Journal of the Operations Research
Society of Japan
Vol. 37, No. 1, March 1994

ON COMPARISON OF ACCURACY OF APPROXIMATIONS IN THE CROSS AGGREGATION METHOD

Yu Song
Fukuoka Institute of Technology

Yukio Takahashi
Tokyo Institute of Technology

(Received June 26, 1992; Revised May 14, 1993)

Abstract The cross aggregation method was developed to get approximate values of stationary probabilities of large Markov chains such as ones used for analyses of queueing systems. In the method, a system is decomposed into several subsystems, and using the aggregation technique a family of approximate models are derived by grouping subsystems in various ways. Each model supplies a set of approximate values. For a given system, usually subsystems are defined in a natural way. However, in some cases, we have several alternatives to choose subsystems. The accuracy of the approximate values depends on the choice. So it is of great interest how we can choose subsystems to get more accurate approximate values. In this paper we propose three indices for roughly estimating the order of accuracy of the candidate approximate models, and test them for Kanban systems or tandem queueing systems with minimal blocking.

1 Introduction

Various approximation methods have been proposed to approximately evaluate performance measures of queueing systems with blocking, since analysis of such a system becomes incredibly difficult as the scale of the system grows large. The cross aggregation method (CRAM) to be discussed here is one of such approximation methods.

The CRAM was first proposed in [5] with an application to tandem queueing systems with communicaton blocking. In the application it decomposes the system into nodes, and provides a nested family of models to get approximate values of the stationary probabilities of the system in several different levels. In Level 1, an approximate model is constructed by looking at one node at a time (Figure 1a). Then a set of approximate values of the stationary probabilities is obtained from the model using the aggregation technique. In Level 2 two adjacent nodes are looked at a time (Figure 1b) and another set of approximate values is obtained. Similarly, Level-3 approximate model is derived by looking at three succesive nodes at a time (Figure 1c). Usually we may expect that the approximate values in higher levels are more accurate. Hence comparing the results of different levels, we can roughly estimate the magnitude of errors contained in approximate values.

Speaking more generally, in the CRAM we decompose a system into several subsystems and introduce a certain number of groups of subsystems so that the groups cover the whole system and are connected with each other like a chain. Usually, it is natural to choose nodes as such subsystems, but is not essential. For systems with communication blocking [5, 6], nodes are taken as the subsystems because the structure of the state spaces of the systems is simple. For systems with production blocking or minimal blocking (See Section 2), however, the structure of the state space of the system is very complicated. So nodes may not be the best choice for subsystems. As we have done in [3], we may take pseudo-nodes instead. Further, even for the same decomposition, there exist several ways of grouping the subsystems. In consequence, we have a lot of approximate models and sets of approximate values. So we are interested in the relation between the accuracy of the approximate values and the model selection.

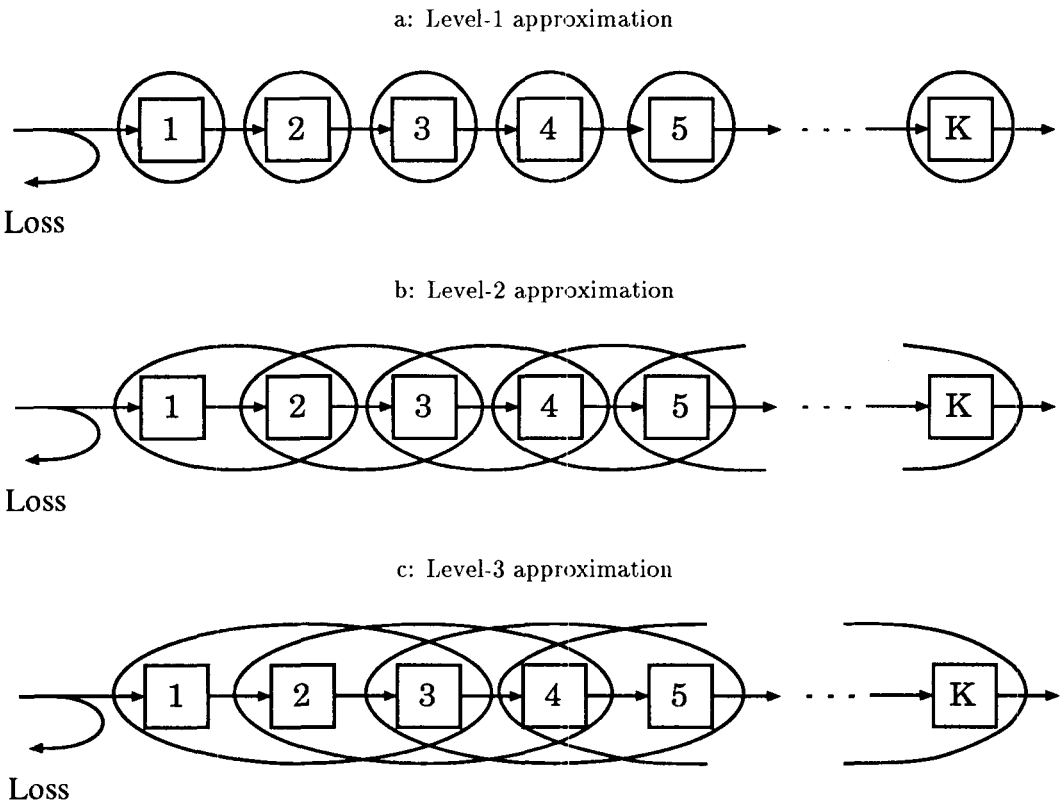

Figure 1: Approximate models for systems with communication blocking

In previous papers on the CRAM [2, 5, 6], we considered the number of aggregate states (variables) dealt with in the models as the most important factor affecting the accuracy of the approximate values. This is based on the expectation that the structure of the original system will be reflected with more fidelity in that of the approximate model with larger number of states. Contrary to our expectation, however, for a tandem queueing system with minimal blocking, decomposition into groups of pseudo-nodes provides better approximate values than decomposition into groups of physical nodes though it uses less number of states. This indicates that we must study the accuracy from other point of view.

