Computational Characteristics of a Numerical Model for Series of Waterway Queues

CHIEN-HUNG WEI, MELODY D. M. DAI, AND PAUL M. SCHONFELD

A numerical method has been developed for estimating delays on congested waterways represented by series of G/G/1 queues (i.e., with generally distributed arrival and service times and one chamber per lock). It is based on a metamodeling approach that develops simple formulas to approximate the results of simulation models. The functional form of the metamodels is derived from queueing theory, whereas their coefficients are statistically estimated from simulation results. The algorithm scans along a waterway and sequentially estimates at each lock the arrival distributions, departure distributions, and delays. It can be applied to systems with two-way traffic through common bidirectional servers as well as to one-way traffic systems. Computational results are presented to illustrate the speed and convergence properties of the algorithm and to investigate some of its variants. The algorithm works satisfactorily and flexibly with different convergence criteria and scanning processes. For an illustrative 20-lock system, parameter estimates converge with five iterations and less than 3 sec of CPU time to differences lower than 0.1 percent between successive iterations. The computation time increases only linearly with the number of locks in the system, thus allowing the analysis of very large systems of interdependent queues.

Inland waterway transportation is important in the United States and elsewhere, especially for heavy or bulky commodities, since it is inexpensive, energy efficient, and safe. Most U.S. waterways consist of stepped navigable pools formed by dams across natural rivers. The lock structures used to raise or lower vessels between adjacent pools constitute the major bottlenecks in the waterway network (*1*) and generate extensive queues. Some locks have only one chamber, whereas others may have two parallel chambers whose characteristics may differ. The service time distributions at locks depend heavily on chamber size and tow size distributions. The lock service time distributions would be affected by the chamber assignment discipline at locks with two dissimilar chambers.

The waterway locks constitute a series of queueing stations. In queueing terms, locks are the servers and tows are customers waiting to be served by locks. Tows from both directions, upstream and downstream, share the same lock servers, whereas in most other queueing systems servers are exclusively one-directional. Hence, the term "two-way traffic operations" characterizes the lock system analyzed later.

Arrival and service time distributions at locks are fairly complex. Carroll et al. (*2*) and Desai (*3*) found that service times are not exponentially distributed, and arrivals are not Poisson distributed. Other standard distributions have been tested for the present study without consistent success. Thus, empirical distributions (specified for 50 intervals) are used here for simulation, whereas general tabular distributions, described usually only by their means and variances, are used for queueing models. Although locks with a single chamber may be modeled as G/G/1 queueing systems (i.e., general arrival/general service times/1 server per station), locks with two parallel chambers may not be treated simply as G/G/2 queueing systems unless these chambers are identical.

Considerable interdependence may exist among locks in a series. The departure distributions differ from the arrival distributions since the service time distributions change the tow headways. Departures from one lock usually affect arrivals at the next lock. Interdependence among locks increases the difficulty in estimating systemwide delays since the interarrival time distributions from adjacent locks must be identified at each lock. Two-way traffic operation through common servers complicates the interdependence of lock delays and precludes the use of some otherwise interesting queueing models.

Random failures (called stalls) contribute significantly to the difficulties in estimating delays. Stalls, which interrupt lock operations and thereby increase delays, are relatively rare compared with other events and difficult to predict. Thus, Kelejian's efforts to model stall frequencies and durations have not yet yielded strong results despite the rigorous statistical methods employed (*4*).

The following special problems are encountered in estimating delays of waterway queues:

1. Arrival and service time distributions are too complex for analytic solutions and do not match known statistical distributions.
2. Parallel chambers are not identical.
3. Service time distributions are affected by the chamber assignment discipline.
4. Considerable interdependence exists among a series of locks.
5. Two-way traffic operates through bidirectional chambers.
6. Arrival distributions depend on distances and speed distributions between locks, as well as departures from adjacent locks.
7. Stalls increase the means and variances of delays.

Delay estimation for a realistic lock queueing system has been undertaken by Dai and Schonfeld (*5,6,7*) using several

Transportation Studies Center, University of Maryland, College Park, Md. 20742.

approaches, including queueing theory, simulation, and numerical methods. Their simulation model deals with all seven problems listed and is efficient for analyzing particular system configurations. However, when large numbers of system alternatives must be evaluated for investment scheduling, a much faster numerical method, which approximates the results of simulations, becomes preferable. The primary purpose of this paper is to assess the computational characteristics of the numerical method developed for this role. In particular, the number of iterations and the computation time required to reach convergence using various criteria and scanning procedures are investigated. The effects of system size (i.e., number of locks) on computational requirements are also examined.

LITERATURE REVIEW

The available analytic solutions for estimating delays in G/G/1 queues are inadequate. Kleinrock (*8*) suggested an approximation solution for a G/G/1 queue with heavy traffic, which is a useful upper bound for average waiting times in G/G/1 queues. Bertsimas (*9*) derived an exact solution for mixed generalized Erlang distributed arrivals and service times. However, without a departure function this result is difficult to extend to a series of locks.

Exact solutions for networks of queues are still limited to Markovian networks. For more general networks of queues, approximation methods are employed by Whitt (*10*) and Albin (*11*) for system performance analysis. The underlying concept is to decompose the network into individual queues that are analyzed independently and then recombine the results. Their efforts are valuable but employ unreasonable coefficients of variation (standard deviation divided by mean) and are not applicable to bidirectional servers.

System simulation models to analyze lock delays and tow travel times were developed by Howe (*12*) and Carroll and Bronzini (*13*). These two models, which did not account for stalls, required considerable data and computer time. However, simulation models can, in principle, represent the complexities of traffic on waterway networks much better than analytic queueing models.

A new waterway simulation model was developed by Dai and Schonfeld (*5*). This model accommodates generally (i.e., arbitrarily) distributed trips and service times. It can also evaluate stall effects. This simulation model requires only a few seconds to a few minutes on a PS/2 computer for each run, depending on traffic volumes, simulation period durations, network size, and so forth. Still, it is hardly affordable for direct application in large combinatorial network investment problems.

