Rapid improvement of stochastic networks using two-moment approximations

Julien Granger\textsuperscript{a}, Ananth Krishnamurthy\textsuperscript{b}, Stephen M. Robinson\textsuperscript{a,}\textsuperscript{*}

\textsuperscript{a} Department of Industrial and Systems Engineering, University of Wisconsin-Madison, 1513 University Avenue, Madison, WI 53706-1572, USA
\textsuperscript{b} Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180-3522, USA

Received 10 March 2005; accepted 4 May 2005

Abstract

This paper investigates a method for rapidly improving the performance of stochastic networks. The method uses queueing network techniques based on parametric decomposition and two-moment approximations. It employs some recently devised developments to the approximations, especially in the context of closed queueing networks with synchronization stations. We describe these developments and explain how the method uses them. It appears to offer a substantial advantage in speed, and reduces the dependence on repetitive simulation runs for identifying opportunities to improve network performance. We present an example from airfield operations, in which the method evaluates the tradeoffs with respect to several strategies for enhancing airfield performance, such as the introduction of concurrent operations, the reduction of service variability, and the increase in airfield capacity.

\copyright 2005 Elsevier Ltd. All rights reserved.

Keywords: Closed queueing networks; Fork/join synchronization station; Parametric decomposition; Two-moment approximations; Airfield operations

1. Introduction

Stochastic networks are commonly used to model the interaction of elements in complex systems found in manufacturing, logistics, health care, and many other areas. Simulation optimization techniques are used to identify ways to optimize performance of such stochastic networks. These approaches typically combine repetitive modeling of the network via simulation with parameter adjustment in order to achieve better performance. One disadvantage of this method is that the repetitive simulation runs may require substantial amounts of computing time, and accordingly the optimization procedure may be costly in terms of time and computing resources.

\textsuperscript{*} Corresponding author. Tel.: +1 608 263 6862.
E-mail addresses: granger@cae.wisc.edu (J. Granger), krisha@rpi.edu (A. Krishnamurthy), smrobins@wisc.edu (S.M. Robinson).


0895-7177/S - see front matter \copyright 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2005.05.023
This paper investigates an alternative method for rapidly improving the performance of stochastic networks. Instead of the usual repetitive simulation runs, queueing network techniques based on parametric decomposition and two-moment approximations are used to evaluate system performance. In the parametric decomposition approach, the traffic processes (arrival, service and departure) in the queueing network are approximated by renewal processes and estimates of performance measures at the individual stations in the network are characterized in terms of the mean and SCV (squared coefficient of variation, defined in Section 2) of the inter-renewal times of those renewal processes. These characterizations are called two-moment approximations. Such approximations permit one to estimate the performance of the system in a particular configuration by solving a system of nonlinear equations, and therefore they appear to offer a substantial advantage in speed compared to simulation.

Our method employs some recently devised developments to the approximations, especially in the context of closed queueing networks with synchronization stations. We describe these developments and explain how the method uses them. To illustrate how one can apply these approximations, we present an example from airfield operations in which the method identifies bottlenecks and shows how to invest in capacity to reduce them and improve performance. The rest of this introduction explains the background of this example and reviews the relevant literature.

1.1. Background

An important component of an air logistics network is the collection of individual airfields through which aircraft in the network must pass. Therefore, airfield performance measures such as the throughput capacity and server bottlenecks are important concerns for logistics planners. In turn, the various operations that aircraft undergo at such airfields, such as landing, parking, cargo handling, refuelling, liquid oxygen servicing, maintenance, taxiing out and takeoff, strongly influence these performance measures. A detailed model of these operations can be very complex, and therefore a common practice in network optimization models is to represent these operations by a few parameters such as aggregate aircraft ground time or airfield throughput.

However, it is usually difficult to obtain reliable estimates of the key parameters characterizing airfield operations, because of the stochastic nature of the various service activities performed on aircraft and the interactions among these activities. Consequently, detailed studies of airfield operations have historically been performed by using simulation models. Often, simulation is the only tool that can model the complex operations in a given setting.

Detailed simulation models typically require long run times. While this might be acceptable when the planner is interested in analyzing airport operations in a fixed setting, a planner is often interested in not only how the system performs in a particular setting, but also in how the system should be changed so that its performance is closer to some set of desired standards. In such situations, the planner is interested in rapid analysis of alternative system designs and evaluating performance tradeoffs. Simulation models become cumbersome for such an analysis. However, substantial insight and reasonable performance estimates can be obtained by using analytical queueing models of airfield operations. Efficient queueing models capture the stochastic nature of airfield operations and can have reasonable computational requirements.

In this paper we discuss a new approach for analyzing queueing models of airfield operations. This approach is not only fast but is also capable of providing reasonably accurate estimates of performance measures at an individual airfield. As an example application, we demonstrate how one can use these queueing models to quantify the potential benefits of conducting operations concurrently instead of sequentially.