In this paper, we study accuracy of the approximate models in connection with the dependency of subsystems in the model. For illustration, we introduce a tandem queueing system with minimal blocking in the next section. Then we select five approximate models in Section 3 and compare the numerical results by the CRAM for various cases in Section 4. In Section 5, we introduce three indices to estimate the order of accuracy of the approximate models. The first one is the ratio of the number of free variables used in the model to the number of free variables in the original system. This index is easy to understand but fails to distinguish accuracy between models derived by grouping subsystems according to physical nodes and psuedo-nodes. The second one is based on the ratio of the number of variables used in the aggregation of the model to the number of variables to be estimated. This index reflects well the order of accuracy among models tested. But it is not easy to calculate the index. The third one is derived in

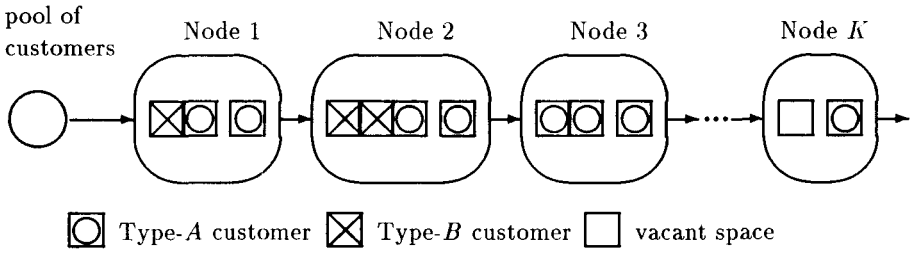

Figure 2: Tandem queues with minimal blocking

a heuristic way by considering the dependency among subsystems. It is difficult to explain its theoretical meaning, but it works well though it is simple and easy to calculate.

2 Tandem Queueing System with Minimal Blocking

In this section, we introduce a tandem queueing system with minimal blocking as an example of the ones having various approximate models. This is the same system as in [3].

We consider a K-node tandem queueing system described in Figure 2. Node k consists of an exponential server and a buffer of size $C_k - 1$, $k = 1, 2, \ldots, K$. At each node, customers are served under the FCFS rule. There always exist customers in front of Node 1 waiting for their services. After service completion at Node K, customers leave the system immediately. But movements after service completion at Node k ($\leq K$) are a little bit complicated. The following rule is known as "*minimal blocking*" discipline.

Upon service completion at Node k ($< K$), a customer immediately enters Node $k+1$ if there is a vacant space in Node $k+1$. Otherwise he is retained in Node k, and may exchange his place with a customer in the buffer waiting for service of Node k. For convenience, we name two typs of customers as Type A and Type B as follows.

Type-A customers are receiving or waiting for service of Node k.

Type-B customers are waiting for vacant spaces in Node $k+1$ after completion of their services at Node k.

The server at Node k is blocked by a Type-B customer if there is neither Type-A customer nor vacant space in Node k at the service completion. Otherwise, the Type-B customer must retire to the buffer and release the server for a new service. When a vacant space appears in Node $k+1$, a Type-B customer proceeds to Node $k+1$ in a First-Retained-First-Released manner. (Under the *production blocking* discipline a Type-B customer *must* block the server of Node k if there is no vacant space in Node $k+1$ at his service completion even if there is a vacant space or a Type-A customer in Node k.)

Let

v_k: number of vacant spaces in Node k,

b_k: number of Type-B customers in Node k.

Note the following:

- Since service distributions in all nodes are exponential and capacity of Node k is C_k, the local state of Node k is represented by the vector (v_k, b_k) such that $v_k + b_k \leq C_k$.

- It is impossible that both vacant space in Node $k+1$ and Type-B customer in Node k exist simultaneously, so $b_k \times v_{k+1} = 0$.
- $v_1 = 0$ because there always exist some customers waiting in front of Node 1.
- $b_K = 0$ because a customer departs from the sytsem as soon as he finishes his service in Node K.

Then the state space of the whole system is given by

$$(1) \qquad S = \{(v_1, b_1; v_2, b_2; \ldots; v_K, b_K) \mid v_1 = b_K = 0,\ v_k + b_k \le C_k,\ k = 1, \ldots, K,\ b_k \times v_{k+1} = 0,\ k = 1, 2, \ldots, K-1\},$$

and we can define a Markov chain on S which describes the stochastic behaviour of the system.

3 Approximation Based on the CRAM

Here we consider five approximate models for the tandem queueing system introduced in Section 2, and conduct some numerical tests.

First we decompose the system into physical nodes and constitute a Level-2 model named Model 1. Secondly we decompose the system into pseudo-nodes and constitute a Level-2 approximate model named Model 2. This model is the same as the one used in [3]. Models 3 through 5 are based on a finer decomposition of the system in which each subsystem is an intersection of a physical node and a pseudo-node. Model 3 is a Level-2 approximate model, while Models 4 and 5 are different type models from Level-2 or Level-3 approximate models.

Detailed model descriptions are as follows.

Model 1. To decompose the system, it is natural to choose physical nodes as subsystems. For convenience, we represent the state of the whole system in the form $(v_1, b_1, \ldots, v_K, b_K)$, with $v_1 = b_K = 0$. Then the local state of Node k is given by (v_k, b_k) with $v_k + b_k \le C_k$, $k = 1, 2, \ldots, K$. We denote the stationary probability of the state $(v_1, b_1, \ldots, v_K, b_K)$ as $x(v_1, b_1, \ldots, v_K, b_K)$.