To avoid the computational expense of simulation, a metamodeling approach (*14*) was developed. This approach consists of (*a*) developing and validating a simulation model to represent waterway networks with queues at locks, (*b*) formulating functions developed from queueing theory for delays through series of locks, (*c*) statistically estimating the parameters of these functions using simulation results, and (*d*) employing an iterative sequential scanning procedure to estimate interarrival and interdeparture time distributions lock by lock until results converge at each lock. Thus, relatively simple equations may serve as a proxy for the simulation model.

SIMULATION MODEL

The simulation model developed for this work is documented in Dai and Schonfeld (*5*). Only a brief description is provided below.

The simulation model was developed using the lock performance monitoring system (PMS) data base, which includes detailed information on traffic through the locks as well as physical aspects of lockages (*15*). The simulation model is programmed in Fortran-77, which provides great flexibility in modeling. Basically, it is a stochastic, microscopic and event-scanning simulation model that can handle any distributions for trip generation, travel speeds, lock service times, and tow sizes. Currently, tabular distributions based on empirical observations are used for most input variables. A FIFO (first-in-first-out) service discipline is currently used. This model simulates two-way traffic through common servers and accounts for stalls.

The validation results (*5,6,7*) show that the overall mechanism of the simulation model is correct, and that the simulated average waiting times for each lock and for the entire series of locks are closely similar to those observed. Dai (*6*) documents the statistical methods used in developing, validating, and applying the simulation model.

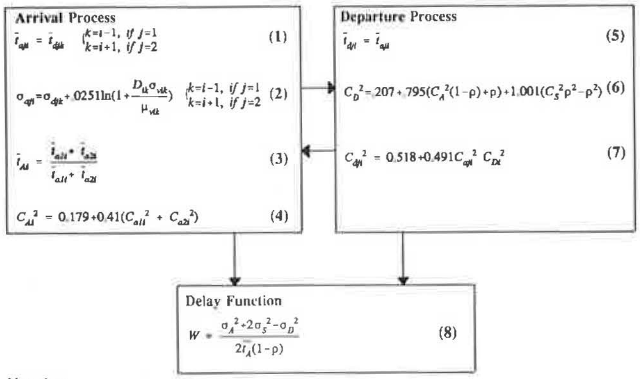
NUMERICAL METHOD

Overview

A numerical method has been developed for estimating delays through a series of queues with bidirectional servers. A brief description of the method follows. Details of its development and validation are provided elsewhere (*6,7*).

The method consists of three major modules, namely arrival processes, departure processes, and delay functions (as summarized in Figure 1), which are applied in that sequence at each lock. The basic concept is to decompose the waterway system into locks (which remain interdependent since they are affected by inflows from adjacent locks), identify the parameters of the interarrival and interdeparture time distributions for each lock, and then estimate the implied waiting times. The structure of the equations used in each module is based as much as possible on queueing theory, and the parameters in those equations are statistically estimated on the basis of simulation results. Currently, the following assumptions are used in the numerical method:

1. Interarrival times and service times are generally distributed.
2. Each lock has one chamber.
3. Inflows and outflows occur only at the two end nodes of a series of locks.
4. The average upstream volumes are equal to the downstream volumes in the long run.
5. The long-run volume to capacity (V/C) ratio is less than 1.0 at every lock.



Notation:

- C_{Ai} : coefficient of variation of interarrival times at Lock i
- C_{aji} : coefficient of variation of directional interarrival times for Direction j and Lock i
- C_{Di} : coefficient of variation of interdeparture times for Direction j and Lock i
- C_{dji} : coefficient of variation of directional interdeparture times for Direction j and Lock i
- C_S : coefficient of variation of service times
- D_{ik} : distance between Locks i and k
- i : index of currently scanned lock
- j : direction index (1 = downstream, 2 = upstream)
- k : index of adjacent locks
- $\bar{t}_{Ai}$: mean interarrival time at Lock i
- $\bar{t}_{aji}$: mean interarrival time for Direction j and Lock i
- $\bar{t}_{djk}$: mean interdeparture time for Direction j and Lock k
- μ_{vik} : mean tow speed between Locks i and k
- σ_{aji} : standard deviation of interarrival times for Direction j and Lock i
- σ_{djk} : standard deviation of interdeparture times for Direction j and Lock k
- σ_{vik} : standard deviation of tow speeds between Locks i and k

FIGURE 1 Structure of numerical method.

Assumptions 2, 3, and 4 are only applicable to the numerical method. The simulation model is not limited by those assumptions. The numerical method can provide a quick and inexpensive analysis of lock delays. However, Assumptions 2, 3, and 4 limit fairly significantly the applicability of the currently developed numerical method and necessitate the substitution of the simulation model when significant deviations from those assumptions must be considered. With some extensions to the numerical method, Assumptions 2 and 3 may be eliminated. Assumption 4 could be relaxed fairly easily even though it is usually realistic for waterways. Assumptions 1 and 5 should be kept since they reflect realities rather than analytic limitations.

Structure of Numerical Method

To estimate delays in a queueing system, we need to know the means and variances of the interarrival, interdeparture, and service time distributions. For series of G/G/1 queues and bidirectional servers, a difficulty arises in identifying the variances of interarrival and interdeparture times. Because the interarrival times at each lock depend on departures from both upstream and downstream locks, the variances of interarrival times cannot be determined from one-directional scans along a series of queues. To overcome such complex interdependence, an iterative scanning procedure is proposed. The core concept is to decompose the system into individual locks and then sequentially analyze each of those locks. At each lock, the tow arrivals from both directions are first combined into an overall arrival distribution and then split into two directional departure distributions.

