1. INTRODUCTION

This paper surveys some issues in the aggregate modelling of queueing networks, from the point of view of aggregation of states in continuous-time Markov chains (CTMC's). It reviews aggregation theory for CTMC's (Section 2) and gives applications of aggregation in queueing networks with product-form (Section 3) and without product-form (Section 4). Emphasis is placed upon the versatility of the aggregation, since any convenient state partition is permitted, and upon the usefulness of aggregation for specifying the form of the aggregate model, thereby identifying (aggregate) parameter estimation as the key technical issue.

2. AGGREGATION IN MARKOV CHAINS

2.1. Introduction

The theory of exact aggregation in finite, stationary, Markov chains, in discrete or continuous time, was introduced by Takahashi [1] [2]. We consider here an N-state irreducible CTMC with state space \( \Omega \equiv \{1,2,\ldots,N\} \), transition rates \( \{\lambda_{ij}: i,j \in \Omega, i \neq j\} \) and unique equilibrium distribution \( \{\pi_i: i \in \Omega\} \), which is the unique solution to the Kolmogoroff equations

\[
\pi_i \sum_{j \in \Omega \setminus i} \lambda_{ij} = \sum_{j \in \Omega \setminus i} \pi_j \lambda_{ji}, \quad i \in \Omega, \quad \sum_{j \in \Omega} \pi_j = 1
\]

2.2 Exact Aggregation Theory

The N states are partitioned into \( \tilde{N} \) groups

\[
(2.2.1) \quad \Omega = \Omega(1) + \Omega(2) + \ldots + \Omega(\tilde{N}).
\]

The partition is arbitrary but in practice \( \tilde{N} \ll N \) and each group \( \Omega(\alpha) \) is defined by a special distinguishing property. Aggregate equilibrium probabilities are defined by

\[
(2.2.2) \quad \bar{\pi}_\alpha \equiv \sum_{j \in \Omega(\alpha)} \pi_j = \Pr[\text{in group } \Omega(\alpha)], \quad \alpha \in \tilde{N} \equiv \{1,2,\ldots,\tilde{N}\}.
\]

Then \( \bar{\pi} = [\bar{\pi}_\alpha] \) is the unique solution to the aggregate Kolmogoroff equations

\[
(2.2.3) \quad \bar{\pi}_\alpha \sum_{\beta \in \Omega(\alpha)} \bar{\lambda}_{\alpha\beta} = \sum_{\beta \in \Omega(\alpha)} \bar{\pi}_\beta \bar{\lambda}_{\beta\alpha}, \quad \alpha \in \tilde{N}, \quad \sum_{\beta=1}^{\tilde{N}} \bar{\pi}_\beta = 1
\]

where the aggregate transition rates are defined by
Disaggregation: Find the individual probabilities \{\pi_i, i \in \Omega(\alpha)\} within one group \(\Omega(\alpha)\) by solution of the \(|\Omega(\alpha)|\) (linear) disaggregation equations

\[
(2.2.5) \quad \pi_i \sum_{j \in \Omega/\alpha} \lambda_{ij} - \sum_{j \in \Omega(\alpha)/\alpha} \pi_j \lambda_{ji} = \sum_{\beta \in \Omega/\alpha} \pi_{\beta} \lambda_{\beta i}^{\#}, \quad i \in \Omega(\alpha)
\]

where \(\lambda_{\beta i}^{\#}\) is the transition rate from group \(\Omega(\beta)\) into state \(i\):

\[
(2.2.6) \quad \lambda_{\beta i}^{\#} = \sum_{j \in \Omega(\beta)} [\pi_j / \pi_{\beta}] \lambda_{ji}.
\]

Equations (2.2.3-6) are equivalent to (2.1.1). Aggregation (2.2.3-4) destroys the Markov property but, since the one-transition behavior is correctly captured by the \(\{\tilde{\lambda}_{\beta}\}\), permits exact computation of the aggregate equilibrium probabilities \(\{\tilde{\pi}_\alpha\}\). Takahashi was the first to note (cf. (2.2.4)) that exact aggregation requires weighting the rows by the conditional probabilities \([\pi_j / \tilde{\pi}_\beta]\) and summing over the columns in the group (see also [3. Theorem 1]).