For this choice of subsystems, we arrange Nodes k and $k+1$ into a group, and look at the two nodes at a time, $k = 1, 2, \ldots, K-1$ (Figure 3a). Thus we construct a Level-2 approximate model. The aggregate variables here are

$$(2) \qquad \begin{aligned} &x_{k,k+1}(v_k, b_k, v_{k+1}, b_{k+1}) \\ &\quad = \sum_{j \ne k, k+1} \sum_{\ell \ne k, k+1} \sum_{v_j} \sum_{b_\ell} x(v_1, b_1, \ldots, v_k, b_k, v_{k+1}, b_{k+1}, \ldots, v_K, b_K), \\ &\qquad\qquad v_k + b_k \le C_k,\ v_{k+1} + b_{k+1} \le C_{k+1},\ b_k \times v_{k+1} = 0, \\ &\qquad\qquad k = 1, 2, \ldots, K-1. \end{aligned}$$

For each fixed k, applying the aggregation technique to aggregate variables $x_{k,k+1}(v_k, b_k, v_{k+1}, b_{k+1})$, we get simultaneous equations corresponding to (13) and (14) in Appendix. To make the aggregate equations solvable, we put the following assumption.

Assumption for Model 1:

$$(3) \qquad \begin{aligned} &x(v_1, b_1, \ldots, v_K, b_K) \\ &\quad \simeq x_{1,2}(v_1, b_1, v_2, b_2) x_{3|2}(v_3, b_3 \mid v_2, b_2) \cdots x_{K|K-1}(v_K, b_K \mid v_{K-1}, b_{K-1}), \end{aligned}$$

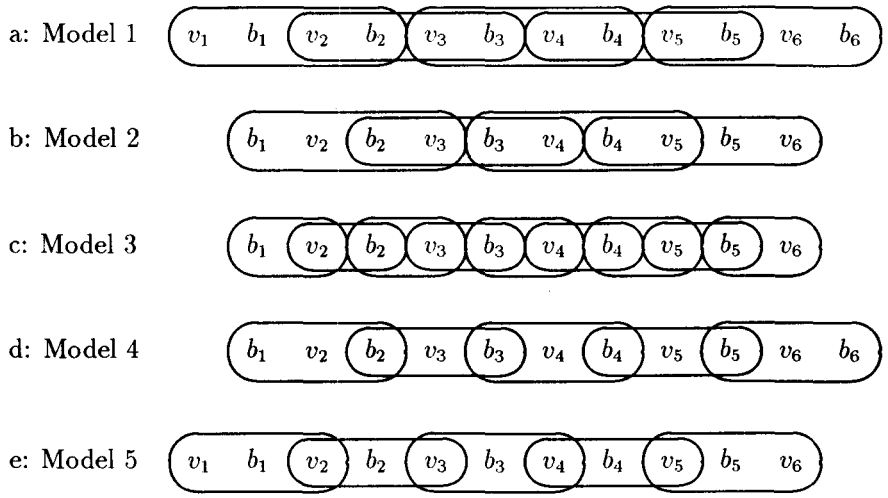

Figure 3: Approximate models

where conditional aggregate variables are defined as

$$x_{k+1|k}(v_{k+1}, b_{k+1} \mid v_k, b_k) = x_{k,k+1}(v_k, b_k, v_{k+1}, b_{k+1}) / \sum_{v'_{k+1}} \sum_{b'_{k+1}} x_{k,k+1}(v_k, b_k, v'_{k+1}, b'_{k+1}). \tag{4}$$

The assumption implies that the stochastic behaviour of a node depends on its neighbouring nodes, but is affected from the other nodes only through a chain of connecting nodes. The assumption (3) enables us to compute the conditional probabilities corresponding to (16), and hence aggregate transition rates corresponding to (15), from other aggregate variables $x_{\ell,\ell+1}(v_\ell, b_\ell, v_{\ell+1}, b_{\ell+1})$, $\ell \neq k$. By solving these $K-1$ sets of equations iteratively, we get approximate values of stationary probabilities.

Model 2. In this model we ignore the physical configuration of the system, and regard the system as the one composed of $K-1$ pseudo-nodes with local states described by the vector $(b_{k-1}, v_k), k = 2, 3, \ldots, K$. By decomposing the system into such pseudo-nodes, we can lead a Level-2 approximate model refered as Model 2 (Figure 3b). The aggregate variables here are

$$x(b_k, v_{k+1}, b_{k+1}, v_{k+2}) = \sum_{j \neq k,k+1} \sum_{\ell \neq k+1,k+2} \sum_{b_j} \sum_{v_\ell} x(v_1, b_1, \ldots, b_k, v_{k+1}, b_{k+1}, v_{k+2}, \ldots, v_K, b_K),$$
$$v_{k+1} + b_{k+1} \leq C_{k+1}, \; b_k \times v_{k+1} = 0, \; b_{k+1} \times v_{k+2} = 0,$$
$$k = 1, 2, \ldots, K-2.$$

Making an assumption corresponding to (3), we can calculate approximate values for the stationary probabilities of the sytsem.

To introduce a finer decomposition for Models 3 – 5, we take both $b_k, k = 1, 2, \ldots, K-1$, and $v_k, k = 2, 3, \ldots, K$, as state descriptions of subsystems.

Model 3. This is a Level-2 approximate model as indicated in Figure 3c. The aggregate

variables here are

$$x(b_k, v_{k+1}) = \sum_{j\neq k} \sum_{\ell\neq k+1} \sum_{b_j} \sum_{v_\ell} x(v_1, b_1, \ldots, b_k, v_{k+1}, \ldots, v_K, b_K),$$
$$b_k \times v_{k+1} = 0,$$
$$k = 1, 2, \ldots, K-1,$$

and

$$x(v_k, b_k) = \sum_{j\neq k} \sum_{\ell\neq k} \sum_{v_j} \sum_{b_\ell} x(v_1, b_1, \ldots, v_k, b_k, \ldots, v_K, b_K),$$
$$v_k + b_k \leq C_k,$$
$$k = 2, 3, \ldots, K-1.$$

Model 4. Arranging the subsystems into groups as shown in Figure 3d, we get Model 4. The aggregate variables are

$$x(b_k, v_{k+1}, b_{k+1}) = \sum_{j\neq k,k+1} \sum_{\ell\neq k+1} \sum_{b_j} \sum_{v_\ell} x(v_1, b_1, \ldots, b_k, v_{k+1}, b_{k+1}, \ldots, v_K, b_K),$$
$$b_k \times v_{k+1} = 0,\ v_{k+1} + b_{k+1} \leq C_{k+1},$$
$$k = 1, 2, \ldots, K-1.$$

Each group consists of three subsystems, but adjacent groups share only one subsystem. Hence Model 4 is neither a Level-2 model nor a Level-3 model.