1.2. Literature review

In this subsection we review several models relevant to air logistics deployment in general, and airfield operations in particular. We also comment on the solution methodologies used in those models.

The NPS/RAND Mobility Optimizer (NRMO) developed in [1] is a large-scale deterministic linear programming model for optimizing strategic airlift capability. The model routes cargo and personnel through a transportation network with a fleet of aircraft subject to physical, policy and time constraints. Operations experienced by aircraft are not modeled explicitly as in this paper, but instead are aggregated as a deterministic waiting time. The deterministic linear programming model is used to build detailed aircraft schedules that minimize a chosen measure, such as the lateness in cargo delivery.
However, one disadvantage of NRMO is that in a real deployment aircraft usually are subject to random delays while performing their missions. To address this limitation, Niemi [2] models travel and repair time in one version of NRMO using a discrete probability distribution. The solution of the resulting stochastic linear program shows that in the presence of stochasticity, system performance degrades significantly.

The need to take randomness into account explicitly without letting the model become computationally intractable motivated our prior work [3,4]. There we investigated the applicability of queueing network approximation methods, which were initially developed for manufacturing and communication systems, for analyzing an airlift system. The methodology uses parametric decomposition and two-moment approximations for an open network of queues, which are originally developed in [5] and [6]. Granger et al. [3,4] demonstrate situations in which such approximations yield sufficient accuracy to allow an improvement procedure, coupled with simulation, for a simple airlift system. The overall method is to start with an initial simulation run, then search for a better network resource allocation by repeatedly solving the approximation model. A final simulation run verifies the improvement in network performance. The quick solution produced by the approximation model greatly reduces the computational effort required to guide the improvement in resource allocation.

The analysis of airfield operations constitute a building block for larger models like NRMO. Since it is important to analyze the impact of randomness in these models, it is natural to model airfield operations by using queueing networks. The papers by Dietz and Jenkins [7], Dietz [8] and Willits and Dietz [9] describe how methods devised for queueing networks can be extended to model airfield operations and provide rapid insights into the relationship between airfield resource level, the flow of aircraft, and airfield throughput capacity. These methods make use of fork and join stations that allow one to model the concurrent use of constrained resources.

In [7], the classical method of mean value analysis (MVA) for closed network of queues is used, in conjunction with an extension of a method originally developed in [10]. The authors of [10] developed a heuristic to analyze a single fork and join station in a network of single-server queues. Dietz and Jenkins in [7] extend this heuristic to accommodate multiple fork and join stations in a larger network with multiple server activities and probabilistic service requirements.

In [8] this heuristic is extended to address the possibility of multiple stations on a fork–join path, including nested fork and join constructs. For the two network configurations considered, Dietz [8] showed that the analytical model yields accurate performance measures as compared with simulation. Restrictive assumptions on service distributions and on the aircraft arrival process are further relaxed in [9] by using the product form approximation (PFA).

The papers [8] and [9] constitute the primary references for the network we analyze in Section 4. However, our work differs from theirs in the following manner. While we share the goal of developing analytical models of airfield operations, our method relies on parametric decomposition and two-moment approximations, instead of MVA and PFA. One advantage of two-moment approximations is that it provides greater flexibility in modeling variability in service and traffic processes, thus allowing the study of more general networks. In addition, the recently developed approximations for fork and join stations allow us to model not only concurrent operations, as in [8] and [9], but also a command-and-control intervention for aircraft seeking to land. This mechanism, that we explain in detail in Section 2, provides the analyst a more realistic representation of airfield operations. Furthermore, we point out that our method allows one to perform computations as rapidly as with MVA and PFA.

Finally we note that Irish [11] and Irish et al. [12] employ an external analytical model to generate control variates, and develop a surrogate search model in which an analytical model is validated against a simulation model. It is quite possible that an adaptation of Irish’s approach could be used with advantage in combination with the approximation methodology suggested here.

1.3. Organization

The rest of the paper is organized as follows. Section 2 provides a description of the queueing model used to describe airfield operations. Section 3 provides an overview of the new analytical approach. Section 4 describes the solution approach and Section 5 presents results from numerical examples. These examples illustrate the accuracy of the proposed approach and also demonstrate the use of the approach for decision-making using an example application. Finally, Section 6 summarizes this research and its conclusions.
2. Airfield operations model

Fig. 1 depicts a typical queueing network model of airfield operations. Each aircraft landing at an airfield can be viewed as a customer entering a capacitated queueing network with population constraints. The capacity \( N \) of the network is simply the maximum number of aircraft that can simultaneously occupy the resources on the airfield.

