UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Iterative Methods for Queueing Networks with Irregular State-spaces

Raymond H. Chan

May 1992

CAM Report 92-21

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Iterative Methods for Queueing Networks with Irregular State-spaces

Raymond H. Chan

May 1992

CAM Report 92-21

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

Iterative Methods for Queueing Networks with Irregular State-spaces*

Raymond H. Chan†

March 25, 1992

Abstract

In this paper, we consider the problem of finding the steady-state probability distribution for Markovian queueing networks with overflow capacity. Our emphasis is on networks with non-rectangular state-spaces. The problem is solved by the preconditioned conjugate gradient method with preconditioners that can be inverted easily by using separation of variables. By relating the queueing problems with elliptic problems, and making use of results from domain decomposition for elliptic problems on irregular domains, we derive three different kinds of such separable preconditioners. Numerical results show that for these preconditioners, the numbers of iterations required for convergence increase very slowly with increasing queue size.

Abbreviated Title. Queueing Networks with Irregular State-spaces.

Key words. Preconditioned conjugate gradient method, queueing networks, domain decomposition, substructuring method, capacitance matrix method, additive Schwarz method.

AMS(MOS) subject classifications. 65N20, 65F10, 60K25.

*To appear in the *Proceedings of the IMA workshop on Linear Algebra, Markov Chains and Queueing Models, Minneapolis, Minnesota, January 12-17, 1992.*

†Department of Mathematics, University of Hong Kong, Hong Kong. Research supported in part by ONR contract no. N00014-90-J-1695 and DOE grant no. DE-FG03-87ER25037.

1 Introduction

In this paper, we are interested in finding the steady-state probability distribution for Markovian queueing networks with overflow capacity. From the steady-state probability distribution, we can compute for examples, the blocking probability of the network, the probability of overflow from one queue to another and the waiting time for customers in various queues.

In matrix terms, finding the steady-state probability distribution is equivalent to finding a right null-vector $\mathbf{p}$ of a matrix A, called the *generating matrix* of the network. The null-vector $\mathbf{p} = (p_1, \cdots, p_N)^t$ that satisfies the probability constraints

$$\begin{array}{ll} \sum_{j=1}^{N} p_j & = 1, \\ p_j & \geq 0, \end{array} \tag{1}$$

will be the required probability distribution vector for the network. The problem is challenging because conventional methods for finding eigenvectors will not be cost effective for such problem as the size N of the matrix A is usually very large. Typically, we have

$$N = \prod_{i=1}^{q} n_i \ ,$$

where q is the number of queues in the network and n_i is the number of buffers in the ith queue, $1 \leq i \leq q$. However, A possesses rich algebraic structures that one can exploit in finding $\mathbf{p}$.

For the networks that we will discuss in this paper, the matrix A is irreducible, has zero column sum, positive diagonal entries and non-positive off-diagonal entries. Thus $(I - A\text{diag}(A)^{-1})$ is a stochastic matrix. From Perron and Frobenius theory, we then know that A has a one-dimensional null-space with a positive null-vector $\mathbf{p}$, see for instance Varga [16]. Another important feature of A is that it is sparse. Its matrix graph is the same as that of the q-dimensional discrete Laplacian. Thus each row of A has at most $2q + 1$ non-zero entries.

One usual approach to the problem is to consider the partition

$$A = \begin{bmatrix} B & \mathbf{d} \\ \mathbf{c}^t & \alpha \end{bmatrix} \quad \text{and} \quad \mathbf{p} = \begin{bmatrix} \mathbf{f} \\ 1 \end{bmatrix}$$

and solve the reduced $(N-1)$-by-$(N-1)$ nonsingular system

$$B\mathbf{f} = -\mathbf{d} \tag{2}$$

by direct or iterative methods, see Funderlic and Mankin [6], Kaufman [9], O'Leary [10] and Plemmons [12]. Sparsity of the matrix B is usually exploited in these methods.

However, from numerical pde's, we know that with suitable domain of definition and compatible boundary conditions, the q-dimensional discrete Laplacian matrix can be inverted efficiently by using the separability of the Laplacian operator; and this approach will not work on any reduced system of size $(N-1)$-by-$(N-1)$. Thus in this paper, instead of reducing the size of the matrix by one, we consider N-by-N singular matrices as preconditioners for the system $A\mathbf{p} = \mathbf{0}$. These preconditioners will cancel the singularity of A in the sense that the resulting preconditioned systems are N-by-N nonsingular systems. By working on matrices of size N-by-N, we can exploit the fast inversion of separable components in A and these separable components will be used as building blocks for constructing preconditioners for A.

The outline of the paper is as follows. In §2, to consolidate the idea, we introduce our method for queueing networks with rectangular state-spaces first. We will illustrate the idea of using separable components in the generating matrix A in constructing preconditioners for A. We will also point out the relationship between queueing networks and elliptic problems which will be useful later in designing preconditioners for queueing problems with irregular state-spaces. Convergence analysis for 2-queue single-server networks with rectangular state-spaces is also given. In §3, we consider queueing networks with irregular state-spaces. We will illustrate how the results from domain decomposition can be used to construct separable preconditioners for these queueing problems. In particular, we will make use of the ideas from the substructuring method, the capacitance matrix method and the additive Schwarz method to derive three different preconditioners for these queueing problems. We remark that domain decomposition methods are most well-suited for these large queueing problems because they can be made parallel easily. Numerical results are given in §4. They show that our preconditioners work very well for the test problems. The numbers of iterations required for convergence increase very slowly as the queue size increases.

2 Networks with Rectangular State-spaces

Let us first introduce the notations that we will be using. Assume that the network has q queues receiving customers from q independent Poisson sources. In the ith queue, $1 \le i \le q$, there are s_i parallel servers and $n_i - s_i - 1$ spaces for customers. Arrival of customers is assumed to be Poisson distributed with rate λ_i and the service time of each server is exponential with mean $1/\mu_i$. To illustrate our method and what we mean by separable components in a generating matrix, we restrict ourselves to queueing networks with rectangular state-spaces in this section.

2.1 A 2-Queue Overflow Network

For simplicity, we begin with a simple 2-queue overflow network discussed in Kaufman [9]. Here customers entering the first queue can overflow to the second queue if the first queue is full and the second queue is not yet full. However, customers entering the second queue will be blocked and lost if the second queue is full, see Figure 1.

Let $p_{i,j}$ be the steady-state probability that there are i and j customers in queues 1 and 2 respectively. The Kolmogorov balance equations, i.e. the equations governing $p_{i,j}$, are given by:

$$\begin{aligned} & \left\{\lambda_1(1-\delta_{in_1-1}\delta_{jn_2-1}) + \lambda_2(1-\delta_{jn_2-1}) + \mu_1 \min(i,s_1) + \mu_2 \min(j,s_2)\right\} p_{ij} \\ = \; & \lambda_1(1-\delta_{i0})p_{i-1,j} + \mu_1(1-\delta_{in_1-1})\min(i+1,s_1)p_{i+1,j} \\ & +(\lambda_1\delta_{in_1-1}+\lambda_2)(1-\delta_{j0})p_{i,j-1} + \mu_2(1-\delta_{jn_2-1})\min(j+1,s_2)p_{i,j+1}, \end{aligned}$$

for $0 \le i < n$, $0 \le j < n_2$. Here δ_{ij} is the Kronecker delta. These equations can be expressed as a matrix equation $A\mathbf{p} = \mathbf{0}$. The matrix A is called the generating matrix of the network while the vector $\mathbf{p}$, after normalization by (1), is called the steady-state probability distribution vector of the network. The matrix A is known to be non-separable with no closed form solution for $\mathbf{p}$. However, we will show below that A can be partitioned into the sum of a separable matrix and a low rank matrix.