The algorithm is initiated by scanning along waterways from either direction, sequentially estimating the interarrival and interdeparture time distributions for each lock. Initially assumed values for the variances of interdeparture times from the opposite direction must be provided for the first scan. Then, the scanning direction is reversed and the process is repeated, using the interdeparture time distributions for the opposite direction estimated in the previous scan. Alternating directions, the scanning process continues until the relative difference in the preselected convergence criteria stays within preset thresholds through successive iterations. Waiting times at locks can be computed in every iteration (and then used as convergence criteria) or just once after all iterations are completed.

Arrival Processes

The mean and standard deviations of interarrival times are estimated in two steps. First, the means and standard deviations of directional interarrival times at a particular lock are estimated from the interdeparture time distributions of the adjacent locks. If flows are conserved between locks and if the V/C ratio is less than 1, such relations are represented in Equation 1 (variables are defined in Figure 1):

$$\bar{t}_{aji} = \bar{t}_{djk} \qquad \begin{cases} k = i - 1 \text{ if } j = 1 \\ k = i + 1 \text{ if } j = 2 \end{cases} \tag{1}$$

Because speed variations change headway distributions between locks, Equation 2 was developed to estimate the standard deviation of directional interarrival times at one lock.

$$\sigma_{aji} = \sigma_{djk} + \underset{(.002)}{.0251} \ln\left(1 + \frac{D_{ik}\sigma_{vik}}{\mu_{vik}}\right) \qquad \begin{cases} k = i - 1 \text{ if } j = 1 \\ k = i + 1 \text{ if } j = 2 \end{cases} \tag{2}$$

$R^2 = 0.999954$ $n = 107$ $S_e = 0.0586$ $\mu = 5.1685$

This suggests that, theoretically, the standard deviation of directional interarrival times should be equal to the standard deviation of directional interdeparture times plus an adjustment factor depending on the speed distribution and distance.

Second, the overall mean and coefficient of variation of interarrival times for this lock are estimated on the basis of the coefficients of variation of directional interarrival times.

$$\bar{t}_{Ai} = \frac{\bar{t}_{a1i} * \bar{t}_{a2i}}{\bar{t}_{a1i} + \bar{t}_{a2i}} \tag{3}$$

$$C_{Ai}^2 = \underset{(0.027)}{0.179} + \underset{(0.014)}{0.41}(C_{a1i}^2 + C_{a2i}^2) \tag{4}$$

$R^2 = 0.9188$ $n = 79$ $S_e = 0.0059$ $\mu = 0.988$

In Equation 4, the coefficients of variation of upstream and downstream interarrival times carry the same weight in estimating the overall variance of interarrival times, since the mean directional trip rates are equal (Assumption 4).

Departure Processes

The departures module estimates the mean and coefficient of variation of interdeparture times. On the basis of the flow conservation law, if capacity is not exceeded, the average directional interdeparture equals the corresponding interarrival time:

$$\bar{t}_{dji} = \bar{t}_{aji} \tag{5}$$

The coefficient of variation of interdeparture times is estimated in two steps. First, the coefficient is estimated for combined two-directional departures. Departure processes with generally distributed arrivals and service times are analyzed using Laplace transforms (*8*). Some analytic relations obtained are shown in Dai (*6*). The following metamodel was eventually developed to bypass the difficulties of determining the variance of the lock idle times:

$$C_D^2 = \underset{(0.065)}{0.207} + \underset{(0.066)}{0.795}[C_A^2(1-\rho)+\rho] + \underset{(0.0046)}{1.001}(C_S^2\rho^2 - \rho^2) \tag{6}$$

$R^2 = 0.9984$ $n = 79$ $S_e = 0.0058$ $\mu = 0.8311$

Next, the coefficient of variation of directional interdeparture times is estimated. The following metamodel was developed for this purpose:

$$C_{dji}^2 = \underset{(0.0056)}{0.518} + \underset{(0.0068)}{0.491}\, C_{aji}^2\, C_{Di}^2 \tag{7}$$

$R^2 = 0.9710$ $n = 158$ $S_e = 0.013$ $\mu = 0.9164$

Delay Function

The delay function is intended to estimate the average waiting time at a lock. By applying Marshall's formula for the variance of interdeparture times (*16*), an exact solution for the average waiting time W was obtained as follows:

$$W = \frac{\sigma_A^2 + 2\sigma_S^2 - \sigma_D^2}{2\bar{t}_A(1-\rho)} \tag{8}$$

In this delay function, the average waiting time increases as the variance of interarrival and service times increases and decreases as the variance of interdeparture times increases. The average waiting time approaches infinity as the V/C ratio approaches 1.0.

Comparison of Simulated and Numerical Results

To validate the numerical method, its results were compared with the results of the previously validated simulation model. Various system configurations were compared, including the relatively large 20-lock system given in Table 1.

The parameter values for this test system (e.g., means and standard deviations of input distributions and distances between locks) were obtained from random number generators, except for traffic volumes, which were assumed to be 10 tows/day in each direction throughout the system. Table 1 gives the input parameters and a comparison of waiting times, which are the output variable of greatest practical interest. It can be seen that the numerical model estimates aggregate waiting times within 7.85 percent of the simulated ones. At individual locks the percentage error can be considerably greater, especially when absolute errors are very small (e.g., in comparisons with zero waiting times). The comparisons of intermediate outputs (e.g., the parameters of directional interarrival and interdeparture time distributions) show that differences below 10 percent are achieved. The detailed validation results are presented in Dai (*6*).

COMPUTATIONAL TESTS

A number of computational tests have been conducted to investigate the speed, accuracy, and convergence properties of the numerical method. Some of the results obtained are presented here. All were obtained with the two-directional iterative algorithm (coded in Fortran-77) compiled and executed on an IBM PS/2 model 70 personal computer with an 80386 processor and an 80387 math coprocessor.

Any variable that is computed in every iteration of the algorithm may be used to check for convergence and stop the algorithm when further changes between iterations become arbitrarily small. The most interesting candidate variables for convergence criteria are the variances in the interdeparture times from each lock (which affect error propagation) and the waiting times in queues (which are the output variables of greatest practical economic interest).