Aircraft entering the airfield proceed through a network of processing stations (indexed by \( i = 1, \ldots, 8 \)) modeling activities such as landing, parking, cargo handling, refueling, liquid oxygen servicing, maintenance, taxi-out and takeoff (see Figs. 1 and 2). The activities at the processing stations require the use of constrained resources resulting in potential queueing delays. For example, the takeoff activity may involve some waiting time for the resource (the runway). Further, the processing times for the different service activities at the different stations could be characterized by random variables with distinct distributions. For instance, cargo handling time might have higher mean or variability than refueling time. Hence, the processing time at station \( i \) is modeled as a random variable, with mean \( \mu_i \) and variance \( \sigma_{i}^2 \). For our purpose in this paper we will instead refer to its rate \( \mu_i \) and squared coefficient of variation (SCV) of \( c^2_i \) (equal to \( \sigma_{i}^2 / \mu_{i}^2 \), its variance divided by the square of its mean).

In the queueing network model, the population constraint of the airfield is modeled using tokens. We assume that there are \( N \) tokens corresponding to the population constraint of the network, and to enter the network each aircraft requires a free token. The token stays with the aircraft during its journey through the different stations in the network and is released when the aircraft leaves the network. The released token then permits the entry of another aircraft into the airfield. Consequently, if an incoming aircraft finds the airfield operating at capacity (i.e., \( N \) aircraft are already receiving service, or equivalently no token is free), then the aircraft is subject to airborne delay (holding). It is granted permission to land only when one of the aircraft currently being serviced finally takes off. The synchronization of aircraft and tokens, and the associated delays prior to entry into the airfield, are modeled using join station \( J_0 \) in Figs. 1 and 2. Usually, airborne delays are expensive: not many aircraft can be held, and not for very long. As a result, the air
traffic controller will have to decide on the maximum number \( K \) of aircraft allowed to hold for landing authorization when \( N \) aircraft are already occupying the airfield. When \( K \) aircraft are holding, the air traffic controller must then divert from the airfield any additional incoming aircraft. Thus, from the point of view of the airfield, the aircraft arrival process shuts down when the number of aircraft waiting for landing authorization reaches \( K \). Subsequently, when ground space is again available and fewer than \( K \) aircraft are holding, the aircraft arrival process resumes.

Note that this model of command-and-control intervention for aircraft seeking to land provides a more realistic representation of airfield operations. It also avoids the subtle counter-intuitive behavior arising in the models of Dietz [8] and Willits and Dietz [9]. More precisely, because those models are based on MVA and PFA, the part of the network residing outside of the airfield is modeled as an artificial \( \cdot/M/1 \) queue, which generates the aircraft arrival process. When analysts need to study the effect of reduced capacity in the airfield, they would need to make the artificial \( \cdot/M/1 \) queue more congested. We feel that this modeling is counter-intuitive because airfield capacity is usually independent of aircraft arrival process. Use of tokens and two-moment approximations for synchronization stations allows the analyst to develop a more precise model for the aircraft arrival process.

Efficiency of airfield operations is typically defined using performance measures such as the throughput of the network, average ground time spent at the airfield by an aircraft, the average queue lengths at various processing stations, the station utilizations, and the average number of aircraft that are held before landing. One possible way of reducing aircraft ground time and minimizing queueing delays is to design airfield operations so that aircraft undergo activities concurrently (in parallel) instead of sequentially (in series). Fig. 2 depicts an airfield network where operations proceed concurrently. Although performing certain activities concurrently has potential benefits, one must respect the precedence constraints imposed on the different activities. For instance, in Fig. 2, although cargo loading and refueling occur concurrently, taxiing can begin only after both refueling and cargo loading are completed. In this queueing network formulation, fork and join stations are used to enforce these conditions. An entity arriving at a fork station in a queueing network can be viewed as generating temporary clones that are rejoined to a single entity at the corresponding join station when all activities along each clone path are complete. We describe this mechanism in more detail in Section 3.2.

Redesigning airfield operations often requires additional investment in expensive resources. Therefore it is important to quantify the potential benefits of alternative designs for airfield operations. In this paper we develop approaches to analyze queueing network models for a wide class of airfield operations, and demonstrate how the approach can be used to evaluate the tradeoffs in system performance with respect to the various designs.

3. Overview of parametric decomposition

Closed queueing networks with synchronization stations such as those described in the previous section belong to the class of non-product form networks. Such networks are hard to solve exactly and approximation methods must be used. In this research we use an approximation approach based on parametric decomposition. In the parametric decomposition approach, the traffic processes (arrival, service and departure) in the queueing network are approximated by renewal processes and estimates of performance measures at the individual stations in the network are characterized in terms of the mean and SCV of the inter-renewal times of those renewal processes. These characterizations are called two-moment approximations. Although parametric decomposition techniques have been used to analyze open queueing networks that do not have product form, they have not yet been used for the analysis of closed queueing networks with synchronization stations. Recent developments with respect to two-moment approximations for synchronization stations permit us to apply parametric decomposition approach to analyze the performance of a wide class of closed queueing networks. The approach consists of four main steps: (1) decomposition, (2) characterization, (3) linkage, and (4) solution. A description of each of these steps is given below.

