcal{F}(G)$, can be recovered from the grouped model. $\mathcal{F}(\cdot)$ is defined as:*

$$\begin{aligned}\mathcal{F}(M_1 \underset{L}{\bowtie} M_2) &:= \mathcal{F}(M_1) \underset{L}{\bowtie} \mathcal{F}(M_2)\\ \mathcal{F}(M/L) &:= \mathcal{F}(M)/L\\ \mathcal{F}(Y\{D\}) &:= \mathcal{F}'(D)\end{aligned}$$

*where for component groups:*

$$\begin{aligned}\mathcal{F}'(D_1 \parallel D_2) &:= \mathcal{F}'(D_1) \parallel \mathcal{F}'(D_2)\\ \mathcal{F}'(P) &:= P\end{aligned}$$

Now in terms of its corresponding standard PEPA model, we can define the notion of *apparent rate* for grouped PEPA models using the flattening function.[3]

**Definition 2.2** (Apparent rate of a grouped PEPA model) *For any grouped PEPA model $G$, its apparent rate for action type $\alpha \in \mathcal{A}$ is:*

$$r_\alpha(G) := r_\alpha(\mathcal{F}(G))$$

As we will see, in generating a fluid model systematically from a PEPA model, we need to be able to extract component generated actions from the top-level PEPA model. This will require us to know which action type a silent $\tau$-action was before it was hidden. The following will augment the existing silent action with an originating action type to allow this to happen.

In the case of standard PEPA components, every evolution will either have a non hidden action type $\beta \neq \tau$ or the hidden action type $\tau$. The hidden action type arises either due to an application of action hiding or because an activity of type $\tau$ was explicitly encoded into the model. We extend the set of action types to consist of the union of *normal action types* (simply the set of action types, except $\tau$, under the original PEPA definition, i.e. $\mathcal{A}\backslash\{\tau\}$) and *hidden action types*, which is the new set of action types $\{\tau^\beta : \beta \in \mathcal{A}\backslash\{\tau\}\}\cup\{\tau\}$. $\tau^\beta$ is the new action type of an evolution which originated locally as a $\beta$-action but was later hidden so that it became a global $\tau$-action. The $\tau$-action type still exists to represent evolution of $\tau$-actions which were explicitly encoded in the grouped PEPA model and did not originate through action hiding. We define an *extended action type* to be an element of the union of normal action types and hidden action types and denote this extended set $\mathcal{A}^\tau$. In keeping with the intended semantics of hidden actions, cooperation sets can only include normal action types (actions in $\mathcal{A} \setminus \{\tau\}$). Furthermore, we do not allow any action types of the form $\tau^\beta$ to be encoded explicitly in a model, they may only arise through action hiding. The exact role of extended action types will be formalised when the operational semantics for grouped PEPA models is presented shortly.

[3] We overload the syntax when equivalent definitions are effectively the same for standard PEPA models and grouped PEPA models, modulo the explicit presence of component groups.

**Grouped model cooperation**

$$\frac{M_1 \xrightarrow{(\alpha, r)} M_1'}{M_1 \underset{S}{\bowtie} M_2 \xrightarrow{(\alpha, r)} M_1' \underset{S}{\bowtie} M_2} \quad (\alpha \notin S) \qquad \frac{M_2 \xrightarrow{(\alpha, r)} M_2'}{M_1 \underset{S}{\bowtie} M_2 \xrightarrow{(\alpha, r)} M_1 \underset{S}{\bowtie} M_2'} \quad (\alpha \notin S)$$

$$\frac{M_1 \xrightarrow{(\alpha, r_1)} M_1' \quad M_2 \xrightarrow{(\alpha, r_2)} M_2'}{M_1 \underset{S}{\bowtie} M_2 \xrightarrow{(\alpha, R)} M_1' \underset{S}{\bowtie} M_2'} \quad (\alpha \in S)$$

where $R = \frac{r_1}{r_\alpha(M_1)} \frac{r_2}{r_\alpha(M_2)} \min(r_\alpha(M_1), r_\alpha(M_2))$

**Grouped model hiding**

$$\frac{M \xrightarrow{(\alpha, r)} M'}{M/L \xrightarrow{(\alpha, r)} M'/L} \quad (\alpha \notin L) \qquad \frac{M \xrightarrow{(\beta, r)} M'}{M/L \xrightarrow{(\tau^\beta, r)} M'/L} \quad (\beta \in L)$$

**Labelled component group**

$$\frac{D \xrightarrow{(\alpha, r)} D'}{Y\{D\} \xrightarrow{(\alpha, r)} Y\{D'\}}$$

**Component group**

$$\frac{D_1 \xrightarrow{(\alpha, r)} D_1'}{D_1 \,||\, D_2 \xrightarrow{(\alpha, r)} D_1' \,||\, D_2}$$

$$\frac{D_2 \xrightarrow{(\alpha, r)} D_2'}{D_1 \,||\, D_2 \xrightarrow{(\alpha, r)} D_1 \,||\, D_2'}$$

Fig. 4: Grouped PEPA structured operational semantics

We also define the utility function $\boldsymbol{t} : \mathcal{A}^\tau \to \mathcal{A}$ which extracts the associated normal action type from an extended action type, i.e. for normal action types $\beta$, $\boldsymbol{t}(\beta) := \beta$, but for hidden action types $\tau^\beta$, $\boldsymbol{t}(\tau^\beta) := \beta$ and $\boldsymbol{t}(\tau) := \tau$.

$\mathcal{F}(G)$ provides information regarding how a grouped PEPA model $G$ evolves, but only by first losing the explicit definition of component groups and appealing to standard PEPA semantics. We of course wish to preserve the explicit nature of component groups in the evolution of grouped models. We therefore extend the PEPA operational semantics of Figure 1 to grouped PEPA models explicitly in Figure 4. The operational semantics for standard PEPA components are not repeated and remain unaltered.

In terms of the operational semantics, we can now define the *activity multiset* of a grouped PEPA model analogously to the standard PEPA case.

**Definition 2.3** (Activity multiset of a grouped PEPA model) *For any grouped PEPA model $G$, its activity multiset is* $\mathcal{A}ct(G) := \{\!|(\alpha, r) : G \xrightarrow{(\alpha, r)} |\!\}$.

The following theorem asserts formally our intention that a grouped PEPA model behaves exactly like the corresponding standard PEPA model.

**Theorem 2.4** *Let $G$ be a grouped PEPA model. Then:*

1. *If $\beta \in \mathcal{A} \setminus \{\tau\}$, transitions $G \xrightarrow{(\beta, r)} G'$ are in one-to-one correspondence with transitions $\mathcal{F}(G) \xrightarrow{(\beta, r)} \mathcal{F}(G')$,*

2. *Transitions $G \xrightarrow{(\tau^{\beta}, r)} G'$ with $\beta \in \mathcal{A}$ are in one-to-one correspondence with transitions $\mathcal{F}(G) \xrightarrow{(\tau, r)} \mathcal{F}(G')$.*

*Proof.* Follows trivially by comparing Figures 1 and 4. □

*2.1. Properties of grouped PEPA models*

Some useful properties of grouped PEPA models are now formalised. To aid the reader, we first present a table giving informal definitions for the notation which is used heavily in later sections.

| | |
|---|---|
| $\mathcal{G}(G)$ | The set of all component group labels in the grouped PEPA model $G$ |
| $\mathcal{B}(G, H)$ | The set of all standard PEPA component states in the component group of $G$ which has group label $H$ |
| $\mathcal{B}(G)$ | The set of all pairs whose first element is a component group label and whose second is a standard PEPA component in the group specified by that label |
| $\mathcal{C}(G, H, Q)$ | The integer count of standard PEPA components in state $Q$ in the component group of $G$ which has group label $H$ |
| $\mathcal{V}(G, H)$ | The set of action types performed locally by a standard PEPA component in group $H$, which are globally hidden in $G$ |

As stated in Section 2, group labels uniquely identify a given parallel grouping of components within a grouped PEPA model. So in all the formal definitions below, for a given group $H \in \mathcal{G}(G)$, the group $H$ can only occur once in the model $G$.

**Definition 2.5** (Set of component group labels) *For any grouped PEPA model $G$, its set of component group labels is $\mathcal{G}(G)$ where $\mathcal{G}(\cdot)$ is defined as:*

$$\begin{aligned} \mathcal{G}(M_1 \underset{L}{\bowtie} M_2) &:= \mathcal{G}(M_1) \cup \mathcal{G}(M_2) \\ \mathcal{G}(M/L) &:= \mathcal{G}(M) \\ \mathcal{G}(Y\{D\}) &:= Y \end{aligned}$$

**Definition 2.6** (Group visibility function) *Let $G$ be a grouped PEPA model and let $H \in \mathcal{G}(G)$ be a component group label. The* group visibility function *$\mathcal{V}(G, H)$ returns the set of normal action types which will be eventually hidden if performed by a standard PEPA component in group $H$ in the context of the system, $G$. The function $\mathcal{V}(\cdot, \cdot)$ is defined over the grouped PEPA syntax as follows:*

$$\begin{aligned} \mathcal{V}(M_1 \underset{L}{\bowtie} M_2, H) &:= \mathcal{V}(M_1, H) \cup \mathcal{V}(M_2, H) \\ \mathcal{V}(M/L, H) &:= \begin{cases} L \cup \mathcal{V}(M, H) & \text{if } H \in \mathcal{G}(M) \\ \emptyset & \text{if } H \notin \mathcal{G}(M) \end{cases} \\ \mathcal{V}(Y\{D\}, H) &:= \emptyset \end{aligned}$$

**Definition 2.7** (Standard PEPA derivative states in a component group) *For any grouped PEPA model $G$, $\mathcal{B}(G, H)$ is the set of standard PEPA component derivative states in a component group, $H \in \mathcal{G}(G)$. The function $\mathcal{B}(\cdot, \cdot)$ is defined over the grouped PEPA syntax as follows:*

$$\begin{aligned}
\mathcal{B}(M_1 \underset{L}{\bowtie} M_2, H) &:= \mathcal{B}(M_1, H) \cup \mathcal{B}(M_2, H) \\
\mathcal{B}(M/L, H) &:= \mathcal{B}(M, H) \\
\mathcal{B}(Y\{D\}, H) &:= \begin{cases} \mathcal{B}'(D) & \text{if } Y = H \\ \emptyset & \text{if } Y \neq H \end{cases}
\end{aligned}$$

*where for component groups:*

$$\begin{aligned}
\mathcal{B}'(D_1 \parallel D_2) &:= \mathcal{B}'(D_1) \cup \mathcal{B}'(D_2) \\
\mathcal{B}'(P) &:= ds(P)
\end{aligned}$$

Furthermore define $\mathcal{B}(G)$ to be the subset of $\mathcal{G}(G) \times \bigcup_{H_i \in \mathcal{G}(G)} \mathcal{B}(G, H_i)$ such that $(H, P) \in \mathcal{B}(G)$ if and only if $H \in \mathcal{G}(G)$ and $P \in \mathcal{B}(G, H)$. That is, there is exactly one element of $\mathcal{B}(G)$ for every standard PEPA component and group in which it occurs in the model. This allows us to specify the standard PEPA components of a particular type occurring in a given component group.

The set of derivative states of a grouped PEPA model $G$, written as standard PEPA model components is easily seen to be $ds(\mathcal{F}(G))$. However, we will need to express the derivative states of a grouped PEPA model themselves as grouped PEPA model components. The next definition achieves this.

**Definition 2.8** (Grouped derivative states) *The set of derivative states of a grouped PEPA model $G$ is $ds(G)$, defined as the smallest set of grouped PEPA model components such that $G \in ds(G)$ and if $G_1 \in ds(G)$ and $G_1 \xrightarrow{(\alpha, r)} G_2$ then $G_2 \in ds(G)$.*

It is clear by virtue of Theorem 2.4 that this definition is identical to that for standard PEPA components (Section 1.1.2) apart from the explicit identification of component groups through their labels. As in the case of standard PEPA (Section 1.1.2), the underlying CTMC of a grouped PEPA model is generated directly from the derivation graph induced by Definition 2.8. Theorem 2.4 tells us that as expected, the underlying CTMC of a grouped PEPA model is trivially isomorphic to that of the corresponding standard PEPA model, i.e. they have identical stochastic behaviour as is our intention.

We now define the *component counting function*. For a given grouped PEPA model $G$, this function takes a component group and a standard PEPA component. It returns the number of standard PEPA components of a particular type currently active in the given group.

**Definition 2.9** (Component counting function) *For any grouped PEPA model $G$ and group label-state pair $(H, Q) \in \mathcal{B}(G)$, the component counting function $\mathcal{C}(G, H, Q)$ counts the number of component members of group $H$ that are in state $Q$. The function $\mathcal{C}(\cdot, \cdot, \cdot)$ is defined over the grouped PEPA syntax as follows:*

$$\begin{aligned}
\mathcal{C}(M_1 \underset{L}{\bowtie} M_2, H, Q) &:= \mathcal{C}(M_1, H, Q) + \mathcal{C}(M_2, H, Q) \\
\mathcal{C}(M/L, H, Q) &:= \mathcal{C}(M, H, Q) \\
\mathcal{C}(Y\{D\}, H, Q) &:= \begin{cases} \mathcal{C}'(D, Q) & \text{if } Y = H \\ 0 & \text{if } Y \neq H \end{cases}
\end{aligned}$$

*where for component groups:*

$$\begin{aligned}
\mathcal{C}'(D_1 \parallel D_2, Q) &:= \mathcal{C}'(D_1, Q) + \mathcal{C}'(D_2, Q) \\
\mathcal{C}'(P, Q) &:= \begin{cases} 1 & \text{if } P = Q \text{ as standard PEPA components} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

It is quite clear from the operational semantics in Figure 4 that as a grouped model evolves, each component group maintains its fixed size in the sense of the number of standard PEPA components it contains. It is therefore easy to see that the component counting function will always be valued between zero and the size of the given component group for any standard PEPA component derivative state in that group.

We now use the component counting function to aggregate a grouped PEPA model state space, combining states where the component counts agree. We begin by defining the relation *groupwise equivalence* on the derivative states of grouped PEPA models.

**Definition 2.10** (Groupwise equivalence) *Let $G$ be a grouped PEPA model. Define $G_1, G_2 \in ds(G)$ to be groupwise equivalent, $G_1 \simeq_G G_2$, if and only if for all $(H, P) \in \mathcal{B}(G)$, we have $\mathcal{C}(G_1, H, P) = \mathcal{C}(G_2, H, P)$.*

**Theorem 2.11** *$\simeq_G$ is an equivalence relation, i.e. it induces a partition $ds(G)/\simeq_G$ on the state space of a grouped PEPA model $G$.*

*Proof.* This is immediate from Definition 2.9. □

For example, in the case of $G = System_G(N_p, N_r)$, writing $P_0$ for $Processor_0$, $R_0$ for $Resource_0$ and so on, one equivalence class $\tilde{G} \in ds(G)/\simeq_G$ is:

$$\begin{aligned}\tilde{G} = \{&\mathbf{Processors}\{P_0 \parallel P_1[N_p - 1]\} \underset{\{task_1\}}{\bowtie} \mathbf{Resources}\{R_0[N_r]\},\\ &\mathbf{Processors}\{P_1 \parallel P_0 \parallel P_1[N_p - 2]\} \underset{\{task_1\}}{\bowtie} \mathbf{Resources}\{R_0[N_r]\},\\ &\mathbf{Processors}\{P_1[2] \parallel P_0 \parallel P_1[N_p - 3]\} \underset{\{task_1\}}{\bowtie} \mathbf{Resources}\{R_0[N_r]\},\\ &\ldots,\\ &\mathbf{Processors}\{P_1[N_p - 1] \parallel P_0\} \underset{\{task_1\}}{\bowtie} \mathbf{Resources}\{R_0[N_r]\}\}\end{aligned}$$

This is the case of one $P_0$ component (and thus $(N_p - 1) \times P_1$ components) and $N_r \times R_0$ components (and thus no $R_1$ components).

Now for a general grouped PEPA model $G$ and any $\tilde{G} \in ds(G)/\simeq_G$, we will find it useful to define the component counting function on equivalence classes, that is, $\tilde{\mathcal{C}}(\tilde{G}, H, P) := \mathcal{C}(\overline{G}, H, P)$ for some representative $\overline{G} \in \tilde{G}$. This is trivially well-defined by definition of the equivalence relation $\simeq_G$.

We wish now to prove that $\simeq_G$ is a strong equivalence. The definition we gave for strong equivalence was in terms of standard PEPA components (Definition 1.2), but it adapts naturally to grouped PEPA models: define for extended action type $\alpha$ and $G_1, G_2 \in ds(G)$, $q(G_1, G_2, \alpha)$ as the total rate of $\alpha$-transitions between derivative states $G_1$ and $G_2$, analogously to the standard PEPA case.

**Theorem 2.12** *$\simeq_G$ is a strong equivalence.*

*Proof.* Let $G$ be a grouped PEPA model with $G_1 \simeq_G G_2$ for $G_1$ and $G_2 \in ds(G)$. Let $\alpha$ be an extended action type and $S \in ds(G)/\simeq_G$. We require to show that $q[G_1, S, \alpha] = q[G_2, S, \alpha]$.

Now $G_1$ and $G_2$ have by definition the same number of each standard PEPA component in each of their component groups. Thus they potentially differ only in that these standard PEPA components might be arranged in a different order within the component groups. Compare for example $(P_1 \parallel P_1 \parallel P_3 \parallel P_2)$ with $(P_1 \parallel P_2 \parallel P_1 \parallel P_3)$.

It is clear from the operational semantics (Figure 4) that transitions $G_1 \xrightarrow{(\alpha, r_1)} G_1'$ where $G_1' \in S$ are in one-to-one correspondence with transitions $G_2 \xrightarrow{(\alpha, r_2)} G_2'$ for some $G_2' \in S$, where $G_2'$ will differ from $G_1'$ in the same fashion as $G_2$ differs from $G_1$, i.e. their component counts will remain the same (so they must both be in $S$). Furthermore, $r_2 = r_1$ since it is clear from the operational semantics that the order of a parallel cooperation does not alter the rate of transitions. □

Therefore $\simeq_G$ induces a lumpable partition $ds(G)/\simeq_G$ on the state space of the underlying CTMC by Theorem 1.3. We refer to this CTMC as the *underlying aggregated CTMC* of the grouped PEPA model $G$.

We have shown that we may aggregate the state space of a grouped PEPA model according to the count of components within component groups that are in a particular derivative state. Thus we do not have to explicitly track the evolution of individual members of component groups. States in our aggregated CTMC are uniquely specified by component counts, so a fluid approximation to the component counting stochastic processes will yield an approximation to the state of the aggregated CTMC.

### *2.2. Evolution rates of grouped PEPA models*

In this section, we present the fluid translation for PEPA models using the grouped PEPA model framework. We will introduce the following key rate and probability functions based on grouped PEPA model evolution.

| | |
|---|---|
| $\mathcal{R}_\alpha(G, E, H, P)$ | The **component rate function** measures the local rate at which component state $P$ in group $H$ performs an $\alpha$ action in the context of the cooperation within the wider grouped PEPA model $G$ (using counting function $E$). |
| $p_\alpha(P, Q)$ | The **derivative weighting function** measures the probability that component $P$ evolves to component $Q$ in one $\alpha$-transition. |
| $r_\alpha(G, E)$ | The **count-oriented apparent rate function** measures the total rate of $\alpha$ being produced by grouped model $G$ (using counting function $E$). |

It is important to know the rates associated with the aggregated CTMC we defined in the previous section. To this end, we define the *component rate function* which calculates the rate of evolution of a local action as experienced by a given component type, i.e. it is used to describe the aggregate rate at which clusters of the same component type evolve into their respective derivatives.