To derive the separable matrix, let us assume for the moment that overflow of customers from queue 1 to queue 2 is not allowed. Thus the two queues are independent of each other. Such network is said to be *free*. Let A_0 be

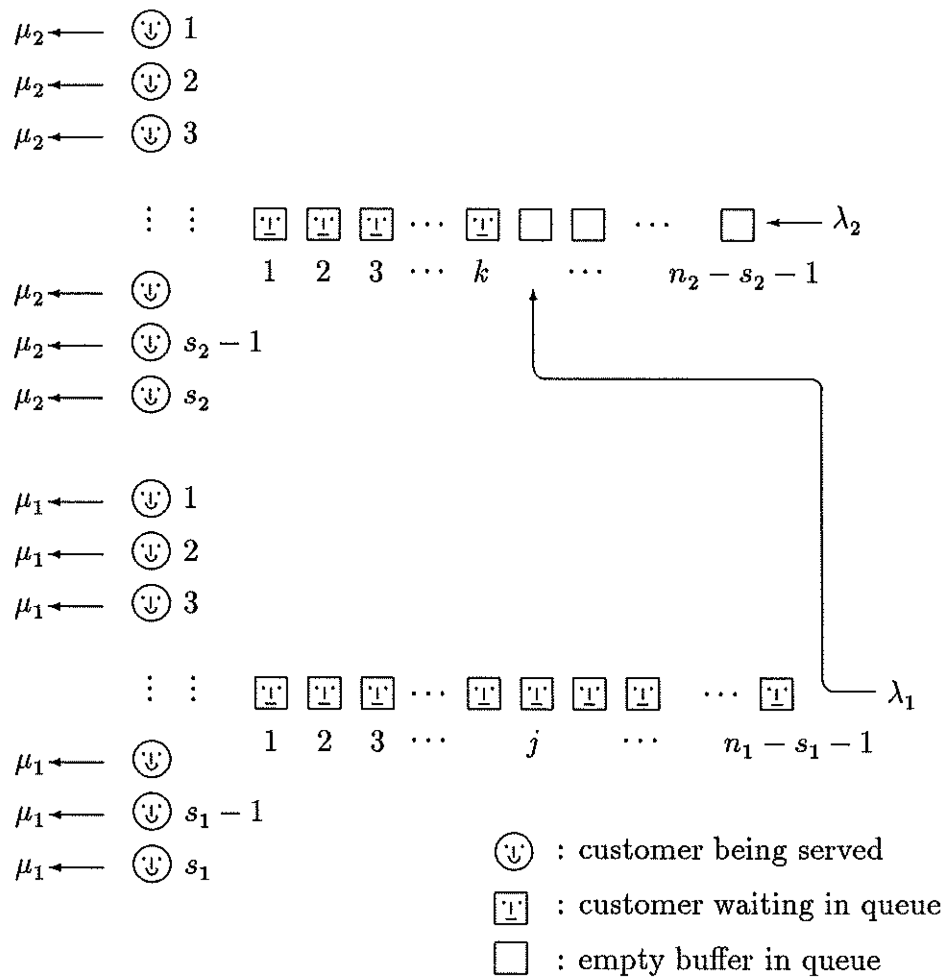

Figure 1: A 2-Queue Overflow Network.

the generating matrix for such 2-queue free network. Then A_0 is separable and is given by

$$A_0 = G_1 \otimes I_2 + I_1 \otimes G_2, \tag{3}$$

where I_i is the identity matrix of size n_i, $\otimes$ is the Kronecker tensor product and G_i, $i = 1, 2$, are n_i-by-n_i matrix given by

$$G_i = \begin{bmatrix} \lambda_i & -\mu_i & & & & & & 0 \\ -\lambda_i & \lambda_i + \mu_i & -2\mu_i & & & & & \\ & -\lambda_i & \lambda_i + 2\mu_i & -3\mu_i & & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & -\lambda_i & \lambda_i + s_i\mu_i & -s_i\mu_i & & \\ & & & & \cdot & \cdot & \cdot & \\ & 0 & & & & -\lambda_i & \lambda_i + s_i\mu_i & -s_i\mu_i \\ & & & & & & -\lambda_i & s_i\mu_i \end{bmatrix}. \tag{4}$$

We note that G_i is just the generating matrix of the ith queue and (3) reflects the fact that the queues are independent of each other.

The difference $R = A - A_0$, which corresponds to the addition of the overflow queueing discipline from queue 1 to queue 2, is given by

$$R = (\mathbf{e}_{\mathbf{n}_1} \mathbf{e}_{\mathbf{n}_1}{}^t) \otimes \lambda_1 \cdot R_2, \tag{5}$$

where $\mathbf{e}_\mathbf{j}$ is the jth unit vector and

$$R_i = \begin{bmatrix} 1 & & & & 0 \\ -1 & 1 & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & 0 & -1 & 1 & \\ & & & -1 & 0 \end{bmatrix}, \tag{6}$$

is an n_i-by-n_i matrix. Clearly R is a matrix of rank n_2.

2.2 Properties of the Preconditioner

Our method is based on the idea of partitioning A into $A_0 + R$ and using A_0 as a preconditioner. In this subsection, we show that the generalized inverse

A_0^+ of A_0 can be obtained easily by using the separability of A_0 and we also discuss some of the properties of A_0^+.

Since the upper and lower diagonals of the matrix G_i given in (4) are nonzero and of the same sign, there exists a diagonal matrix D_i such that $D^{-1}G_iD_i$ is symmetric. In fact $D_i = \text{diag}(d_1^i, \cdots, d_{n_i}^i)$ with

$$d_j^i = \begin{cases} 1 & j = 1, \\ \prod_{k=1}^{j-1} \left(\frac{\lambda_i}{\min(k,s_i)\mu_i}\right)^{\frac{1}{2}} & 1 < j \le n_i. \end{cases} \tag{7}$$

As $D_i^{-1}G_iD_i$ is symmetric and tri-diagonal, we can find in $O(n_i^2)$ operations an orthogonal matrices Q_i such that

$$Q_i^t D_i^{-1} G_i D_i Q_i = \Lambda_i$$

is diagonal, see Golub and van Loan [7]. Thus by (3), we see that

$$(Q_1 \otimes Q_2)^t (D_1 \otimes D_2)^{-1} A_0 (D_1 \otimes D_2)(Q_1 \otimes Q_2) = \Lambda_1 \otimes I_2 + I_1 \otimes \Lambda_2 \equiv \Lambda,$$

where Λ is a diagonal matrix. Therefore we can define a generalized inverse A_0^+ of A_0 as

$$A_0^+ = (D_1 \otimes D_2)(Q_1 \otimes Q_2)\Lambda^+ (Q_1 \otimes Q_2)^t (D_1 \otimes D_2)^{-1}. \tag{8}$$

Since G_i has zero column sum and $D_i^{-1}G_iD_i$ is symmetric, it follows that

$$G_i D_i^2 \mathbf{1}_i = D_i^2 G_i^t \mathbf{1}_i = \mathbf{0}, \tag{9}$$

where $\mathbf{1}_i$ is the n_i-vector of all ones. From (3), we thus see that a right null-vector of A_0 is given by

$$\mathbf{p}_0 = (D_1^2 \otimes D_2^2)(\mathbf{1}_1 \otimes \mathbf{1}_2), \tag{10}$$

which after normalization by (1), will give us the steady-state probability distribution of the free network.

Using the spectral decomposition of A_0 in (8) and the fact that

$$Im(A_0) = Im(A_0^+) = \langle \mathbf{1} \rangle^\perp, \tag{11}$$

we can easily verify the following property of A_0.