(1) Decomposition

The closed queueing network is decomposed into its constituent stations. Figs. 1 and 2 show how the queueing model representation of airfield operations is composed of several kinds of stations: processing stations, fork stations, and join synchronization stations.
Each station is analyzed in isolation under the assumption that its arrival and service processes are renewal processes. Further it is assumed that the corresponding inter-renewal times are fully characterized by two parameters only: their common mean and SCV. Assuming that the parameters characterizing the arrival and service process at each station are known, two-moment approximations are used to characterize the departure process, the mean waiting time, and the mean queue length at the station. In particular, the departure process is approximated by a renewal process and the corresponding parameters describing the mean and the SCV of the inter-departure times are determined.

In reality, the arrival and departure processes in the network need not be renewal processes, and in particular successive inter-arrival or inter-departure times need not be independent. In this sense, the two-parameter characterization is an approximate representation of these traffic processes by “equivalent” renewal processes. The first parameter, corresponding to the mean, is equal to the mean of the corresponding inter-arrival or inter-departure times. The role of the second parameter, corresponding to the SCV of the equivalent renewal process, is to account for the complex structure of the traffic processes in an aggregate way.

Next, the stations in the closed queueing network are linked together using known relationships between the traffic processes at the different stations. For instance, the process of arrival at each station is formed by the superposition of the processes of departure from preceding stations. These relationships yield simple closed-form expressions for linking the mean and SCV of the different traffic processes. Finally, Little’s law (for which see [6]) is applied to ensure that the mean queue lengths sum up to \( N \), the capacity of the network. These expressions result in a set of nonlinear equations in the different parameters.

The system of nonlinear equations is solved using an iterative algorithm to obtain performance metrics such as the system throughput, mean waiting times and queue lengths at the different stations. The algorithm stops when successive iterations yield a solution of the nonlinear equations. This solution is then taken to be an approximate solution of the closed queueing network.

The decomposition step described above is straightforward. We discuss steps 2 through 4 in the following subsections.

3.1. Characterization of a processing station

Fig. 3 considers in isolation a processing station \( S \) from the airfield network, as well as the parameters of the traffic process at the station. Station \( S \) represents one of the processing stations in the airfield network (stations \( i = 1, \ldots, 8 \) in Figs. 1 and 2). \( SN \) denotes the rest of the closed queueing network to which station \( S \) belongs. Station \( S \) is assumed to be composed of one fixed rate server. Subsequent to processing at station \( S \), the entity is routed to subnetwork \( SN \). In \( SN \) the entity could be subject to random delays before it revisits \( S \). As there is a finite population \( N \) of entities in the network, the process of arrival at station \( S \) shuts down when there are \( N \) entities at the station. Although the arrival and service processes at station \( S \) could have general characteristics, in the parametric decomposition approach we approximate these traffic processes by renewal processes.
Table 1
Inputs for the characterization of a processing station S

\( \mu \) = service rate
\( c_s^2 \) = SCV of service process
\( \lambda \) = arrival rate when the arrival process is not shut down
\( c_a^2 \) = SCV of inter-arrival times
\( N \) = finite population of the closed queueing network containing station S

Table 2
Outputs and equations for the characterization of a processing station S

\( \lambda_d \) = departure rate
\( c_d^2 \) = SCV of inter-departure times
\( L \) = mean queue length at station S

Tables 1 and 2 summarize the input and output parameters relative to Fig. 3. We assume that \( \lambda^{-1} \) and \( c_a^2 \) denote the mean and SCV of the inter-arrival times respectively, and that \( \mu^{-1} \) and \( c_s^2 \) denote the mean and SCV of the service times. With these assumptions, our model of the processing station is completely characterized by the parameter 5-tuple \( (\lambda, c_a^2, \mu, c_s^2, N) \). Assuming that this information about the arrival and service processes is known, characterization of the processing station S involves deriving two-moment approximations for the mean and SCV of the inter-departure times, \( \lambda_d^{-1} \) and \( c_d^2 \) respectively, and the mean queue length \( L \).

Characterization of a processing station with the above characteristics has been reported in [5] and [6]. We report the final expressions here for the sake of completeness.

Call \( \rho \) the utilization of the station S. By definition,

\( \rho = \lambda / \mu. \)

Flow conservation yields

\( \lambda_d = \lambda. \) \hspace{2cm} (1)

The SCV \( c_d^2 \) of the inter-departure times is given by

\( c_d^2 = (1 - \rho^2) c_a^2 + \rho^2 c_s^2, \) \hspace{2cm} (2)

and the expression for the mean queue length \( L \) is

\( L = \rho + C \left( \frac{c_a^2 + c_s^2}{2} \right) \left( \frac{\rho^2}{1 - \rho} \right). \) \hspace{2cm} (3)

In Eq. (3), \( C \) is a correction factor applied to account for the fact that station S is part of a closed queueing network. Kamath et al. [13] derived the expression

\( C = \left( \frac{N - 1}{N} \right) \left( \frac{1 - \rho}{1 - \rho + \rho/N} \right) \)

for analyzing cyclic closed queueing networks.