**Definition 2.13** (Component rate function) *Let $G$ be a grouped PEPA model. Let $\tilde{G} \in ds(G)/\simeq_G$ be a groupwise equivalence class of $G$ and let $\overline{G} \in \tilde{G}$ be a representative of the equivalence class. Then for $(H, P) \in \mathcal{B}(G)$ and extended action type $\alpha \in \mathcal{A}^\tau$, the component rate for $\alpha$ of members in state $P$ of component group $H$ of $\tilde{G}$ is $\tilde{\mathcal{R}}_\alpha(\tilde{G}, H, P) := \mathcal{R}_\alpha(\overline{G}, H, P)$ where $\mathcal{R}_\cdot(\cdot, \cdot, \cdot)$ is defined as:*

$$\mathcal{R}_\alpha(M_1 \underset{L}{\bowtie} M_2, H, P) := \begin{cases} \frac{\mathcal{R}_\alpha(M_i, H, P)}{r_\alpha(M_i)} \min(r_\alpha(M_1), r_\alpha(M_2)) & \text{if } H \in \mathcal{G}(M_i) \text{ and } \alpha \in L, \text{ for } i = 1 \text{ or } 2 \\ \mathcal{R}_\alpha(M_i, H, P) & \text{if } H \in \mathcal{G}(M_i) \text{ and } \alpha \notin L, \text{ for } i = 1 \text{ or } 2 \end{cases}$$

$$\mathcal{R}_\alpha(M/L, H, P) := \begin{cases} 0 & \text{if } \alpha \in L \text{ (and } \alpha \in \mathcal{A}) \\ \mathcal{R}_\alpha(M, H, P) & \text{if } \alpha \notin L \text{ and } \alpha \in \mathcal{A} \\ \mathcal{R}_{\tau^\beta}(M, H, P) & \text{if } \alpha = \tau^\beta \text{ and } \beta \notin L \\ \mathcal{R}_{\tau^\beta}(M, H, P) + \mathcal{R}_\beta(M, H, P) & \text{if } \alpha = \tau^\beta \text{ and } \beta \in L \end{cases}$$

$$\mathcal{R}_\alpha(Y\{D\}, H, P) := \begin{cases} \mathcal{C}(Y\{D\}, H, P)\, r_\alpha(P) & \text{if } H = Y \\ 0 & \text{if } H \neq Y \end{cases}$$

*The terms of the form $\frac{\mathcal{R}_\alpha(M_i, H, P)}{r_\alpha(M_i)} \min(r_\alpha(M_1), r_\alpha(M_2))$ are defined as 0 when $r_\alpha(M_i) = 0$.*

*That $\tilde{\mathcal{R}}_\cdot(\cdot, \cdot, \cdot)$ is well-defined follows immediately from the definition of $\simeq_G$.*

The component rate function gives the overall aggregate rate at which a given standard PEPA component type within a grouped PEPA model does a certain action type. We are ultimately interested however in how that rate is shared between enabled transitions into different derivative states. For example, for a given grouped PEPA model and particular component group containing standard PEPA components $P$ and $Q$,

we wish to know the rate at which $P$ components make transitions to become $Q$ components, as opposed to just the overall rate of transition out of $P$. To this end, we define the *derivative weighting function* which calculates the probability that given that a standard PEPA component does an action, it transits to another specified standard PEPA component state.

**Definition 2.14** (Derivative weighting function) *Let $P$ and $Q$ be standard PEPA components and let $\alpha \in \mathcal{A}$. Then:*

$$p_\alpha(P, Q) := \frac{1}{r_\alpha(P)} \sum_{P \xrightarrow{(\alpha, \lambda_i)} Q} \lambda_i$$

*This is defined to be zero when $r_\alpha(P) = 0$.*

We now wish to relate the component rate function and derivative weighting function to the underlying aggregated CTMC induced by $ds(G)/\simeq_G$ by characterising the outgoing transitions from a given state in terms of these functions.

**Theorem 2.15** *Let $G$ be a grouped PEPA model. Let $\tilde{G} \in ds(G)/\simeq_G$ be a groupwise equivalence class of $G$. Consider the component group with label $H \in \mathcal{G}(G)$ and let $\alpha \in \mathcal{A}^\tau$ be an extended action type. For each standard PEPA component $P \in \mathcal{B}(G, H)$, all $\alpha$-transitions from $\tilde{G}$ to some other state $\tilde{G}' \in ds(G)/\simeq_G$ where $\tilde{\mathcal{C}}(\tilde{G}', H, P) \neq \tilde{\mathcal{C}}(\tilde{G}, H, P)$ are such that exactly one of the following holds:*

1. *$\tilde{\mathcal{C}}(\tilde{G}', H, P) = \tilde{\mathcal{C}}(\tilde{G}, H, P) - 1$, and furthermore, the sum of the rates of all $\alpha$-transitions to states $\tilde{G}'$ for which this holds is:*
$$\sum_{\substack{Q_i \in \mathcal{B}(G, H) \\ Q_i \neq P}} p_{t(\alpha)}(P, Q_i)\, \tilde{\mathcal{R}}_\alpha(\tilde{G}, H, P)$$
*or*
2. *$\tilde{\mathcal{C}}(\tilde{G}', H, P) = \tilde{\mathcal{C}}(\tilde{G}, H, P) + 1$ and furthermore, the sum of the rates of all $\alpha$-transitions to states $\tilde{G}'$ for which this holds is:*
$$\sum_{\substack{Q_i \in \mathcal{B}(G, H) \\ Q_i \neq P}} p_{t(\alpha)}(Q_i, P)\, \tilde{\mathcal{R}}_\alpha(\tilde{G}, H, Q_i)$$

*Also in either case, if there is no such $\alpha$-transition, the rates given above are zero.*

*Proof.* See Appendix A.1. □

## 3. Fluid analysis of grouped PEPA models

We now proceed to show how a system of coupled ODEs may be derived from a grouped PEPA model. These ODEs describe a continuous approximation to the time evolution of the numbers of each type of standard PEPA component within a grouped PEPA model. This section claims only that the rates used to define the ODEs match those of the underlying aggregated CTMC (Theorem 2.15). The actual relationship between the ODE solution and the underlying CTMC is considered in the next section.

Since the component counts uniquely determine groupwise equivalence classes, we are effectively representing the state of the system with a sequence of integers specifying the number of each standard PEPA component in each component group. Given such a definition, there is nothing preventing us from allowing, more generally, a sequence of real numbers and naturally extending the definition of component rate accordingly. Although this would have no relationship to the original grouped PEPA model since it makes no sense to have a non-integer number of components, it is exactly what we need for the fluid approximation, where integer component counts are approximated by real variables.

We will call these alternative definitions *count-oriented*. Let $G$ be a grouped PEPA model and define the set of all functions $\mathcal{E}(G) := \mathcal{B}(G) \to \mathbb{R}_{\geq 0}$. This represents a superset of the possible states of the aggregated CTMC of a grouped PEPA model, i.e. a subset[4] of elements of $\mathcal{E}(G)$ specify counts for all standard PEPA components within the grouped model $G$. This construction plays the role of the sequence of real numbers mentioned above. In order to present the count-oriented definition of component rate, we first need a count-oriented definition of apparent rate.

**Definition 3.1** (Count-oriented apparent rate) *Let $G$ be a grouped PEPA model. Let $\alpha \in \mathcal{A}^\tau$ be an extended action type and $E \in \mathcal{E}(G)$ specify the component counts. Then the count-oriented apparent rate is $r_\alpha(G, E)$ where $r_\cdot(\cdot, \cdot)$ is defined as:*

$$r_\alpha(M_1 \underset{L}{\bowtie} M_2, E) := \begin{cases} \min(r_\alpha(M_1, E), r_\alpha(M_2, E)) & \text{if } \alpha \in L \\ r_\alpha(M_1, E) + r_\alpha(M_2, E) & \text{otherwise} \end{cases}$$

$$r_\alpha(M/L, E) := \begin{cases} r_\alpha(M, E) & \text{if } \alpha \notin L \\ 0 & \text{otherwise} \end{cases}$$

$$r_\alpha(Y\{D\}, E) := \sum_{P_i \in \mathcal{B}(Y\{D\}, Y)} E(Y, P_i)\, r_\alpha(P_i)$$

This matches the original definition of apparent rate (Definition 2.2), or more formally:

**Theorem 3.2** *Let $G$ be a grouped PEPA model. Then for all extended action types $\alpha \in \mathcal{A}^\tau$ and groupwise equivalence classes $\tilde{G} \in ds(G)/\simeq_G$, we have that $r_\alpha(G, C(\tilde{G})) = r_\alpha(\overline{G})$ for all $\overline{G} \in \tilde{G}$, where $C(\tilde{G}) \in \mathcal{E}(G)$ is given by $C(\tilde{G}) := \lambda(H, P) \to \tilde{\mathcal{C}}(\tilde{G}, H, P)$.*

*Proof.* Immediate from the definitions. □

**Definition 3.3** (Count-oriented component rate function) *Let $G$ be a grouped PEPA model. For $(H, P) \in \mathcal{B}(G)$, extended action type $\alpha \in \mathcal{A}^\tau$ and $E \in \mathcal{E}(G)$ specifying the component counts, the count-oriented component rate is $\mathcal{R}_\alpha(G, E, H, P)$ where $\mathcal{R}_\cdot(\cdot, \cdot, \cdot, \cdot)$ is defined as:*

$$\mathcal{R}_\alpha(M_1 \underset{L}{\bowtie} M_2, E, H, P) := \begin{cases} \frac{\mathcal{R}_\alpha(M_i, E, H, P)}{r_\alpha(M_i, E)} \min(r_\alpha(M_1, E), r_\alpha(M_2, E)) & \text{if } H \in \mathcal{G}(M_i) \text{ and } \alpha \in L, \text{ for } i = 1 \text{ or } 2 \\ \mathcal{R}_\alpha(M_i, E, H, P) & \text{if } H \in \mathcal{G}(M_i) \text{ and } \alpha \notin L, \text{ for } i = 1 \text{ or } 2 \end{cases}$$

$$\mathcal{R}_\alpha(M/L, E, H, P) := \begin{cases} 0 & \text{if } \alpha \in L \text{ (and } \alpha \in \mathcal{A}) \\ \mathcal{R}_\alpha(M, E, H, P) & \text{if } \alpha \notin L \text{ and } \alpha \in \mathcal{A} \\ \mathcal{R}_{\tau^\beta}(M, E, H, P) & \text{if } \alpha = \tau^\beta \text{ and } \beta \notin L \\ \mathcal{R}_{\tau^\beta}(M, E, H, P) + \mathcal{R}_\beta(M, E, H, P) & \text{if } \alpha = \tau^\beta \text{ and } \beta \in L \end{cases}$$

$$\mathcal{R}_\alpha(Y\{D\}, E, H, P) := \begin{cases} E(H, P)\, r_\alpha(P) & \text{if } H = Y \text{ and } P \in \mathcal{B}(G, H) \\ 0 & \text{otherwise} \end{cases}$$

*As before, terms with zero-valued denominators are defined to be zero.*

This matches the original definition of component rate (Definition 2.13), or more formally:

**Theorem 3.4** *Let $G$ be a grouped PEPA model. Then for all $(H, P) \in \mathcal{B}(G)$, extended action types $\alpha \in \mathcal{A}^\tau$ and groupwise equivalence classes $\tilde{G} \in ds(G)/\simeq_G$, we have that $\mathcal{R}_\alpha(G, C(\tilde{G}), H, P) = \tilde{\mathcal{R}}_\alpha(\tilde{G}, H, P)$, where $C(\tilde{G}) \in \mathcal{E}(G)$ is given by $C(\tilde{G}) := \lambda(H, P) \to \tilde{\mathcal{C}}(\tilde{G}, H, P)$.*

[4] Non-negative integer-valued, in the correct range and preserving of the component group sizes. Note also that even if a given element of $\mathcal{E}(G)$ has these properties, the CTMC state which it uniquely defines may not actually be reachable. This is worth bearing in mind but has no explicit effect on this work.

*Proof.* Immediate from the definitions. □

Now consider a grouped PEPA model $G$. Let $B = (H, P) \in \mathcal{B}(G)$. In the style of Example 1.5, we will introduce $v_B(t)$ as a deterministic, continuous approximation to the integer-valued stochastic process $N_B(t) := \tilde{\mathcal{C}}(\tilde{G}_t, H, P)$ where $\tilde{G}_t \in ds(G)/\simeq_G$ is the state of the underlying aggregated CTMC at time $t$. As in the example, the definition of $v_B(t)$ will be by means of a system of coupled, first-order ODEs.

**Definition 3.5** (ODE system associated with a grouped PEPA model) *Let $G$ be a grouped PEPA model. We define the evolution of the $v_{H,P}(t)$ over time for $(H, P) \in \mathcal{B}(G)$ by the system of first-order coupled ODEs:*

$$\dot{v}_{H,P}(t) = \sum_{\alpha_i \in \mathcal{A}^\tau} \left( \sum_{Q_j \in \mathcal{B}(G,H)} p_{\boldsymbol{t}(\alpha_i)}(Q_j, P)\, \mathcal{R}_{\alpha_i}(G, V(t), H, Q_j) \right) - \mathcal{R}_{\alpha_i}(G, V(t), H, P)$$
$$\text{for all } (H, P) \in \mathcal{B}(G)$$

*where for $t \in \mathbb{R}_{\geq 0}$, $V(t) \in \mathcal{E}(G)$ is given by $V(t) := (\lambda(H, P) \to v_{H,P}(t))$ for all $(H, P) \in \mathcal{B}(G)$.*

*The initial conditions, $V_0 \in \mathcal{E}(G)$, for this system of ODEs are those naturally defined by the initial state of $G$. That is:*

$$V_0 := (\lambda(H, P) \to \mathcal{C}(G, H, P))$$

Definition 3.5 requires some explanation. As discussed, it is motivated by a more general version of the method applied in Example 1.5. Indeed, for each component type in each component group, we consider the rates of the underlying CTMC at which "copies" of the component are lost and gained, forming an ODE by balancing these quantities, informally:

$$\dot{v}_{H,P}(t) = \sum_{\alpha \in \mathcal{A}^\tau} \Big[ \sum \text{ of rates of all } \alpha \text{ transitions } \textbf{increasing} \text{ no. } P \text{ components in group } H$$
$$- \sum \text{ of rates of all } \alpha \text{ transitions } \textbf{decreasing} \text{ no. } P \text{ components in group } H \Big]$$

Theorem 2.15 tells us that the first sum of the above is:

$$\sum_{\substack{Q_j \in \mathcal{B}(G,H) \\ Q_j \neq P}} p_{\boldsymbol{t}(\alpha)}(Q_j, P)\, \tilde{\mathcal{R}}_\alpha(\tilde{G}_t, H, Q_j)$$

and the second sum:

$$\sum_{\substack{Q_j \in \mathcal{B}(G,H) \\ Q_j \neq P}} p_{\boldsymbol{t}(\alpha)}(P, Q_j)\, \tilde{\mathcal{R}}_\alpha(\tilde{G}_t, H, P)$$

Since we are concerned with the difference of these two sums, we may drop the condition $Q_j \neq P$ on each since the corresponding terms will cancel. We may then note that $\sum_{Q_j \in \mathcal{B}(G,H)} p_{\boldsymbol{t}(\alpha)}(P, Q_j) = 1$, which gives the equation of Definition 3.5.

## 4. Fluid analysis as a first moment approximation

There is currently very limited work in the literature relating the deterministic, real-valued quantities $v_{H,P}(t)$ defined by the systems of ODEs presented in the last section to the integer-valued stochastic processes $N_{H,P}(t)$, which count the number of standard PEPA model components in the component group with label $H$ in derivative state $P$ at time $t$. In this section, we show, using the Chapman–Kolmogorov equations which govern the evolution of the underlying CTMC, that the quantities $v_{H,P}(t)$ are each an approximation

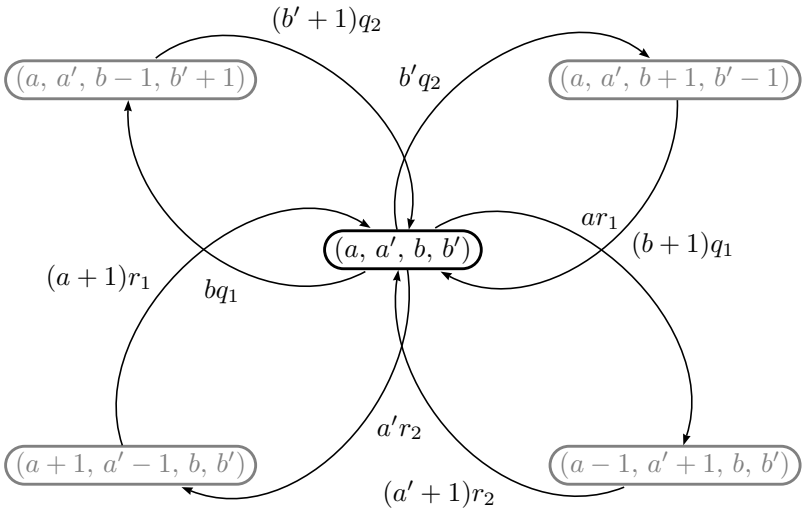



Fig. 5: A central state of the underlying aggregated CTMC of Example 4.2.

to the corresponding expectations, $\mathbb{E}[N_{H,\,P}(t)]$. Then, in the following section, we will see how this point of view allows us to develop similar approximate analyses for (joint) higher order moments of the stochastic processes $N_{H,\,P}(t)$.

We begin our analysis by considering a very small class of model, the *purely concurrent* models, for which we have an exact result.

### *4.1. Purely concurrent models*

In the case of *purely concurrent* grouped PEPA models, that is, those without synchronisation between component groups, we can show an exact correspondence with the underlying CTMC. In particular, we will show that the solution of the ODEs of Definition 3.5 has an exact interpretation as the first moment of the stochastic process $N_{H,\,P}(t)$.

**Definition 4.1** (Purely concurrent grouped PEPA model) *A purely concurrent grouped PEPA model is a grouped PEPA model that has no shared actions (between component groups), that is, it can still be expressed in the less general grammar, obtained by modifying Equation (2.3):*

$$M ::= M \parallel M \;\mid\; M/L \;\mid\; Y\{D\}$$

We motivate what follows through a simple example.

Example 4.2 Exact transient correspondence of fluid analysis for a simple model
Consider the following purely concurrent grouped PEPA model representing two unsynchronised groups of

processors ($N \times Processor_a$ and $M \times Processor_b$):

$$\begin{aligned}
Processor_a &\stackrel{def}{=} (task_1,\, r_1).Processor'_a \\
Processor'_a &\stackrel{def}{=} (task_2,\, r_2).Processor_a \\
Processor_b &\stackrel{def}{=} (task_1,\, q_1).Processor'_b \\
Processor'_b &\stackrel{def}{=} (task_2,\, q_2).Processor_b
\end{aligned}$$

$$System \stackrel{def}{=} \mathbf{A}\{Processor_a[N]\} \parallel \mathbf{B}\{Processor_b[M]\}$$

Writing $a$, $a'$, $b$ and $b'$ for the number of $Processor_a$, $Processor'_a$, $Processor_b$ or $Processor'_b$ components respectively, a central state of the aggregated CTMC of this model is shown in Figure 5. We write $p_{(a,\,a',\,b,\,b')}(t)$ for the transient probability of being in the CTMC state representing the groupwise equivalence class of $a \times Processor_a$, $a' \times Processor'_a$, $b \times Processor_b$ and $b' \times Processor'_b$ components at time $t$. For this case, the Chapman–Kolmogorov forward equations, that govern the evolution of the state probabilities of the underlying aggregated CTMC, have the general form:

$$\begin{aligned}
\dot{p}_{(a,\,a',\,b,\,b')}(t) = {} & (a+1)r_1 \cdot p_{(a+1,\,a'-1,\,b,\,b')}(t) \\
& + (a'+1)r_2 \cdot p_{(a-1,\,a'+1,\,b,\,b')}(t) \\
& + (b+1)q_1 \cdot p_{(a,\,a',\,b+1,\,b'-1)}(t) \\
& + (b'+1)q_2 \cdot p_{(a,\,a',\,b-1,\,b'+1)}(t) \\
& - ar_1 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& - a'r_2 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& - bq_1 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& - b'q_2 \cdot p_{(a,\,a',\,b,\,b')}(t)
\end{aligned}$$

where each of the first four summands appear only when the state $(a,\, a',\, b,\, b')$ has the corresponding incoming transitions, in the aggregated state space, say $S$. Multiplying $p_{(a,\,a',\,b,\,b')}(t)$ by $a$, summing for all states $(a,\, a',\, b,\, b') \in S$, then re-indexing the sums for the first four terms on the right-hand side, we obtain:

$$\begin{aligned}
\sum_{(a,\,a',\,b,\,b') \in S} a \cdot \dot{p}_{(a,\,a',\,b,\,b')}(t) = \sum_{(a,\,a',\,b,\,b') \in S} \Big[ & (a-1)ar_1 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& + (a+1)a'r_2 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& + abq_1 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& + ab'q_2 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& - aar_1 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& - aa'r_2 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& - abq_1 \cdot p_{(a,\,a',\,b,\,b')}(t) \\
& - ab'q_2 \cdot p_{(a,\,a',\,b,\,b')}(t) \Big]
\end{aligned}$$

This re-indexing is possible since the sums of the first four terms on the right-hand side only omit terms which are equal to zero. If $(A(t),\, A'(t),\, B(t),\, B'(t)) \in S$ is the state of the underlying aggregated CTMC at time $t$, cancelling terms then yields:

$$\dot{\mathbb{E}}[A(t)] = -r_1 \cdot \mathbb{E}[A(t)] + r_2 \cdot \mathbb{E}[A'(t)]$$

Using a similar technique, we obtain also:

$$\begin{aligned}\dot{\mathbb{E}}[A'(t)] &= -r_2 \cdot \mathbb{E}[A'(t)] + r_1 \cdot \mathbb{E}[A(t)] \\ \dot{\mathbb{E}}[B(t)] &= -q_1 \cdot \mathbb{E}[B(t)] + q_2 \cdot \mathbb{E}[B'(t)] \\ \dot{\mathbb{E}}[B'(t)] &= -q_2 \cdot \mathbb{E}[B'(t)] + q_1 \cdot \mathbb{E}[B(t)]\end{aligned}$$

These coincide exactly with the ODEs of Definition 3.5 when $v_{H,\,P}(t)$ is taken to be the expectations of the component counts in question at time $t$.

□

We will see that this result extends to general purely concurrent grouped models, but we first present a more general result for arbitrary grouped PEPA models which will also be of later use.