The convergence threshold may be specified as a relative change in the value of a variable from one iteration to the next (i.e., a ratio or percentage change) or an absolute difference. The ratios may be large if and when some variable values approach zero even though absolute differences may be insignificant.

Convergence may be sought on the basis of aggregate or systemwide outputs (e.g., total delay per tow through a series of locks) or may be based on localized outputs (e.g., delay at each lock). In principle, it should be easier to reduce changes between iterations to x percent for a systemwide variable than for every single location in that system.

The original algorithm used the squared coefficients of variation of directional interdeparture times (VARDEP) as the convergence criteria. In this work, the individual lock waiting times (LOCWAIT) and system weighted waiting times (SYSWAIT) are also tested as convergence criteria. Waiting times must then be computed in every iteration rather than just at the end.

The required inputs for the algorithm include the inflow rates, V/C ratio and service time variance at each lock, distances between locks, means and standard deviations of tow speed distributions, and the choice of convergence criterion. We generally used 0.001 as the convergence threshold (i.e.,

TABLE 1 VALIDATION OF NUMERICAL METHOD FOR 20-LOCK TEST CASE

Lock	σ_A[a]	C_A[b]	σ_D	C_D	σ_S	C_S	V/C	Dist[c]
1	1.21	1.01	0.92	0.77	0.52	0.56	0.78	7.04
2	1.18	0.98	1.17	0.98	0.10	0.70	0.12	49.04
3	1.20	1.00	0.91	0.76	0.74	0.69	0.90	46.05
4	1.19	0.99	1.05	0.88	0.69	0.76	0.75	47.74
5	1.20	1.00	1.02	0.85	0.81	0.78	0.86	105.56
6	1.19	0.99	0.90	0.75	0.61	0.60	0.84	71.76
7	1.19	0.99	1.05	0.88	0.91	0.84	0.90	39.91
8	1.21	1.01	1.20	1.00	0.13	0.57	0.19	91.12
9	1.19	0.99	0.94	0.79	0.65	0.65	0.83	60.55
10	1.17	0.97	0.99	0.83	0.75	0.74	0.85	22.44
11	1.21	1.01	1.08	0.90	0.56	0.71	0.66	53.38
12	1.22	1.02	1.20	1.00	0.23	0.67	0.28	89.78
13	1.22	1.02	1.19	0.99	0.28	0.69	0.34	103.77
14	1.23	1.02	0.89	0.74	0.57	0.57	0.83	125.02
15	1.21	1.01	1.16	0.97	0.35	0.71	0.41	105.41
16	1.22	1.02	1.18	0.99	0.37	0.80	0.39	80.29
17	1.21	1.01	1.02	0.85	0.45	0.57	0.66	99.98
18	1.20	1.00	0.95	0.79	0.62	0.64	0.81	65.54
19	1.18	0.99	1.13	0.94	0.45	0.74	0.51	42.38
20	1.22	1.02	0.96	0.80	0.71	0.70	0.85	96.75

	Estimated Waiting Time, hrs/tow			
Lock	Numerical	Simulation	Difference	%
1	2.15	2.04	0.11	5.31
2	0.00	0.01	-0.01	--[d]
3	6.91	6.37	0.54	8.48
4	1.91	1.78	0.12	6.88
5	4.71	4.20	0.50	11.96
6	3.39	2.67	0.73	27.19
7	7.76	7.23	0.53	7.38
8	0.00	0.04	-0.03	--
9	3.30	2.83	0.47	16.48
10	4.21	3.73	0.49	13.02
11	1.10	1.08	0.01	1.04
12	0.08	0.10	-0.03	-27.02
13	0.13	0.17	-0.04	-22.84
14	3.33	3.20	0.13	4.19
15	0.22	0.26	-0.04	-16.41
16	0.21	0.26	-0.05	-19.91
17	0.95	0.99	-0.04	-4.26
18	2.80	2.74	0.06	2.29
19	0.42	0.45	-0.03	-7.54
20	4.34	4.27	0.08	1.81
System	47.92	44.44	3.49	7.85

[a]σ_i: Standard deviation of interarrival time, interdeparture time, and service time distributions, respectively.
[b]C_i: Coefficients of variation of interarrival time, interdeparture time, and service time distributions, respectively.
[c]Dist: Distance to the next lock, in miles.
[d]Not applicable.

results were considered sufficiently accurate and additional iterations were deemed unnecessary when the variables chosen as convergence criteria changed by less than 0.1 percent from the previous iteration).

Three-Lock Systems

The first test concerns the eight three-lock systems analyzed in Dai (*6*). These eight systems (described in Table 2) were originally used to show the performance of various algorithms. The distances and speed distributions between locks were kept equal within each of these eight systems. Using VARDEP, LOCWAIT, and SYSWAIT as convergence criteria, the estimated individual lock delays and system delays and number of iterations required are listed in Table 3. Also included are the simulated waiting times. Generally, the three criteria perform equally well for each of the eight systems in terms of number of iterations required for convergence. The SYSWAIT criterion produces slightly faster convergence than the others.

While assessing the differences in the number of iterations required with various criteria in System 1, we found that delays at low V/C ratios are so small that relative differences may be large and unstable even for very small changes in the absolute magnitudes of delays. Consequently, more iterations are required to satisfy a relative threshold. If, instead, we set an absolute threshold for delay (e.g., less than 0.001 hr/tow difference between successive iterations), System 1 converges at the fourth iteration for both LOCWAIT and SYSWAIT.