Lemma 1 (Chan [3]) *Let A_0^+ be the generalized inverse of A_0 as defined in (8). Then*

(i) $\mathbf{R}^N = \langle \mathbf{p}_0 \rangle \oplus Im(A_0)$.

(ii) For all $\mathbf{x} \in Im(A_0^+)$, *there exists a unique* $\mathbf{y} \in Im(A_0)$ *such that* $A_0^+\mathbf{y} = \mathbf{x}$.

(iii) For all $\mathbf{x} \in Im(A_0)$, *we have* $A_0 A_0^+ \mathbf{x} = A_0^+ A_0 \mathbf{x} = \mathbf{x}$.

Note that by (8), given any vector $\mathbf{y}$, the matrix-vector multiplication $A_0^+\mathbf{y}$ can be computed in $O(n_1 n_2(n_1+n_2))$ operations. Moreover, there is no need to generate A_0^+ explicitly, all we need are storages for each individual Q_i, D_i and Λ_i, $i = 1, 2$. We emphasize that if $s_i = 1$, then G_i has constant upper and lower diagonals and hence for any vector $\mathbf{x}$, the product $Q_i\mathbf{x}$ can be computed by using Fast Fourier Transform in $O(n_i \log n_i)$ operations, see Chan [3]. In particular, if $s_1 = s_2 = 1$, then $A_0^+\mathbf{y}$ can be obtained in $O(n_1 n_2 \log(n_1 n_2))$ operations.

2.3 The Method

Let us now go back to the problem of finding the null-vector $\mathbf{p}$ for A. By Lemma 1, we see that for any null-vector $\mathbf{p}$ of A, there exist unique real number α, $\mathbf{x} \in Im(A_0^+)$ and $\mathbf{y} \in Im(A_0)$ such that

$$\mathbf{p} = \alpha \mathbf{p}_0 + \mathbf{x} = \alpha \mathbf{p}_0 + A_0^+ \mathbf{y}.$$

Since $\mathbf{1}^t\mathbf{p} \neq 0$ and $\mathbf{1}^t\mathbf{x} = 0$, we see that $\alpha \neq 0$.

For the moment, let us concentrate on finding the null-vector $\mathbf{p}$ with $\alpha = 1$, i.e.

$$\mathbf{p} = \mathbf{p}_0 + A_0^+\mathbf{y}. \tag{12}$$

Putting this into the expression $A\mathbf{p} = \mathbf{0}$, we get $(A_0 + R)(\mathbf{p}_0 + A_0^+\mathbf{y}) = \mathbf{0}$. After simplification, we then have

$$A_0 A_0^+ \mathbf{y} + R A_0^+ \mathbf{y} = -R\mathbf{p}_0.$$

Since $\mathbf{y} \in Im(A_0)$, by Lemma 1(iii), we have $A_0 A_0^+ \mathbf{y} = \mathbf{y}$. Therefore the above equation reduces to

$$(I + RA_0^+)\mathbf{y} = -R\mathbf{p}_0. \tag{13}$$

Note that the equation has unique solution $\mathbf{y}$ in $Im(A_0)$. In particular, the mapping $(I + RA_0^+)$ is invertible on $Im(A_0)$. In fact, we can show further that the matrix $(I + RA_0^+)$ is indeed invertible on $\mathbf{R}^N$, see Chan [3]. Hence the system (13) can be solved by iterative methods or even direct methods without any restriction onto the subspace $Im(A_0)$. Thus we see that by preconditioning A from the left by A_0, we have basically cancel the singularity in A and reduce the singular system $A\mathbf{p} = \mathbf{0}$ to a nonsingular system (13). Once we have the solution $\mathbf{y}$ from (13), the null-vector $\mathbf{p}$ can be obtained by (12) and by using the normalization constraints in (1), we then have the steady-state probability distribution.

Because of the sparsity of R, we see that the matrix $(I + RA_0^+)$ is also sparse and has at most n_2 eigenvalues different from 1. Thus the matrix is an ideal candidate for the conjugate gradient type methods whose convergence rate depends on how clustered the spectrum of the matrix is, see Golub and van Loan [7]. Notice that the cost of forming the matrix-vector product $(I + RA_0^+)\mathbf{x}$, which is required in every iteration of the conjugate gradient method, can be further reduced if sparsity of R is exploited. Since the matrix $(I + RA_0^+)$ is nonsymmetric, one can apply the conjugate gradient method to the normal equation

$$(I + RA_0^+)^t(I + RA_0^+)\mathbf{y} = -(I + RA_0^+)^t R\mathbf{p}_0 \tag{14}$$

or apply other generalized conjugate gradient type methods, see for instance Young and Jea [17] or Saad and Schultz [15].

2.4 Convergence Analysis

Clearly, the total cost of our method depends on its convergence rate. In this section, we give the convergence rate of our method in the single server case, i.e. $s_1 = s_2 = 1$. Notice that in analyzing the convergence rate, which is a function of N, we need to know the relationship between the parameters of the queues as a function of N. To this end, let us consider a typical 1-queue, single server, Markovian network. It can be seen from (9) that in this case, the steady state probabilities are given by

$$p_j = \frac{\rho - 1}{\rho^n - 1}\rho^j, \quad 0 \le j < n,$$

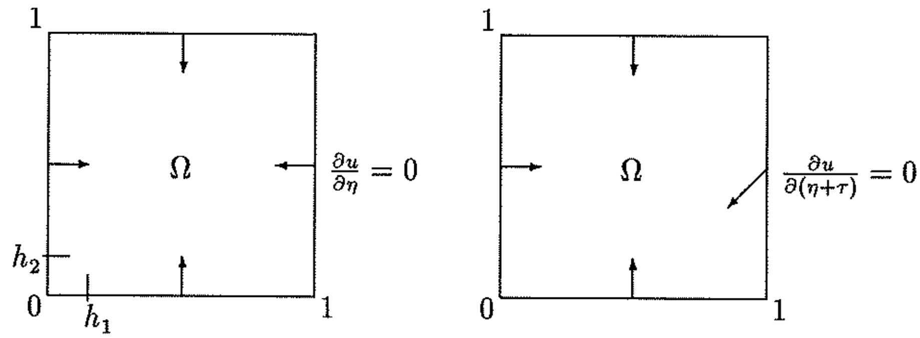

Figure 2: Neumann and Oblique Problems on Unit Square.

where $\rho = \lambda/\mu$ is the traffic density of the queue. Thus as we increase the queue size n, in order that the probabilities p_j for $j \approx n$ are not exponentially small, one possible limit to consider is $\rho = 1 + cn^{-\alpha}$ for some constants $c \in \mathbf{R}$ and $\alpha \geq 1$.

We remark that we can obtain the same limit if we consider the continuous analogue of queueing networks. We begin by noting that in the 2-queue single-server case, when $\lambda_i = \mu_i = 1$, the matrix A_0 is exactly equal to the two-dimensional discrete Laplacian matrix on the unit square Ω with Neumann boundary conditions on every sides and mesh size $h_i = 1/(n_i - 1)$, see Figure 2. When $\lambda_i \neq \mu_i$, A_0 resembles the finite difference approximation of the second order elliptic equation

$$\begin{aligned} & (\lambda_1 + \mu_1)p_{xx} + (\lambda_2 + \mu_2)p_{yy} \\ + \quad & 2(n_1 - 1)(\mu_1 - \lambda_1)p_x + 2(n_2 - 1)(\mu_2 - \lambda_2)p_y = 0 \end{aligned} \tag{15}$$

with the equation defined on the same unit square Ω and having the same Neumann boundary conditions on every sides. Moreover, we observe that R_2 in (6) is a first order differencing matrix. Hence the matrix R in (5) resembles a tangential operator on the edge $x = 1$ where overflow occurs. Compared to A_0, we see that the generating matrix A of the overflow problem resembles the same second order elliptic equation on the unit square with the Neumann boundary condition on the edge $x = 1$ being replaced by an oblique derivative, see Figure 2.