Model 5. Here we arrange the subsystems into groups as shown in Figure 3e. The aggregate variables are

$$x(v_k, b_k, v_{k+1}) = \sum_{j\neq k} \sum_{\ell\neq k,k+1} \sum_{b_j} \sum_{v_\ell} x(v_1, b_1, \ldots, b_k, v_{k+1}, b_{k+1}, \ldots, v_K, b_K),$$
$$v_k + b_k \leq C_k,\ b_k \times v_{k+1} = 0,$$
$$k = 1, 2, \ldots, K-1.$$

In [2] we discussed an applicability condition for the CRAM on the structure of the state spaces of subsystems. It ensures the assumption like (3) does not lead a positive approximate value to an infeasible state $(v_1, b_1, \ldots, b_k, v_{k+1}, b_{k+1}, \ldots, v_K, b_K)$. From the condition, it is easy to check that Level-1 approximate models are not applicable for all the three decompositions of the system. The applicability of the CRAM to Model 2 has been shown in [2], and that to the other four models is easily checked.

In some cases, Models 1 and 2 might be too large to deal with. Models 3, 4 and 5 are devised to introduce smaller models than Models 1 or 2.

4 Numerical Results

For the five approximate models introduced in the last section, we numerically test more than 500 cases with $K = 6$ fixed and C_k and μ_k varied. Here we show some results of the test.

Table 1 shows throughput of 12 cases with $\mu_k = 1, k = 1, 2, \ldots, 6$, and varying C_k. It also shows relative errors of the approximate values. Tables 2 and 3 show throughput and relative errors of 18 cases with fixed C_k and varying μ_k.

The approximate values of the throughput are underestimating in almost all cases, except in Cases 17, 18, 26 and 27. In these exceptional cases, the service rates become large for servers in the middle of the system.

Table 1: Relative errors in throughput with $K = 6$, $\mu_k = 1$ $(k = 1, 2, ..., K)$

No.	$(C_1, \ldots, C_K)$	Exact	Error in %				
			Model 1	Model 2	Model 3	Model 4	Model 5
1	(1,1,1,1,1,1)	0.4667	-5.87	-2.23	-8.94	-7.34	-7.34
2	(2,1,1,1,1,1)	0.4805	-5.37	-1.92	-8.37	-6.67	-6.95
3	(1,2,1,1,1,1)	0.4965	-4.73	-1.78	-7.68	-5.86	-6.51
4	(1,1,2,1,1,1)	0.5078	-4.61	-1.75	-7.16	-5.62	-5.98
5	(1,1,1,2,1,1)	0.5078	-4.71	-1.78	-7.16	-5.95	-5.66
6	(1,1,1,1,2,1)	0.4965	-4.98	-1.75	-7.68	-6.49	-5.90
7	(1,1,1,1,1,2)	0.4805	-5.47	-1.95	-8.37	-6.95	-6.67
8	(1,1,2,2,1,1)	0.5453	-3.79	-1.37	-6.14	-4.83	-4.91
9	(1,2,2,1,2,1)	0.5823	-3.09	-1.38	-5.47	-4.12	-4.42
10	(1,2,2,2,2,1)	0.6158	-2.74	-1.21	-5.05	-3.79	-3.99
11	(1,2,3,2,2,1)	0.6354	-2.28	-1.03	-4.51	-3.25	-3.59
12	(1,3,2,2,3,1)	0.6600	-2.23	-1.07	-4.11	-3.05	-3.23

Table 2: Relative errors in throughput with $K = 6$, $C_k = 1$ $(k = 1, 2, ..., K)$

No.	$(\mu_1, \ldots, \mu_K)$	Exact	Error in %				
			Model 1	Model 2	Model 3	Model 4	Model 5
13	(1,2,3,4,5,6)	0.8350	-1.22	-0.20	-2.23	-1.58	-1.80
14	(6,5,4,3,2,1)	0.8350	-1.22	-0.19	-2.23	-1.80	-1.58
15	(1,1,2,2,1,1)	0.5500	-4.16	-0.65	-4.68	-4.41	-4.41
16	(1,1.5,2,2,1.5,1)	0.6429	-3.90	-0.55	-4.51	-4.19	-4.19
17	(1,2,3,3,2,1)	0.7213	-1.52	1.26	-0.01	-0.84	-0.84
18	(1,4,7,7,4,1)	0.8129	3.00	4.00	8.73	5.44	5.44
19	(3,2,1,1,2,3)	0.6203	-3.26	-0.85	-6.15	-4.62	-4.62
20	(1,1.5,1,1.5,1,1.5)	0.5491	-5.34	-1.78	-7.72	-6.41	-6.53
21	(1,2,1,2,1,2)	0.5903	-4.52	-1.14	-5.70	-4.97	-5.14

Table 3: Relative errors in throughput with $K = 6$, $C_1 = C_3 = C_5 = 1$, $C_2 = C_4 = C_6 = 2$