We also sought to check whether the convergence was monotonic (i.e., whether the changes always decrease through successive iterations). We found that relative changes decrease monotonically for all systems when SYSWAIT, but not VARDEP or LOCWAIT, is the convergence criterion. However, the magnitudes of various criterion variables change monotonically through successive iterations for all systems, as shown in Table 4. It seems that monotonic convergence is

TABLE 2 PHYSICAL CHARACTERISTICS OF THREE-LOCK SYSTEMS

System	Lock	Two-way Flow Rate tows/day	V/C	Distance miles	Tow Speed miles/day μ_v[a]	σ_v[b]	Variance of Service Time hr²/tow²
1	1	6.0	0.01	5	270	85	0.0007
	2	6.0	0.07	5	270	85	0.0360
	3	6.0	0.17	5	270	85	0.1897
2	1	12.0	0.15	5	325	102	0.0309
	2	12.0	0.34	5	325	102	0.1620
	3	12.0	0.25	5	325	102	0.0915
3	1	18.0	0.22	5	108	34	0.0309
	2	18.0	0.03	5	108	34	0.0006
	3	18.0	0.50	5	108	34	0.1618
4	1	24.0	0.50	5	162	51	0.1883
	2	24.0	0.29	5	162	51	0.0646
	3	24.0	0.67	5	162	51	0.3330
5	1	27.0	0.75	10	108	34	0.2271
	2	27.0	0.57	10	108	34	0.1279
	3	27.0	0.89	10	108	34	0.3167
6	1	27.0	0.75	20	216	68	0.1616
	2	27.0	0.57	20	216	68	0.0909
	3	27.0	0.89	20	216	68	0.2259
7	1	28.5	0.60	5	325	102	0.1557
	2	28.5	0.05	5	325	102	0.0011
	3	28.5	0.80	5	325	102	0.2738
8	1	28.5	0.35	60	162	51	0.0645
	2	28.5	0.60	60	162	51	0.1882
	3	28.5	0.80	60	162	51	0.3332

[a]μ_v: Average tow speed.
[b]σ_v: Standard deviation of tow speeds.

more difficult to achieve for local variables when the algorithm scans along the series of locks in alternating directions. When an iteration is defined as a two-way scan (e.g., first upstream, then downstream, and only afterwards compare results to the previous iteration), monotonic convergence is achieved for the local variables LOCWAIT and VARDEP. It is achieved without two-way iterations for the aggregate variable SYSWAIT which, incidentally, requires 3 to 12 percent less CPU time than the local criteria.

The algorithm was also allowed to run for 100 iterations to check the convergence and CPU times for various criteria. The results were quite satisfactory since no system ever diverged in this experiment. This is illustrated in Figure 2 using System 6 as the example.

Twenty-Lock Systems

To further check the behavior of the algorithm, we randomly generated parameter values for a 20-lock system in which the values of the V/C ratio were uniformly distributed between 0 and 1, and the coefficients of variation of service time were uniformly distributed between 0.2 and 1.0. This test system was assumed to have equal mean inflow rates in the two directions, as well as identical tow speed distributions and distances between any pair of locks. Table 5 describes this 20-lock system.

The aggregate results for the 20-lock system are summarized in Table 6. We found that the number of iterations required for convergence within 0.001 is almost identical to the numbers in Table 3, even though this 20-lock system is more than six times larger. This suggests that the algorithm may be applicable for very large systems. Comparisons of CPU times required for convergence again confirm that the aggregate criterion SYSWAIT saves iterations compared with the local criteria LOCWAIT and VARDEP and reaches convergence with approximately 25 percent less CPU time. As in 3-lock systems, the 20-lock system never diverges, and the monotonic properties with various criteria are similar. With the LOCWAIT criterion a single violation of monotonic convergence was found at Lock 2 in the fourth iteration. Consequently, one more scan is desired to bring the entire system into convergence. Such violations were never found when the aggregate convergence criterion SYSWAIT was used or when iterations were defined to consist of two scans in alternate directions.

The relation between system size and computational requirements was also examined using the 20-lock system and arbitrarily chosen subsets of that system. The CPU times and number of iterations required for convergence in various system sizes are shown in Figure 3. It again seems promising that the number of iterations does not change much for different criteria and system sizes. The CPU times seem roughly proportional to system sizes in all cases. Figure 3 demonstrates the apparently linear relations. We sought to statistically estimate the relations between CPU time and the number of locks in the system, using the following structural form:

TABLE 3 COMPUTATIONAL COMPARISON FOR VARIOUS CRITERIA IN THREE-LOCK SYSTEMS

		Estimated Waiting Time, hrs/tow						
			VARDEP		LOCWAIT		SYSWAIT	
System	Lock	Wsim[a]	Wv[b]	Dv[c]	WL	DL	Ws	Ds
1	1	0.0003	0.0001	-0.0002	0.0001	-0.0002	0.0001	-0.0002
	2	0.0153	0.0175	0.0022	0.0176	0.0023	0.0176	0.0023
	3	0.0989	0.0990	0.0001	0.0990	0.0001	0.0990	0.0001
	Total	0.1145	0.1166	0.0021	0.1167	0.0022	0.1167	0.0022
Required Iterations			5		7		6	
2	1	0.0334	0.0290	-0.0044	0.0290	-0.0044	0.0290	-0.0044
	2	0.2316	0.2289	-0.0027	0.2289	-0.0027	0.2289	-0.0027
	3	0.1139	0.1099	-0.0040	0.1099	-0.0040	0.1099	-0.0040
	Total	0.3789	0.3678	-0.0111	0.3678	-0.0111	0.3678	-0.0111
Required Iterations			5		5		5	
3	1	0.0542	0.0528	-0.0014	0.0528	-0.0014	0.0528	-0.0014
	2	0.0008	0.0001	-0.0007	0.0001	-0.0007	0.0001	-0.0007
	3	0.4621	0.4660	0.0039	0.4660	0.0039	0.4659	0.0038
	Total	0.5171	0.5189	0.0018	0.5189	0.0018	0.5188	0.0017
Required Iterations			5		5		4	
4	1	0.4355	0.4404	0.0049	0.4404	0.0049	0.4404	0.0049
	2	0.0962	0.0999	0.0037	0.0999	0.0037	0.0999	0.0037
	3	1.2028	1.1844	-0.0184	1.1844	-0.0184	1.1844	-0.0184
	Total	1.7345	1.7247	-0.0098	1.7247	-0.0098	1.7247	-0.0098
Required Iterations			4		4		4	
5	1	1.3926	1.4693	0.0767	1.4693	0.0767	1.4693	0.0767
	2	0.3901	0.4127	0.0226	0.4127	0.0226	0.4127	0.0226
	3	4.9837	4.7980	-0.1857	4.7980	-0.1857	4.7980	-0.1857
	Total	6.7664	6.6800	-0.0864	6.6800	-0.0864	6.6800	-0.0864
Required Iterations			4		4		4	
6	1	1.2203	1.3038	0.0835	1.3038	0.0835	1.3038	0.0835
	2	0.3286	0.3416	0.0130	0.3416	0.0130	0.3416	0.0130
	3	4.4608	4.2983	-0.1625	4.2983	-0.1625	4.2983	-0.1625
	Total	6.0097	5.9437	-0.0660	5.9437	-0.0660	5.9437	-0.0660
Required Iterations			4		4		4	
7	1	0.5430	0.5900	0.0470	0.5899	0.0469	0.5899	0.0469
	2	0.0012	0.0001	-0.0011	0.0001	-0.0011	0.0001	-0.0011
	3	2.0874	2.0906	0.0032	2.0906	0.0032	2.0906	0.0032
	Total	2.6316	2.6807	0.0491	2.6806	0.0490	2.6806	0.0490
Required Iterations			4		3		3	
8	1	0.1372	0.1405	0.0033	0.1405	0.0033	0.1405	0.0033
	2	0.6381	0.6592	0.0211	0.6592	0.0211	0.6592	0.0211
	3	2.3165	2.3146	-0.0019	2.3146	-0.0019	2.3146	-0.0019
	Total	3.0918	3.1143	0.0225	3.1143	0.0225	3.1143	0.0225
Required Iterations			4		4		4	

[a]Wsim: Waiting time estimated from simulation.
[b]Wi: Waiting time estimated when criterion i used.
[c]Di: Difference between numerically estimated waiting time at a given iteration and simulated waiting time = Wi - Wsim.

$$CPU_i = K_i N^{P_i} \quad (9)$$

In Equation 9 CPU_i is the central processing time using Convergence Criterion i, K_i and P_i are statistically estimated parameters associated with Criterion i, and N is the number of locks in the system. The P_i parameter was expected to be very close to 1.0, on the basis of the nearly linear relations shown in Figure 3, and indeed turned out to be nearly 1.0, confirming the essentially linear relation. The value of P_i was, therefore, fixed at 1.0, and the remaining parameter K_i was estimated as indicated in Table 7.

The small standard errors and high R^2 again confirm that CPU time is essentially linear with respect to the number of locks in the system. Among the three criteria, the aggregate criterion SYSWAIT has the smallest standard error and highest R^2, suggesting it yields not only the fastest but also the most predictable computer times. The structural form of Equation 9 forces the computer time function through the origin, since Equation 9 has no intercept. When an intercept A_i is provided in Equation 10 (presumably to reflect the fixed times required for setup or input and output functions), even better fits were obtained, as indicated in Table 7.

$$CPU_i = A_i + K_i N \quad (10)$$

The best fit is again obtained for the SYSWAIT criterion. Thus, based on our very small sample, the best estimate of CPU time (in seconds to reach convergence within 0.001) for N-lock systems is obtained with the SYSWAIT criterion as

$$CPU = 0.107 + 0.0853N \quad (11)$$

Table 6 shows that convergence to within 0.1 percent difference between successive iterations is reached in 1.75 sec of CPU time for the 20-lock system and SYSWAIT criterion. The corresponding time for the simulation model to analyze the same 20-lock system on the same computer is 53 min per replication (i.e., 1,590 min or 95,400 sec for 30 replications).

TABLE 4 CONVERGENCE PROPERTIES FOR VARIOUS CRITERIA IN SYSTEM 2

	Criterion: VARDEP					
	Magnitude, Dir 1			Relative Difference, Dir 1		
Iter	Lock 1	Lock 2	Lock 3	Lock 1	Lock 2	Lock 3
1	0.9930	0.9629	0.9715	--[a]	--	--
2	1.0027	0.9724	0.9715	0.0098	0.0098	0.0000
3	1.0027	0.9778	0.9802	0.0000	0.0056	0.0090
4	1.0030	0.9780	0.9802	0.0003	0.0002	0.0000
5	1.0030	0.9782	0.9804	0.0000	0.0002	0.0002
	Magnitude, Dir 2			Relative Difference, Dir 2		
Iter	Lock 1	Lock 2	Lock 3	Lock 1	Lock 2	Lock 3
1	0.9456	0.9214	0.9889	--	--	--
2	0.9884	0.9705	0.9889	0.0453	0.0533	0.0000
3	0.9884	0.9715	0.9907	0.0000	0.0011	0.0018
4	0.9897	0.9725	0.9907	0.0013	0.0010	0.0000
5	0.9897	0.9726	0.9907	0.0000	0.0000	0.0000
	Criterion: LOCWAIT					
	Magnitude			Relative Difference		
Iter	Lock 1	Lock 2	Lock 3	Lock 1	Lock 2	Lock 3
1	0.0177	0.1994	0.1065	--	--	--
2	0.0287	0.2254	0.1065	0.6167	0.1304	0.0000
3	0.0287	0.2283	0.1098	0.0000	0.0126	0.0312
4	0.0290	0.2288	0.1098	0.0110	0.0023	0.0000
5	0.0290	0.2289	0.1099	0.0000	0.0004	0.0007
	Criterion: SYSWAIT					
	Magnitude			Relative Difference		
Iter	System			System		
1	0.3236			--		
2	0.3606			0.1142		
3	0.3667			0.0171		
4	0.3676			0.0023		
5	0.3677			0.0004		

[a]Not applicable.