**Theorem 4.3** *Let $G$ be a grouped PEPA model and $(H,\,P) \in \mathcal{B}(G)$. Then:*

$$\dot{\mathbb{E}}[N_{H,\,P}(t)] = \sum_{\alpha_i \in \mathcal{A}^\tau} \left( \sum_{Q_j \in \mathcal{B}(G,\,H)} p_{\boldsymbol{t}(\alpha_i)}(Q_j,\,P)\mathbb{E}[\mathcal{R}_{\alpha_i}(G,\,N(t),\,H,\,Q_j)] \right) - \mathbb{E}[\mathcal{R}_{\alpha_i}(G,\,N(t),\,H,\,P)]$$

*where for $t \in \mathbb{R}_{\geq 0}$, $N(t) \in \mathcal{E}(G)$ is given by $N(t) := (\lambda(H,\,P) \to N_{H,\,P}(t))$ for all $(H,\,P) \in \mathcal{B}(G)$.*

*Proof.* See Appendix A.2 □

**Theorem 4.4** *Let $G$ be a purely concurrent grouped PEPA model. Then the expectations of the component counts at time $t$, $\mathbb{E}[N_{H,\,P}(t)]$ satisfy the ODEs of Definition 3.5.*

*Proof.* Since $G$ is purely concurrent, we have by Definition 3.3 that for all $(H,\,P) \in \mathcal{B}(G)$, the term $\mathcal{R}_\alpha(G,\,N(t),\,H,\,P)$ is nothing more than a linear combination of component counts. Therefore by linearity of expectation, Theorem 4.3 reduces to:

$$\dot{\mathbb{E}}[N_{H,\,P}(t)] = \sum_{\alpha_i \in \mathcal{A}^\tau} \left( \sum_{Q_j \in \mathcal{B}(G,\,H)} p_{\boldsymbol{t}(\alpha_i)}(Q_j,\,P)\mathcal{R}_{\alpha_i}(G,\,E(t),\,H,\,Q_j) \right) - \mathcal{R}_{\alpha_i}(G,\,E(t),\,H,\,P)$$

where for all $(H,\,P) \in \mathcal{B}(G)$, $E(t) := (\lambda(H,\,P) \to \mathbb{E}[N_{H,\,P}(t)])$. □

Despite being quantitatively very strong, this result has limited use since it applies only to the simplest of models where components evolve independently of each other. Such models are of course already easily tractable and not particularly interesting.

Interesting models are those with cooperation between component groups. In this case, Theorem 4.3 tells us that we lose the exact correspondence with the expectations of the underlying CTMC since the component rate function is no longer necessarily linear in the component counts. The ODEs of Definition 3.5 for a system with cooperation between component groups will be seen to yield an approximation to the expectations of component counts. In the next section, we consider a large class of models (the *split-free models*) for which this approximation has a particularly simple form.

### *4.2. Split-free models*

The class of *split-free models* is general enough to include most large concurrent models encountered in practice. Specifically it excludes models involving cooperations $M_1 \underset{L}{\bowtie} M_2$ where, for some shared action $\alpha \in L$, either $M_1$ or $M_2$ contains more than one type of standard PEPA component which enables $\alpha$. That is, at least one side of the cooperation *splits* between two different types of standard PEPA component. This is formalised in the next definition.

**Definition 4.5** (Split-free grouped PEPA model) *A grouped PEPA model $G$ is split-free if and only if $\mathcal{Q}(G) = \mathbf{true}$, where $\mathcal{Q}(\cdot)$ is defined as:*

$$\begin{aligned}\mathcal{Q}(M_1 \underset{L}{\bowtie} M_2) &:= \mathcal{Q}(M_1) \wedge \mathcal{Q}(M_2) \wedge \mathcal{Q}'(M_1, L) \wedge \mathcal{Q}'(M_2, L)\\ \mathcal{Q}(M/L) &:= \mathcal{Q}(M)\\ \mathcal{Q}(Y\{D\}) &:= \mathbf{true}\end{aligned}$$

*where $\mathcal{Q}'(M, L) = \mathbf{true}$ if and only if for all $\alpha \in L$, there do not exist distinct $(H_1, P_1), (H_2, P_2) \in \mathcal{B}(M)$ with $\alpha \notin \mathcal{V}(M, H_1)$ and $\alpha \notin \mathcal{V}(M, H_2)$, such that $(\alpha, \cdot) \in \mathcal{A}ct(P_1)$ and $(\alpha, \cdot) \in \mathcal{A}ct(P_2)$.*

Recall that for a general grouped PEPA model $G = M_1 \underset{L}{\bowtie} M_2$, $(H, P) \in \mathcal{B}(G)$ and $E \in \mathcal{E}(G)$, the (count-oriented) component rate function (Definition 3.3) is defined as:

$$\mathcal{R}_\alpha(M_1 \underset{L}{\bowtie} M_2, E, H, P) := \begin{cases} \frac{\mathcal{R}_\alpha(M_i, E, H, P)}{r_\alpha(M_i, E)} \min(r_\alpha(M_1, E), r_\alpha(M_2, E)) & \text{if } H \in \mathcal{G}(M_i) \text{ and } \alpha \in L, \text{ for } i = 1 \text{ or } 2\\ \mathcal{R}_\alpha(M_i, E, H, P) & \text{if } H \in \mathcal{G}(M_i) \text{ and } \alpha \notin L, \text{ for } i = 1 \text{ or } 2 \end{cases}$$

The term $\frac{\mathcal{R}_\alpha(M_i, E, H, P)}{r_\alpha(M_i, E)}$ may introduce rational functions in terms of the component counts into the component rate function and thus the system of differential equations associated with the model. The desirable property of a split-free model, on the other hand, is that the rational expression will always cancel, that is, $\mathcal{R}_\alpha(M_i, E, H, P) = r_\alpha(M_i, E)$ for all $\alpha \in L$. This follows immediately from the definition of split-free.

For a general split-free grouped PEPA model $G$, there may be cooperation between component groups, and we see by Definition 3.3 that the general form of $\mathcal{R}_\alpha(G, \cdot, \cdot, \cdot)$ may involve minimums and linear combinations (but not rational functions) of component counts, i.e. it is a piecewise linear function of the component counts. Since the only source of non-linearity is the use of minimums, we see that the equation of Theorem 4.3 matches that of Definition 3.5 after the following simple approximation is applied, perhaps more than once. For two stochastic processes, $M(t)$ and $N(t)$:

$$\mathbb{E}[\min(M(t), N(t))] \approx \min(\mathbb{E}[M(t)], \mathbb{E}[N(t)])$$

Note that in general:

$$\mathbb{E}[\min(M(t), N(t))] \leq \min(\mathbb{E}[M(t)], \mathbb{E}[N(t)])$$

We now illustrate this by means of a simple concrete example.

Example 4.6 Fluid analysis as a first moment approximation for a simple synchronised (split-free) model

Consider again the processor/resource model $System_G(N_p, N_r)$. It is trivially split-free.

Writing $P_0$, $P_1$, $R_0$ or $R_1$ for the number of $Processor_0$, $Processor_1$, $Resource_0$ or $Resource_1$ components respectively, a central state of the aggregated CTMC of this model is shown in Figure 6. We can write $p_{(P_0, P_1, R_0, R_1)}(t)$ for the transient probability of being in the CTMC state representing the groupwise equivalence class of $P_0 \times Processor_0$, $P_1 \times Processor_1$, $R_0 \times Resource_0$ and $R_1 \times Resource_1$ components at time $t$. The Chapman–Kolmogorov forward equations are in this case:

$$\begin{aligned}\dot{p}_{(P_0, P_1, R_0, R_1)}(t) = {} & \min((P_0+1)r_1, (R_0+1)r_2) \cdot p_{(P_0+1, P_1-1, R_0+1, R_1-1)}(t)\\ & + (P_1+1)q \cdot p_{(P_0-1, P_1+1, R_0, R_1)}(t)\\ & + (R_1+1)s \cdot p_{(P_0, P_1, R_0-1, R_1+1)}(t)\\ & - \min(P_0 r_1, R_0 r_2) \cdot p_{(P_0, P_1, R_0, R_1)}(t)\\ & - P_1 q \cdot p_{(P_0, P_1, R_0, R_1)}(t)\\ & - R_1 s \cdot p_{(P_0, P_1, R_0, R_1)}(t)\end{aligned} \tag{4.1}$$

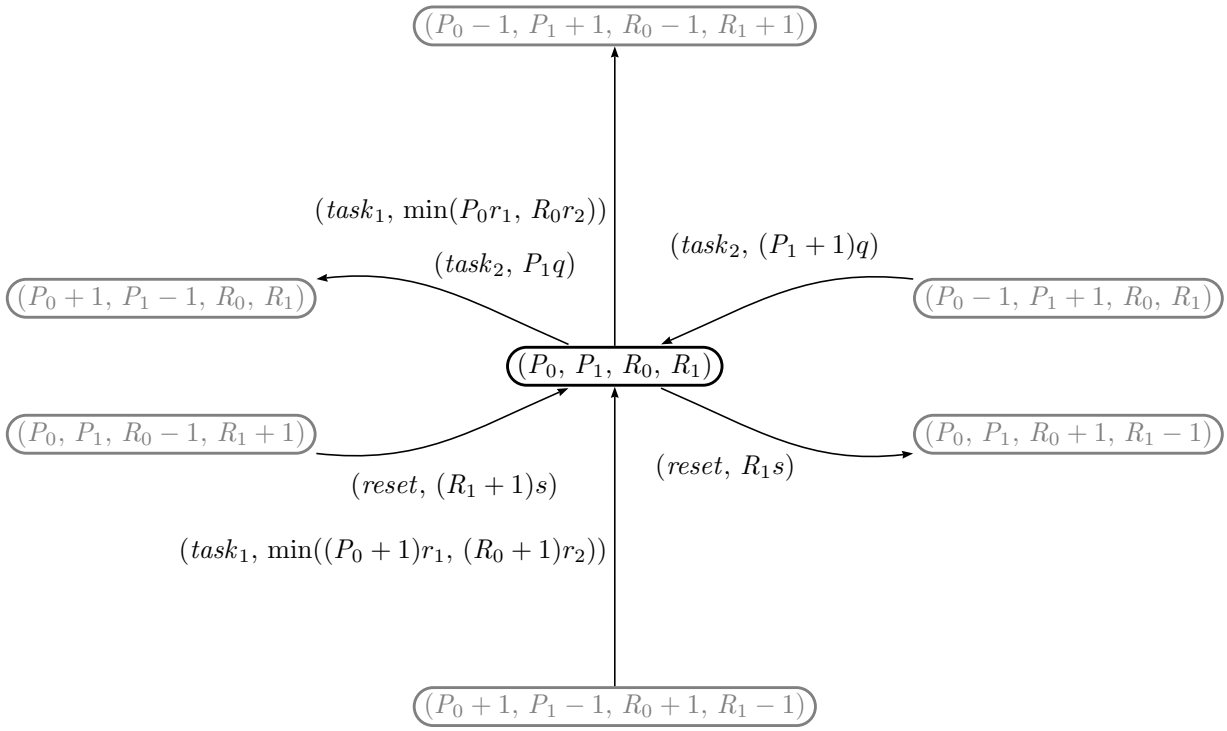



Fig. 6: A central state of the underlying aggregated CTMC of Example 4.6.

where again, each of the first four summands appear only when the state $(a,\, a',\, b,\, b')$ has the corresponding incoming transitions, in the aggregated state space, say $S$. Multiplying $p_{(P_0,\, P_1,\, R_0,\, R_1)}(t)$ by $P_0$ and summing for all states $(P_0,\, P_1,\, R_0,\, R_1) \in S$ as in the previous example, we obtain:

$$\begin{aligned}\sum_{(P_0,\, P_1,\, R_0,\, R_1) \in S} P_0 \cdot \dot{p}_{(P_0,\, P_1,\, R_0,\, R_1)}(t) = \sum_{(P_0,\, P_1,\, R_0,\, R_1) \in S} \Big[ & (P_0 - 1) \min(P_0 r_1,\ R_0 r_2) \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ & + (P_0 + 1) P_1 q \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ & + P_0 R_1 s \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ & - P_0 \min(P_0 r_1,\ R_0 r_2) \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ & - P_0 P_1 q \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ & - P_0 R_1 s \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \Big]\end{aligned}$$

If $(P_0(t),\, P_1(t),\, R_0(t),\, R_1(t)) \in S$ is the state of the underlying aggregated CTMC at time $t$, cancelling terms then yields:

$$\dot{\mathbb{E}}[P_0(t)] = -\mathbb{E}[\min(P_0(t) r_1,\ R_0(t) r_2)] + q \cdot \mathbb{E}[P_1(t)]$$

Similarly we may obtain also:

$$\dot{\mathbb{E}}[P_1(t)] = \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] - q \cdot \mathbb{E}[P_1(t)]$$
$$\dot{\mathbb{E}}[R_0(t)] = -\mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] + s \cdot \mathbb{E}[R_1(t)]$$
$$\dot{\mathbb{E}}[R_1(t)] = \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] - s \cdot \mathbb{E}[R_1(t)]$$

These equations could be obtained systematically by applying Theorem 4.3 directly and not considering the Chapman–Kolmogorov equations explicitly. Then when the $v_{H,\,P}(t)$ are taken to be the expectations of the component counts in question at time $t$ and the approximation $\mathbb{E}[\min(\cdot,\,\cdot)] \approx \min(\mathbb{E}[\cdot],\, \mathbb{E}[\cdot])$ is applied once for each equation, we do indeed obtain the ODEs of Definition 3.5. □

*4.2.1. Nature of the approximation*

We have identified that for split-free grouped PEPA models, the fluid analysis relies on the approximation $\mathbb{E}[\min(X,\, Y)] \approx \min(\mathbb{E}[X],\, \mathbb{E}[Y])$, where $X$ and $Y$ are random variables. In simple instances, these are linear combinations of component count random variables for some time $t$, possibly multiplied by positive real rates. In more complicated situations, the $X$ and $Y$ random variables can themselves involve minimums of such terms.

We now define the notion of a *switch point* associated to a given grouped PEPA model $G$. Consider a $(H,\, P) \in \mathcal{B}(G)$, then for any $E \in \mathcal{E}(G)$, the component rate function $\mathcal{R}_\alpha(G,\, E,\, H,\, P)$ defines switch points of $G$. Indeed, for each $\min(f(E),\, g(E))$ term that occurs in the component rate function, the associated set of switch points is $\{E \in \mathcal{E}(G) \,:\, f(E) = g(E)\}$. Intuitively, these are the points in the phase space of the associated ODE system at which the $\min(f(E),\, g(E))$-term switches from $f(E)$ to $g(E)$ and vice-versa.

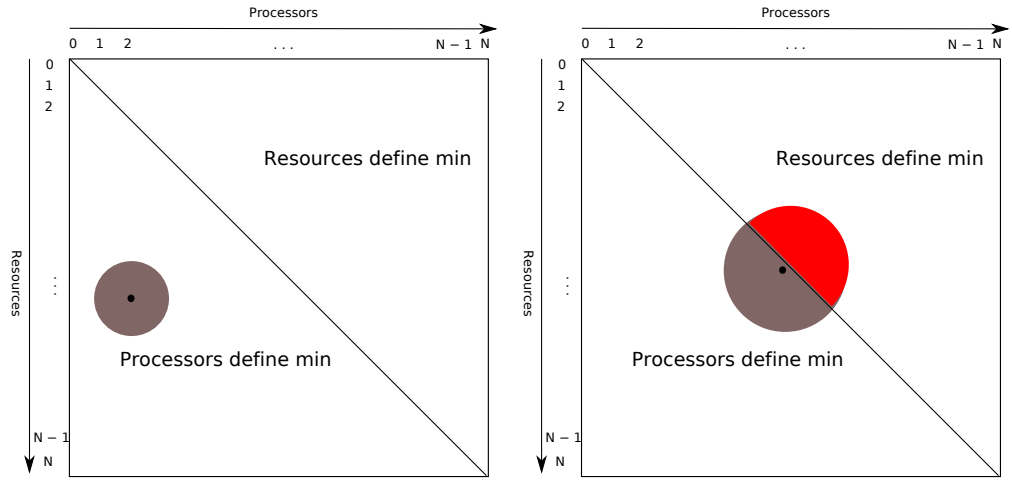
In the case of the processor/resource model $System_G(N_p,\, N_r)$, the only switch points are those of the expression $\min(v_{P_0}(t)r_1,\, v_{R_0}(t)r_2)$, where we write $v_{P_0}(t)$ for $v_{\mathbf{Processors},\,P_0}(t)$ and so on. Indeed, in this simple case, the set of switch points is the line $y = \frac{r_2}{r_1}x$ where the $y$-axis counts the number of $P_0$ components and the $x$-axis, the number of $R_0$ components.

It is easy to see that far away from switch points, we would expect the fluid approximation to remain good unless the variabilities of the component counting stochastic processes are high. Figure 7a illustrates this scenario for the case of $N_p = N_r = N$ and $r_1 = r_2 = 1$. The black dot illustrates the expected values of the number of processors and resources with the grey area representing informally the extent of the variability or spread of the distribution. The diagonal line splits the state space into the areas in which each side of the $\min(\cdot,\,\cdot)$ is the defining term, that is, it is the line of switch points. At the point of the state space illustrated, $\mathbb{E}[\min(N_{P_0}(t),\, N_{R_0}(t))]$ is clearly very well approximated by $\mathbb{E}[N_{P_0}(t)] = \min(\mathbb{E}[N_{P_0}(t)],\, \mathbb{E}[N_{R_0}(t)])$, since $N_{P_0}(t) \ll N_{R_0}(t)$.

Around the switch points, the quality of the approximation decreases with increasing variability of $N_{P_0}(t)$ and $N_{R_0}(t)$. This is illustrated in Figure 7b. The red part of the circle illustrates the part of the distribution which has been incorrectly assigned to the processors rather than the resources, under the approximation $\mathbb{E}[\min(N_{P_0}(t),\, N_{R_0}(t))] \approx \mathbb{E}[N_{P_0}(t)] = \min(\mathbb{E}[N_{P_0}(t)],\, \mathbb{E}[N_{R_0}(t)])$.

## 5. Fluid approximations for higher-order moments

In this section, we show how the results of the last section can be extended to develop a similar style of fluid approximation for the (joint) higher order moments of the component counts. This provides inexpensive access to key characteristics of the probability distribution of the component counts, such as their variance. As well as having application in itself, critically, this will allow us to predict the accuracy of the original first-order fluid analysis. Furthermore, knowledge of the variance of component counts can provide an indication as to whether the expectations of component counts are likely to reflect the actual trace behaviour of the system being modelled.




(a) Far from a switch point.

(b) Close to a switch point.

Fig. 7: Accuracy of the $\mathbb{E}[\min(N_{P_0}(t),\, N_{R_0}(t))] \approx \min(\mathbb{E}[N_{P_0}(t)],\, \mathbb{E}[N_{R_0}(t)])$ approximation.

### *5.1. An example*

We introduce this idea by again considering the model $System_G(N_p,\, N_r)$ defined in Section 1.1.3.

Example 5.1 Fluid variance approximation for a simple synchronised (split-free) model
As in Example 4.6, we will work with the Chapman–Kolmogorov forward equations (Equation (4.1)) of the underlying aggregated CTMC of $System_G(N_p,\, N_r)$.