No.	$(\mu_1, \ldots, \mu_K)$	Exact	Error in %				
			Model 1	Model 2	Model 3	Model 4	Model 5
22	(1,2,3,4,5,6)	0.9217	-0.31	-0.04	-0.73	-0.49	-0.52
23	(6,5,4,3,2,1)	0.9287	-0.38	-0.04	-0.55	-0.46	-0.38
24	(1,1,2,2,1,1)	0.6570	-1.74	-0.36	-1.26	-1.34	-1.34
25	(1,1.5,2,2,1.5,1)	0.7626	-1.18	-0.01	-0.72	-0.79	-0.80
26	(1,2,3,3,2,1)	0.8282	1.17	1.42	3.46	2.41	2.45
27	(1,4,7,7,4,1)	0.8833	4.35	2.13	8.92	6.57	6.55
28	(3,2,1,1,2,3)	0.7245	-1.44	-0.27	-2.87	-2.12	-2.02
29	(1,1.5,1,1.5,1,1.5)	0.6526	-3.00	-1.01	-4.65	-3.74	-3.91
30	(1,2,1,2,1,2)	0.6856	-2.32	-0.48	-3.17	-2.73	-2.90

For all cases shown in the tables, relative errors of approximate values of the throughput do not exceed 6%, 4%, 9%, 7% and 7% in Models 1 through 5, respectively. This is also true for about 500 cases which did not appear in the tables.

From Tables 1 through 3, we see in most cases

$$e^{(2)} < e^{(1)} < \left\{ \begin{array}{c} e^{(4)} \\ e^{(5)} \end{array} \right\} < e^{(3)}, \tag{5}$$

where $e^{(i)}$ symbolically represents the magnitude of relative errors in Model i. We cannot judge clear superiority of Model 4 to Model 5 or vice versa. Both models are better than Model 3 and worse than Model 1.

In some cases this order is violated. In Cases 18 and 26, Model 2 is worse than Model 1, and in Case 17, Model 3 is the best among the five approximate models. By comparing results for Cases 15 through 18 and Cases 24 through 27, we see that this happens in the course of changing signs of the errors from minus to plus as the service rates of middle servers become large. Hence, we may consider the magnitude of errors in these models being basically subject to the order (5).

5 Indices of Accuracy

In this section, we propose three indices for accuracy of the models which are expected to roughly estimate the order of magnitudes of errors (5).

5.1 Index $\alpha^{(i)}$: The Number of Variables

The accuray of approximate values are expected to depend largely on the number of variables dealt with in a model. The more number of variables a model uses, the more information about the structure of the system is taken in, and the better the accuray would be. Based on this expectation, it is conjectured in [2, 5, 6] and in Section 1 that higher level approximations have better accuracy.

Table 4 lists $n^{(i)}$, the number of variables used in Model i, for Cases 1 through 12 in Table 1 together with the number of variables (states) $n^{(0)}$ of the original system. It is seen from Tables 1 and 4 that $e^{(1)} > e^{(2)}$ though $n^{(1)} > n^{(2)}$. And in Cases 1 through 9, $e^{(3)} > e^{(4)}, e^{(5)}$ though $n^{(3)} > n^{(4)}, n^{(5)}$. These conflict with our expectation.

It is easily checked on the aggregate equations (13) and (14) in Appendix that variables in CRAM have to satisfy some marginal constraints other than the total probability constraint. For example, aggregate variables $x_{k,k+1}(v_k, b_k, v_{k+1}, b_{k+1})$ of Model 1 in (2) satisfy the relation

$$\sum_{v_{k-1}} \sum_{b_{k-1}} x_{k-1,k}(v_{k-1}, b_{k-1}, v_k. b_k) = \sum_{v_{k+1}} \sum_{b_{k+1}} x_{k,k+1}(v_k, b_k, v_{k+1}, b_{k+1}).$$

Both sides are equal to the marginal probability $x_k(v_k, b_k)$.

Hence the number of variables which can take their values independently to other variables is less than the total number of variables. The number $m^{(i)}$ of such *free* variables, or the degree of freedom, is given by $n^{(i)}$ minus the number of constraints. In order to discuss accuracy of approximations, we might refer to $m^{(i)}$ rather than $n^{(i)}$. Table 5 lists $m^{(i)}$ of each model for Cases 1 through 12 together with $m^{(0)} = n^{(0)} - 1$, the number of *free* variables of the original system.

We shall consider the ratio

$$\alpha^{(i)} = m^{(i)}/m^{(0)}$$

Table 4: Number of variables

No.	$n^{(0)}$	$n^{(1)}$	$n^{(2)}$	$n^{(3)}$	$n^{(4)}$	$n^{(5)}$	No.	$n^{(0)}$	$n^{(1)}$	$n^{(2)}$	$n^{(3)}$	$n^{(4)}$	$n^{(5)}$
1	144	34	32	27	23	23	7	199	36	35	28	24	25
2	199	36	35	28	25	24	8	407	67	56	37	36	36
3	241	45	40	32	29	28	9	660	75	62	42	41	41
4	247	48	43	32	29	29	10	1037	99	78	47	49	49
5	247	48	43	32	29	29	11	1482	129	95	53	58	58
6	241	45	40	32	28	29	12	2160	139	102	59	64	64

Table 5: Number of free variables

No.	$m^{(0)}$	$m^{(1)}$	$m^{(2)}$	$m^{(3)}$	$m^{(4)}$	$m^{(5)}$	No.	$m^{(0)}$	$m^{(1)}$	$m^{(2)}$	$m^{(3)}$	$m^{(4)}$	$m^{(5)}$
1	143	21	22	10	14	14	7	198	23	25	11	15	16
2	198	23	25	11	16	15	8	406	48	42	16	25	25
3	240	29	29	13	19	18	9	659	53	48	19	29	29
4	246	32	31	13	19	19	10	1036	74	62	22	36	36
5	246	32	31	13	19	19	11	1481	100	77	26	44	44
6	240	29	29	13	18	19	12	2159	106	84	30	49	49