From (15), we see that if for large n_i, λ_i and μ_i are related by $\lambda_i/\mu_i = 1 + c_i n_i^{-\alpha_i}$ for some constants $c_i \in \mathbf{R}$ and $\alpha_i \geq 1$, then the second order terms in (15) are the dominant terms. In the following, we will analyze the convergence rate of our method under such limit. We note that if $\alpha_i < 1$ and $\lambda_i < \mu_i$, then $||A\mathbf{p}_0||_2$ tends to zero exponentially fast as n_i increases, see Chan [3]. In particular, $\mathbf{p}_0$ will already be a good estimate for $\mathbf{p}$ in such cases.

Notice that because of the low rank of R, the actual number of unknowns in the vector $\mathbf{y}$ in (13) is equal to n_2. Let the last n_2-by-n_2 principal submatrix of $(I + RA_0^+)$ be denoted by S. For ease of presentation, we assume in the following Lemma and Theorem that the queueing parameters λ_i, μ_i, s_i and n_i are the same for both queues with $n_1 = n_2 = n$. Full version and proof of the Lemma and Theorem are given in Chan [3].

Lemma 2 *Assume that for both queues, $s_i = 1$ and $\mu_i = \lambda_i + cn^{-\alpha}$ for some constants $c \in \mathbf{R}$ and $\alpha \geq 1$. Then for sufficiently large n, $||S^{-1}||_2 < O(n^3)$ and*

$$S^t S = 2I + L + U,$$

where $||U||_2 < O(n^{1-\alpha}/\log n)$ and rank $L = O(\log n)$.

The Lemma states that the singular values of S are clustered around $\sqrt{2}$ except for at most $O(\log n)$ outlying ones. Applying standard error analysis of conjugate gradient method to the above results on S, we obtain the following Theorem.

Theorem 1 *Assume that for both queues, $s_i = 1$ and $\mu_i = \lambda_i + cn^{-\alpha}$ for some constants $c \in \mathbf{R}$ and $\alpha \geq 1$. Then for large n, the conjugate gradient method applied to the normal equation (14) will converge within $O(\log^2 n)$ steps.*

The proof of the above Lemma involves purely linear algebra. However, the same result is anticipated if we look at the continuous analogue of the overflow problem. We recall that the matrix $(I+RA_0^+) \approx AA_0^+$ represents the mapping that maps the Neumann boundary data to the oblique boundary data. By using regularity theorem and trace theorem in elliptic theory, we have the following result on this mapping.

Theorem 2 (Chan [2]) *Let Ω be a bounded region in $\mathbf{R}^2$ with a smooth boundary $\partial\Omega$. Let*

$$E \equiv \left\{ g \in H^{-\frac{1}{2}}(\partial\Omega) \mid \int_{\partial\Omega} g d\tau = 0 \right\}$$

be equipped with the Sobolev $H^{-\frac{1}{2}}(\partial\Omega)$ norm. Let T be the Neumann-to-oblique mapping that maps g_1 in E to g_2 where g_1 and g_2 are boundary values of the problems

$$(N): \begin{cases} \Delta u = 0 & \text{in } \Omega \ , \\ \dfrac{\partial u}{\partial \eta} = g_1 & \text{on } \partial\Omega \ , \end{cases} \quad \textit{and} \quad (O): \begin{cases} \Delta u = 0 & \text{in } \Omega \ , \\ \dfrac{\partial u}{\partial(\eta+\tau)} = g_2 & \text{on } \partial\Omega \ , \end{cases}$$

with both problems normalized by $\int_\Omega u = 0$. Here η and τ are the normal and tangential vectors respectively. Then T is a one-one onto mapping on E and satisfies

$$c||Tg||_{H^{-\frac{1}{2}}(\partial\Omega)} \le ||g||_{H^{-\frac{1}{2}}(\partial\Omega)} \le C||Tg||_{H^{-\frac{1}{2}}(\partial\Omega)}$$

for some constants $c = c(\Omega) > 0$ and $C = C(\Omega) > 0$.

Thus the Neumann-to-oblique mapping T is well-conditioned. Hence, we expect the matrix $(I + RA_0^+)$ to be also well-conditioned for large n and if the conjugate gradient method is used, we expect fast convergence.

In the multi-server case, i.e. $s_i > 1$, instead of (15), the underlying continuous equation is of the form

$$\begin{aligned} &(\lambda_1 + s_1\mu_1)p_{xx} + (\lambda_2 + s_2\mu_2)p_{yy} \\ + \ \ &2(n_1 - 1)(s_1\mu_1 - \lambda_1)p_x + 2(n_2 - 1)(s_1\mu_2 - \lambda_2)p_y = 0 \end{aligned} \qquad (16)$$

in the region where the states (i,j) satisfy $s_1 \le i < n_1$ and $s_2 \le j < n_2$. In other part of the rectangular state-space, the equation will be one with variable coefficients, with the coefficients of the second order terms decreasing in magnitude with decreasing i and j. Hence for large n_i, a reasonable limit to consider is

$$\frac{\lambda_i}{s_i\mu_i} = 1 + c_i n_i^{-\alpha} \qquad (17)$$

for some constants $c_i \in \mathbf{R}$ and $\alpha \ge 1$. We note however that under such limit, by Stirling's formula, the last entry $d_{n_i}^i$ of D_i in (7) is given

$$d_{n_i}^i = \left[\frac{1}{s_i!}\left(\frac{\lambda_i}{\mu_i}\right)^{s_i}\left(\frac{\lambda_i}{s_i\mu_i}\right)^{n_i-s_i-1}\right]^{\frac{1}{2}} \approx \left[\frac{s_i^{s_i}}{s_i!}\right]^{\frac{1}{2}} \approx (2\pi s_i)^{-1/4}e^{s_i/2}. \qquad (18)$$

Thus for large s_i, the matrix D_i will be ill-conditioned and hence the spectral decomposition of A_0^+ in (8) will be unstable. Experimental results show that our method works very well under limit (17) for small s_i but will break down when s_i becomes large, see Chan [3] and also §4.

2.5 General Overflow Queueing Networks with Rectangular State-space

The above method for finding $\mathbf{p}$ in $A\mathbf{p} = \mathbf{0}$ can be readily generalized to overflow queueing networks with more than two queues, provided that all overflows occur when and only when the queue is full. In this case, the state-space of the problem is the q-dimensional unit cube and one can automate the whole procedure for finding $\mathbf{p}$. More precisely, one can write a program that accepts queue parameters and overflow disciplines as input and outputs the steady-state probability distribution.

To see how this can be done automatically, we first note that the generating matrix for any q-queue free network is given by

$$A_0 = G_1 \otimes I_2 \otimes \cdots \otimes I_q + I_1 \otimes G_2 \otimes \cdots \otimes I_q + \cdots + I_1 \otimes I_2 \otimes \cdots \otimes G_q,$$

where G_i are given by (4). Clearly A_0 is still separable with a null-vector given by

$$\mathbf{p}_0 = (D_1^2 \otimes \cdots \otimes D_q^2)(\mathbf{1}_1 \otimes \cdots \otimes \mathbf{1}_q).$$

Any addition of overflow queueing disciplines to this free network corresponds to addition of matrices of the form (6) to the matrix $R = A - A_0$. With R and A_0 known, we can apply our method to obtain the system (13) which can then be solved by conjugate gradient type methods.