2.3. Iterative Aggregation-Disaggregation (IAD)

In practice \(\tilde{\lambda}\) and \(\lambda^{\#}\) must be estimated since the right sides of (2.2.4) and (2.2.6) are unknown because \(\pi\) is unknown. Takahashi's iterative scheme starts with an estimate of \(\pi\), estimates \(\tilde{\lambda}\) and \(\lambda^{\#}\) from (2.2.4) and (2.2.6), then estimates \(\tilde{\pi}\) by solving (2.2.3), and obtains a (hopefully improved) estimate of \(\pi\) by solving (2.2.5), namely by disaggregating one group \(\Omega(\alpha)\) at a time. In many cases this iterative algorithm converges in a few score iterations, providing better and better estimates of \(\pi, \tilde{\pi}, \tilde{\lambda}\) and \(\lambda^{\#}\).

2.4. Approximate Aggregation

In a few cases (e.g., product-form queueing networks, see Section 3), the conditional probabilities \(\pi_j / \tilde{\pi}_\beta\) can be estimated perfectly, and the IAD converges in one iteration to the exact solution. In most cases, however, (e.g., non-product form queueing networks), several iterations of IAD are needed. In many realistic problems, the state space is so large that disaggregation is impractical and omitted; only the aggregate probabilities are of interest. The following one-pass approximation is the consequence: first estimate \(\{\tilde{\lambda}_{\alpha\beta}\}\) by any heuristic, then solve (2.2.3) for approximate \(\{\tilde{\pi}_\alpha\}\) and stop. Note that this approximation separates the two features of parameter estimation and of model solution.

2.5. Bounding the Aggregate Probabilities

In many cases, good bounds on \(\{\tilde{\lambda}_{\alpha\beta}\}\) can be obtained either heuristically, or by inspection of (2.2.4), e.g.

\[
(2.5.1) \quad \tilde{\lambda}_{\alpha\beta} \leq \max_{j \in \Omega(\beta)} \sum_{i \in \Omega(\alpha)} \lambda_{ji}
\]
These can lead in a straightforward way to bounds on the aggregate probabilities, e.g., an upper bound on $\pi_\alpha$ is obtained by replacing aggregate transition rates $\lambda_{cp}$ out of group $\Omega(\alpha)$ by their lower bounds, and all other aggregate transition rates by their upper bounds. See also [2], [4] and [5].

2.6. Nearly Completely-Decomposable Markov Chains [2], [6], [7], and [8]


2.7. Exact Aggregation-Disaggregation for Transient Behavior of Markov Chains


3. AGGREGATION IN PRODUCT-FORM QUEUEING NETWORKS

3.1. Introduction

These queueing networks have the feature that the conditional probabilities $\pi_j/\pi_B$ appearing in (2.2.4) are known exactly, so that an exact one-pass aggregation is possible.


See Section 4.2 for discussion of aggregation.

3.3. Other Aggregations

These are treated by suitable partitions of the state space. For example, aggregation of servers 1, 2 and 3 into a single server, thereby obtaining a reduced network [9], [10] and [11] is obtained by partitioning $((n_{1r}, n_{2r}, \ldots, n_{M_r}))_{r=1}^R$ onto $((n_{1r} + n_{2r} + n_{3r}, n_{4r}, \ldots, n_{M_r}))_{r=1}^R$. Similarly, aggregation of customer classes [3] is obtained by partitioning $(n_{ir})_{ir=1}^{MR}$ onto $\{n_i = \sum n_{ir} \}_{ir=1}^M$. These partitions may be nested, leading to coarser models.

4. AGGREGATION IN NON-PRODUCT-FORM NETWORKS

4.1. Introduction

In non-product-form networks, approximate aggregation (cf. Section 2.4) is employed because exact results are unavailable. Aggregation theory is useful because it reveals the types of parameters which are needed and permits us to separate parameter estimation from model solution.

In what follows, we consider a network with $M$ service centers and $R$ customer classes. The state description is $(z_1, z_2, \ldots, z_M)$ and behaves as a Markov chain in continuous time. Here $z_i$, the local state for center $i$, is given by $z_i = (n_{i1}, n_{i2}, \ldots, n_{iR}, \ell_i)$ where $n_{ir}$ is the number of class $r$ customers at center $i$ and $\ell_i$ is all additional local information including phase of service at each of the servers in center $i$, the exact allocation of the $n_{ir} = \sum n_{ir}$ customers among the various queues including their sequence, the phase of any external arrival process to center $i$, etc.
4.2. Single Server Decomposition

These decompositions examine one service center, say center 1, in the network [12]. The partition (2.2.1) here takes the form

$$\Omega = \bigcup_{z_1'} \Omega(z'_1)$$

where $\Omega(z'_1)$ is the set of states with $z_1 = z'_1$.