Table 6: Index $\alpha^{(i)}$

No.	$\alpha^{(1)}$	$\alpha^{(2)}$	$\alpha^{(3)}$	$\alpha^{(4)}$	$\alpha^{(5)}$	No.	$\alpha^{(1)}$	$\alpha^{(2)}$	$\alpha^{(3)}$	$\alpha^{(4)}$	$\alpha^{(5)}$
1	.147	.154	.070	.098	.098	7	.116	.126	.056	.076	.081
2	.116	.126	.056	.081	.076	8	.118	.103	.039	.062	.062
3	.121	.121	.054	.079	.075	9	.080	.073	.029	.044	.044
4	.130	.126	.053	.077	.077	10	.071	.060	.021	.035	.035
5	.130	.126	.053	.077	.077	11	.068	.052	.018	.030	.030
6	.121	.121	.054	.075	.079	12	.049	.039	.014	.023	.023

Table 7: Index $\beta^{(i)}$

No.	$\beta^{(1)}$	$\beta^{(2)}$	$\beta^{(3)}$	$\beta^{(4)}$	$\beta^{(5)}$	No.	$\beta^{(1)}$	$\beta^{(2)}$	$\beta^{(3)}$	$\beta^{(4)}$	$\beta^{(5)}$
1	.396	.496	.130	.222	.222	7	.340	.458	.108	.183	.200
2	.340	.458	.108	.200	.183	8	.354	.443	.085	.162	.162
3	.382	.462	.108	.203	.190	9	.338	.394	.068	.140	.142
4	.373	.465	.106	.189	.191	10	.316	.371	.053	.118	.118
5	.373	.465	.106	.191	.189	11	.307	.356	.046	.106	.107
6	.382	.462	.108	.190	.203	12	.298	.325	.039	.096	.096

as an index for accuracy of Model i. It is expected that the error $e^{(i)}$ is small if $\alpha^{(i)}$ is large. Table 6 lists $\alpha^{(i)}$. It shows that

$$\left\{ \begin{array}{c} \alpha^{(1)} \\ \alpha^{(2)} \end{array} \right\} > \left\{ \begin{array}{c} \alpha^{(4)} \\ \alpha^{(5)} \end{array} \right\} > \alpha^{(3)}. \tag{6}$$

This index seems to reflect the order of errors (5) rather well except that it does not indicate any superiority of Model 2 to Model 1.

5.2 Index $\beta^{(i)}$: Number of Variables in the Weights of Aggregate Equations

To investigate the relation $e^{(2)} < e^{(1)}$, we have to take a closer look at the aggregate equations (13) and (14) in Appendix. For deriving the aggregate equations, we have to approximate the weight $w_{\alpha i}$ in (16) and then calculate the coefficients $p^*_{\alpha\beta}$ in (15) of aggregate variables. The probabilistic meaning of the weight $w_{\alpha i}$ is the conditional state probability of state i of the system given the state α of the individual group of the model.

For example, the coefficients $p^*_{\alpha\beta}$ of the aggregate variable $x_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1})$ of Model 1 in (2) are given as weighted sums of transition probabilities with weights

$$\begin{aligned} w_{\alpha i} &= x_{\cdot|k,k+1}(v_1, b_1, \ldots, v_{k-1}, b_{k-1}, v_{k+2}, b_{k+2}, \ldots, v_K, b_K | v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1}) \\ &= x(v_1, b_1, \ldots, v_{k-1}, b_{k-1}, v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1}, v_{k+2}, b_{k+2}, \ldots, v_K, b_K) \\ &\qquad / x_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1}), \\ &\quad v_j + b_j \le C_j,\ v_{j+1} + b_{j+1} \le C_{j+1},\ b_j \times v_{j+1} = 0, \\ &\quad j = 1, 2, \ldots, k-1, k+2, \ldots, K-1, \end{aligned} \tag{7}$$

and the weights in turn can be approximated by products of conditional aggregate variables as

$$\begin{aligned} w_{\alpha i} \simeq\ & x_{1|2}(v_1, b_1 | v_2, b_2) \cdots x_{k-2|k-1}(v_{k-2}, b_{k-2} | v_{k-1}, b_{k-1}) \\ & \cdot x_{k-1|k}(v_{k-1}, b_{k-1} | v^*_k, b^*_k) \cdot x_{k+2|k+1}(v_{k+2}, b_{k+2} | v^*_{k+1}, b^*_{k+1}) \\ & \cdot x_{k+3|k+2}(v_{k+3}, b_{k+3} | v_{k+2}, b_{k+2}) \cdots x_{K|K-1}(v_K, b_K | v_{K-1}, b_{K-1}). \end{aligned} \tag{8}$$

Let $m^{(0)}_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1})$ be the number of *free* variables of the form (7), i.e., the number of variables of the form (7) minus 1 (the total probability constraint), and $m^{(1)}_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1})$ be the number of *free* conditional aggregate variables used in (8), i.e., the number of conditional aggregate variables used in (8) minus the number of constraints for total probabilities. $m^{(1)}_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1})$ is also calculated by subtracting the number of constraints for total probabilities and marginal probabilities from the number of aggregate variables used for defining the conditional aggregate variables in (8).

Then the ratio

$$\beta^{(1)}_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1}) = m^{(1)}_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1}) / m^{(0)}_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1})$$

is an index of the accuracy of the coefficients of $x_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1})$ in the aggregate equations, and the minimum of them

$$\beta^{(1)}_{k,k+1} = \min_{v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1}} \beta^{(1)}_{k,k+1}(v^*_k, b^*_k, v^*_{k+1}, b^*_{k+1})$$

is considered as an index of the accuracy of aggregate variables $x_{k,k+1}(v_k, b_k, v_{k+1}, b_{k+1})$ of Group k. Then the average

$$\beta^{(1)} = \frac{1}{g^{(1)}} \sum_{k=1}^{g^{(1)}} \beta_{k,k+1}^{(1)}$$

can serve as an index of accuracy of Model 1, where $g^{(1)}$ is the number of groups in Model 1.

We define index $\beta^{(i)}$ for other models in a similar manner.