Multiplying $p_{(P_0,\, P_1,\, R_0,\, R_1)}(t)$ by $P_0^2$ and summing for all states $(P_0,\, P_1,\, R_0,\, R_1) \in S$ in a similar fashion to the previous examples, we obtain:

$$\begin{aligned}\sum_{(P_0,\, P_1,\, R_0,\, R_1)\in S} P_0^2 \cdot \dot{p}_{(P_0,\, P_1,\, R_0,\, R_1)}(t) = \sum_{(P_0,\, P_1,\, R_0,\, R_1)\in S} \Big[ &(P_0-1)^2 \min(P_0 r_1,\, R_0 r_2) \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ &+ (P_0+1)^2 P_1 q \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ &+ P_0^2 R_1 s \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ &- P_0^2 \min(P_0 r_1,\, R_0 r_2) \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ &- P_0^2 P_1 q \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ &- P_0^2 R_1 s \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \Big] \end{aligned} \tag{5.1}$$

If $(P_0(t),\, P_1(t),\, R_0(t),\, R_1(t)) \in S$ is the state of the underlying aggregated CTMC at time $t$, this simplifies to:

$$\begin{aligned}\dot{\mathbb{E}}[P_0^2(t)] = \sum_{(P_0,\, P_1,\, R_0,\, R_1)\in S} \Big[ &-2\min(P_0^2 r_1,\, P_0 R_0 r_2) \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ &+ \min(P_0 r_1,\, R_0 r_2) \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ &+ 2 P_0 P_1 q \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \\ &+ P_1 q \cdot p_{(P_0,\, P_1,\, R_0,\, R_1)}(t) \Big] \end{aligned}$$

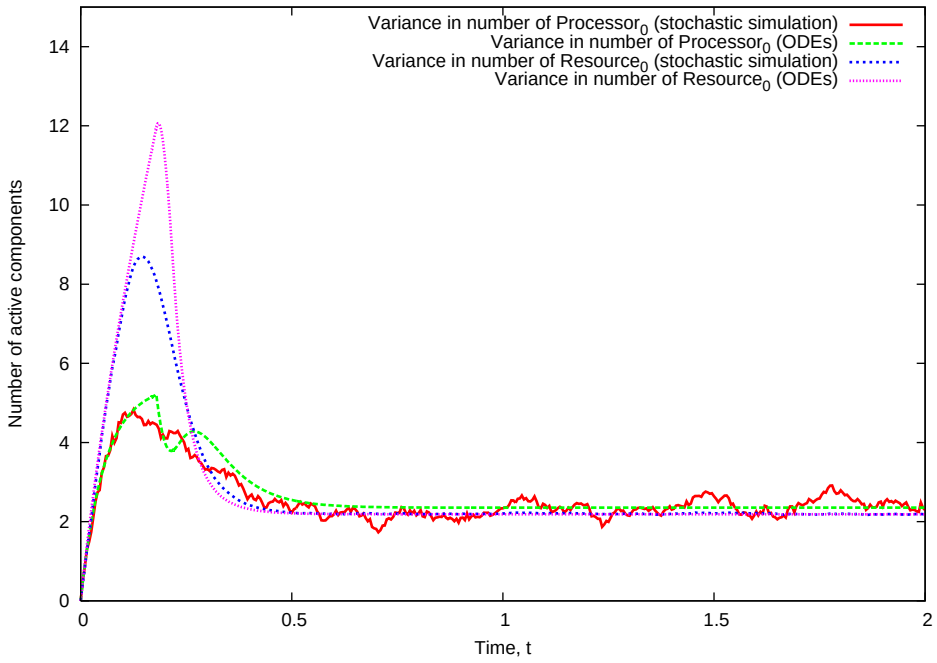



Fig. 8: Comparison of ODE solutions with those obtained through stochastic simulation for the variance of the number of processors and of resources in the simple processor/resource model. Rates used are $r_1 = 2.0$, $r_2 = 14.0$, $q = 14.0$ and $s = 2.0$. Initial conditions are 50 $Processor_0$ and 20 $Resource_0$ components.

or:

$$\begin{aligned}\dot{\mathbb{E}}[P_0^2(t)] = & -2\mathbb{E}[\min(P_0(t)^2 r_1,\, P_0(t)R_0(t)r_2)] + \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] \\ & + 2\mathbb{E}[P_0(t)P_1(t)]q + \mathbb{E}[P_1(t)]q\end{aligned} \tag{5.2}$$

Following this programme for all moments of orders one and two (the resulting system of equations is given in Appendix B.1), and then applying the previous approximation $\mathbb{E}[\min(\cdot,\,\cdot)] \approx \min(\mathbb{E}[\cdot],\, \mathbb{E}[\cdot])$ to the entire system we get a system of 14 coupled first-order (piecewise-linear) ODEs.

Now that we have access to second order moments, we may of course compute quantities such as the variance as well as the mean. The variance of the number of processors and the number of resources are shown in Figure 8, with the results obtained through integrating the systems of ODEs[5] against those obtained using Gillespie's algorithm [17] to generate 1,000,000 independent replications. It is clear we obtain a good agreement, at least of the qualitative features of the variances, between the ODE solution and the averaged stochastic simulation. □

### *5.2. General treatment of higher-order moments*

In this section, the aim is to give a general definition of the system of ODEs which allow systematic fluid analysis of higher order moments for grouped PEPA models. For this, we will need to evaluate how components evolve jointly so that we can derive equations for $\dot{\mathbb{E}}[X(t)Y(t)]$, for instance, as well as other higher order cross-terms.

[5]Specifically, we compute $\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ using fluid analysis.

*5.2.1. Joint component rates*

To proceed, we need to generalise the notion of the component rate function for a grouped PEPA model, an action type and *one* given standard PEPA component (Section 2.2) to the *joint component rate* for a given *set* of standard PEPA components. This will arise naturally in the consideration of higher order moments involving more than one component count.

The joint component rate is the aggregate rate at which all of the given standard PEPA components complete an action of a given action type in cooperation together. It is the sum of the rates of all transitions of that action type in the aggregated CTMC, in which the specified components, in $\mathcal{J}$, are the *only* joint participants.

**Definition 5.2** (Count-oriented joint component rate function) *Let $G$ be a grouped PEPA model. Let $\mathcal{J} \subseteq \mathcal{B}(G)$ be a non-empty set of component group label and corresponding standard PEPA component pairs. Let $\alpha \in \mathcal{A}^\tau$ be an extended action type and $E \in \mathcal{E}(G)$ specify the component counts. Then the count-oriented joint component rate is $\mathcal{R}_\alpha(G, E, \mathcal{J})$ where $\mathcal{R}_\cdot(\cdot, \cdot, \cdot)$ is defined as:*

$$\mathcal{R}_\alpha(M_1 \underset{L}{\bowtie} M_2, E, \mathcal{J}) := \begin{cases} \frac{\mathcal{R}_\alpha(M_1, E, \mathcal{J}_1)}{r_\alpha(M_1, E)} \frac{\mathcal{R}_\alpha(M_2, E, \mathcal{J}_2)}{r_\alpha(M_2, E)} \min(r_\alpha(M_1, E), r_\alpha(M_2, E)) & \text{if } \alpha \in L,\ \mathcal{J} \not\subseteq \mathcal{B}(M_j) \\ & \quad \text{for } j = 1 \text{ and } 2 \\ \mathcal{R}_\alpha(M_j, E, \mathcal{J}) & \text{if } \alpha \notin L,\ \mathcal{J} \subseteq \mathcal{B}(M_j) \\ & \quad \text{for } j = 1 \text{ or } 2 \\ 0 & \text{otherwise} \end{cases}$$

*In the first line of the definition, we define $\mathcal{J}_1$ and $\mathcal{J}_2$ to be the unique partition of $\mathcal{J}$ such that $\mathcal{J}_1 \subseteq \mathcal{B}(M_1)$ and $\mathcal{J}_2 \subseteq \mathcal{B}(M_2)$. As before, terms with zeros in the denominator are defined as zero.*

$$\mathcal{R}_\alpha(M/L, E, \mathcal{J}) := \begin{cases} 0 & \text{if } \alpha \in L \text{ (and } \alpha \in \mathcal{A}) \\ \mathcal{R}_\alpha(M, E, \mathcal{J}) & \text{if } \alpha \notin L \text{ and } \alpha \in \mathcal{A} \\ \mathcal{R}_{\tau^\beta}(M, E, \mathcal{J}) & \text{if } \alpha = \tau^\beta \text{ and } \beta \notin L \\ \mathcal{R}_{\tau^\beta}(M, E, \mathcal{J}) + \mathcal{R}_\beta(M, E, \mathcal{J}) & \text{if } \alpha = \tau^\beta \text{ and } \beta \in L \end{cases}$$

$$\mathcal{R}_\alpha(H\{D\}, E, \mathcal{J}) := \begin{cases} 0 & \text{if } |\mathcal{J}| \neq 1 \\ E(H, P)\, r_\alpha(P) & \text{if } \mathcal{J} = \{(H, P)\} \end{cases}$$

In the final case of the count-oriented formula, the cardinality of $\mathcal{J}$ has to be 1 to return a non-zero value, since there is no cooperation between the components of a component group by definition, and we are computing the *joint* rate of evolution between cooperating components.

For brevity, we have presented the count-oriented version only, but for a given $\tilde{G} \in ds(G)/\simeq_G$, we can of course define an equivalent version on groupwise equivalence classes directly by:

$$\mathcal{R}_\alpha(\tilde{G}, \mathcal{J}) := \mathcal{R}_\alpha(G, C(\tilde{G}), \mathcal{J})$$

where $C(\tilde{G}) \in \mathcal{E}(G)$ is given by $C(\tilde{G}) := \lambda(H, P) \to \tilde{\mathcal{C}}(\tilde{G}, H, P)$. Or indeed on elements $\overline{G} \in \tilde{G}$ of groupwise equivalence classes:

$$\mathcal{R}_\alpha(\overline{G}, \mathcal{J}) := \mathcal{R}_\alpha(\tilde{G}, \mathcal{J})$$

We now relate the joint component rate function to the underlying CTMC. For a given grouped PEPA model $G$, and $\tilde{G} \in ds(G)/\simeq_G$, we will specify pairs of non-empty sets $\mathcal{J}_-$ and $\mathcal{J}_+ \subseteq \mathcal{B}(G)$. These will be useful to characterise the possible states reachable in one step from $\tilde{G}$ and are much used in the remainder of the paper, so we present a summary:

| | |
|---|---|
| $\mathcal{J}_-$ | A set of component types that must all cooperate together to produce a particular action type. When they cooperate, the number of each component in this set will be decremented (unless the component also appears in $\mathcal{J}_+$) |
| $\mathcal{J}_+$ | A set of component types that is a possible 1-step evolution of a set of cooperating components, $\mathcal{J}_-$. When a cooperation occurs, the number of each component in this set will be incremented (unless the component also appears in $\mathcal{J}_-$) |

For a given action type, we will be interested in the sum of the rates of all transitions from $\tilde{G}$ of that action type decrementing the count of all components in $\mathcal{J}_- \setminus \mathcal{J}_+$, incrementing the count of all components in $\mathcal{J}_+ \setminus \mathcal{J}_-$ and leaving the count of all other components alone. Note that for each transition out of $\tilde{G}$, there is exactly one pair $(\mathcal{J}_-, \mathcal{J}_+)$, which represents it in this manner. Transitions where $\mathcal{J}_- \cap \mathcal{J}_+ \neq \emptyset$ are possible since there may be components which are involved in a cooperation that does not decrement the count of that component. Consider, for example, components which may loop such as $P \stackrel{def}{=} (\alpha, r).P + (\beta, q).Q$ in a component group on one side of a cooperation.

We will impose two conditions restricting the allowed pairs $(\mathcal{J}_-, \mathcal{J}_+)$. Firstly, no two derivative states in the same component group can cooperate on a transition, so there can be no $H \in \mathcal{G}(G)$ such that there exist distinct $(H, P)$ and $(H, Q)$ both in $\mathcal{J}_-$ or both in $\mathcal{J}_+$. Secondly, any state change in a component group must be between derivative states within the group, thus for every $H \in \mathcal{G}(G)$, if $(H, P) \in \mathcal{J}_-$, there must be some corresponding $(H, Q) \in \mathcal{J}_+$ (not necessarily distinct from $(H, P)$) and vice-versa. Define $\mathcal{J}(G)$ to be the set of all such pairs of non-empty sets $(\mathcal{J}_-, \mathcal{J}_+)$, satisfying these two conditions. The conditions do not change anything since the operational semantics assert that there can be no such transition if either condition does not hold. They do however simplify the following theorem, which gives the rates associated with these transitions.

**Theorem 5.3** *Let $G$ be a grouped PEPA model. Let $\tilde{G}, \tilde{G}' \in ds(G)/\simeq_G$ be groupwise equivalence classes of $G$. Let $\alpha \in \mathcal{A}^\tau$ be an extended action type.*

*Then the sum of the rates of all $\alpha$-transitions from $\tilde{G}$ to $\tilde{G}'$ is the sum of all quantities:*

$$\rho_{t(\alpha)}(\mathcal{J}_-, \mathcal{J}_+)\mathcal{R}_\alpha(\tilde{G}, \mathcal{J}_-) \tag{5.3}$$

*taken over all pairs $(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)$, which characterise $\tilde{G}'$ in the sense given above, that is:*

1. $\tilde{\mathcal{C}}(\tilde{G}', H, P) = \tilde{\mathcal{C}}(\tilde{G}, H, P) - 1$ *for all* $(H, P) \in \mathcal{J}_- \setminus \mathcal{J}_+$,
2. $\tilde{\mathcal{C}}(\tilde{G}', H, P) = \tilde{\mathcal{C}}(\tilde{G}, H, P) + 1$ *for all* $(H, P) \in \mathcal{J}_+ \setminus \mathcal{J}_-$ *and*
3. $\tilde{\mathcal{C}}(\tilde{G}', H, P) = \tilde{\mathcal{C}}(\tilde{G}, H, P)$ *for all* $(H, P) \in \mathcal{B}(G) \setminus (\mathcal{J}_- \ominus \mathcal{J}_+)$.[6]

*For $\alpha \in \mathcal{A}$, $\rho_\alpha(\cdot, \cdot)$ is the generalised derivative weighting function, defined by:*

$$\rho_\alpha(\mathcal{J}_-, \mathcal{J}_+) := \prod_{(H,Q) \in \mathcal{J}_+} p_\alpha(\boldsymbol{g}(H, \mathcal{J}_-), Q)$$

*This is the joint probability taken over all $(H, Q) \in \mathcal{J}_+$, that given that each standard PEPA component, $(H, \boldsymbol{g}(H, \mathcal{J}_-)) \in \mathcal{J}_-$ does an $\alpha$-action, it transits to the component, $(H, Q) \in \mathcal{J}_+$, where $\boldsymbol{g}(H, \mathcal{J}_-)$ is defined as the unique standard PEPA component $P$, such that $(H, P) \in \mathcal{J}_-$. That it will exist and be unique is guaranteed by the fact that $(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)$.*

*Each term given by Equation (5.3) is zero if no such $\alpha$-transitions exist.*

[6] Where $A \ominus B := (A \cup B) \setminus (A \cap B)$ is the symmetric difference of the two sets $A$ and $B$.

*Proof.* See Appendix A.3. □

As we will see in Section 6, we will generate a $\mathcal{J}_-$ set of components for every possible local transition of a model, whether shared or not. For each $\mathcal{J}_-$, there will be at least one and possibly several resulting $\mathcal{J}_+$ sets of components. The $\rho_\alpha(\mathcal{J}_-, \mathcal{J}_+)$ weighting function generates the probability that a specified $\mathcal{J}_+$ is the actual 1-step joint $\alpha$-evolution from a $\mathcal{J}_-$ set of cooperating components.

Making use of the characterisation of the underlying CTMC given by Theorem 5.3, we are now able to present the following key result, which generalises Theorem 4.3 to arbitrary joint moments of component counts.

**Theorem 5.4** *Let $G$ be a grouped PEPA model and $\mathcal{M} \in (\mathcal{B}(G) \to \mathbb{Z}_{\geq 0})$ specify a particular joint moment of component counting stochastic processes that is of interest, $\mathbb{E}[M(t)]$, by:*

$$M(t) := \prod_{B_i \in \mathcal{B}(G)} N_{B_i}(t)^{\mathcal{M}(B_i)}$$

*Then we have:*

$$\dot{\mathbb{E}}[M(t)] = \sum_{\alpha_i \in \mathcal{A}^\tau} \sum_{(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)} \rho_{t(\alpha_i)}(\mathcal{J}_-, \mathcal{J}_+)\, \mathbb{E}\left[(F_+(t)F_-(t)F(t) - M(t))\, \mathcal{R}_{\alpha_i}(G, N(t), \mathcal{J}_-)\right] \tag{5.4}$$

*where:*

$$F_\pm(t) := \prod_{B_i \in \mathcal{J}_\pm \setminus \mathcal{J}_\mp} (N_{B_i}(t) \pm 1)^{\mathcal{M}(B_i)} \qquad \text{and} \qquad F(t) := \prod_{B_i \in \mathcal{B}(G) \setminus (\mathcal{J}_- \ominus \mathcal{J}_+)} N_{B_i}(t)^{\mathcal{M}(B_i)}$$

*and for $t \in \mathbb{R}_{\geq 0}$, $N(t) \in \mathcal{E}(G)$ is given by $N(t) := \lambda(B) \to N_B(t)$ for all $B \in \mathcal{B}(G)$.*

*Proof.* See Appendix A.4 □

For the incoming transitions, the functions $F_+(t)$ and $F_-(t)$ reflect the contributions of $\mathcal{J}_-$ and $\mathcal{J}_+$, respectively from the underlying Chapman–Kolmogorov equations, and $F(t)$ represents the contribution of the other components (in $\mathcal{B}(G) \setminus (\mathcal{J}_- \ominus \mathcal{J}_+)$). $M(t)$ expresses the contributions for the outgoing transitions for all components. This is best illustrated by considering Example 5.1, where we are constructing an equation for $\dot{\mathbb{E}}[P_0^2(t)]$. For action type $task_1$, the only value of $\mathcal{J}_-$ for which the joint component rate is not identically zero is:

$$\mathcal{J}_- = \{(\textbf{Processors}, P_0), (\textbf{Resources}, R_0)\}$$

Indeed, in this case, the joint component rate is:

$$\mathcal{R}_{task_1}(G, N(t), \mathcal{J}_-) = \min(P_0(t)r_1, R_0(t)r_2)$$

Furthermore, for this $\mathcal{J}_-$, the only value of $\mathcal{J}_+$ for which $\rho_{task_1}(\mathcal{J}_-, \mathcal{J}_+)$ is not zero (it is one) is:

$$\mathcal{J}_+ = \{(\textbf{Processors}, P_1), (\textbf{Resources}, R_1)\}$$

For this pair $(\mathcal{J}_-, \mathcal{J}_+)$:

$$\begin{aligned} F_+(t) &= (P_1(t) + 1)^0 (R_1(t) + 1)^0 = 1 \\ F_-(t) &= (P_0(t) - 1)^2 (R_0(t) - 1)^0 = (P_0(t) - 1)^2 \\ F(t) &= 1 \end{aligned}$$

so:

$$\begin{aligned}(F_+(t)F_-(t)F(t) - M(t))\mathcal{R}_{task_1}(G,\, N(t),\, \mathcal{J}_-) &= ((P_0(t)-1)^2 - P_0^2(t))\min(P_0(t)r_1,\, R_0(t)r_2) \\ &= -2\min(P_0^2(t)r_1,\, P_0(t)R_0(t)r_2) \\ &\quad + \min(P_0(t)r_1,\, R_0(t)r_2)\end{aligned}$$

This corresponds with lines one and four of the right-hand side of Equation (5.1) and the first two summands of the right-hand side of Equation (5.2). The other terms come from considering the two remaining actions, $task_2$ and *reset*.

### *5.2.2. Fluid approximation*

The previous section presented an equation for the derivative of a general moment of component counting functions (Equation (5.4) of Theorem 5.4). In this section, we show how a fluid approximation to this equation can be constructed using a suitable system of ODEs. In order to achieve this, me must rearrange Equation (5.4) to make the individual component count moments explicit on the right hand side. Each of these component count moments is then approximated by its own differential equation (as finally achieved in Equation (5.9)).

We begin by using the binomial theorem to see that the term $F_+(t)F_-(t)F(t)$ of Equation (5.4) is the sum of all monomials of the form:

$$\prod_{B_i \in \mathcal{J}_- \setminus \mathcal{J}_+} (-1)^{\mathcal{M}(B_i)-K(B_i)} \prod_{B_i \in \mathcal{J}_- \ominus \mathcal{J}_+} \binom{\mathcal{M}(B_i)}{K(B_i)} \prod_{B_i \in \mathcal{B}(G)} N_{B_i}(t)^{K(B_i)} \tag{5.5}$$

over all elements $K \in (\mathcal{B}(G) \to \mathbb{Z}_{\geq 0})$, such that $0 \leq K(B) \leq \mathcal{M}(B)$ for all $B \in \mathcal{J}_- \ominus \mathcal{J}_+$, and $K(B) = \mathcal{M}(B)$ for all $B \in \mathcal{B}(G) \setminus (\mathcal{J}_- \ominus \mathcal{J}_+)$. Now consider the term $F_+(t)F_-(t)F(t) - M(t)$ and note that $M(t)$ simply serves to cancel the highest order Equation (5.5) monomial, i.e. the case of $K(B) = \mathcal{M}(B)$ for all $B \in \mathcal{B}(G)$, so after this cancellation, all monomials in the sum will have order at most $M - 1$, where $M := \sum_{B_i \in \mathcal{B}(G)} \mathcal{M}(B_i)$. That is, $F_+(t)F_-(t)F(t) - M(t)$ is equal to the sum of all monomials of the form of Equation (5.5) taken over all elements $K \in (\mathcal{B}(G) \to \mathbb{Z}_{\geq 0})$, such that $0 \leq K(B) \leq \mathcal{M}(B)$ for all $B \in \mathcal{J}_- \ominus \mathcal{J}_+$, and $K(B) = \mathcal{M}(B)$ for all $B \in \mathcal{B}(G) \setminus (\mathcal{J}_- \ominus \mathcal{J}_+)$, and $\sum_{B_i \in \mathcal{B}(G)} K(B_i) \neq M$. Call this subset of $(\mathcal{B}(G) \to \mathbb{Z}_{\geq 0})$, $\mathcal{K}(\mathcal{J}_-,\, \mathcal{J}_+,\, \mathcal{M})$.