As an example, consider a 3-queue network where customers from queue i, $i = 1, 2$, can overflow to queue $i+1$ provided that queue i is full. Then the matrix $R = A - A_0$ is given by

$$\begin{aligned} R \;=\; & (\mathbf{e}_{n_1}\mathbf{e}_{n_1}{}^t) \otimes \lambda_1 \cdot R_2 \otimes I_3 + I_1 \otimes (\mathbf{e}_{n_2}\mathbf{e}_{n_2}{}^t) \otimes \lambda_2 \cdot R_3 \\ & + (\mathbf{e}_{n_1}\mathbf{e}_{n_1}{}^t) \otimes (\mathbf{e}_{n_2}\mathbf{e}_{n_2}{}^t) \otimes \lambda_1 \cdot R_3, \end{aligned} \tag{19}$$

with R_i given by (6). The last term above represents the flow from queue 1 to queue 3 when both queues 1 and 2 are full. We further note that one does not have to generate R nor the terms in R explicitly. For in conjugate gradient type methods, one only needs to compute the product $R\mathbf{x}$ which can be computed term-wise and the product for each term can be computed easily without forming each individual term explicitly.

The cost per iteration of our method will mainly depend on the cost of computing the matrix-vector multiplication $A_0^+\mathbf{y}$. However, in view of (8),

Method	Operations	Storage
Normal equation	$O(n^{q+1})$ per iteration	$O(n^q)$
Point SOR	$O(n^q)$ per iteration	$O(n^q)$
Block SOR	$O(n^{q+1})$ per iteration	$O(n^q)$
Band solver	$O(n^{3q-2})$	$O(n^{2q-1})$

Table 1: Cost Comparison.

this can be done in

$$O(N(n_1 + n_2 + \cdots + n_q))$$

operations where $N = \prod_{i=1}^{q} n_i$. In the case of single-server, i.e. $s_1 = s_2 = \cdots = s_q = 1$, the cost is reduced to $O(N \log N)$ by using FFT. Storage for few N-vectors and the small dense matrices Q_i will be required and will be of order $O(N)$.

Table 1 below compares the cost of our method as applied to the normal equation (14) with the cost of other methods as applied to system (2). For simplicity, we let $n_i = n$ for all $i = 1, \cdots, q$. We emphasize again that FFT can be used to speed up the computation of $A_0^{+}\mathbf{x}$ if any one of the s_i is equal to 1.

2.6 Some Remarks on the Method

We remark that our method is a generalization of the method in (2). In the simplest case if we partition $A = A_0 + R$ with A_0 given by

$$A_0 = I_N - \mathbf{e_N}\mathbf{e_N}^t$$

we then have $\mathbf{p}_0 = \mathbf{e_N}$ and equation (13) is reduced basically to (2). If we choose

$$A_0 = \text{diag}(A) - (\mathbf{e_N}^t A \mathbf{e_N}) \cdot \mathbf{e_N}\mathbf{e_N}^t$$

then (13) is similar to the Jacobi method applied to (2). The main difference between our method and that of (2) is that our preconditioner A_0 is an N-by-N matrix and any preconditioners for (2) will be of size $(N-1)$-by-$(N-1)$. The ability of using N-by-N matrices as preconditioners enables us to exploit the separable components of the original generating matrix A and hence speed up the inversion of the preconditioners.

We finally note that one can also precondition the system $A\mathbf{p} = \mathbf{0}$ from the left by A_0^+. More precisely, we expand $\mathbf{p}$ as $\mathbf{p} = \beta\mathbf{p}_0 + \mathbf{y}$ where $\mathbf{y} \in Im(A_0)$. By (1), $\beta \neq 0$ and by Lemma 1(iii), the system $A\mathbf{p} = \mathbf{0}$ is reduced to

$$A_0^+ A\mathbf{y} = (I + A_0^+ R)\mathbf{y} = -A_0^+ R\mathbf{p}_0. \qquad (20)$$

One can also prove that the matrix $(I + A_0^+ R)$ is nonsingular. We note further that in view of (11), $\mathbf{p}$ can be expanded as $\mathbf{p} = \gamma\mathbf{1} + \mathbf{x}$ where $\mathbf{x} \in Im(A_0)$. In particular, we can replace the right hand side of (20) by $-A_0^+ A\mathbf{1}$. However, we note that if A_0 is expected to be a close approximation to A, then we also expect $\mathbf{p}_0$ to be a better approximation to $\mathbf{p}$ than $\mathbf{1}$.

3 Networks with Irregular State-spaces

In this section, we consider queueing networks where the state-spaces are no longer rectangular and we will make use of the results from elliptic solvers for irregular domains to design preconditioners for these networks. To illustrate the idea, we consider the 2-queue one-way overflow network in §2.1 again and assume additionally that customers waiting for service in queue 1 must be served at queue 2 once servers in queue 2 become available. This network is almost the same as the one discussed in Kaufman et. al. [8], except now that we also allow the one-way overflow of customers from queue 1 to queue 2 when queue 1 is full. The Kolmogorov equations of the network are given in Chan [4].

According to the overflow discipline, we see that

$$p_{ij} = 0, \qquad s_1 < i < n_1, \quad 0 \le j < s_2 \,.$$

Hence the state-space of the network is no longer rectangular, but is given by an L-shaped region, see Figure 3. The subregions Ω_1 and Ω_2 are defined to be the set of states (i, j) where $0 \le i \le s_1$ and $s_1 < i < n_1$ respectively. They are separated by the interface $\tau = \{(s_1, j)\}_{j=s_2}^{n_2-1}$.

The generating matrix A of the network is in the block form

$$A = \begin{bmatrix} T_1 & D_{11} & 0 \\ D_{12} & C & D_{22} \\ 0 & D_{21} & T_2 \end{bmatrix}$$

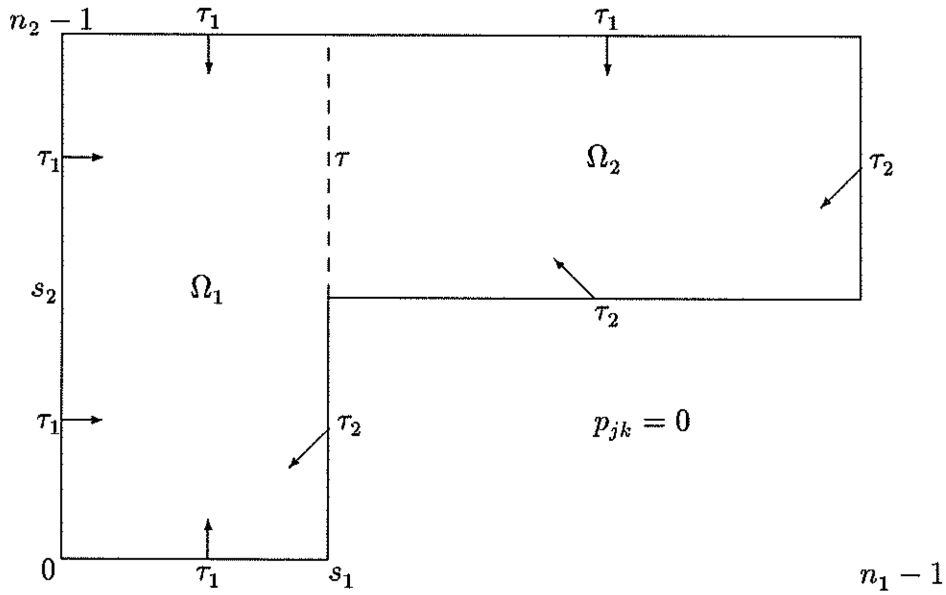

Figure 3: The State-space of the Queueing Network.

where T_k, $k = 1, 2$, correspond to the interactions between the states within the subregions $\Omega_k \setminus \tau$, C corresponds to states in τ and D_{kl} correspond to the interactions between the states in the subregion Ω_k and the states in τ. Notice that there is no interaction between the states in $\Omega_1 \setminus \tau$ and states in Ω_2. Since A is no longer of size $n_1 n_2$, A_0 cannot be used as preconditioner.