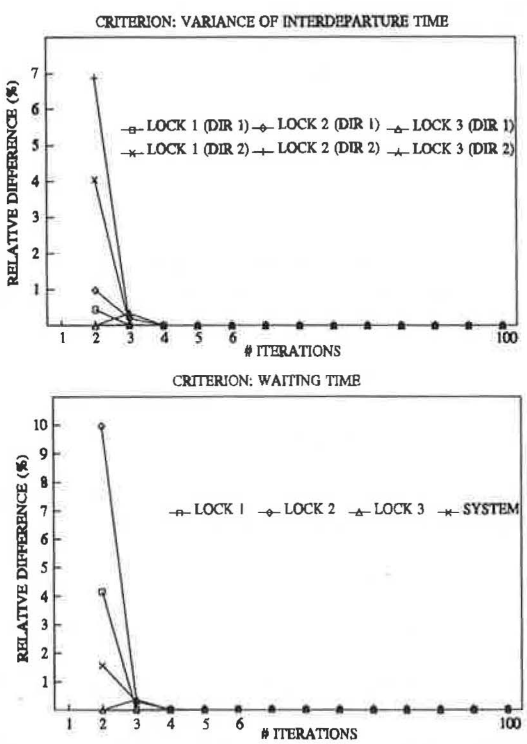


FIGURE 2 Convergence for three-lock system with various criteria.

Thus, in this case simulation requires 54,514 times more CPU time than the numerical method. However, it should be noted that our simulation runs were designed to extract very precise estimates for estimating new metamodels. We usually simulated 22,000 tows, discarded the first 10,000 of those, and replicated the simulation 30 to 80 times for each "data point." For practical application, the simulation would require 10^4 to 10^5 times more CPU time than the numerical method.

Double Scanning Versus Single Scanning

In the baseline algorithm an iteration consists of scanning the waterway from one end to the other (i.e., in one direction). The next iteration would then scan in the opposite direction. The results obtained so far suggest that a smoother convergence may be obtained by double scanning (i.e., checking for convergence only after two full scans in opposite directions are completed). With such double scanning, the changes in variables are always found to decrease (or at least not increase) with each successive convergence check, which is performed every second iteration by comparing Iteration i with Iteration $i - 2$ (instead of $i - 1$).

However, double scanning imposes a computer time penalty by increasing the number of iterations required for convergence to a specified threshold. That is indicated in Table 8, where the convergence threshold is still 0.001. There are

TABLE 5 RELEVANT DATA FOR THE 20-LOCK SYSTEM

Lock	V/C	C_s[a]	Cap[b]	μ_s[c]	σ_s^2[d]
1	0.5625	0.2482	48	0.5000	0.0154
2	0.2473	0.2591	109	0.2198	0.0032
3	0.4505	0.3725	60	0.4004	0.0223
4	0.4098	0.2942	66	0.3643	0.0115
5	0.9865	0.8953	27	0.8769	0.6163
6	0.2148	0.2328	126	0.1909	0.0020
7	0.8315	0.5422	32	0.7391	0.1606
8	0.7088	0.3447	38	0.6300	0.0472
9	0.8563	0.9832	32	0.7612	0.5601
10	0.5989	0.8641	45	0.5324	0.2116
11	0.2065	0.6823	131	0.1836	0.0157
12	0.0510	0.5392	529	0.0453	0.0006
13	0.9894	0.8309	27	0.8795	0.5340
14	0.5051	0.4834	53	0.4490	0.0471
15	0.6715	0.6363	40	0.5969	0.1442
16	0.6728	0.7805	40	0.5980	0.2179
17	0.9475	0.6943	28	0.8422	0.3419
18	0.8662	0.4078	31	0.7700	0.0986
19	0.9074	0.4017	30	0.8066	0.1050
20	0.8711	0.9968	31	0.7743	0.5957

Inflow Rate of Direction 1 (tows/day)	13.5
Inflow Rate of Direction 2 (tows/day)	13.5
Convergence Threshold	0.001
Tow Speed (miles/day)	213.48
Standard Deviation of Speed (miles/day)	67.68
Distance between Locks (miles)	20.0

[a]C_s: Coefficient of variation of service time distribution.
[b]Cap: Lock capacity, tows/day.
[c]μ_s: Mean of service time distribution, hrs/tow.
[d]σ_s^2: Variance of service time distribution, hrs^2/tow^2.

TABLE 6 COMPUTATION RESULTS FOR THE 20-LOCK SYSTEM

	VARDEP	LOCWAIT	SYSWAIT
Required Iterations for Convergence Within 0.001	4	5	4
CPU Time (seconds)	2.15	2.36	1.75
Total Waiting Time (hrs/tow)	151.2056	151.2044	151.2056
100 Iterations			
Divergence	None	None	None
CPU Time (seconds)	31.25	34.38	27.14
Total waiting time (hrs/tow)	151.2043	151.2043	151.2043

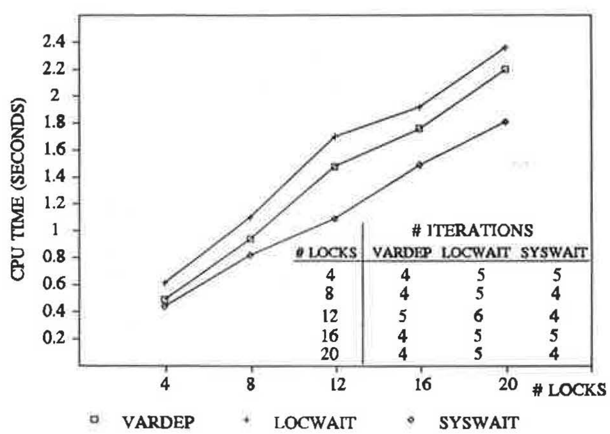


FIGURE 3 Relations between system size and computational speed.

two reasons for the penalty. First, an even number of iterations is required in double scanning, even when convergence is reachable with one less iteration. Second, a larger change may be expected after two iterations than after one, making the same threshold (e.g., 0.001) harder to satisfy.