Next we give the two-moment approximation equations for fork and join stations.

3.2. Characterization of fork and join stations

Fig. 4 shows a typical fork station \( F \) and join station \( J \) obtained from decomposition. The join station \( J \) has two input buffers \( B_i \) indexed by \( i = 1, 2 \). \( SN_i \) denotes the rest of the queueing network for entities that arrive at buffer \( B_i \). Departures from \( J \) occur only when there are entities in both buffers. As soon as there is at least one entity in each buffer, one entity is removed from each buffer and joined together. The joined entity immediately departs from \( J \) and the contents of each input buffer are reduced by one. Subsequent to departure, the joined entity visits subnetwork \( SN_3 \) where it is subjected to random delays, before visiting the fork station \( F \) where it is forked back into two original
Table 3
Input parameters for characterization of join station

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>population of entities arriving at buffer $B_1$</td>
</tr>
<tr>
<td>$\lambda_1$</td>
<td>rate of arrivals at buffer $B_1$ when the arrival process is not shut down</td>
</tr>
<tr>
<td>$c_1^2$</td>
<td>SCV of the inter-arrival times for $B_1$</td>
</tr>
<tr>
<td>$K_2$</td>
<td>population of entities arriving at buffer $B_2$</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>rate of arrival at buffer $B_2$ when the arrival process is not shut down</td>
</tr>
<tr>
<td>$c_2^2$</td>
<td>SCV of the inter-arrival times for $B_2$</td>
</tr>
</tbody>
</table>

Table 4
Output parameters for characterization of join station

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_D$</td>
<td>departure rate</td>
<td>Eqs. (6) and (9)</td>
</tr>
<tr>
<td>$c_D^2$</td>
<td>SCV of inter-departure times</td>
<td>Eq. (12)</td>
</tr>
<tr>
<td>$L_1$</td>
<td>mean queue length at buffer $B_1$</td>
<td>Eqs. (7) and (10)</td>
</tr>
<tr>
<td>$L_2$</td>
<td>mean queue length at buffer $B_2$</td>
<td>Eqs. (8) and (11)</td>
</tr>
</tbody>
</table>

entities. The two entities are routed back to $SN_i$, $i = 1, 2$ respectively, where they are subjected to random delays before revisiting $J$.

Moreover the process of arrival at buffer $B_i$ shuts down when there are $K_i$ units in buffer $B_i$. Call $L_i$ the mean queue length at buffer $B_i$ ($i = 1, 2$), and $L_{S,i}$ the mean queue length in subnetwork $SN_i$ ($i = 1, \ldots, 3$). Since there is a finite population of size $K_i$ for each entity $i$, we must have

$$K_i = L_{S,3} + L_{S,i} + L_i, \quad i = 1, 2.$$  \hspace{1cm} (4)

Since the subnetworks $SN_i$ can have different configurations resulting in arbitrary delays, the processes of arrival at the join station can have general characteristics. However, we assume that the processes of arrival at the buffers $B_i$, $i = 1, 2$, are independent renewal processes, and we denote the mean and SCV of the inter-arrival times for buffer $B_i$ by $\lambda_i^{-1}$ and $c_i^2$. Since we assume that the process of arrival at buffer $B_i$ shut down once it has $K_i$ units, the arrival process is renewal between shutdowns. With these assumptions, our model of this fork station and join station is completely characterized by the parameter 6-tuple $(\lambda_1, c_1^2, K_1, \lambda_2, c_2^2, K_2)$. Assuming that this information about the arrival and service processes is known, characterization of the fork station and join station involves deriving approximations for the mean ($\lambda_D^{-1}$) and the SCV $c_D^2$ of the inter-departure times, and the mean queue lengths $L_i$, $i = 1, 2$. Tables 3 and 4 summarize the inputs and outputs of the analysis.

Characterization equations for fork and join stations were developed in [14]. We report below the results derived in those works for the sake of completeness. For notational simplicity, we define

$$r = \lambda_1/\lambda_2, \quad c^2 = (c_1^2 + c_2^2)/2.$$  \hspace{1cm} (5)

We distinguish two cases.
• Case 1. If \( r \neq 1 \),
\[
\lambda_D = \lambda_1 \left( \frac{1 - r^{K_1+K_2}}{1 - r^{K_1+K_2+1}} \right) \left[ 1 - 0.5(c^2 - 1) \frac{(1 - r)r^{K_1+K_2}}{1 - r^{2(K_1+K_2)+1}} \right].
\]
(6)

\[
L_1 = \frac{1}{1 - r^{-(K_1+K_2+1)}} \left[ K_1 - \frac{1 - r^{-K_1}}{r - 1} \right] \left[ 1 + \left( \frac{1 - r}{1 + r} \right) \left( \frac{r^4}{1 + r^8} \right) (c^2 - 1) \right],
\]
(7)

\[
L_2 = \frac{1}{1 - r^{-(K_1+K_2+1)}} \left[ K_2 - r \frac{1 - r^{K_2}}{1 - r} \right] \left[ 1 - \left( \frac{1 - r}{1 + r} \right) \left( \frac{r^4}{1 + r^8} \right) (c^2 - 1) \right].
\]
(8)