The continuous analogue of this problem is a second order elliptic equation on the L-shaped region with oblique derivatives on boundary τ_2 and Neumann boundary conditions on τ_1. There are many domain decomposition techniques for solving elliptic equations on irregular regions. In the following subsections, we will apply some of the ideas there to design preconditioners for our queueing networks.

3.1 Substructuring Method

For elliptic problem on an L-shaped domain as in Figure 3, the theory of substructuring suggests the following Dirichlet-Neumann map as a preconditioner:

1. solve the problem defined in Ω_1 with Neumann boundary conditions on every side,

2. then use the value on the interface τ to solve a Dirichlet-Neumann problem in Ω_2,

see for instance Bjϕrstad and Widlund [1].

In term of the matrix A, that means we write $A = A_1 + R$, where

$$A_1 = \begin{bmatrix} T_1 & D_{11} & 0 \\ D_{12} & \tilde{C} & 0 \\ 0 & D_{21} & \tilde{T}_2 \end{bmatrix}.$$

The matrix $\tilde{C}$ is chosen such that the submatrix

$$W \equiv \begin{bmatrix} T_1 & D_{11} \\ D_{12} & \tilde{C} \end{bmatrix}$$

corresponds to a Neumann problem in subregion Ω_1. In terms of the queues, W will be the generating matrix of a 2-queue free network with s_1 spaces in the first queue and $n_2 - 1$ spaces in the second queue. Hence W will be separable and has a 1-dimensional null-space.

The submatrix $\tilde{T}_2$ in A_1 above corresponds to a Dirichlet-Neumann problem in subregion Ω_2, with Dirichlet data being transported from τ by D_{21}. More precisely, we have

$$\tilde{T}_2 = V_1 \otimes I_{n_2 - s_2} + I_{n_1 - s_1 - 1} \otimes V_2$$

with

$$V_1 = \text{tridiag}(-\lambda_1, \lambda_1 + s_1\mu_1, -s_1\mu_1) - \lambda_1 \cdot \mathbf{e}_{\mathbf{n_1 - s_1 - 1}}\mathbf{e}_{\mathbf{n_1 - s_1 - 1}}{}^t, \tag{21}$$

and

$$V_2 = \text{tridiag}(-\lambda_2, \lambda_2 + s_2\mu_2, -s_2\mu_2) - s_2\mu_2 \cdot \mathbf{e}_1\mathbf{e}_1{}^t - \lambda_2 \cdot \mathbf{e}_{\mathbf{n_2 - s_2}}\mathbf{e}_{\mathbf{n_2 - s_2}}{}^t. \tag{22}$$

Thus $\tilde{T}_2$ will also be separable but it will be nonsingular.

Since A_1 is in block lower-triangular form, we see that A_1 is singular with a one-dimensional null-space and a null-vector

$$\mathbf{p}_1 = \begin{bmatrix} \mathbf{w} \\ -\tilde{T}_2^{-1}(0,\ D_{21})\mathbf{w} \end{bmatrix},$$

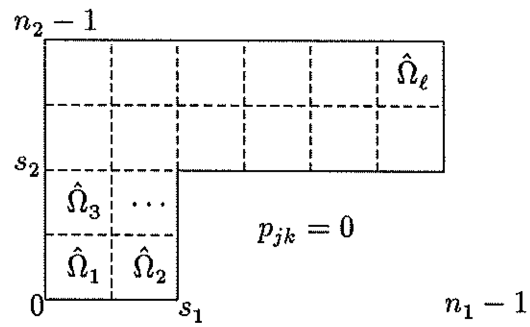

Figure 4: Partition into Many Subdomains.

where $\mathbf{w}$ is a null-vector of W. We can easily check that A_1 also satisfies the properties listed in Lemma 1 for A_0. Thus we can again expand $\mathbf{p} = \alpha\mathbf{p}_1 + A_1^+\mathbf{y}$ where $\mathbf{y} \in Im(A_1)$ and reduce the homogeneous system $A\mathbf{p} = \mathbf{0}$ to a nonsingular system

$$(I + RA_1^+)\mathrm{y} = -R\mathbf{p}_1, \tag{23}$$

where $R = A - A_1$, see Chan [4].

We emphasize that the matrix R is sparse. In fact, $\tilde{C}$ and $\tilde{T}_2$ differ from C and T_2 only on the rows that correspond to those states on the interface τ and on the edges τ_2; and on such rows, they can only differ by at most three entries. The matrix-vector multiplication $A_0^+\mathbf{x}$ can be done efficiently by using the separability of W and $\tilde{T}_2$. We emphasize that for all arbitrary values of s_i, the submatrix $\tilde{T}_2$ can be inverted easily by using FFT, for it corresponds to a constant-coefficient Dirichlet-Neumann problem in Ω_2, see (16). In fact, we see from (21) and (22) that both matrices V_1 and V_2 have constant upper and lower diagonals.

The idea of using two subregions can easily be extended to many subregions. Consider the domain in Figure 3 being partitioned into many subregions, see Figure 4. In $\hat{\Omega}_1$, we solve a Neumann problem. With the values obtained on the boundary $\partial\hat{\Omega}_1$, we solve mixed problems in $\hat{\Omega}_2$ and $\hat{\Omega}_3$ and so on.

The preconditioner will be of the block lower-triangular form:

$$A_1 = \begin{bmatrix} T_{11} & & & 0 \\ D_{21} & T_{22} & & \\ & \ddots & \ddots & \\ D_{\ell 1} & \cdots & D_{\ell(\ell-1)} & T_{\ell\ell} \end{bmatrix}$$

The submatrix T_{11} corresponds to a Neumann problem on $\hat{\Omega}_1$, the submatrices T_{ii}, $2 \le i \le \ell$ correspond to mixed problems in $\hat{\Omega}_i$ and the off-diagonal block matrices D_{ij} will transport the required boundary data from one subregion to another. Notice that for each $i = 1, \cdots, \ell$, only two D_{ij} will be non-zero, and for those $\hat{\Omega}_i$ that lie inside Ω_2, their corresponding T_{ii} can be inverted by using FFT.

3.2 Capacitance Matrix Method

Another method of solving elliptic problem on irregular domain is to embed the whole domain into a rectangular domain and use the preconditioners on the rectangular domain as preconditioners for the embedded system, see Proskurowski and Widlund [13]. To illustrate the idea, let us order the states in the L-shaped domain first and denote A_0 to be the generating matrix of the 2-queue free network on the rectangular domain $[0, n_1 - 1] \times [0, n_2 - 1]$. We then partition A_0 as

$$A_0 = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where A_{22} gives the interactions between states that are both in the region $[s_1 + 1, n_1 - 1] \times [0, s_2 - 1]$. We note that A_{22} corresponds to a mixed problem in that region and can easily be proved to be nonsingular.