Consider now the expectation term:

$$\mathbb{E}\left[(F_+(t)F_-(t)F(t) - M(t))\mathcal{R}_{\alpha_i}(G,\, N(t),\, \mathcal{J}_-)\right] \tag{5.6}$$

from Equation (5.4). Using the fact above that $F_+(t)F_-(t)F(t) - M(t)$ consists of a sum of monomials of the form of Equation (5.5), we may expand Equation (5.6) into a sum of expectations, one for each monomial term, i.e. it is equal to the sum of the terms:

$$\left[\prod_{B_j \in \mathcal{J}_- \setminus \mathcal{J}_+} (-1)^{\mathcal{M}(B_j)-K(B_j)} \prod_{B_j \in \mathcal{J}_- \ominus \mathcal{J}_+} \binom{\mathcal{M}(B_j)}{K(B_j)}\right] \mathbb{E}\left[\prod_{B_j \in \mathcal{B}(G)} N_{B_j}(t)^{K(B_j)} \mathcal{R}_{\alpha_i}(G,\, N(t),\, \mathcal{J}_-)\right] \tag{5.7}$$

over all $K \in \mathcal{K}(\mathcal{J}_-,\, \mathcal{J}_+,\, \mathcal{M})$.

The joint component rate function enjoys a *homogeneity* property, which is useful in presenting the system of ODEs which will facilitate the fluid analysis.

**Lemma 5.5** *Let $G$ be a grouped PEPA model. Let $\mathcal{J} \subseteq \mathcal{B}(G)$ be a non-empty set of component group label and corresponding standard PEPA component pairs. Let $\alpha \in \mathcal{A}^\tau$ be an extended action type and let $E \in \mathcal{E}(G)$. Then for any real $C \geq 0$:*

$$C \times \mathcal{R}_\alpha(G,\, E,\, \mathcal{J}) = \mathcal{R}_\alpha(G,\, E^*,\, \mathcal{J})$$

*where $E^* \in \mathcal{E}(G)$ is defined by $E^*(B) = C \times E(B)$ for all $B \in \mathcal{B}(G)$.*

*Proof.* See Appendix A.5. □

Lemma 5.5 allows us to write Equation (5.7) as:

$$\left[\prod_{B_j \in \mathcal{J}_- \setminus \mathcal{J}_+} (-1)^{\mathcal{M}(B_j)-K(B_j)} \prod_{B_j \in \mathcal{J}_- \ominus \mathcal{J}_+} \binom{\mathcal{M}(B_j)}{K(B_j)}\right] \mathbb{E}\left[\mathcal{R}_{\alpha_i}(G,\, N^*(t),\, \mathcal{J}_-)\right] \tag{5.8}$$

where for $t \in \mathbb{R}_{\geq 0}$, $N^*(t) \in \mathcal{E}(G)$ is given by:

$$N^*(t) := \lambda(B) \to N_B(t) \times \prod_{B_i \in \mathcal{B}(G)} N_{B_i}(t)^{K(B_i)}$$

for all $B \in \mathcal{B}(G)$.

Now for a given moment, specified by $\mathcal{M} \in (\mathcal{B}(G) \to \mathbb{Z}_{\geq 0})$, write $v_{\mathcal{M}}(t)$ as its fluid approximation. For split-free models, the joint component rate function (Definition 5.2), as in the case of the component rate function (Definition 3.3), may involve minimums and linear combinations (but not rational functions) of component counts. So applying the approximation $\mathbb{E}[\min(\cdot,\, \cdot)] \approx \min(\mathbb{E}[\cdot],\, \mathbb{E}[\cdot])$, potentially repeatedly to Equation (5.8),[7] yields in terms of the fluid approximations:

$$\left[\prod_{B_j \in \mathcal{J}_- \setminus \mathcal{J}_+} (-1)^{\mathcal{M}(B_j)-K(B_j)} \prod_{B_j \in \mathcal{J}_- \ominus \mathcal{J}_+} \binom{\mathcal{M}(B_j)}{K(B_j)}\right] \mathcal{R}_{\alpha_i}(G,\, V(t,\, K),\, \mathcal{J}_-)$$

where for $t \in \mathbb{R}_{\geq 0}$, $V(t,\, K) \in \mathcal{E}(G)$ is given by, for all $B \in \mathcal{B}(G)$:

$$V(t,\, K) := \lambda(B) \to v_{\mathcal{I}}(t)$$

and $\mathcal{I}(B_i) = K(B_i)$ for all $B_i \neq B$ and $\mathcal{I}(B) = K(B) + 1$. Note that since $\sum_{B_i \in \mathcal{B}(G)} K(B_i) \leq M - 1$ for all $K \in \mathcal{K}(\mathcal{J}_-,\, \mathcal{J}_+,\, \mathcal{M})$, fluid approximations of joint moments of order at most $M$ occur in such terms.

Combining these terms to approximate Equation (5.4) gives the following ODE where the right-hand side is in terms of the fluid approximations of the joint moments of order $M$ and below:

$$\begin{aligned}\dot{v}_{\mathcal{M}}(t) = \sum_{\alpha_i \in \mathcal{A}^\tau} \sum_{(\mathcal{J}_-,\, \mathcal{J}_+) \in \mathcal{J}(G)} \rho_{\boldsymbol{t}(\alpha_i)}(\mathcal{J}_-,\, \mathcal{J}_+) \\ \sum_{K \in \mathcal{K}(\mathcal{J}_-,\, \mathcal{J}_+,\, \mathcal{M})} \left(\left[\prod_{B_j \in \mathcal{J}_- \setminus \mathcal{J}_+} (-1)^{\mathcal{M}(B_j)-K(B_j)} \prod_{B_j \in \mathcal{J}_- \ominus \mathcal{J}_+} \binom{\mathcal{M}(B_j)}{K(B_j)}\right] \mathcal{R}_{\alpha_i}(G,\, V(t,\, K),\, \mathcal{J}_-)\right)\end{aligned} \tag{5.9}$$

The initial condition for each moment, $\mathcal{M} \in (\mathcal{B}(G) \to \mathbb{Z}_{\geq 0})$, is that naturally implied by the initial state of $G$. Specifically, we define:

$$v_{\mathcal{M}}(0) := \prod_{B_i \in \mathcal{B}(G)} \mathcal{C}(G,\, B_i)^{\mathcal{M}(B_i)}$$

In the case of first-order moments, Equation (5.9) reduces to the ODE given by Definition 3.5.

For a grouped PEPA model $G$ and a given order $n \geq 1$, the system of ODEs consisting of Equation (5.9) taken for every $m$th ($1 \leq m \leq n$) joint moment of component counts thus yields a complete Lipschitz continuous system of:

$$\sum_{i=1}^{n} \binom{P+i-1}{i} = \binom{P+n}{n} - 1 \leq P^{n+1} \text{ for } P > 1$$

[7] For models which are not split-free, the fluid approximation is still defined but the nature of the approximation is more complicated, as in the first-order case.

ODEs where $P = |\mathcal{B}(G)|$, which can be solved for their unique solutions, yielding approximations for all joint moments of orders $n$ and below. So for a given order $n$, the growth in the number of ordinary differential equations is no worse than polynomial in the number of standard PEPA components in the model. This will obviously have a practical implication for the degree of moment analysis that will be possible for a given model, however first and second order moment analysis should certainly be very tractable in most cases.

## 6. Worked example

In this section, we show explicitly how Equation (5.9) can be used in practice to construct fluid approximations to the expectations and higher order moments (specifically, the variance) of component counts of grouped PEPA models.

### *6.1. Client–Server model with 2 stage fetch and server breakdowns*

In the (split-free) grouped PEPA model $G$ below, we have a population of $C$ Clients and a population of $S$ Servers. The system uses a 2-stage fetch mechanism: a client requests data from the pool of servers; one of the servers receives the request, another server may then fetch the data for the client. At any stage, a server in the pool may fail.

$$
\begin{aligned}
\mathit{Client} &\stackrel{def}{=} (\mathit{request}, r_{req}).\mathit{Client\_waiting} \\
\mathit{Client\_waiting} &\stackrel{def}{=} (\mathit{data}, r_{data}).\mathit{Client\_think} \\
\mathit{Client\_think} &\stackrel{def}{=} (\mathit{think}, r_{think}).\mathit{Client} \\
\\
\mathit{Server} &\stackrel{def}{=} (\mathit{request}, r_{req}).\mathit{Server\_get} + (\mathit{break}, r_{break}).\mathit{Server\_broken} \\
\mathit{Server\_get} &\stackrel{def}{=} (\mathit{data}, r_{data}).\mathit{Server} + (\mathit{break}, r_{break}).\mathit{Server\_broken} \\
\mathit{Server\_broken} &\stackrel{def}{=} (\mathit{reset}, r_{reset}).\mathit{Server} \\
\\
G &\stackrel{def}{=} \mathbf{Clients}\{\mathit{Client}[C]\} \underset{L}{\bowtie} \mathbf{Servers}\{\mathit{Server}[S]\}
\end{aligned}
$$

where $L = \{\mathit{request}, \mathit{data}\}$.

We will be considering this model for $C = 100$ and $S = 50$. Its unaggregated state space would therefore have of the order of $3^{150}$ states. This places the model out of the immediate reach of existing analysis techniques for numerically obtaining steady-state or transient probabilities. The only other option is to simulate the model many times for the random quantity of interest using, for example, Gillespie's algorithm [17], and take the average to obtain the desired expectation. This is of course still many orders of magnitude more costly than the simple operation of integrating a small system of ODEs, as is required for the fluid analysis but is how we will validate the results of the fluid approximation.

### *6.2. Construction of first and second order moment fluid approximation*

We are now going to evaluate Equation (5.9) explicitly to obtain a system of ODEs allowing fluid analysis of all first and second order moments of the various client and server component counts for the above model. We will write $N_C(t)$ for the *Client* counting stochastic process, $N_{S_g}(t)$ for the *Server_get* counting stochastic process and similarly for all other components. Also define $N(t) \in \mathcal{E}(G)$ by:

$$N(t)(\mathbf{Clients},\ \mathit{Client}) := N_C(t)$$

and so on. We will write $v_{C^2}(t)$ for the fluid approximation to the second moment of the *Client* counting process and $v_{C \cdot S_g}(t)$ for the fluid approximation to the joint moment of the *Client* and *Server_get* counting processes, and again, similarly for all other first and second order moment approximations.

In order to evaluate Equation (5.9), it is easiest to first enumerate all of the non identically zero joint component rates:

1. *request*-transitions of $\{$*Client*, *Server*$\}$ occur at rate $\min(N_C(t), N_S(t))r_{req}$, that is:

$$\mathcal{R}_{request}(G, N(t), \{(\textbf{Clients}, \textit{Client}), (\textbf{Servers}, \textit{Server})\}) = \min(N_C(t), N_S(t))r_{req}$$

2. *data*-transitions of $\{$*Client_waiting*, *Server_get*$\}$ occur at rate $\min(N_{C_w}(t), N_{S_g}(t))r_{data}$, that is:

$$\mathcal{R}_{data}(G, N(t), \{(\textbf{Clients}, \textit{Client\_waiting}), (\textbf{Servers}, \textit{Server\_get})\}) = \min(N_{C_w}(t), N_{S_g}(t))r_{data}$$

3. *think*-transitions of $\{$*Client_think*$\}$ occur at rate $N_{C_t}(t)r_{think}$, that is:

$$\mathcal{R}_{think}(G, N(t), \{(\textbf{Clients}, \textit{Client\_think})\}) = N_{C_t}(t)r_{think}$$

4. *break*-transitions of $\{$*Server*$\}$ occur at rate $N_S(t)r_{break}$, that is:

$$\mathcal{R}_{break}(G, N(t), \{(\textbf{Servers}, \textit{Server})\}) = N_S(t)r_{break}$$

5. *break*-transitions of $\{$*Server_get*$\}$ occur at rate $N_{S_g}(t)r_{break}$, that is:

$$\mathcal{R}_{break}(G, N(t), \{(\textbf{Servers}, \textit{Server\_get})\}) = N_{S_g}(t)r_{break}$$

6. *reset*-transitions of $\{$*Server_broken*$\}$ occur at rate $N_{S_b}(t)r_{reset}$, that is:

$$\mathcal{R}_{reset}(G, N(t), \{(\textbf{Servers}, \textit{Server\_broken})\}) = N_{S_b}(t)r_{reset}$$

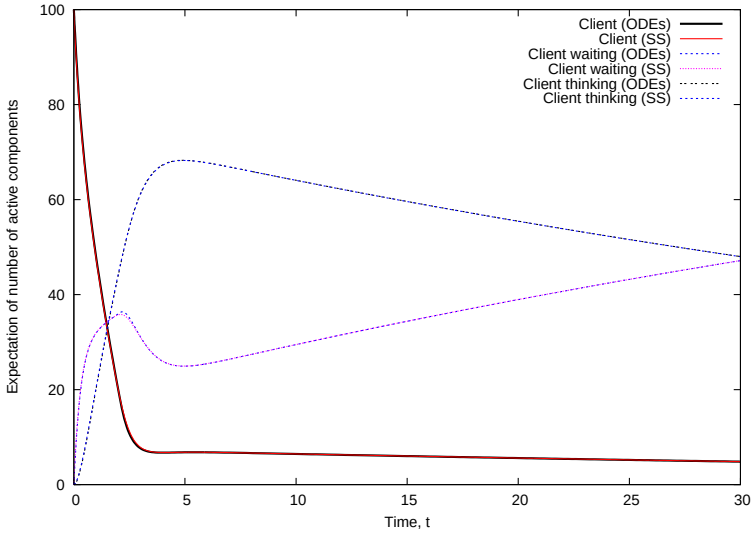
Having identified the subsets $\mathcal{J}_- \subseteq \mathcal{B}(G)$ for which the joint component rate function is not identically zero, we then wish to identify the corresponding subsets $\mathcal{J}_+ \subseteq \mathcal{B}(G)$, such that $(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)$, for which the coefficient $\rho_\alpha(\mathcal{J}_-, \mathcal{J}_+)$ is non-zero. That is, we are interested in the sets of components $\mathcal{J}_+ \subseteq \mathcal{B}(G)$, whose counts can all be incremented by simultaneous evolution of (only) the components $\mathcal{J}_-$. For this model, it is clear that there is exactly one such $\mathcal{J}_+$ corresponding to each of the above transition classes. In the same order of enumeration, we thus now give the corresponding $\mathcal{J}_+$. Note that for this model, if $\rho_\alpha(\mathcal{J}_-, \mathcal{J}_+)$ is non-zero, it is always one.

1. $\mathcal{J}_+ = \{(\textbf{Clients}, \textit{Client\_waiting}), (\textbf{Servers}, \textit{Server\_get})\}$
2. $\mathcal{J}_+ = \{(\textbf{Clients}, \textit{Client\_think}), (\textbf{Servers}, \textit{Server})\}$
3. $\mathcal{J}_+ = \{(\textbf{Clients}, \textit{Client})\}$
4. $\mathcal{J}_+ = \{(\textbf{Servers}, \textit{Server\_broken})\}$
5. $\mathcal{J}_+ = \{(\textbf{Servers}, \textit{Server\_broken})\}$
6. $\mathcal{J}_+ = \{(\textbf{Servers}, \textit{Server})\}$

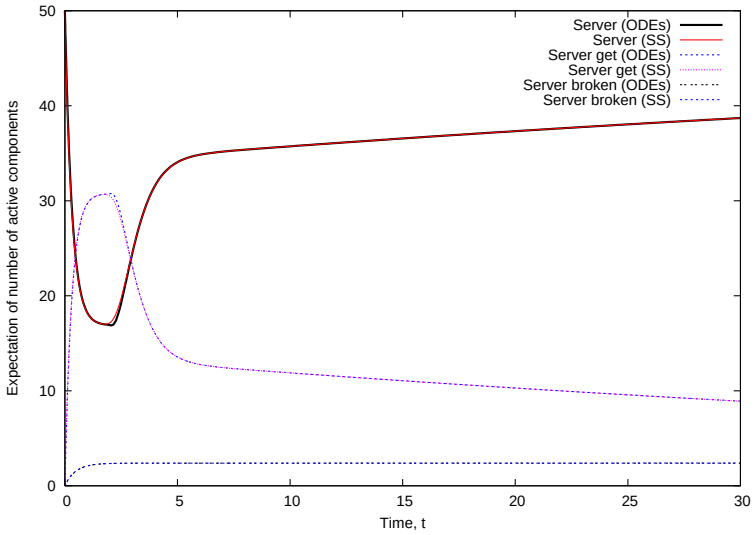
Let us begin by considering the joint second order moment $\mathbb{E}[N_{C_w}(t)N_{S_g}(t)]$. The only transitions of interest are 1, 2 and 5 in the above enumeration since they are the only ones for which either:

$$(\textbf{Clients}, \textit{Client\_waiting}) \in \mathcal{J}_- \ominus \mathcal{J}_+ \text{ or } (\textbf{Servers}, \textit{Server\_get}) \in \mathcal{J}_- \ominus \mathcal{J}_+$$

Indeed, if $\mathcal{J}_- \ominus \mathcal{J}_+$ does not include at least one $B \in \mathcal{B}(G)$ for which $\mathcal{M}(B) > 0$, $\mathcal{K}(\mathcal{J}_-, \mathcal{J}_+, \mathcal{M}) = \emptyset$ and the pair $(\mathcal{J}_-, \mathcal{J}_+)$ would contribute nothing to the right hand side of Equation (5.9).




(a) Expectations of component counts for the **Clients** group.




(b) Expectations of component counts for the **Servers** group.

Fig. 9: Comparison of expectations obtained through fluid analysis with those obtained through stochastic simulation for client–server model. Rates used are: $r_{req} = 2.0$, $r_{think} = 0.2$, $r_{break} = 0.1$, $r_{data} = 1.0$, $r_{reset} = 2.0$ and initial component counts are 100 *Client* and 50 *Server* components.

Now for the first transition, $\mathcal{K}(\mathcal{J}_-, \mathcal{J}_+, \mathcal{M})$ consists of three elements, $K_1$, $K_2$ and $K_3$, where $K_1(B) = 0$ for all $B \in \mathcal{B}(G)$, $K_2((\textbf{Clients}, \mathit{Client\_waiting})) = 1$, $K_2((\textbf{Servers}, \mathit{Server\_get})) = 0$ and $K_2(B) = 0$ otherwise, and $K_3((\textbf{Clients}, \mathit{Client\_waiting})) = 0$, $K_3((\textbf{Servers}, \mathit{Server\_get})) = 1$ and $K_3(B) = 0$ otherwise. $K_1$ contributes to Equation (5.9):

$$r_{req} \min(v_C(t), v_S(t))$$

$K_2$ contributes:

$$r_{req} \min(v_{C_w \cdot C}(t), v_{C_w \cdot S}(t))$$

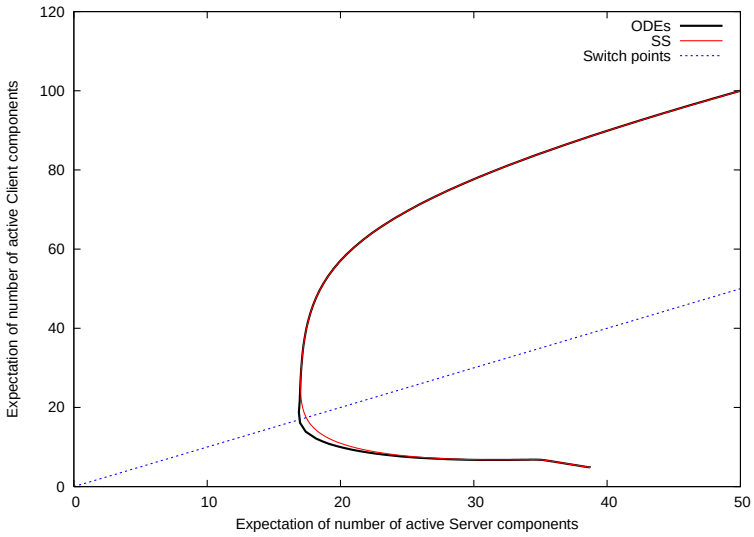



Fig. 10: Phase plot of expectations of *Client* components against *Server* components obtained through fluid analysis and through stochastic simulation for client–server model. Rates used are: $r_{req} = 2.0$, $r_{think} = 0.2$, $r_{break} = 0.1$, $r_{data} = 1.0$, $r_{reset} = 2.0$ and initial component counts are 100 *Client* and 50 *Server* components.

and $K_3$ contributes:

$$r_{req} \min(v_{S_g \cdot C}(t),\, v_{S_g \cdot S}(t))$$

So together, transition 1 contributes:

$$r_{req}(\min(v_C(t),\, v_S(t)) + \min(v_{C_w \cdot C}(t),\, v_{C_w \cdot S}(t)) + \min(v_{S_g \cdot C}(t),\, v_{S_g \cdot S}(t)))$$

For transition 2, $\mathcal{K}(\mathcal{J}_-, \mathcal{J}_+, \mathcal{M})$ is as for transition 1 and the contribution is similarly:

$$r_{data}(\min(v_{C_w}(t),\, v_{S_g}(t)) - \min(v_{C_w^2}(t),\, v_{C_w \cdot S_g}(t)) - \min(v_{C_w \cdot S_g}(t),\, v_{S_g^2}(t)))$$

For transition 5, $\mathcal{K}(\mathcal{J}_-, \mathcal{J}_+, \mathcal{M})$ consists only of $K_2$ as defined above, and the contribution is thus just:

$$-r_{break} v_{C_w \cdot S_g}(t)$$

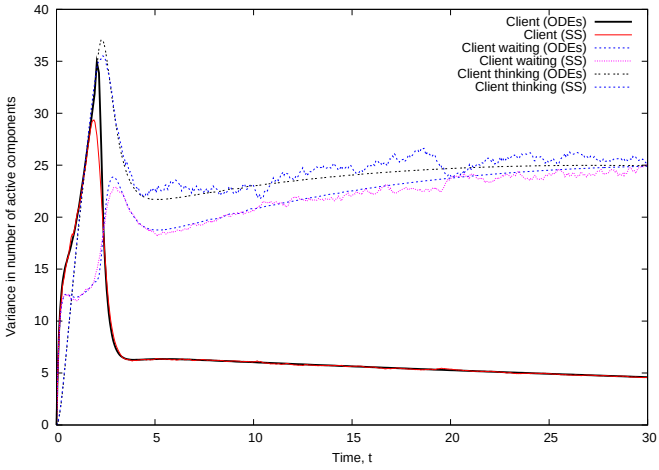
Collecting all of these individual contributions to form the ODE of Equation (5.9) for $\dot{v}_{C_w \cdot S_g}(t)$ gives:

$$\begin{aligned} \dot{v}_{C_w \cdot S_g}(t) = {} & r_{req}(\min(v_C(t),\, v_S(t)) + \min(v_{C_w \cdot C}(t),\, v_{C_w \cdot S}(t)) + \min(v_{S_g \cdot C}(t),\, v_{S_g \cdot S}(t))) \\ & + r_{data}(\min(v_{C_w}(t),\, v_{S_g}(t)) - \min(v_{C_w^2}(t),\, v_{C_w \cdot S_g}(t)) - \min(v_{C_w \cdot S_g}(t),\, v_{S_g^2}(t))) \\ & - r_{break} v_{C_w \cdot S_g}(t) \end{aligned} \tag{6.1}$$

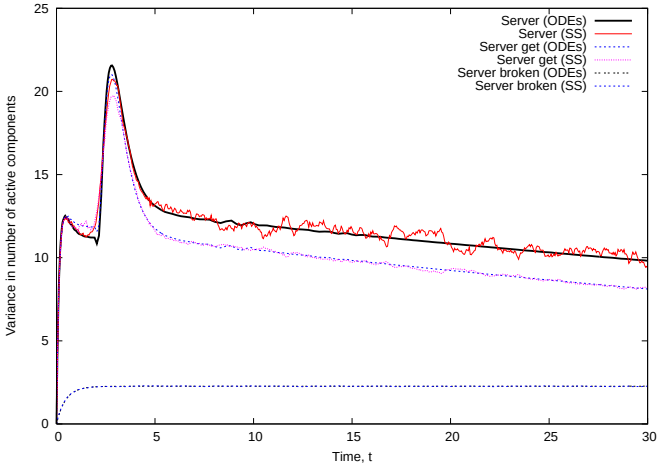
Repeating this procedure for the first-order and other joint second order moments yields a complete system of 27 piecewise-linear ODEs, which uniquely determine the 27 approximations.