Table 7 lists the values of $\beta^{(i)}$. If $\beta^{(i)}$ is large, the magnitude of errors $e^{(i)}$ is expected to be small. Table 7 shows that

$$\beta^{(2)} > \beta^{(1)} > \left\{ \begin{array}{c} \beta^{(4)} \\ \beta^{(5)} \end{array} \right\} > \beta^{(3)}, \tag{9}$$

and this entirely agrees with the order of magnitudes of errors (5).

$\beta^{(i)}$ seems to be an appropriate index for the accuracy of the approximate models, but unfortunately it is not easy to calculate. In order to get the value of $\beta^{(i)}$, we have to write a computer program which counts the number of free variables in (7) and (8). It is not difficult but not an easy task. The situation is almost the same for index $\alpha^{(i)}$. Our purpose of introducing an index is to know rough estimate of the magnitude of errors contained in the approximate models without conducting a test computation. So the index should be easily calculated. The next one is derived in a heuristic way, but it has the same order as $\beta^{(i)}$ in (9) even though its value can be calculated without a computer.

5.3 Index $\gamma^{(i)}$: Dependence of Subsystems among Groups

Index $\beta^{(i)}$ succeeds in distinguishing Models 1 and 2 by taking the structure of the model into account through conditional probabilities. Here we propose another index which reflects the magnitude of dependence among subsystems in a group.

In order to apply the CRAM, we have to introduce a chain of groups of subsystems according to a certain criterion, and make an assumption on the independence among subsystems in different groups. If the assumption is not far from the dependence among the subsystems, we may expect good approximation. In other words, the accuracy of approximation may be affected largely by how much the structure of the system is reflected in or taken into the approximate model under the assumption of independence.

In the approximate models considered in Section 3, the state space of a group is not a product space of individual state spaces of subsystems in the group. For instance, Table 8 shows the state space of Group 1 of Model 3. Symbol ○ represents a combination (b_1, v_2) of feasible state of the group, and × represents a combination of infeasible state. Among $(C_1 + 1) \times (C_2 + 1)$ combinations in the product space, only $C_1 + C_2 + 1$ are feasible. Similarly, Table 9 shows the state space of Group 2 of Model 3. In this case, out of $(C_2 + 1)^2$ combinations, $(C_2 + 2) \times (C_2 + 1)/2$ combinations are state descriptions of feasible states of the group.

Let $\gamma_k^{(i)}$ be the ratio of the number of infeasible states in the product space (the number of ×'s) to the number of elements in the product space (the sum of the numbers of ○'s and ×'s) of Group k of Model i. Namely,

$$\gamma_k^{(i)} = \frac{\text{number of infeasible states in Group } k \text{ of Model } i}{\text{number of combinations in the product space of Group } k \text{ of Model } i}$$

$$= 1 - \frac{\text{number of local states in Group } k \text{ of Model } i}{\text{number of combinations in the product space of Group } k \text{ of Model } i}.$$

Table 8: The state space of Group 1 of Model 3

	$b_1 = 0$	$b_1 = 1$	$\cdots$	$b_1 = C_1$
$v_2 = 0$	○	○	○	○
$v_2 = 1$	○	×	×	×
$\vdots$	○	×	×	×
$v_2 = C_2$	○	×	×	×

Table 9: The state space of Group 2 of Model 3

	$b_2 = 0$	$b_2 = 1$	$\cdots$	$b_2 = C_2$
$v_2 = 0$	○	○	○	○
$v_2 = 1$	○	○	○	×
$\vdots$	○	○	×	×
$v_2 = C_2$	○	×	×	×

Table 10: Index $\gamma^{(i)}$

No.	$\gamma^{(1)}$	$\gamma^{(2)}$	$\gamma^{(3)}$	$\gamma^{(4)}$	$\gamma^{(5)}$	No.	$\gamma^{(1)}$	$\gamma^{(2)}$	$\gamma^{(3)}$	$\gamma^{(4)}$	$\gamma^{(5)}$
1	.450	.500	.250	.350	.350	7	.458	.510	.259	.367	.358
2	.458	.510	.259	.358	.367	8	.517	.613	.309	.419	.419
3	.492	.545	.278	.383	.392	9	.567	.648	.336	.461	.461
4	.483	.556	.278	.383	.383	10	.600	.707	.370	.500	.500
5	.483	.556	.278	.383	.383	11	.617	.734	.387	.519	.519
6	.492	.545	.278	.392	.383	12	.642	.750	.401	.540	.540

We may consider that $\gamma_k^{(i)}$ indexes the structural dependence among subsystems in the group. If $\gamma_k^{(i)}$ is large, the subsystems in the group are insisted to depend on each other largely. Hence, we might consider that a large portion of the dependence between subsystems is took into the structure of the group, and expect that $\gamma_k^{(i)}$ roughly estimate the fitness between the indepence assumption and the structure of the system.

Define the average of $\gamma_k^{(i)}$ as

$$\gamma^{(i)} = \frac{1}{g^{(i)}} \sum_{k=1}^{g^{(i)}} \gamma_k^{(i)}$$

where $g^{(i)}$ is the number of groups in Model i. Then $\gamma^{(i)}$ is an index for evaluating the dependence among subsystems and hence it could serve as an index for the accuracy of Model i, too. If $\gamma^{(i)}$ is large, we expect the magnitude of errors being small.

Table 10 lists the values of $\gamma^{(i)}$ for the 12 cases in Table 1. It shows that

$$\gamma^{(2)} > \gamma^{(1)} > \left\{ \begin{array}{c} \gamma^{(4)} \\ \gamma^{(5)} \end{array} \right\} > \gamma^{(3)}. \tag{10}$$

$\gamma^{(i)}$ has the same order as $\beta^{(i)}$ in (9) and the reverse order to the magnitude of errors in (5). In this sense $\gamma^{(i)}$ can be a substitute of $\beta^{(i)}$ though it is derived in a heuristic manner and difficult to explain explicitly its direct relation to the magnitude of errors. We note that $\gamma^{(i)}$ is calculated from the number of local states and the calculation is easily achieved with paper and pencil.