We now embed the generating matrix A for the L-shaped domain into the whole rectangular domain as

$$A_e = \begin{bmatrix} A & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

Since A_{22} is nonsingular, we see that A_e has a one-dimensional null-space and $\mathbf{p}_e = [\mathbf{p}, \mathbf{0}]^t$ is a null vector to A_e. Again we expand $\mathbf{p}_e$ as $\mathbf{p}_e = \mathbf{p}_0 + A_0^+ \mathbf{y}$

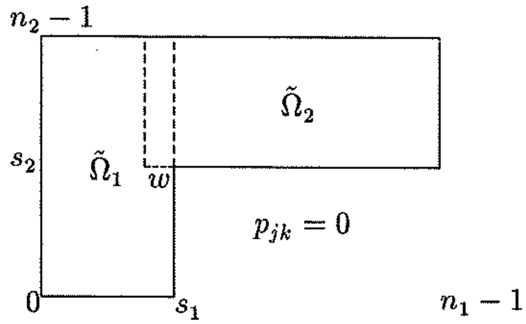

Figure 5: Partition into Overlapping Subdomains.

with $\mathbf{y} \in Im(A_0)$. Then the vector $\mathbf{y}$ can be obtained by solving

$$(I + RA_0^+)\mathbf{y} = -R\mathbf{p}_0, \tag{24}$$

where $R = A_e - A_0$. Once again the matrix $(I + RA_0^+)$ can be shown to be nonsingular, see Chan [4].

3.3 Additive Schwarz Method

One of the basic ideas of the additive Schwarz method is to extract easily invertible components from the matrix A, inverse each of them individually and then add their inverses together to form an approximate inverse of the matrix A, see Dryja and Widlund [5]. To illustrate the idea, let us partition the L-shaped domain in Figure 3 into two overlapping subdomains, see Figure 5. For $i = 1, 2$, let B_i be the matrix that corresponds to the Neumann problem in $\tilde{\Omega}_i$ and zero elsewhere. Clearly, both matrices are singular and separable. Similar to (11), we can prove that

$$\mathbf{1}^t B_1^+ = \mathbf{1}^t B_2^+ = \mathbf{1}^t (B_1^+ + B_2^+) = \mathbf{0},$$

or equivalently,

$$Im(B_1^+ + B_2^+) \subseteq \langle \mathbf{1} \rangle^\perp. \tag{25}$$

The matrix $(B_1^+ + B_2^+)$ will be used to precondition our system. Notice that it is in general difficult to find a null-vector for the preconditioner. Thus to find a null-vector $\mathbf{p}$ for A, we expand $\mathbf{p} = \alpha\mathbf{1} + \mathbf{x}$ where $\mathbf{x} \in \langle \mathbf{1} \rangle^\perp$. By the normalization constraints on $\mathbf{p}$, we see that $\alpha \neq 0$. Therefore we can set $\alpha = 1$ and rewrite $A\mathbf{p} = \mathbf{0}$ as

$$A\mathbf{x} = -A\mathbf{1}. \tag{26}$$

The system is then preconditioned by $(B_1^+ + B_2^+)$ to form

$$(B_1^+ + B_2^+)A\mathbf{x} = -(B_1^+ + B_2^+)A\mathbf{1}. \tag{27}$$

Notice that unlike the methods mentioned in previous sections, the preconditioned system in this case is still singular. However, because of (25), if the above equation is solved by conjugate gradient type methods with initial guess in $\langle \mathbf{1} \rangle^\perp$, then each subsequent iterant will automatically be in $\langle \mathbf{1} \rangle^\perp$ again.

Since the matrix $(B_1^+ + B_2^+)$ is singular, any matrix-vector product of the form $(B_1^+ + B_2^+)\mathbf{x}$ will have no component along the null vector of $(B_1^+ + B_2^+)$, which is in general difficult to find. To partially remedy this, we add to the product a vector of the form $\beta\mathbf{q}_1 + \gamma\mathbf{q}_2$, where $\mathbf{q}_i$ are the normalized null-vectors of B_i^+, $i = 1, 2$ (extended by zeros outside their respective domains of definition). By the separability of B_i^+, $\mathbf{q}_i$ can be obtained easily. In order that the vector

$$(B_1^+ + B_2^+)\mathbf{x} + \beta\mathbf{q}_1 + \gamma\mathbf{q}_2$$

is still in $\langle \mathbf{1} \rangle^\perp$, we need $\gamma = -\beta$. The remaining degree of freedom can be determined by imposing extra conditions on the vector at the intersection of the subdomains. One successful choice is to equate the mean values of the solutions of the subdomains at the intersection, i.e.

$$\sum_{k \in \tilde{\Omega}_1 \cap \tilde{\Omega}_2} (B_1^+\mathbf{x} + \beta\mathbf{q}_1)_k = \sum_{k \in \tilde{\Omega}_1 \cap \tilde{\Omega}_2} (B_2^+\mathbf{x} + \gamma\mathbf{q}_2)_k, \tag{28}$$

see Mathew [11]. Our numerical results show that the addition of the vector $\beta\mathbf{q}_1 + \gamma\mathbf{q}_2$ does improve the convergence rate especially when the size of the overlapping is small, see §4.

Obviously, because of the overlapping, the cost per iteration is higher than that in the substructuring method, but this is usually compensated by the advantage that $B_1^+A\mathbf{y}$ and $B_2^+A\mathbf{y}$ can be computed in parallel. Clearly the idea can be generalized to the case of many subdomains if more processors are available. However, it is already noted in domain decomposition literature

Method	Operations
Substructuring	$O(sn^2 + (n-s)^2 \log(n-s))$ per iteration
Capacitance Matrix	$O(n^3)$ per iteration
Additive Schwarz	$O(sn^2 + (n-s+w)^2(n-s))$ per iteration
Point-SOR	$O(ns + (n-s)^2)$ per iteration
Band Solver	$O([ns + (n-s)^2]n^2)$

Table 2: Comparison of Cost for Different Methods.

that unless a coarse grid component is added, the additive Schwarz method will not converge as fast as the other domain decomposition preconditioners. This fact is also verified in the numerical results in §4 for queueing networks. It is an interesting project to formulate and implement a suitable coarse grid structure to the queueing problem. One possible way is to aggregate the states together to form superstates and use the balance equations for the superstates to derive a coarse grid formulation for the queueing problem. Another possible approach is to use the idea of algebraic multigrid method to construct the coarse grid matrix directly from the given generating matrix, see for instance, Ruge and Stüben [14].

We conclude this section by listing the costs of different methods in Table 2. For simplicity, we assume that $n_1 = n_2 = n$ and $s_1 = s_2 = s$. The variable w in the additive Schwarz method denotes the width of the overlapping region, see Figure 5.

4 Numerical Results

In this section, we apply our method to the queueing networks considered in previous sections. The parameters of the queues are assumed to be the same for all queues and are related by

$$s_i \mu_i = \lambda_i + (n_i - 1)^{-\alpha}, \quad i = 1, 2, \cdots, q,$$

with $\lambda_i \equiv 1$. In the examples, the preconditioned systems were solved by a generalized conjugate gradient method, called the Orthodir method which does not require the formation of the normal equation, see Young and Jea [17]. We chose zero vector to be the initial guess and the stopping criterion to be $||r_k||_2/||r_0||_2 \leq 10^{-6}$, where r_k is the residual vector after k iterations.

n_i	N	s_i	α = 1	2	3	s_i	α = 1	2	3	s_i	α = 1	2	3
4	64	1	10	10	10	3	9	9	9	3	9	9	9
8	512	1	14	14	14	3	14	14	14	6	13	13	13
16	4096	1	18	18	18	3	18	18	18	9	17	17	17

Table 3: Numbers of Iterations required for Convergence.

n_i	s_i	N	ω^*	number of iterations: point-SOR	number of iterations: orthodir	time in seconds: point-SOR	time in seconds: orthodir
4	1	64	1.700	69	10	1.176	0.461
4	3	64	1.593	30	9	0.529	0.420
8	1	512	1.831	242	14	31.282	3.815
8	7	512	1.715	49	12	5.997	3.274

Table 4: Comparison with point-SOR method when $\alpha = 2$.