### *6.3. Results for first-order moments*

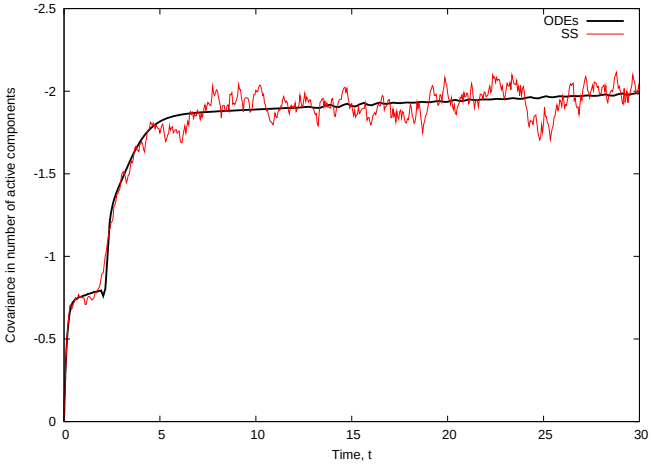
Figure 9 shows a comparison of component count expectations for this model generated by repeated stochastic simulation and the first-order moment fluid approximation. We see that the correspondence is so close that the fluid approximation almost overlays the stochastic simulation result for all components.




(a) Variance of component counts for the **Clients** group.




(b) Variance of component counts for the **Servers** group.




(c) Covariance of *Server* and *Server_broken* component counts.

Fig. 11: Comparison of variances and covariances obtained through fluid analysis with those obtained through stochastic simulation for client–server model. Rates used are: $r_{req} = 2.0$, $r_{think} = 0.2$, $r_{break} = 0.1$, $r_{data} = 1.0$, $r_{reset} = 2.0$ and initial component counts are 100 *Client* and 50 *Server* components.

It is possible however to discern a slight but visible quantitative discrepancy for the *Client*, *Client_waiting*, *Server* and *Server_get*, components approximately in the time interval $t \in [3, 4]$. It is clear that there is a switch point around this time interval for the term $\min(v_C(t), v_S(t))r_{req}$ (since there is a point at which the number of *Client* and *Server* components becomes equal). To see this more clearly, Figure 10 shows a phase plot of the expectation of the number of *Client* components against *Server* components, again computed both by fluid analysis and repeated stochastic simulation. The straight line is the line of switch points for this term. It is clear that the quantitative error is introduced as the switch point is approached, in line with the predictions of Section 4.2.1.

Furthermore, it is worth noting that Figure 10 suggests that the accuracy of the fluid approximation is restored again as the solution moves away from the switch point and approaches the steady state limit of the underlying CTMC. Indeed, if the steady state of the underlying CTMC is far from any switch points, it at least makes sense by the arguments of Section 4.2.1 that if the fluid approximation is started in the CTMC steady state, it will remain very close to it and thus perform well as an approximation for that configuration. If the fluid approximation always reaches the same long-time limit for any given initial configuration which has the same underlying (irreducible) CTMC, we are provided with intuition as to why the fluid approximation appears to improve in the long-time limit in, for example, Figure 10. Whether or not this one-to-one correspondence between the fixed point(s) of the approximating ODE system and the steady state of the CTMC does always hold is unresolved in general.

### *6.4. Results for second order moments: variance and covariance*

Figure 11 shows a comparison of component count variances and covariances for this model generated by repeated stochastic simulation (1,000,000 independent replications) and the second order moment fluid approximation. We see again impressive correspondence with all qualitative features of the (co)variances being exhibited by the fluid approximation. Again, visible quantitative discrepancy occurs roughly in the time interval $t \in [3, 4]$, around when the $\min(v_C(t), v_S(t))r_{req}$ switch point mentioned in the previous section is approached. This is of course also going to be close to a switch point for the second order terms such as $\min(v_{C_w \cdot C}(t), v_{C_w \cdot S}(t))r_{req}$ which occur in the differential equations for the higher order moments (see for example Equation (6.1)) and which will thus also contribute to this error.

It is interesting to note that the variance of all components apart from *Server_broken* appears to reach a local maximum around this switch point. This is a reasonable, if not even expected, outcome. As a switch point is approached, there is increasing stochastic uncertainty as to which side of the corresponding cooperation is fastest. Since this quantity determines the rate of the cooperation, it makes sense that this uncertainty leads to increased variability in the component counts affected by this cooperation (in this case all the components apart from *Server_broken*, since servers may break in any of their states).

Furthermore, we observe a completely negative covariance of the *Server* and *Server_broken* components. This is also to be expected since when the number of *Server* components is above its expected value, we would usually expect a corresponding negative difference in the count of *Server_broken* components.

## 7. Conclusion

We have described a systematic transformation of the stochastic process algebra, PEPA, to a system of ordinary differential equations, which measure the mean dynamic evolution of the PEPA model. In doing this, we have shown that the state space explosion, which inhibits performance analysis as it does functional analysis, can effectively be beaten for large classes of massively parallel models.

Importantly, we have established a relationship between the fluid approximation of the system of ODEs and the traditional underlying discrete performance model. We have shown that fluid-flow analysis is a good approximation of the mean of the transient distribution of components in a system. In some simple cases, the technique captures the mean precisely and for a much larger class of models it gives us a close approximation.

We have shown that for a large class of models, the fluid-flow approximation hinges on the accuracy of the statement, $\mathbb{E}[\min(X, Y)] \approx \min(\mathbb{E}[X], \mathbb{E}[Y])$, where $X$ and $Y$ are component rate random variables. This arises from the synchronisation of component groups in PEPA, although other similar approximations would also be needed to take account of synchronisation in, say, stochastic $\pi$-calculus. We are able to predict theoretically that the error will be at its worst when one component group changes from being the faster to the slower group within a cooperation. We have supported these predictions empirically using stochastic simulation. In conjunction with the complementary result [25], we can have confidence that this error cannot grow unboundedly in any finite time range.

For the first time for any synchronously communicating stochastic process algebra, we have formally derived a formulation for the second moment (and thus variance) of the evolution of a model. This is essential if we are to quantify the error in the mean fluid-flow approximation. Also and perhaps more importantly, this gives an idea of the sense in which the expectation is representative of the behaviour of the system, that is, how concentrated the probability mass is around the mean.

For the first time for any process algebra, we have also been able to extend the formulation to derive a fluid analysis for any higher moment or joint moment. The value of these contributions is highlighted by the fact that any simulation-based attempt at accurately obtaining quantities like variance, which are of relatively small magnitude, will require many thousands or even millions of independent replications. This is evident in Figure 11, where visible fluctuations are still present in the variances derived by stochastic simulation, even after 1,000,000 independent replications (an extremely time-consuming endeavour compared with the single integration of a small system of ODEs). Finally, we have extended the subset of PEPA that can be translated to a fluid model from that presented in [16, 25].

Through this work, we have a more precise understanding of what fluid-flow analysis of PEPA models means in terms of traditional performance measures. This allows performance modellers to use a powerful tool that permits analysis of massive state space models that were beyond the capability of traditional techniques. We aim to develop libraries which support these transformations and new fluid analysis techniques, for integration into existing popular toolsets, such as PEPA plugin [40], ipc [41] and Möbius [42].

*Acknowledgements*

The authors would like to thank Anton Stefanek, Luca Bortolussi and Jane Hillston for helpful insights and comments which helped put the work in context. We would also like to thank the referees for their time and efforts in reviewing the original submission and their helpful comments which have certainly improved the paper. Richard Hayden and Jeremy Bradley are supported by the Engineering and Physical Sciences Research Council on the AMPS project (reference EP/G011737/1).

## References

[1] J. Hillston, *A Compositional Approach to Performance Modelling*, vol. 12 of *Distinguished Dissertations in Computer Science*. Cambridge University Press, 1996.

[2] C. Priami, "A stochastic $\pi$-calculus," *The Computer Journal*, vol. 38(7), pp. 578–589, June 1995.

[3] M. Bernardo and R. Gorrieri, "Extended Markovian Process Algebra," in *CONCUR'96, Proceedings of the 7th International Conference on Concurrency Theory* (U. Montanari and V. Sassone, eds.), vol. 1119 of *Lecture Notes in Computer Science*, pp. 315–330, Springer-Verlag, Pisa, August 1996.

[4] H. Hermanns, *Interactive Markov Chains*. PhD thesis, Universität Erlangen–Nürnberg, July 1998.

[5] L. Bortolussi, "Stochastic concurrent constraint programming," in *QAPL'06, 4th Workshop on Quantitative Aspects of Programming Languages*, vol. 164 of *Electronic Notes in Theoretical Computer Science*, pp. 65–80, 2006.

[6] A. Duguid, "Coping with the parallelism of BitTorrent: Conversion of PEPA to ODEs in dealing with state space explosion," in *FORMATS'06, 4th Int. Conference on Formal Modeling and Analysis of Timed Systems*, vol. 4202 of *Lecture Notes in Computer Science*, pp. 156–170, Springer, 2006.

[7] P. Buchholz, "Hierarchical Markovian models: symmetries and reduction," *Performance Evaluation*, vol. 22, no. 1, pp. 93–110, 1995.

[8] V. Mertsiotakis, *Approximate analysis methods for stochastic process algebras*. PhD thesis, Universität Erlangen–Nürnberg, 1998.

[9] P. G. Harrison, "Turning back time in Markovian process algebra," *Theoretical Computer Science*, vol. 290, pp. 1947–1986, 2003.
[10] J. Hillston and N. Thomas, "Product form solution for a class of PEPA models," *Performance Evaluation*, vol. 35, no. 3–4, pp. 171–192, 1999.
[11] R. J. Boucherie, "A characterization of independence for competing Markov chains with applications to stochastic Petri nets," *IEEE Transactions on Software Engineering*, vol. 20, no. 7, pp. 536–544, 1994.
[12] P. Buchholz, "Exact and ordinary lumpability in finite markov chains," *Journal of Applied Probability*, vol. 31, pp. 59–75, March 1994.
[13] M. Siegle, "Using structured modelling for efficient performance prediction of parallel systems," in *Parallel Computing: Trends and Applications – Proc. of the Int. Conf. on Advances in Parallel Computing (Parco93)* (G. R. Joubert, D. Trystram, and F. J. Peters, eds.), pp. 453–460, North-Holland, May 1994.
[14] G. Chiola, "Manual and automatic exploitation of symmetries in SPN models," in *ICATPN'98, Proceedings of the International Conference on the Application and Theory of Petri Nets* (J. Desel and M. Silva, eds.), vol. 1420 of *Lecture Notes in Computer Science*, pp. 28–43, Springer-Verlag, 1998.
[15] S. T. Gilmore, J. Hillston, and M. Ribaudo, "An efficient algorithm for aggregating PEPA models," *IEEE Transactions on Software Engineering*, vol. 27, pp. 449–464, May 2001.
[16] J. Hillston, "Fluid flow approximation of PEPA models," in *QEST'05, Proceedings of the 2nd International Conference on Quantitative Evaluation of Systems*, (Torino), pp. 33–42, IEEE Computer Society Press, September 2005.
[17] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.
[18] J. T. Bradley, S. T. Gilmore, and J. Hillston, "Analysing distributed internet worm attacks using continuous state-space approximation of process algebra models," *Journal of Computer and System Sciences*, vol. 74, pp. 1013–1032, September 2008.
[19] L. Bortolussi and A. Policriti, "Stochastic concurrent constraint programming and differential equations," in *QAPL'07, 5th Workshop on Quantitative Aspects of Programming Languages*, vol. 190 of *Electronic Notes in Theoretical Computer Science*, pp. 27–42, September 2007.
[20] L. Bortolussi, "On the approximation of stochastic concurrent constraint programming by master equation," in *QAPL'08, 6th Workshop on Quantitative Aspects of Programming Languages*, (Budapest), March 2008.
[21] C. A. R. Hoare, "Communicating sequential processes," *Communications of the ACM*, vol. 21, pp. 666–677, August 1978.
[22] L. Cardelli, "From processes to ODEs by Chemistry," in *TCS 2008, Fifth IFIP International Conference on Theoretical Computer Science*, (Milan), Springer, 2008.
[23] L. Cardelli, "On process rate semantics," *Theoretical Computer Science*, vol. 391, pp. 190–215, February 2008.
[24] L. Cardelli, "A process algebra master equation," in *QEST 2007, Fourth International Conference on the Quantitative Evaluation of Systems*, (Edinburgh), IEEE, 2007.
[25] N. Geisweiller, J. Hillston, and M. Stenico, "Relating continuous and discrete PEPA models of signalling pathways," *Theoretical Computer Science*, vol. 404, pp. 97–111, November 2008.
[26] T. Kurtz, "Solutions of ordinary differential equations as limits of pure jump Markov processes," *Applied Probability*, vol. 7, pp. 49–58, April 1970.
[27] J.-Y. Le Boudec, D. McDonald, and J. Mundinger, "A generic mean field convergence result for systems of interacting objects," in *QEST'07, Proceedings of the 3rd IEEE Conference on the Quantitative Evaluation of Systems*, (Edinburgh), pp. 3–18, IEEE Computer Society, September 2007.
[28] M. Benaïm and J.-Y. Le Boudec, "A class of mean field interaction models for computer and communication systems," *Performance Evaluation*, vol. 65, no. 11–12, pp. 823–838, 2008.
[29] G. Horton, V. G. Kulkarni, D. M. Nicol, and K. S. Trivedi, "Fluid stochastic Petri nets: Theory, applications, and solution techniques," *European Journal of Operational Research*, vol. 105, no. 1, pp. 184–201, 1998.
[30] O. J. Boxma and V. Dumas, "The busy period in the fluid queue," *ACM SIGMETRICS'98, Proceedings of Joint International Conference on Measurement and Modeling of Computer Systems. SIGMETRICS Performance Evaluation Review*, vol. 26, no. 1, pp. 100–110, 1998.
[31] A. J. Field and P. G. Harrison, "An approximate compositional approach to the analysis of fluid queue networks," *Performance Evaluation*, vol. 64, pp. 1137–1152, September 2007.
[32] W. Whitt, *Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and their Application to Queues*. Springer Series in Operations Research and Financial Engineering, Springer, 2002.
[33] H. Bowman, J. W. Bryans, and J. Derrick, "Analysis of a multimedia stream using stochastic process algebras," *The Computer Journal*, vol. 44, no. 4, pp. 230–245, 2001.
[34] J.-M. Fourneau, L. Kloul, and F. Valois, "Performance modelling of hierarchical cellular networks using PEPA," *Performance Evaluation*, vol. 50, pp. 83–99, November 2002.
[35] N. Thomas, J. T. Bradley, and W. J. Knottenbelt, "Stochastic analysis of scheduling strategies in a GRID-based resource model," *IEE Software Engineering*, vol. 151, pp. 232–239, September 2004.
[36] D. R. W. Holton, "A PEPA specification of an industrial production cell," *The Computer Journal*, vol. 38(7), pp. 542–551, June 1995.
[37] J. T. Bradley, N. J. Dingle, S. T. Gilmore, and W. J. Knottenbelt, "Derivation of passage-time densities in PEPA models using ipc: the Imperial PEPA Compiler," in *MASCOTS'03, Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pp. 344–351, IEEE Computer Society, October 2003.
[38] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Van Nostrand, 1960.

[39] R. A. Hayden and J. T. Bradley, "Fluid semantics for passive stochastic process algebra cooperation," in *VALUETOOLS'08, Third International Conference on Performance Evaluation Methodologies and Tools*, (Athens), 2008.
[40] M. Tribastone, "The PEPA plug-in project," in *QEST'07, Proceedings of the 4th Int. Conference on the Quantitative Evaluation of Systems*, pp. 53–54, IEEE Computer Society, September 2007.
[41] J. T. Bradley and W. J. Knottenbelt, "The ipc/HYDRA tool chain for the analysis of PEPA models," in *QEST'04, Proceedings of the 1st IEEE Conference on the Quantitative Evaluation of Systems*, (University of Twente, Enschede), pp. 334–335, IEEE Computer Society, September 2004.
[42] D. D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. G. Webster, "The Möbius framework and its implementation," *IEEE Transactions on Software Engineering*, vol. 28, pp. 956–969, October 2002.

## A. Proofs

### *A.1. Proof of Theorem 2.15*

It is clear from the operational semantics of parallel cooperation that a single grouped PEPA model transition may evolve at most one component in each component group. Therefore if $\tilde{\mathcal{C}}(\tilde{G}', H, P) \neq \tilde{\mathcal{C}}(\tilde{G}, H, P)$, it must be the case that either $\tilde{\mathcal{C}}(\tilde{G}', H, P) = \tilde{\mathcal{C}}(\tilde{G}, H, P) - 1$ or $\tilde{\mathcal{C}}(\tilde{G}', H, P) = \tilde{\mathcal{C}}(\tilde{G}, H, P) + 1$. We begin with the former case.

Let $\tilde{G} \in ds(G)/\simeq_G$ and let $\overline{G} \in \tilde{G}$. Then let $\tilde{\mathcal{G}}' \subseteq ds(G)/\simeq_G$ be the set of all $\tilde{G}' \in ds(G)/\simeq_G$ such that $\tilde{\mathcal{C}}(\tilde{G}', H, P) = \tilde{\mathcal{C}}(\tilde{G}, H, P) - 1$. We require to show:

$$\sum_{\tilde{G}' \in \tilde{\mathcal{G}}'} q[\overline{G}, \tilde{G}', \alpha] = \sum_{\substack{Q \in \mathcal{B}(G, H) \\ Q \neq P}} p_{t(\alpha)}(P, Q) \mathcal{R}_\alpha(\overline{G}, H, P)$$

We proceed by induction on the structure of $\overline{G}$.

If $\overline{G} = H\{D\}$, $\mathcal{R}_\alpha(\overline{G}, H, P) = \mathcal{C}(H\{D\}, H, P)\, r_\alpha(P)$ by definition. Now there is exactly one $\tilde{G}' \in \tilde{\mathcal{G}}'$ with $q[\overline{G}, \tilde{G}', \alpha] \neq 0$ for each $Q \in \mathcal{B}(G, H)$, $Q \neq P$ with $p_\alpha(P, Q) \neq 0$. Also for any $\overline{G}' \in \tilde{G}'$, $q(\overline{G}, \overline{G}', \alpha) = p_\alpha(P, Q)\, r_\alpha(P)$.