• Case 2. If \( r = 1 \),
\[
\lambda_D = \lambda_1 \left[ \frac{K_1 + K_2}{K_1 + K_2 + 1} \right] \left[ 1 - \frac{0.5(c^2 - 1)}{2(K_1 + K_2 + 1)} \right],
\]
(9)

\[
L_1 = \frac{K_1(K_1 + 1)}{2(K_1 + K_2 + 1)},
\]
(10)

\[
L_2 = \frac{K_2(K_2 + 1)}{2(K_1 + K_2 + 1)}.
\]
(11)

Further, in [15] and [16], an expression for the SCV of the departure process \( (c_D^2) \) is derived:
\[
c_D^2 = \left( \frac{\lambda_2^5}{\lambda_1^2 + \lambda_2^2} \right) c_i^2 + \left( \frac{\lambda_1^5}{\lambda_1^2 + \lambda_2^2} \right) c_i^2 \left[ 1 - \frac{1}{K_1 + K_2 + 1} - \frac{1}{(K_1 + K_2 + 1)^2} \right] \left[ \frac{\lambda_1^5 + \lambda_2^5}{\lambda_1^2 + \lambda_2^2} \right].
\]
(12)

We see from Eqs. (6) and (9) that the rate out of the join station is always less than the minimum of the input arrival rates for all finite values of \( K_1 \) and \( K_2 \). Similarly, we notice from Eq. (12) that the SCV of the process of departure from a join station is always less than the maximum SCV of the two input arrival processes. In this sense, the join station brings robustness to the network. Finally, we observe that higher values in the input SCV parameters lead to increased queue lengths and synchronization delays.

The fork station \( F \) has zero service time and simply splits an arrival stream into identical streams. Let \( \lambda_{a,i}^{-1} \) and \( c_a^2 \) denote the mean and the SCV of the arrival process at \( F \) and let \( \lambda_{a,i}^{-1} \) and \( c_{a,i}^2 \) for \( i = 1, 2 \) denote the mean and the SCV of the departure process for the two streams leaving \( F \). Then the fork mechanism implies that
\[
\lambda_a = \lambda_{a,1} = \lambda_{a,2},
\]
(13)
\[
c_a^2 = c_{a,1}^2 = c_{a,2}^2.
\]
(14)

Finally, if we want to model a clone path situation as described in Section 2 then we must enforce the constraint
\[
L_{S,1} + L_1 = L_{S,2} + L_2;
\]
(15)
that is, the number of forked entities must be the same (in addition to Eq. (4), which must also remain true).

4. Solution algorithm for an airfield operations network

4.1. Airfield queueing network model without concurrent operation

Fig. 1 represents the queueing network of an airfield where operations are performed sequentially. The inputs for the analysis include the mean \( \mu_i^{-1} \) and SCV \( c_i^2 \) characterizing the service at each station \( i \), for \( i = 1, \ldots, 8 \), the mean \( \lambda_{0,1}^{-1} \) and variability parameter \( c_{0,1}^2 \) characterizing the process of arrival of aircraft at queue \( B_{0,2} \) at synchronization station \( J_0 \), and the maximum number \( K_{0,2} \) of aircraft that can be permitted to hold. The outputs of interest are the expected steady-state throughput of the airfield network given by
\[
\lambda_{D,0} := \{\lambda_{d,i}, \ i = 1, \ldots, 8\},
\]
the mean queue lengths at each processing station and at each input queue of station \( J_0 \), the utilization of each station and the mean time spent by an aircraft in the airfield for service. The characterization equations for the stations in the
network (processing, fork, and join stations) were obtained assuming that the parameters of the processes of arrival at the different stations were known. In general the parameters characterizing the different processes of arrival at the different stations in a network are not known. However, as the stations are part of a cyclic closed queueing network, the process of departure from a station contributes to the processes of arrival at the following stations in the network. Using this fact and flow conservation principles, we derive specific equations linking the traffic process parameters at the different stations of the network and we then use the characterization equations to solve for the various performance measures of interest. For the network shown in Fig. 1, we obtain the following set of linking equations.