Example 1 is taken from Chan [4] to illustrate the effectiveness of our method over the point-SOR method. Computations in Example 2 were done by Matlab on SPARC stations at UCLA.

Example 1: We consider the 3-queue network with overflow queueing disciplines $q_1 \to q_2$, $q_2 \to q_3$ and $q_1 \to q_2 \to q_3$, see (19). Table 3 shows the number of iterations required for convergence for our preconditioned system (13). Table 4 compares the performance of our method with that of the point-SOR method as applied to (2) with $\mathbf{p}_0$ as the initial guess. The optimal relaxation factors ω^* were obtained numerically. We see that our method performs much better than the point-SOR method especially for small s_i. We note that in the test, we have not used FFT to speed up our algorithm in the case where all $s_i = 1$.

Example 2: We consider solving the 2-queue network with line-jumping in §3 by the different techniques we mentioned there. For the additive Schwarz method, two kinds of overlapping were tested: the maximum one with $\tilde{\Omega}_2 = [0, n_1-1] \times [s_2, n_2-1]$ and the minimum one with $\tilde{\Omega}_2 = [s_1-1, n_1-1] \times [s_2, n_2-1]$, see Figure 5. No coarse grid components were added however. Table 5 shows the numbers of iterations required for convergence for three different

n_i	s_i	α	N	No	DN	CM	ASmax	ASmin	ASmin*
10	2	1	86	41	12	12	16	18	19
20	4	1	340	81	16	15	19	25	27
40	8	1	1352	153	18	18	25	35	39
80	16	1	5392	>200	21	21	31	48	56
10	5	2	80	37	11	10	15	16	17
20	5	2	330	78	16	15	20	24	27
40	5	2	1430	161	19	19	25	34	38
80	5	2	6030	>200	23	21	30	49	54
10	4	3	80	39	12	10	15	16	18
20	8	3	312	74	14	14	19	23	26
40	16	3	1232	147	17	18	24	31	36
80	32	3	4896	>200	19	20	>60	>60	>60

Table 5: Numbers of Iterations for Convergence.

sets of queueing parameters. In the tables, N denotes the total number of states in the L-shaped domain, the symbols No, DN, CM, ASmax and ASmin stand for no-preconditioning (see (26)), preconditioning by Dirichlet-Neumann preconditioner (23), capacitance matrix method (24) and additive Schwarz method (27) with maximum and minimum overlap respectively. In both additive Schwarz methods, we have added the null-vectors according to (28). As a comparison, we also tested the case of minimum overlapping without adding the null-vector components (i.e. $\beta = \gamma = 0$) and the results are shown under the column ASmin*. To check the accuracy of our computed solution $\mathbf{p}_c$, we have computed $||A\mathbf{p}_c||_2$ and found it to be less than 10^{-7} in all the cases we tested.

From the numerical results, we see that the numbers of iterations for the non-preconditioned systems grow linearly with the queue size, a well-known phenomenon for second order elliptic problems. However, for the Dirichlet-Neumann preconditioner and the capacitance matrix preconditioner, the numbers of iterations grow very slowly with increasing queue size. Notice that when s_i are smaller, the L-shaped domain is more rectangular. Thus it is not surprising to see that the matrix capacitance method is a better choice for networks with small s_i.

For the additive Schwarz method, as already mentioned, it will not be as competitive as the other preconditioners if coarse grid components are not added. We note that adding the null-vectors $\mathbf{q_1} - \mathbf{q_2}$ speeds up the convergence rate significantly. However, the speed-ups are less significant in the maximum overlapping case where the numbers of iterations with or without adding the null-vectors differ by only one iteration in all the cases we tested. Also the method shows instability in the last test problem when $s_i = 32$. We have tried GMRES(40), see Saad and Schultz [15], and it still did not converge after 30 iterations. The instability may partly due to the fact that the matrix D_i in (8) is ill-conditioned with condition number about 10^6 in this case, see (18). Another possible reason is that we are not able to recover exactly the components of $(B_1^+ + B_2^+)\mathbf{x}$ along the null vector of $(B_1^+ + B_2^+)$, see §3.3. Further research will be carried out in this direction.

Acknowledgments: I would like to thank Professors Olof Widlund, Tony Chan and Tarek Mathew for their guidance in the preparation of this paper.

References

[1] P. Bjφrstad and O. Widlund, *Iterative Methods for the Solutions of Elliptic Problems on Regions Partitioned into Substructures*, SIAM J. Numer. Anal., V23, pp. 1097-1120 (1986).

[2] R. Chan, *Iterative Methods for Overflow Queueing Models*, NYU Comput. Sci. Dept., Tech Report No. 171, 1985.

[3] R. Chan, *Iterative Methods for Overflow Queueing Models I*, Numer. Math. V51, pp.143-180 (1987).

[4] R. Chan, *Iterative Methods for Overflow Queueing Models II*, Numer. Math. V54, pp.57-78 (1988).

[5] M. Dryja and O. Widlund, *Some Domain Decomposition Algorithms for Elliptic Problems*, Proceedings of the Conference on Iterative Methods for Large Linear Systems, Austin, Texas, October, 1989, Academic Press.

[6] R. Funderlic and J. Mankin, *Solution of Homogeneous Systems of Linear Equations Arising from Compartmental Models*, SIAM J. Sci. Statist. Comput., V2, pp.375-383, (1981).

[7] G. Golub and C. van Loan, *Matrix Computations*, 2nd Ed., The Johns Hopkins University Press, Maryland, 1989.

[8] L. Kaufman, J. Seery and J. Morrison, *Overflow Models for Dimension PBX Feature Package*, Bell Syst. Tech. J., V60, pp. 661-676 (1981).

[9] L. Kaufman, *Matrix Methods for Queueing Problems*, SIAM J. Sci. Stat. Comput., V4, pp. 525-552 (1982).

[10] D. O'Leary, *Iterative Methods for Finding the Stationary Vector for Markov Chains*, Proceedings of the IMA Workshop on Linear Algebra, Markov Chains and Queueing Models, Minneapolis, Minnesota, January 1992, Springer-Verlag.

[11] T. Mathew, *Schwarz Alternating and Iterative Refinement Methods for Mixed Formulations of Elliptic Problems, Part I: Algorithms and Numerical Results*, UCLA CAM Report 92-04, January 1992.

[12] R. Plemmons, *Matrix Iterative Analysis for Markov Chains*, Proceedings of the IMA Workshop on Linear Algebra, Markov Chains and Queueing Models, Minneapolis, Minnesota, January 1992, Springer-Verlag.

[13] W. Proskurowski and O. Widlund, *A Finite Element-Capacitance Matrix Method for the Neumann Problem for Laplace's Equation*, SIAM J. Sci. Statist. Comput., V1, pp. 410-425 (1980).

[14] J. Ruge and K. Stüben, *Algebraic Multigrid* in Multigrid Methods, Frontier in Applied Mathematics, ed. S. McCormick, Philadelphia, SIAM, 1987.

[15] Y. Saad and M. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM J. Sci. Statist. Comput., V7, pp. 856-869 (1986).

[16] R. Varga, *Matrix Iterative Analysis*, New Jersey, Prentice-Hall, 1962.

[17] D. Young and K. Jea, *Generalized Conjugate Gradient Acceleration of Non-symmetric Iterative Method*, Linear Algebra Appl, V34, pp.159-194 (1980).