Now by definition, $q[\overline{G}, \tilde{G}', \alpha] = \sum_{\overline{G}' \in \tilde{G}'} q(\overline{G}, \overline{G}', \alpha)$, and to each $P$ component in group $H$ of $\overline{G}$, there is exactly one $\overline{G}' \in \tilde{G}'$ for which $q(\overline{G}, \overline{G}', \alpha)$ is non-zero, so:

$$q[\overline{G}, \tilde{G}', \alpha] = \mathcal{C}(H\{D\}, H, P)\, p_\alpha(P, Q)\, r_\alpha(P)$$

as required.

If $\overline{G} = \overline{M}/L$ then by definition:

$$\mathcal{R}_\alpha(\overline{G}, H, P) := \begin{cases} 0 & \text{if } \alpha \in L \text{ ( and } \alpha \in \mathcal{A}) \\ \mathcal{R}_\alpha(\overline{M}, H, P) & \text{if } \alpha \notin L \text{ and } \alpha \in \mathcal{A} \\ \mathcal{R}_{\tau^\beta}(\overline{M}, H, P) & \text{if } \alpha = \tau^\beta \text{ and } \beta \notin L \\ \mathcal{R}_{\tau^\beta}(\overline{M}, H, P) + \mathcal{R}_\beta(\overline{M}, H, P) & \text{if } \alpha = \tau^\beta \text{ and } \beta \in L \end{cases}$$

Let $\tilde{\mathcal{M}}' \subseteq ds(\overline{M})/\simeq_{\overline{M}}$ be the set of all $\tilde{M}' \in ds(\overline{M})/\simeq_{\overline{M}}$ such that $\tilde{\mathcal{C}}(\tilde{M}', H, P) = \mathcal{C}(\overline{M}, H, P) - 1$. It is clear from the operational semantics that since $\overline{G} = \overline{M}/L$ then any $\overline{G}' \in \tilde{G}' \in ds(G)/\simeq_G$ has the form $\overline{G}' = \overline{M}'/L$ for some $\overline{M}' \in \tilde{M}' \in ds(\overline{M})/\simeq_{\overline{M}}$. So the elements of $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{G}}$ are in one-to-one correspondence, agreeing on component counts. Furthermore, the elements of each $\tilde{G}'$ and $\tilde{M}'$ are in one-to-one correspondence also.

If $\alpha \in L$, it is immediate that $q(\overline{G}, \overline{G}', \alpha) = 0$. If $\alpha \notin L$ and $\alpha \in \mathcal{A}$, or if $\alpha = \tau^\beta$ where $\beta \notin L$, it is immediate that $q(\overline{G}, \overline{G}', \alpha) = q(\overline{M}, \overline{M}', \alpha)$. Furthermore, we also then have $q[\overline{G}, \tilde{G}', \alpha] = q[\overline{M}, \tilde{M}', \alpha]$, and then since the elements of $\tilde{\mathcal{G}}'$ are in one-to-one correspondence with those of $\tilde{\mathcal{M}}'$ and they agree on component counts, the desired result follows by induction in all three of these cases.

Otherwise, if $\alpha = \tau^\beta$ where $\beta \in L$, it is clear from the operational semantics that:

$$q(\overline{G},\, \overline{G}',\, \tau^\beta) = q(\overline{M},\, \overline{M}',\, \tau^\beta) + q(\overline{M},\, \overline{M}';,\, \beta)$$

Furthermore, by the one-to-one correspondence between the elements of $\tilde{G}'$ and $\tilde{M}'$, and the one-to-one correspondence between those of $\tilde{\mathcal{G}}'$ and $\tilde{\mathcal{M}}'$:

$$\sum_{\tilde{G}' \in \tilde{\mathcal{G}}'} q[\overline{G},\, \tilde{G}',\, \tau^\beta] = \sum_{\tilde{M}' \in \tilde{\mathcal{M}}'} q[\overline{M},\, \tilde{M}',\, \tau^\beta] + \sum_{\tilde{M}' \in \tilde{\mathcal{M}}'} q[\overline{M},\, \tilde{M}',\, \beta]$$

The result then follows by induction and the definition of the component rate function.

If $\overline{G} = \overline{M} \underset{L}{\bowtie} \overline{R}$ then assume without loss of generality that $H \in \mathcal{G}(\overline{M})$. By definition:

$$\mathcal{R}_\alpha(\overline{G},\, H,\, P) = \begin{cases} \frac{\mathcal{R}_\alpha(\overline{M},\, H,\, P)}{r_\alpha(\overline{M})} \min(r_\alpha(\overline{M}),\, r_\alpha(\overline{R})) & \text{if } \alpha \in L \\ \mathcal{R}_\alpha(\overline{M},\, H,\, P) & \text{if } \alpha \notin L \end{cases}$$

Let $\tilde{\mathcal{M}}' \subseteq ds(\overline{M})/\simeq_{\overline{M}}$ be the set of all $\tilde{M}' \in ds(\overline{M})/\simeq_{\overline{M}}$ such that $\tilde{\mathcal{C}}(\tilde{M}',\, H,\, P) = \mathcal{C}(\overline{M},\, H,\, P) - 1$. Consider first the case $\alpha \notin L$. It is clear from the operational semantics that the following two sets:

$$\tilde{\mathcal{G}}^* := \{\tilde{G}' \in \tilde{\mathcal{G}}' \,:\, q[\overline{G},\, \tilde{G}',\, \alpha] \neq 0 \text{ and for any } \overline{G}' \in \tilde{G}',\, \overline{G}' = \overline{M}' \underset{L}{\bowtie} \overline{R}, \text{ for some } \overline{M}' \in ds(\overline{M})\}$$
$$\tilde{\mathcal{M}}^* := \{\tilde{M}' \in \tilde{\mathcal{M}}' \,:\, q[\overline{M},\, \tilde{M}',\, \alpha] \neq 0\}$$

are in one-to-one correspondence in the sense that if $\overline{G}' \in \tilde{G}' \in \tilde{\mathcal{G}}^*$ with $\overline{G}' = \overline{M}' \underset{L}{\bowtie} \overline{R}$, then $\overline{M}' \in \tilde{M}' \in \tilde{\mathcal{M}}^*$. Also, $q(\overline{G},\, \overline{G}',\, \alpha) = q(\overline{M},\, \overline{M}',\, \alpha)$. Furthermore, the elements of $\tilde{G}_i$ are also in one-to-one correspondence with those of $\tilde{M}'$ in the above sense, so $q[\overline{G},\, \tilde{G}',\, \alpha] = q[\overline{M},\, \tilde{M}',\, \alpha]$. Thus the result follows by induction and the definition of the component rate function.

For the case $\alpha \in L$, define $\tilde{\mathcal{M}}^*$ as before, but:

$$\tilde{\mathcal{G}}^* := \{\tilde{G}' \in \tilde{\mathcal{G}}' \,:\, q[\overline{G},\, \tilde{G}',\, \alpha] \neq 0\}$$

Consider $\overline{G}' \in \tilde{G}' \in \tilde{\mathcal{G}}^*$. From the operational semantics, we see that $\overline{G}' = \overline{M}' \underset{L}{\bowtie} \overline{R}'$ for some $R' \in ds(\overline{R})$ and $\overline{M}' \in \tilde{M}' \in \tilde{\mathcal{M}}^*$. Write also $\tilde{R}'$ for the element of $ds(\overline{R})/\simeq_{\overline{R}}$ to which $\overline{R}'$ belongs. We have from the operational semantics:

$$q(\overline{G},\, \overline{G}',\, \alpha) = \frac{q(\overline{M},\, \overline{M}',\, \alpha)}{r_\alpha(\overline{M})} \frac{q(\overline{R},\, \overline{R}',\, \alpha)}{r_\alpha(\overline{R})} \min(r_\alpha(\overline{M}),\, r_\alpha(\overline{R}))$$

and thus also:

$$\begin{aligned} q[\overline{G},\, \tilde{G}',\, \alpha] &= \sum_{\overline{G}' \in \tilde{G}'} q(\overline{G},\, \overline{G}',\, \alpha) \\ &= \sum_{\overline{M}' \in \tilde{M}'} \sum_{\overline{R}' \in \tilde{R}'} q(\overline{G},\, \overline{M}' \underset{L}{\bowtie} \overline{R}',\, \alpha) \\ &= \sum_{\overline{M}' \in \tilde{M}'} \sum_{\overline{R}' \in \tilde{R}'} \frac{q(\overline{M},\, \overline{M}',\, \alpha)}{r_\alpha(\overline{M})} \frac{q(\overline{R},\, \overline{R}',\, \alpha)}{r_\alpha(\overline{R})} \min(r_\alpha(\overline{M}),\, r_\alpha(\overline{R})) \\ &= \frac{q[\overline{M},\, \tilde{M}',\, \alpha]}{r_\alpha(\overline{M})} \frac{q[\overline{R},\, \tilde{R}',\, \alpha]}{r_\alpha(\overline{R})} \min(r_\alpha(\overline{M}),\, r_\alpha(\overline{R})) \end{aligned}$$

We now need to sum this expression over all elements $\tilde{G}' \in \tilde{\mathcal{G}}^*$. This is the same as summing over all elements $\tilde{M}' \in \tilde{\mathcal{M}}^*$ and all elements $\tilde{R}' \in ds(\overline{R})/\simeq_{\overline{R}}$ because this will definitely include all elements of $\tilde{\mathcal{G}}^*$

and for any additional ones also included, the above expression must be zero (or they would be in $\tilde{\mathcal{G}}^*$ by definition). Noting that $\sum_{\tilde{R}' \in ds(\overline{R})/\simeq_{\overline{R}}} q[\overline{R},\, \tilde{R}',\, \alpha] = r_\alpha(\overline{R})$ then gives the desired result by induction and completes the proof for the $\tilde{\mathcal{C}}(\tilde{G}',\, H,\, P) = \tilde{\mathcal{C}}(\tilde{G},\, H,\, P) - 1$ case.

The proof for the second case, when $\tilde{\mathcal{C}}(\tilde{G}',\, H,\, P) = \tilde{\mathcal{C}}(\tilde{G},\, H,\, P) + 1$, is very similar and is omitted for brevity.

*A.2. Proof of Theorem 4.3*

Let $S \subseteq \mathcal{E}(G)$ be the aggregated state space. For $s \in S$ write $p_s(t)$ as the transient probability of being in state $s$ at time $t$.

Consider the following sum, i.e. $\dot{\mathbb{E}}[N_{H,\, P}(t)]$:

$$\sum_{s_i \in S} \dot{p}_{s_i}(t) s_i(H,\, P)$$

This quantity can be constructed by summing over the left-hand sides of the individual Chapman–Kolmogorov forward equations for all $s \in S$ after first multiplying them each by $s(H,\, P)$. Therefore it is equal to the sum of the corresponding right-hand sides, also each multiplied by $s(H,\, P)$. We now aim to compute this quantity.

Consider those $\alpha$-transitions into each state $s \in S$ which decrease the count of $P$ components. By Theorem 2.15, these will together contribute the following term:

$$\sum_{s_i \in S} p_{s_i}(t)(s_i(H,\, P) - 1) \sum_{\substack{Q_j \in \mathcal{B}(G,\, H) \\ Q_j \neq P}} p_{\boldsymbol{t}(\alpha)}(P,\, Q_j) \mathcal{R}_\alpha(G,\, s_i,\, H,\, P)$$

Now consider those $\alpha$-transitions into each state $s \in S$ which increase the count of $P$ components. By Theorem 2.15, these will together contribute the following term:

$$\sum_{s_i \in S} p_{s_i}(t)(s_i(H,\, P) + 1) \sum_{\substack{Q_j \in \mathcal{B}(G,\, H) \\ Q_j \neq P}} p_{\boldsymbol{t}(\alpha)}(Q_j,\, P) \mathcal{R}_\alpha(G,\, s_i,\, H,\, Q)$$

The $\alpha$-transitions out of each state $s \in S$ which decrease the count of $P$ components will similarly together contribute the following term:

$$-\sum_{s_i \in S} p_{s_i}(t) s_i(H,\, P) \sum_{\substack{Q_j \in \mathcal{B}(G,\, H) \\ Q_j \neq P}} p_{\boldsymbol{t}(\alpha)}(P,\, Q_j) \mathcal{R}_\alpha(G,\, s_i,\, H,\, P)$$

Finally, the $\alpha$-transitions out of each state $s \in S$ which increase the count of $P$ components will together contribute the following term:

$$-\sum_{s_i \in S} p_{s_i}(t) s_i(H,\, P) \sum_{\substack{Q_j \in \mathcal{B}(G,\, H) \\ Q_j \neq P}} p_{\boldsymbol{t}(\alpha)}(Q_j,\, P) \mathcal{R}_\alpha(G,\, s_i,\, H,\, Q_j)$$

Combining now the left- and right-hand sides, we have:

$$\sum_{s_i \in S} \dot{p}_{s_i}(t) s_i(H,\, P) = \sum_{\alpha_i \in \mathcal{A}^\tau} \Bigg[ \sum_{s_j \in S} p_{s_j}(t)(s_j(H,\, P) - 1) \sum_{\substack{Q_k \in \mathcal{B}(G,\, H) \\ Q_k \neq P}} p_{\boldsymbol{t}(\alpha_i)}(P,\, Q_k) \mathcal{R}_{\alpha_i}(G,\, s_j,\, H,\, P)$$

$$+ \sum_{s_j \in S} p_{s_j}(t)(s_j(H,\, P) + 1) \sum_{\substack{Q_k \in \mathcal{B}(G,\, H) \\ Q_k \neq P}} p_{\boldsymbol{t}(\alpha_i)}(Q_k,\, P) \mathcal{R}_{\alpha_i}(G,\, s_j,\, H,\, Q_k)$$

$$- \sum_{s_j \in S} p_{s_j}(t) s_j(H,\, P) \sum_{\substack{Q_k \in \mathcal{B}(G,\, H) \\ Q_k \neq P}} p_{\boldsymbol{t}(\alpha_i)}(P,\, Q_k) \mathcal{R}_{\alpha_i}(G,\, s_j,\, H,\, P)$$

$$- \sum_{s_j \in S} p_{s_j}(t) s_j(H,\, P) \sum_{\substack{Q_k \in \mathcal{B}(G,\, H) \\ Q_k \neq P}} p_{\boldsymbol{t}(\alpha_i)}(Q_k,\, P) \mathcal{R}_{\alpha_i}(G,\, s_j,\, H,\, Q_k) \Bigg]$$

$$+ \ldots$$

where the omitted terms $(\ldots)$ are those due to all of the remaining transitions that do not involve changes in the number of $P$ components.

Noticing that all terms in $(\ldots)$ also cancel, we obtain:

$$\sum_{s_i \in S} \dot{p}_{s_i}(t) s_i(H,\, P) = \sum_{\alpha_i \in \mathcal{A}^\tau} \Bigg[ \sum_{s_j \in S} p_{s_j}(t) \sum_{\substack{Q_k \in \mathcal{B}(G,\, H) \\ Q_k \neq P}} p_{\boldsymbol{t}(\alpha_i)}(Q_k,\, P) \mathcal{R}_{\alpha_i}(G,\, s_j,\, H,\, Q)$$

$$- \sum_{s_j \in S} p_{s_j}(t) \sum_{\substack{Q_k \in \mathcal{B}(G,\, H) \\ Q_k \neq P}} p_{\boldsymbol{t}(\alpha_i)}(P,\, Q_k) \mathcal{R}_{\alpha_i}(G,\, s_j,\, H,\, P) \Bigg]$$

which, after simplification, is the desired result. □

*A.3. Proof of Theorem 5.3*

Let $\tilde{G}$, $\tilde{G}' \in ds(G)/\simeq_G$ be any two groupwise equivalence classes and let $\overline{G} \in \tilde{G}$. Let $\alpha \in \mathcal{A}^\tau$. We require to show that the sum of the rates of all $\alpha$-transitions from $\tilde{G}$ to $\tilde{G}'$, that is, $q[\overline{G},\, \tilde{G}',\, \alpha]$ is the sum of all quantities:

$$\prod_{(H,\, Q) \in \mathcal{J}_+} \left[ p_{\boldsymbol{t}(\alpha)}(\boldsymbol{h}(H,\, \mathcal{J}_-),\, Q) \right] \mathcal{R}_\alpha(\overline{G},\, \mathcal{J}_-) \tag{A.1}$$

taken over all $(\mathcal{J}_-,\, \mathcal{J}_+) \in \mathcal{J}(G)$, which characterise $\tilde{G}'$ in the sense that:

1. $\tilde{\mathcal{C}}(\tilde{G}',\, H,\, P) = \tilde{\mathcal{C}}(\tilde{G},\, H,\, P) - 1$ for all $(H,\, P) \in \mathcal{J}_- \setminus \mathcal{J}_+$,
2. $\tilde{\mathcal{C}}(\tilde{G}',\, H,\, P) = \tilde{\mathcal{C}}(\tilde{G},\, H,\, P) + 1$ for all $(H,\, P) \in \mathcal{J}_+ \setminus \mathcal{J}_-$ and
3. $\tilde{\mathcal{C}}(\tilde{G}',\, H,\, P) = \tilde{\mathcal{C}}(\tilde{G},\, H,\, P)$ for all $(H,\, P) \in \mathcal{B}(G) \setminus (\mathcal{J}_- \ominus \mathcal{J}_+)$.

We proceed by induction on the structure of $\overline{G}$.

If $\overline{G} = H\{D\}$ and there are $\alpha$-transitions between $\tilde{G}$ and $\tilde{G}'$, there are two possible cases. The first case is that $\tilde{G} \neq \tilde{G}'$ and there is some $P \neq Q \in \mathcal{B}(G,\, H)$, such that a $P$ component in $\overline{G}$ performs an $\alpha$-transition to become a $Q$ component in some $\overline{G}' \in \tilde{G}'$. In this case:

$$q[\overline{G},\, \tilde{G}',\, \alpha] = p_\alpha(P,\, Q)\, \mathcal{C}(H\{D\},\, H,\, P)\, r_\alpha(P) \tag{A.2}$$

The only $(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)$ characterising $\tilde{G}'$ is then given by $\mathcal{J}_- = \{(H, P)\}$ and $\mathcal{J}_+ = \{(H, Q)\}$. So by definition of the joint component rate function, Equation (A.2) agrees with the sum of Equation (A.1) terms, as required.

The second case is that $\tilde{G} = \tilde{G}'$ and there are possibly many $P \in \mathcal{B}(G, H)$ such that some $P$ component in $\overline{G}$ performs an $\alpha$-transition back to itself in $\overline{G}$. In this case:

$$q[\overline{G}, \tilde{G}', \alpha] = \sum_{P \in \mathcal{B}(G, H)} p_\alpha(P, P)\, \mathcal{C}(H\{D\}, H, P)\, r_\alpha(P) \tag{A.3}$$

The possible pairs $(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)$ which characterise $\tilde{G}'$ are now given by $\mathcal{J}_- = \mathcal{J}_+ = \{(H, P)\}$ for any $P \in \mathcal{B}(G, H)$. So by definition of the joint component rate function, Equation (A.3) also agrees with the sum of Equation (A.1) terms, as required.

If there are no $\alpha$-transitions between $\tilde{G}$ and $\tilde{G}'$ and $\tilde{G} \neq \tilde{G}'$, there can be no components $P, Q \in \mathcal{B}(G, H)$, such that $P \xrightarrow{(\alpha, \cdot)} Q$ and:

1. $\tilde{\mathcal{C}}(\tilde{G}', H, P) = \tilde{\mathcal{C}}(\tilde{G}, H, P) - 1$,
2. $\tilde{\mathcal{C}}(\tilde{G}', H, Q) = \tilde{\mathcal{C}}(\tilde{G}, H, Q) + 1$ and
3. $\tilde{\mathcal{C}}(\tilde{G}', H, R) = \tilde{\mathcal{C}}(\tilde{G}, H, R)$ for all $R \in \mathcal{B}(G, H)$, where $R \neq P$ and $R \neq Q$.

But Equation (A.1) is only non-zero for $(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)$, which characterise states, $\tilde{G}'$ where this is true.

If there are no $\alpha$-transitions between $\tilde{G}$ and $\tilde{G}'$ and $\tilde{G} = \tilde{G}'$, there can be no component $P \in \mathcal{B}(G, H)$, such that $P \xrightarrow{(\alpha, \cdot)} P$. As above, if $(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)$ characterises $\tilde{G}' = \tilde{G}$, $\mathcal{J}_- = \mathcal{J}_+ = \{(H, P)\}$ for some $P \in \mathcal{B}(G, H)$, so Equation (A.1) must be zero by definition.