- Equations linking station $J_0$ to station 1:
  \[
  \lambda_{d,1} = \lambda_{D,0}, \quad c_{a,1}^2 = c_{D,0}^2; \tag{16}
  \]

- Equations linking station $i$ to station $i+1$, for $i = 1, \ldots, 7$:
  \[
  \lambda_{d,i+1} = \lambda_{d,i}, \quad c_{a,i+1}^2 = c_{a,i}^2; \tag{17}
  \]

- Equations linking station 8 to station $J_0$:
  \[
  \lambda_{0,1} = \lambda_{d,8}/(1 - \pi_{0,1}), \quad c_{0,1}^2 = \frac{c_{d,8}^2}{(1 - \pi_{0,1})^2} - \frac{c_{0,2}^2}{(1 - \pi_{0,1})^2} \left( \frac{\lambda_{d,8}}{\lambda_{0,2}} \right) \left( \frac{2\pi_{0,1}}{1 + c_{0,2}^2} \right). \tag{18}
  \]

In Eqs. (20) and (21), $\pi_{0,1}$ denotes the probability that the process of arrival at queue $B_{0,1}$ shuts down. This probability is determined using flow conservation, which implies that the throughput $\lambda_{D,0}$ from the join station $J_0$ characterized by the 6-tuple ($\lambda_{0,1}, c_{0,1}^2, N, \lambda_{0,2}, c_{0,2}^2, K_{0,2}$) should equal $\lambda_{d,8}$. Algorithm 1 describes a bisection search algorithm that helps to determine $\pi_{0,1}, \lambda_{0,1},$ and $c_{0,1}^2$.

**Algorithm 1** Bisection search for $\pi_{0,1}$

1. **Given**: $c_{d,8}^2, \lambda_{d,8}, K, \lambda_{0,2}, c_{0,2}^2,$ and $K_{0,2}$.
2. **Initialize**: $LOW = 0, \quad HIGH = 1, \quad \epsilon = 0.001$
3. **for** $j = 1, \ldots, 10$ **do**
4.   **Set** $\pi_{0,1}^{(j)} = 0.5(LOW + HIGH)$
5.   **Compute** $\lambda_{0,1}$ by setting $\pi_{0,1} = \pi_{0,1}^{(j)}$ in equation 20
6.   **Compute** $c_{0,1}^2$ using equation 21
7.   **Compute** $\lambda_{D,0}$ using equations 6 and 9, and set $\lambda_{D,0}^{(j)} = \lambda_{D,0}$
8.   **Compute** $\delta = \lambda_{d,8} - \lambda_{D,0}^{(j)}$
9.   **if** $\delta < -\epsilon$ **then**
   10.   **Set** $LOW = \pi_{0,1}^{(j)}$
11.   **else if** $\delta > \epsilon$ **then**
   12.   **Set** $HIGH = \pi_{0,1}^{(j)}$
13.   **else if** $|\delta| \leq \epsilon$ **then**
   14.       **Stop**
15.   **end if**
16. **end for**

The linking equations above yield closed-form expressions that relate the mean and SCV parameters characterizing the performance measures at the different stations in the network. Finally, in the solution step we apply Little’s law to the entire network to ensure that the mean queue lengths of the different stations in the network add up to the network capacity $N$. This results in a set of nonlinear equations in the set of unknown parameters defining the traffic processes in the closed queueing network. This set of equations is solved for the unknowns using an iterative algorithm.
The algorithm is initialized with an estimate of the parameters characterizing the process of departure from one of the stations: e.g., if station $J_{0,1}$ is chosen, the algorithm begins with an initial estimate of $(\lambda_{D,0}, c_{D,0}^2)$. Starting with this initial estimate, the iterative procedure progressively updates the estimates of the traffic process parameters until they converge and are consistent with the input parameters. The iterative procedure uses a combination of bisection scheme and fixed point iteration. While we do not yet have a formal proof of the convergence of the algorithm in this context, in all the numerical experiments conducted (see Section 5 for details) the algorithm converged. To improve the convergence guarantees, a bisection scheme is used wherever bounds on the unknown parameters are available, i.e. to compute the unknown throughput and probabilities. For the SCV, proven bounds are not available and hence fixed point iteration is used. However, in all the numerical experiments conducted, the fixed point iteration scheme converged very quickly. Finally the iterative procedure is described in Algorithm 2.