By examining the aggregate Kolmogoroff equations (2.2.3-4), we see that center 1 can be studied in isolation, with (as expected) the original description of the service process and internal transfers within center 1, but with a Poisson arrival rate $a_1(z'_1)$ that depends upon the full local state $z'_1$. This holds for any external arrival process, for any service discipline, for finite capacity cases (see Section 4.3), etc. This result is somewhat surprising (but convenient) because the successive interarrival times at a given center in a queueing network are generally not statistically independent.

This justifies one-server decompositions and includes the special result, for one class networks, that each server behaves as a birth-and-death process with queue-dependent arrival and departure rates. Note however that "behaves as" only implies equilibrium probabilities are computed correctly. It does not imply that $z_1$ truly behaves as a Markov chain. Note also that the arrival rate $a_1(z'_1)$ must depend on the full local state $z'_1$, including $\ell_1$, not merely upon $(n_{11}, n_{12}, \ldots, n_{1R})$ [2], [13]. However, for the special case of product-form networks, $\ell_1$ is absent and the arrival rate depends only upon $(n_{1R})$.

This in turn leads directly to Norton's theorem [12] for such networks.

4.3. Blocking Calculations

We consider here finite-capacity queueing networks with (production) blocking: a customer who has completed service cannot transfer to the next service center on his route if that center is full. Instead he waits at his current center, blocking it from serving others, until his next center is willing to accept him. A full description of the process requires a rule by which a filled-up center chooses among several blocked customers contending for acceptance.

The analysis of the case of tandem queues [14], [15] generalizes to arbitrary queueing networks as follows: Look at one service center at a time, say center 1, and partition as in (4.2.1), where the local information $\ell_1$ now also includes whether each server in center 1 is blocked or unblocked, and if blocked identifies the desired next center. Then a one-center model as in Section 4.2 is possible with the following changes and insights.

(i) The (normal) arrival rates depend upon the full local state $z_1$, including whether center 1 is currently blocked or not blocked.

(ii) When a server at center 1 completes service, there will be a (blocking) probability that it cannot eject its customer. This blocking probability is not a constant, as attempted in popular heuristics, but depends on the full local state $z_1$.

(iii) If center 1 contains a blocked customer, who is unable to transfer out, such a customer is described by a deblocking rate which depends on the full local state $z_1$. Since the local state $z_1$ evolves, this is more
complex than merely giving a static distribution of deblocking times.

(iv) If center 1 ejects a customer while full, there is a probability of simultaneous instantaneous arrival of a customer who was hitherto being blocked by center 1. This probability depends upon the full local state \( z_1 \) at server 1, and whether the ejection was normal (i.e., of a just-completed customer) or by deblocking. These deblocking arrivals to center 1 are ignored by many other models, which instead attempt to capture them by using a GI interarrival process to center 1 (rather than by a \( z_1 \)-dependent Poisson arrival process). The "exact" treatment by aggregation concepts will produce better accuracy in the congested case, where plateaus of deblocking transfers will occur. (The author has seen tandem-queueing cases where 40% of the total arrivals are due to deblocking.)

4.4. Buffer Overflow Calculations

Consider the case of one customer class with external arrivals to centers 1, 2, ..., \( M-1 \), which have finite space \( B_1, B_2, \ldots, B_{M-1} \). Overflow traffic is sent to server \( M \) which, for simplicity of exposition, has unlimited space. Aggregation onto server \( M \) alone provides a description of the overflow traffic.

The full state is \( (n_1, \ell_1, n_2, \ell_2, \ldots, n_M, \ell_M) \) where \( n_i \) and \( \ell_i \) are the population and service/external arrival phases at server \( i \). Employ the partition

\[
\Omega(n,\ell) = \bigcup_{n_i,\ell_i} \Omega(n_i,\ell_i)
\]

where \( \Omega(n,\ell) \) is the set of states with \( n_M = n \) and \( \ell_M = \ell \). According to (2.2.4), server \( M \) may be treated as an isolated server with given phase-type service distribution and a Poisson arrival rate (of overflow traffic)

\[
(4.4.2) \quad a_j(n,\ell) = \sum_{k=1}^{M-1} \sum \Pr[n_j = B_j, \ell_j = k | n_M = n, \ell_M = \ell] a_{jk}(B_j)
\]

where \( a_{jk}(B_j) \) is the arrival rate to server \( j \) when in service phase \( k \).