If $\overline{G} = \overline{M}/L$ then by definition:

$$\mathcal{R}_\alpha(\overline{G}, \mathcal{J}_-) := \begin{cases} 0 & \text{if } \alpha \in L \text{ (and } \alpha \in \mathcal{A}) \\ \mathcal{R}_\alpha(\overline{M}, \mathcal{J}_-) & \text{if } \alpha \notin L \text{ and } \alpha \in \mathcal{A} \\ \mathcal{R}_{\tau^\beta}(\overline{M}, \mathcal{J}_-) & \text{if } \alpha = \tau^\beta \text{ and } \beta \notin L \\ \mathcal{R}_{\tau^\beta}(\overline{M}, \mathcal{J}_-) + \mathcal{R}_\beta(\overline{M}, \mathcal{J}_-) & \text{if } \alpha = \tau^\beta \text{ and } \beta \in L \end{cases}$$

Firstly, it is clear from the operational semantics that since $\overline{G} = \overline{M}/L$, any $\overline{G}' \in \tilde{G}'$ has the form $\overline{G}' = \overline{M}'/L$, and furthermore, $\overline{M} \in \tilde{M} \in ds(\overline{M})/\simeq_{\overline{M}}$ and $\overline{M}' \in \tilde{M}' \in ds(\overline{M})/\simeq_{\overline{M}}$. Furthermore, the pairs $\tilde{M}$, $\tilde{G}$ and $\tilde{M}'$, $\tilde{G}'$ have identical component counts.

For the case $\alpha \in L$, it is immediate that $q[\overline{G}, \tilde{G}', \alpha] = 0$, as required. If $\alpha \notin L$ and $\alpha \in \mathcal{A}$, or if $\alpha = \tau^\beta$ where $\beta \notin L$, it is immediate that $q[\overline{G}, \tilde{G}', \alpha] = q[\overline{M}, \tilde{M}', \alpha]$ and the result follows by induction and the definition of the joint component rate function, since the same pairs $(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)$ characterise $\tilde{G}'$ as do $\tilde{M}'$.

Otherwise, if $\alpha = \tau^\beta$ where $\beta \in L$, it is clear from the operational semantics that:

$$q[\overline{G}, \tilde{G}', \alpha] = q[\overline{M}, \tilde{M}', \tau^\beta] + q[\overline{M}, \tilde{M}', \beta]$$

The result then follows as in the previous case.

If $\overline{G} = \overline{M} \underset{L}{\bowtie} \overline{R}$ then by definition:

$$\mathcal{R}_\alpha(\overline{M} \underset{L}{\bowtie} \overline{R},\, \mathcal{J}_-) :=$$
$$\begin{cases} \frac{\mathcal{R}_\alpha(\overline{M},\, \mathcal{J}_-^{\overline{M}})}{r_\alpha(\overline{M})} \frac{\mathcal{R}_\alpha(\overline{R},\, \mathcal{J}_-^{\overline{R}})}{r_\alpha(\overline{R})} \min(r_\alpha(\overline{M}),\, r_\alpha(\overline{R})) & \text{if } \alpha \in L,\, \mathcal{J}_- \nsubseteq \mathcal{B}(\overline{M}) \\ & \quad \text{and } \mathcal{J}_- \nsubseteq \mathcal{B}(\overline{R}) \\ \mathcal{R}_\alpha(\overline{M},\, \mathcal{J}_-) & \text{if } \alpha \notin L,\, \mathcal{J}_- \subseteq \mathcal{B}(\overline{M}) \\ \mathcal{R}_\alpha(\overline{R},\, \mathcal{J}_-) & \text{if } \alpha \notin L,\, \mathcal{J}_- \subseteq \mathcal{B}(\overline{R}) \\ 0 & \text{otherwise} \end{cases}$$

where in the first line of the above definition, the set $\mathcal{J}_-$ is partitioned into $\mathcal{J}_-^{\overline{M}}$ and $\mathcal{J}_-^{\overline{R}}$, such that $\mathcal{J}_-^{\overline{M}} \subseteq \mathcal{B}(\overline{M})$ and $\mathcal{J}_-^{\overline{R}} \subseteq \mathcal{B}(\overline{R})$.

Now it is clear from the operational semantics that since $\overline{G} = \overline{M} \underset{L}{\bowtie} \overline{R}$, any $\overline{G}' \in \tilde{G}'$ has the form $\overline{G}' = \overline{M}' \underset{L}{\bowtie} \overline{R}'$. We also have that $\overline{M} \in \tilde{M} \in ds(\overline{M})/\simeq_{\overline{M}}$, $\overline{R} \in \tilde{R} \in ds(\overline{R})/\simeq_{\overline{R}}$, $\overline{M}' \in \tilde{M}' \in ds(\overline{M})/\simeq_{\overline{M}}$ and $\overline{R}' \in \tilde{R}' \in ds(\overline{R})/\simeq_{\overline{R}}$. Furthermore, $\tilde{G}$ agrees with $\tilde{M}$ respectively $\tilde{R}$ on the counts of all components in $\mathcal{B}(\overline{M})$ respectively $\mathcal{B}(\overline{R})$. The same is true for $\tilde{G}'$, $\tilde{M}'$ and $\tilde{R}'$.

Firstly, we consider the case $\alpha \notin L$. If both $\tilde{M} \neq \tilde{M}'$ and $\tilde{R} \neq \tilde{R}'$, the operational semantics assert that there can be no such $\alpha$-transition. For any pairs $(\mathcal{J}_-,\, \mathcal{J}_+) \in \mathcal{J}(G)$ characterising $\tilde{G}'$, it is clear that $\mathcal{J}_-$ has elements in both $\mathcal{B}(\overline{M})$ and $\mathcal{B}(\overline{R})$, and thus the joint component rate function is zero by definition, as required.

In the case $\tilde{M} = \tilde{M}'$ and $\tilde{R} = \tilde{R}'$ (so $\tilde{G} = \tilde{G}'$), it is clear from the operational semantics that:

$$q[\overline{G},\, \tilde{G}',\, \alpha] = q[\overline{M},\, \tilde{M},\, \alpha] + q[\overline{R},\, \tilde{R},\, \alpha]$$

Furthermore, the only elements of $\mathcal{J}(G)$ characterising $\tilde{G}'$, for which the joint component rate function is not immediately zero is simply the union of the elements of $\mathcal{J}(\overline{M})$ and $\mathcal{J}(\overline{R})$, which characterise $\tilde{M}'$ and $\tilde{R}'$, respectively. The equality of $q[\overline{G},\, \tilde{G}',\, \alpha]$ and the sum of Equation (A.1) terms then follows by induction and the definition of the joint component rate function.

It remains to consider, without loss of generality, just the case $\tilde{M} \neq \tilde{M}'$ and $\tilde{R} = \tilde{R}'$. Then, the operational semantics assert that:

$$q[\overline{G},\, \tilde{G}',\, \alpha] = q[\overline{M},\, \tilde{M}',\, \alpha]$$

Now the only elements of $\mathcal{J}(G)$ characterising $\tilde{G}'$, for which the joint component rate function is not immediately zero are the elements of $\mathcal{J}(\overline{M})$, which characterise $\tilde{M}'$. The equality of $q[\overline{G},\, \tilde{G}',\, \alpha]$ and the sum of Equation (A.1) terms then follows again by induction and the definition of the joint component rate function.

Now it remains to consider the case $\alpha \in L$. Let $\{\overline{G}'_{ij} = \overline{M}'_i \underset{L}{\bowtie} \overline{R}'_j\}_{ij}$ enumerate the elements of $\tilde{G}'$, so $\tilde{M}' = \{\overline{M}'_i\}_i$ and $\tilde{R}' = \{\overline{R}'_j\}_j$ enumerate the elements of $\tilde{M}'$ and $\tilde{R}'$, respectively. Then the operational semantics assert:

$$q(\overline{G},\, \overline{G}'_{ij},\, \alpha) = \frac{q(\overline{M},\, \overline{M}'_i,\, \alpha)}{r_\alpha(\overline{M})} \frac{q(\overline{R},\, \overline{R}'_j,\, \alpha)}{r_\alpha(\overline{R})} \min(r_\alpha(\overline{M}),\, r_\alpha(\overline{R}))$$

Then we have:

$$\begin{aligned} q[\overline{G},\, \tilde{G}',\, \alpha] &= \sum_{\overline{G}'_{ij} \in \tilde{G}'} q(\overline{G},\, \overline{G}'_{ij},\, \alpha) = \sum_{\overline{M}'_i \in \tilde{M}'} \sum_{\overline{R}'_j \in \tilde{R}'} q(\overline{G},\, \overline{M}'_i \underset{L}{\bowtie} \overline{R}'_j,\, \alpha) \\ &= \sum_{\overline{M}'_i \in \tilde{M}'} \sum_{\overline{R}'_j \in \tilde{R}'} \frac{q(\overline{M},\, \overline{M}'_i,\, \alpha)}{r_\alpha(\overline{M})} \frac{q(\overline{R},\, \overline{R}'_j,\, \alpha)}{r_\alpha(\overline{R})} \min(r_\alpha(\overline{M}),\, r_\alpha(\overline{R})) \\ &= \frac{q[\overline{M},\, \tilde{M}',\, \alpha]}{r_\alpha(\overline{M})} \frac{q[\overline{R},\, \tilde{R}',\, \alpha]}{r_\alpha(\overline{R})} \min(r_\alpha(\overline{M}),\, r_\alpha(\overline{R})) \end{aligned}$$

Now let $\{(\mathcal{J}_-^{k,\,\tilde{M}},\, \mathcal{J}_+^{k,\,\tilde{M}})\}_k \in \mathcal{J}(\overline{M})$ characterise $\tilde{M}'$ and $\{(\mathcal{J}_-^{l,\,\tilde{M}},\, \mathcal{J}_+^{l,\,\tilde{M}})\}_l \in \mathcal{J}(\overline{R})$ characterise $\tilde{R}'$. Then the subset of $\mathcal{J}(G)$ characterising $\tilde{G}'$ is:

$$\mathcal{J}^{\tilde{G}} := \{(\mathcal{J}_-^{kl},\, \mathcal{J}_+^{kl}) \in \mathcal{J}(G) \,:\, \mathcal{J}_-^{kl} = \mathcal{J}_-^{k,\,\tilde{M}} \cup \mathcal{J}_-^{l,\,\tilde{R}} \text{ and } \mathcal{J}_+^{kl} = \mathcal{J}_+^{k,\,\tilde{M}} \cup \mathcal{J}_+^{l,\,\tilde{R}}\}_{kl}$$

By induction, we then have:

$$\begin{aligned} &q[\overline{M},\, \tilde{M}',\, \alpha]\, q[\overline{R},\, \tilde{R}',\, \alpha] = \\ &\qquad \sum_{k,\,l} \prod_{(H,\,Q) \in \mathcal{J}_+^{k,\,\tilde{M}} \cup \mathcal{J}_+^{l,\,\tilde{R}}} \Big[p_{\boldsymbol{t}(\alpha)}(\boldsymbol{h}(H,\, \mathcal{J}_-^{k,\,\tilde{M}} \cup \mathcal{J}_-^{l,\,\tilde{R}}),\, Q)\Big]\, \mathcal{R}_\alpha(\overline{M},\, \mathcal{J}_-^{k,\,\tilde{M}}) \mathcal{R}_\alpha(\overline{R},\, \mathcal{J}_-^{l,\,\tilde{R}}) \end{aligned}$$

The equality of $q[\overline{G},\, \tilde{G}',\, \alpha]$ and the sum of Equation (A.1) terms then follows by the definition of the joint component rate function.

□

*A.4. Proof of Theorem 5.4*

Let $S \subseteq \mathcal{E}(G)$ be the aggregated state space. For $s \in S$ write $p_s(t)$ as the transient probability of being in state $s$ at time $t$.

Consider the following sum, i.e. $\dot{\mathbb{E}}[M(t)]$:

$$\sum_{s_i \in S} \dot{p}_{s_i}(t) \prod_{B_j \in \mathcal{B}(G)} s_i(B_j)^{\mathcal{M}(B_j)}$$

Similarly to the proof of Theorem 4.3, this quantity can be constructed by summing over the left-hand sides of the individual Chapman–Kolmogorov forward equations for all $s \in S$ after first multiplying them each by $\prod_{B_i \in \mathcal{B}(G)} s(B_i)^{\mathcal{M}(B_i)}$. Therefore it is equal to the sum of the corresponding right-hand sides, also each multiplied by $\prod_{B_i \in \mathcal{B}(G)} s(B_i)^{\mathcal{M}(B_i)}$. We now aim to compute this quantity.

For $(\mathcal{J}_-,\, \mathcal{J}_+) \in \mathcal{J}(G)$, consider those $\alpha$-transitions into each state $s \in S$ which decrease the count of all components specified by $\mathcal{J}_- \setminus \mathcal{J}_+$, increase the count of all components specified by $\mathcal{J}_+ \setminus \mathcal{J}_-$ and do not modify the count of any components specified by $\mathcal{B}(G) \setminus (\mathcal{J}_- \ominus \mathcal{J}_+)$. By Theorem 5.3, these will together contribute the following term:

$$\rho_\alpha(\mathcal{J}_-,\, \mathcal{J}_+) \sum_{s_k \in S} \dot{p}_{s_k}(t)\, [F_+(s_k) F_-(s_k) F(s_k) \mathcal{R}_\alpha(G,\, s_k,\, \mathcal{J}_-)]$$

The $\alpha$-transitions out of each state $s \in S$ which decrease the count of all components specified by $\mathcal{J}_- \setminus \mathcal{J}_+$, increase the count of all components specified by $\mathcal{J}_+ \setminus \mathcal{J}_-$ and do not modify the count of any components specified by $\mathcal{B}(G) \setminus (\mathcal{J}_- \ominus \mathcal{J}_+)$, contribute by Theorem 5.3, the following term:

$$-\rho_\alpha(\mathcal{J}_-,\, \mathcal{J}_+) \sum_{s_k \in S} \dot{p}_{s_k}(t)\, [M(s_k) \mathcal{R}_\alpha(G,\, s_k,\, \mathcal{J}_-)]$$

where:

$$F_{\pm}(s) := \prod_{B_i \in \mathcal{J}_{\pm} \setminus \mathcal{J}_{\mp}} (s(B_i) \pm 1)^{\mathcal{M}(B_i)}$$

$$F(s) := \prod_{B_i \in \mathcal{B}(G) \setminus (\mathcal{J}_- \ominus \mathcal{J}_+)} s(B_i)^{\mathcal{M}(B_i)}$$

and:

$$M(s) := \prod_{B_i \in \mathcal{B}(G)} s(B_i)^{\mathcal{M}(B_i)}$$

Combining the left- and right-hand sides and noting that every transition in the aggregated state space corresponds to exactly one element of $(\mathcal{J}_-, \mathcal{J}_+) \in \mathcal{J}(G)$, gives the desired result. □

*A.5. Proof of Lemma 5.5*

By induction on the structure of $G$.

If $G = H\{D\}$, then the result is trivial since the joint component rate is simply a linear combination of component counts.

If $G = M/L$, the result is again trivial from the definition and by induction.

If $G = M_1 \underset{L}{\bowtie} M_2$, the only interesting case is when $\alpha \in L$ and $\mathcal{J} \nsubseteq \mathcal{B}(M_j)$ for $j = 1$ and 2, in which case:

$$\mathcal{R}_\alpha(M_1 \underset{L}{\bowtie} M_2, E, \mathcal{J}) = \frac{\mathcal{R}_\alpha(M_1, E, \mathcal{J}_1)}{r_\alpha(M_1, E)} \frac{\mathcal{R}_\alpha(M_2, E, \mathcal{J}_2)}{r_\alpha(M_2, E)} \min(r_\alpha(M_1, E), r_\alpha(M_2, E))$$

where $\mathcal{J}$ is partitioned uniquely into (non-empty) $\mathcal{J}_1$ and $\mathcal{J}_2$, such that $\mathcal{J}_1 \subseteq \mathcal{B}(M_1)$ and $\mathcal{J}_2 \subseteq \mathcal{B}(M_2)$. By a similar induction argument on apparent rate and this induction, we see that both $\mathcal{R}_\alpha(M_j, E, \mathcal{J}_j)$ and $r_\alpha(M_j, E)$ have this homogeneity property for $j = 1, 2$, so:

$$\frac{\mathcal{R}_\alpha(M_j, E^*, \mathcal{J}_j)}{r_\alpha(M_j, E^*)} = \frac{\mathcal{R}_\alpha(M_j, E, \mathcal{J}_j)}{r_\alpha(M_j, E)}$$

for $j = 1, 2$. The result then follows since for any non-negative $a$, $b$ and $c$, $c \times \min(a, b) = \min(c \times a, c \times b)$.

## B. Systems of equations

*B.1. First and second order moment equations for processor/resource model*

$$
\begin{aligned}
\dot{\mathbb{E}}[P_0(t)] &= -\mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] + q\cdot\mathbb{E}[P_1(t)] \\
\dot{\mathbb{E}}[P_1(t)] &= \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] - q\cdot\mathbb{E}[P_1(t)] \\
\dot{\mathbb{E}}[R_0(t)] &= -\mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] + s\cdot\mathbb{E}[R_1(t)] \\
\dot{\mathbb{E}}[R_1(t)] &= \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] - s\cdot\mathbb{E}[R_1(t)] \\
\dot{\mathbb{E}}[P_0^2(t)] &= -2\cdot\mathbb{E}[\min(P_0^2(t)r_1,\, P_0(t)R_0(t)r_2)] + \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] \\
&\quad + 2\cdot\mathbb{E}[P_0(t)P_1(t)]\cdot q + \mathbb{E}[P_1(t)]\cdot q \\
\dot{\mathbb{E}}[P_0(t)P_1(t)] &= \mathbb{E}[\min(P_0^2(t)r_1,\, P_0(t)R_0(t)r_2)] - \mathbb{E}[\min(P_1(t)P_0(t)r_1,\, P_1(t)R_0(t)r_2)] \\
&\quad - \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] - \mathbb{E}[P_0(t)P_1(t)]\cdot q + \mathbb{E}[P_1^2(t)]\cdot q - \mathbb{E}[P_1(t)]\cdot q \\
\dot{\mathbb{E}}[P_0(t)R_0(t)] &= -\mathbb{E}[\min(P_0^2(t)r_1,\, P_0(t)R_0(t)r_2)] - \mathbb{E}[\min(R_0(t)P_0(t)r_1,\, R_0^2(t)r_2)] \\
&\quad + \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] + \mathbb{E}[P_1(t)R_0(t)]\cdot q + \mathbb{E}[P_0(t)R_1(t)]\cdot s \\
\dot{\mathbb{E}}[P_0(t)R_1(t)] &= \mathbb{E}[\min(P_0^2(t)r_1,\, P_0(t)R_0(t)r_2)] - \mathbb{E}[\min(R_1(t)P_0(t)r_1,\, R_1(t)R_0(t)r_2)] \\
&\quad - \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] + \mathbb{E}[P_1(t)R_1(t)]\cdot q - \mathbb{E}[P_0(t)R_1(t)]\cdot s \\
\dot{\mathbb{E}}[P_1^2(t)] &= 2\cdot\mathbb{E}[\min(P_1(t)P_0(t)r_1,\, P_1(t)R_0(t)r_2)] + \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] \\
&\quad - 2\cdot\mathbb{E}[P_1^2(t)]\cdot q + \mathbb{E}[P_1(t)]\cdot q \\
\dot{\mathbb{E}}[P_1(t)R_0(t)] &= -\mathbb{E}[\min(P_1(t)P_0(t)r_1,\, P_1(t)R_0(t)r_2)] + \mathbb{E}[\min(R_0(t)P_0(t)r_1,\, R_0^2(t)r_2)] \\
&\quad - \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] - \mathbb{E}[P_1(t)R_0(t)]\cdot q + \mathbb{E}[P_1(t)R_1(t)]\cdot s \\
\dot{\mathbb{E}}[P_1(t)R_1(t)] &= \mathbb{E}[\min(P_1(t)P_0(t)r_1,\, P_1(t)R_0(t)r_2)] + \mathbb{E}[\min(R_1(t)P_0(t)r_1,\, R_1(t)R_0(t)r_2)] \\
&\quad + \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] - \mathbb{E}[P_1(t)R_1(t)]\cdot q - \mathbb{E}[P_1(t)R_1(t)]\cdot s \\
\dot{\mathbb{E}}[R_0^2(t)] &= -2\cdot\mathbb{E}[\min(R_0(t)P_0(t)r_1,\, R_0^2(t)r_2)] + \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] \\
&\quad + 2\cdot\mathbb{E}[R_0(t)R_1(t)]\cdot s + \mathbb{E}[R_1(t)]\cdot s \\
\dot{\mathbb{E}}[R_0(t)R_1(t)] &= \mathbb{E}[\min(R_0(t)P_0(t)r_1,\, R_0^2(t)r_2)] - \mathbb{E}[\min(R_1(t)P_0(t)r_1,\, R_1(t)R_0(t)r_2)] \\
&\quad - \mathbb{E}[\min(P_0(t)r_1,\, R_0(t)r_2)] - \mathbb{E}[R_0(t)R_1(t)]\cdot s + \mathbb{E}[R_1^2(t)]\cdot s - \mathbb{E}[R_1(t)]\cdot s \\
\dot{\mathbb{E}}[R_1^2(t)] &= 2\cdot\mathbb{E}[\min(R_1(t)P_0(t)r_1,\, R_1(t)R_0(t)r_2)] + \mathbb{E}[\min(P_0(t)]r_1,\, R_0(t)r_2)] \\
&\quad - 2\cdot\mathbb{E}[R_1^2(t)]\cdot s + \mathbb{E}[R_1(t)]\cdot s
\end{aligned}
$$