6 Concluding Remarks

In the last section we introduced three indices $\alpha^{(i)}$, $\beta^{(i)}$ and $\gamma^{(i)}$ and showed numerically that $\beta^{(i)}$ and $\gamma^{(i)}$ have reverse order to the magnitude of errors in the approximate models for 12 cases listed in Table 1. The authors think that $\gamma^{(i)}$ will also work well in other systems where the state space is not the product space of the local state spaces. However to say this more affirmatively we need much more numerical experience.

If the state space of the system is the product space of the local state spaces as in tandem queueing systems with communication blocking, $\gamma^{(i)}$ does not work since it is identically equal to 0. Even in such a case $\alpha^{(i)}$ and $\beta^{(i)}$ will give us some information about the order of magnitude of errors. Of course in such a case $\alpha^{(i)}$ and $\beta^{(i)}$ are easily calculated.

These indices will provide only relative orders and not provide any quantitative information about the magnitude of errors. But they will help us greatly in the choice of approximate models in practical analyses of large-scale stochastic systems by the CRAM.

Appendix. The Aggregation Method

Let us consider an ergodic Markov chain $\boldsymbol{X}(t)$ on a state space $S = \{1, 2, \ldots, n\}$. We denote the transition probabilities by $p_{ij}, i, j = 1, 2, \ldots, n$, and the stationary state probabilities by $\pi_i, i = 1, 2, \ldots, n$. Then $\{\pi_i\}$ is given as a unique non-negative solution of the the equations

$$\pi_i = \sum_{j \in S} \pi_j p_{ji}, \quad i \in S, \tag{11}$$

$$\sum_{i \in S} \pi_i = 1. \tag{12}$$

Suppose that the state space is partitioned into N mutually exclusive subsets:

$$S = \bigcup_{\alpha=1}^{N} S_\alpha, \quad S_\alpha \bigcap S_\beta = \phi, \ \alpha \neq \beta.$$

We regard the set S_α as an *aggregate state*, and define an stochastic process $\boldsymbol{Y}(t)$ on the aggregate state space $S^* = \{1, 2, \ldots, N\}$ by

$$\boldsymbol{Y}(t) = \alpha \Longleftrightarrow \boldsymbol{X}(t) \in S_\alpha.$$

Though $\boldsymbol{Y}(t)$ may not be a Markov chain, it has stationary state probabilities given by

$$\pi^*_\alpha = \sum_{i \in S_\alpha} \pi_i.$$

It is easily checked that $\{\pi^*_\alpha\}$ satisfies the *aggregate equations*

$$\pi^*_\alpha = \sum_{\beta=1}^{N} \pi^*_\beta p^*_{\beta\alpha}, \quad \alpha = 1, 2, \ldots, N, \tag{13}$$

$$\sum_{\alpha=1}^{N} \pi^*_\alpha = 1. \tag{14}$$

where *aggregate transition probabilities* $p^*_{\alpha\beta}$ are given by

$$p^*_{\alpha\beta} = \sum_{i\in S_\alpha}\sum_{j\in S_\beta} \pi_i p_{ij}/\pi^*_\alpha = \sum_{i\in S_\alpha} w_{\alpha i} \sum_{j\in S_\beta} p_{ij}, \tag{15}$$

and the *weights* $w_{\alpha i}$ are given by

$$w_{\alpha i} = \pi_i/\pi^*_\alpha. \tag{16}$$

This means that the stationary state probabilities of $\boldsymbol{Y}(t)$ coincide with those of a Markov chain $\boldsymbol{Y}'(t)$ with transition probabilities $\{p^*_{\alpha\beta}\}$.

Note that if we know the weights $w_{\alpha i}$ in (16) in advance, we can get $\{\pi^*_i\}$ by solving the aggregate equations (13) and (14). The CRAM exploits this property. It makes the aggregate equations solvable by introducing an assumption on the independence among subsystems and approximating $w_{\alpha i}$ by using it.

Acknowledgements

The authors wish to thank the anonymous referees for their comments which improve the description of the paper greatly.

References

[1] SCHWEITZER, P.: Aggregation Methods for Large Markov Chains, in *Mathematical Computer Performance and Reliability*, (IAZEOLLA, G., COURTOIS, P. and HORDIJK, A., eds.), North Holland (1984). 275–286.

[2] SONG, Y. and TAKAHASHI, Y.: Aggregate Approximation for Tandem Queueing Systems with Production Blocking, *Journal of Operations Research Society of Japan*, Vol. 34 (1991). 329–354.

[3] SONG, Y. and TAKAHASHI, Y.: "Cross Aggregation Method for Kanban Systems," submitted for publication.

[4] TAKAHASHI, Y.: A Lumping Method for Numerical Calculation of Stationary Distributions of Markov chains, *Research Reports on Information Sciences* B-18, Department of Information Sciences, Tokyo Institute of Technology, Tokyo (1975).

[5] TAKAHASHI, Y.: A New Type Aggregation Method for Large Markov Chains and Its Application to Queueing Networks, in *Proceedings of ITC 11*, (AKIYAMA, M., ed.), North Holland (1985), 3.4 A.1.1–4.

[6] TAKAHASHI, Y.: Aggregate Approximation for Acyclic Queueing Networks with Communication Blocking, in *Queueing Networks with Blocking*, (PERROS, H. and ALTIOK, T., eds.), North Holland (1989), 33–46.

Yu SONG:
Department of Management Engineering
Fukuoka Institute of Technology
Wajiro-higashi, Higashi-ku
Fukuoka 811-02, Japan

Yukio TAKAHASHI:
Department of Information Science
Tokyo Institute of Technology
Oh-okayama, Meguro-ku
Tokyo 152, Japan