Thus, it seems that double scanning provides added reassurance that the algorithm converges in a smooth and well-behaved way. However, since convergence seems so assured regardless of scanning procedure, it seems preferable to opt for the computation savings of single scanning.

CONCLUSIONS

A numerical method has been developed to estimate waterway travel times through a series of lock queues. This nu-

TABLE 7 PARAMETERS FOR CPU TIME VERSUS SYSTEM SIZE

Criterion	K_1	Standard Error of K_1	Standard Error of CPU Estimate	R^2	A_1
Eq. 9					
VARDEP	0.1129	0.0026	0.0774	0.9867	
LOCWAIT	0.1245	0.005	0.1482	0.9537	
SYSWAIT	0.0925	0.0019	0.0578	0.9885	
Eq. 10					
VARDEP	0.106	0.0055	0.0696	0.9919	0.102
LOCWAIT	0.108	0.0084	0.1074	0.9817	0.242
SYSWAIT	0.0853	0.0024	0.0315	0.9974	0.107

TABLE 8 ITERATIONS REQUIRED FOR VARIOUS SCANNING PROCESSES

3-Lock System 1	VARDEP	LOCWAIT	SYSWAIT
Single Scan	5	7	6
Double Scan	6	8	8
20-Lock System			
Single Scan	4	5	4
Double Scan	6	6	6

merical method was estimated from simulation results. It can approximately duplicate simulation results for complex systems of interdependent queues, while requiring 10^4 to 10^5 times less computer time than simulation. The basic approach used in this numerical method and several of its components (or "metamodels") should lead to numerical analysis methods for other types of queueing networks with greater complexity.

This paper focused on the main computational characteristics of the baseline numerical method and some its variations. The main computational findings are as follows:

1. Variables other than the original interdeparture time variance VARDEP are suitable as convergence criteria. In particular, the aggregate waiting time SYSWAIT yields convergence faster than the other variables considered. Not surprisingly, more iterations may be needed if a specified convergence threshold (e.g., 0.1 percent) is to be satisfied at every location and in every direction rather than for an aggregate criterion.
2. Convergence to within 0.1 percent of values in the previous iteration is achieved relatively quickly (typically in four to six iterations), even when that 0.1 percent threshold must be satisfied everywhere in a 20-lock system.
3. Convergence is achieved smoothly and, with rare exceptions, differences in the variable values decrease with each successive iteration. The exceptions are all traceable to scans in alternating directions and can be avoided by double scanning before convergence checks or by always scanning in the same directions. However, since convergence seems always assured, the single scanning in alternating directions seems preferable to save computer time.
4. The computer time required by the algorithm seems to be linear with respect to the number of locks in the system. It also seems to be predictable. Thus, the numerical method should analyze efficiently relatively large systems of interdependent queues.

ACKNOWLEDGMENT

The authors wish to thank George Antle and the Institute for Water Resources of the U.S. Army Corps of Engineers for advice and support in this work.

REFERENCES

1. *National Waterways Study—A Framework For Decision Making—Final Report*. Water Resource Support Center, Institute for Water Resources, U.S. Army Corps of Engineers, Fort Belvoir, Va., 1983.
2. J. L. Carroll, M. S. Bronzini, and J. C. Hayward. *Analysis of Waterway Systems: Summary Report*. Pennsylvania Transportation and Traffic Safety Center, University Park, Pa., 1972.
3. K. J. Desai. *Analysis of Waterway Systems: A Study of Tow Arrival Processes at Inland Waterway Locks*. Pennsylvania Transportation and Traffic Safety Center, University Park, Pa., 1972.
4. H. Kelejian. *A Study of Lockage Stall Probabilities*. Institute for Water Resources, U.S. Army Corps of Engineers, Fort Belvoir, Va., 1989.
5. M. D. M. Dai and P. M. Schonfeld. Simulation of Waterway Transportation Reliability. Forthcoming in *Transportation Research Record*, TRB, National Research Council, Washington, D.C., 1991.
6. M. D. M. Dai. *Delay Estimation on Congested Waterways*. Ph.D. dissertation. University of Maryland, College Park, Md., 1991.
7. M. D. M. Dai and P. M. Schonfeld. Metamodels for Estimating Waterway Delays Through Series of Queues. Forthcoming in *Transportation Research, Part B*, 1991.
8. L. Kleinrock. *Queueing Systems Volume I: Theory*. John Wiley and Sons, Inc., New York, 1975.
9. D. Bertsimas. An Exact FCFS Waiting Time Analysis for a General Class of G/G/s Queueing Systems. *Queueing Systems Theory and Applications*, Vol. 3, 1988, pp. 305–320.
10. W. Whitt. The Queueing Network Analyzer. *The Bell System Technical Journal*, Vol. 62, 1983, pp. 2779–2815.
11. S. L. Albin. Approximating a Point Processes by a Renewal Process, II: Superposition Arrival Processes to Queues. *Operations Research*, Vol. 32, 1984, pp. 1133–1162.
12. C. W. Howe et al. *Inland Waterway Transportation Studies in Public and Private Management and Investment Decisions*. Johns Hopkins Press, Baltimore, Md., 1969.
13. J. L. Carroll and M. S. Bronzini. Waterway Transportation Simulation Models: Development and Application. *Waterway Resources Research*, Vol. 9, No. 1, 1973, pp. 51–63.
14. A. Law and W. Kelton. *Simulation Modeling and Analysis* (2nd ed.). McGraw Hill, Inc., New York, 1991.
15. M. Fleming, D. Wood, and R. Goodwin. *Lock Performance Monitoring System User's Manual for Data Analysis*. Institute for Water Resources, U.S. Army Corps of Engineers, Fort Belvoir, Va., 1985.
16. K. T. Marshall. Some Inequalities in Queuing. *Operations Research*, Vol. 16, 1968, pp. 651–665.

Publication of this paper sponsored by Committee on Inland Water Transportation.