Note that \( a(n,\ell) \) depends upon both \( n \) and \( \ell \), and that this conclusion holds for parallel servers at each center and for any arrival processes (Poisson, phase type, etc.) or any service time distributions at the first \( M-1 \) servers. Note also that it is not appropriate to treat interarrival times to server \( M \) as coming from some GI distribution such as gamma. Aggregation theory specifies both the form of the arrival process (local-state dependent arrivals) as well as the numerical values via (4.4.2).

The analysis of the overflow traffic now reduces to the (hard) problem of estimating the conditional probabilities in (4.4.2). This can be done either heuristically or exactly (if \( M \) is not too big) because the equilibrium vectors

\[
(4.4.3) \quad \chi(n) = \chi(1) R^{-1} \quad n \geq 1
\]

(4.1B.1.5)
where $R$ is the minimal non-negative solution to a matrix quadratic equation.

This also leads to insights into the geometric tail probabilities at server $M$. The asymptotic form of (4.4.3) is

$$
\hat{\chi}(n) \approx f \, \rho(R)^n \quad n \to \infty \quad (\rho(R) < 1).
$$

Then the conditional probabilities in (4.4.2) are ultimately independent of $n$

$$
\tilde{a}(n, \ell) \to b(\ell) \quad \text{as} \quad n \to \infty. 
$$

If one further aggregates over the phase $\ell = \ell_M$ at server $M$, one sees that the population $n = n_M$ behaves, for large $n$, as a birth-and-death process with some asymptotically constant arrival rate

$$
\bar{b} = \lim_{n \to \infty} \frac{n}{\ell} \sum \tilde{a}(n, \ell) \Pr[\ell_M = \ell | n_M = n] 
$$

and constant service rate $\bar{\mu} = \lim_{n \to \infty} \Sigma[\text{service completion rate while in phase } \ell \text{ of service}] \Pr[\ell_M = \ell | n_M = n]$. Thus the queue length distribution at server $M$ has a geometric tail [17]

$$
\Pr[n_M = n] \sim (\bar{b}/\bar{\mu})^{n-n_0} \Pr[n_M = n_0], \quad n \gg n_0.
$$

While this result may be derived more directly from (4.4.4), with the identification $\bar{b}/\bar{\mu} = \rho(R)$, the derivation from aggregation theory has the appeal of relating a more intuitive description of a single-server $M/M/1$ queue with the less-intuitive description in terms of $\rho(R)$.

4.5 Replacing General Service Time Distributions by Exponentials [2, Section 9]

Consider a one-class queueing network with state $(n_1, \ell_1, n_2, \ell_2, \ldots, n_M, \ell_M)$ and routing matrix $P_{ij}$, $i \neq j$. If all phase information is discarded, the resulting aggregated Markov chain has an aggregate state $(n_1, n_2, \ldots, n_M)$, the same routing matrix, and an aggregate service rate at server $j$ given by

$$
\tilde{\mu}_j(n_1, n_2, \ldots, n_M) = \Sigma[\text{service completion rate at server } j \text{ while in local state } (n_j, \ell)] \Pr[\ell_j = \ell | n_1, n_2, \ldots, n_M].
$$

In general this depends upon the full aggregate state $(n_1, n_2, \ldots, n_M)$.

However, Marie [18] has shown that the approximation

$$
\tilde{\mu}_j(n_1, n_2, \ldots, n_M) \sim \bar{\mu}_j(n_j) \quad 1 \leq k \leq M
$$

is often very satisfactory, in which case the aggregate model reduces to a readily-solved product-form network. The hard part is estimation of $\bar{\mu}_j(n_j)$. If this approximation proves unsuitable, it can be improved (albeit with loss of product-form) by allowing the right side of (4.5.1) to depend not only on $n_j$ but also upon the population of nearby other servers which whom server $j$ is interacting strongly.
REFERENCES