Algorithm 2 Solution algorithm for network without concurrent operation

1: Input: $\mu_{s,i}, c_{s,i}^2, i = 1, \ldots, 8, \lambda_{0,2}, c_{0,2}^2, K_{0,2}, N$
2: Initialize: $LOW = 0, HIGH = \min(\mu_{s,i}, i = 1, \ldots, 8), \epsilon = 0.001$
3: Begin Loop 1
4: for $j = 1, \ldots, 10$
5: $\lambda_{D,0}^{(j)} = 0.5(LOW + HIGH)$ and $c_{D,0}^2 = 1$
6: Begin Loop 2
7: while $|\delta_2| > \epsilon$
8: For Station $i = 1$
9: Compute $\lambda_{a,i}$ and $c_{a,i}^2$ using linking equations 16 and 17
10: Compute $\lambda_{d,i}, c_{d,i}^2$ and $L_{S,i}$ using characterization equations 1, 2, and 3
11: For Station $i = 2, \ldots, 8$
12: Compute $\lambda_{a,i}$ and $c_{a,i}^2$ using linking equations 18 and 19
13: Compute $\lambda_{d,i}, c_{d,i}^2$ and $L_{S,i}$ using characterization equations 1, 2, and 3
14: For Station $J_0$
15: Solve for $\pi_{0,1}, \lambda_{0,1}$, and $c_{0,1}^2$ using Algorithm 1 and current estimates of $\lambda_{d,8}$ and $c_{d,8}^2$
16: Compute $\lambda_{D,0}, c_{D,0}^2, L_{0,1}$, and $L_{0,2}$ using characterization equations 6 to 12
17: Compute $\delta_2 = |c_{D,0}^2 - c_{D,0}^{(j)}|$. Set $c_{D,0}^{(j)} = c_{D,0}^2$
18: end while
19: End Loop 2
20: Compute $\delta_1 = \sum_{s=1}^{8} L_s + L_{0,1} - N$
21: if $\delta_1 < -\epsilon$ then
22: Set $LOW = \lambda_{D,0}^{(j)}$
23: else if $\delta_1 > \epsilon$ then
24: Set $HIGH = \lambda_{D,0}^{(j)}$
25: else if $|\delta_1| \leq \epsilon$ then
26: Stop
27: end if
28: end for
29: End Loop 1

4.2. Airfield queueing network model with concurrent operations

Fig. 2 represents the queueing network of an airfield where operations are performed concurrently. As in the case of the network represented in Fig. 1, the inputs for the analysis include the mean $\mu_{i}^{-1}$ and SCV $c_{s,i}^2$ characterizing the service at each station $i$ for $i = 1, \ldots, 8$, the mean $\lambda_{0,2}^{-1}$ and SCV $c_{0,2}^2$ characterizing the process of arrival of aircraft at queue $B_{0,2}$ at synchronization station $J_0$, and the maximum number $K_{0,2}$ of aircraft that can be permitted to be on
hold. As before, the outputs of interest are the throughput of the airfield network given by
\[ \lambda_{D,0} = \{\lambda_{d,i}, \ i = 1, \ldots, 8\}, \]
the mean queue lengths at each processing station and at each input queue of station \( J_0 \), the utilization of each station, and the mean time spent by an aircraft in the airfield for service. We obtain these performance measures by analyzing the network using parametric decomposition equations.

One of the key distinguishing features of the network shown in Fig. 2 is the use of fork and join stations, and the cloning of entities along paths, to model concurrent operations. The clone entities traverse the network along clone paths \( A_1 \) and \( A_2 \) after fork station \( F_1 \), \( A_1 \) consists of station 4 (LOX), 5 (Fuel) and input queue \( B_{1,1} \) of join station \( J_1 \), while \( A_2 \) consists of station 3 (Cargo), 6 (Maintenance) and input queue \( B_{1,2} \) of join station \( J_1 \). Clone entities traversing the network along these paths possess the same attributes as the original entity. For example, the mean service time of a clone at a station equals the mean service time the original entity would require at this station. As soon as there is at least one clone in each queue \( B_{1,1} \) and \( B_{1,2} \), a clone entity is removed from each queue. The original entity is then considered rejoined and departs from the synchronization station \( J_1 \). Using this fact and flow conservation principles, we derive specific equations linking the traffic process parameters at the different stations of the network and then solve for the various performance measures of interest. For the network shown in Fig. 2, we obtain the following set of linking equations.

- **Equations linking station \( J_0 \) to station 1:**
  \[ \lambda_{a,1} = \lambda_{D,0}, \]
  \[ c_{a,1}^2 = c_{D,0}^2; \] (22)

- **Equations linking station 1 to station 2:**
  \[ \lambda_{a,2} = \lambda_{d,1}, \]
  \[ c_{a,2}^2 = c_{d,1}^2; \] (23)

- **Equations linking arrival and departure processes at fork station \( F_1 \):**
  \[ \lambda_{a,2} = \lambda_{a,3} = \lambda_{a,4}, \]
  \[ c_{a,2}^2 = c_{a,3}^2 = c_{a,4}^2; \] (24)

- **Equations linking station 4 to station 5:**
  \[ \lambda_{a,5} = \lambda_{d,4}, \]
  \[ c_{a,5}^2 = c_{d,4}^2; \] (25)

- **Equations linking station 3 to station 6:**
  \[ \lambda_{a,6} = \lambda_{d,3}, \]
  \[ c_{a,6}^2 = c_{d,3}^2; \] (26)

- **Equations linking stations 5 and 6 to station \( J_1 \):**
  \[ \lambda_{1,1} = \frac{\lambda_{d,5}}{1 - \pi_{1,1}}, \]
  \[ \lambda_{1,2} = \frac{\lambda_{d,6}}{1 - \pi_{1,2}}, \] (27)
  \[ c_{1,1}^2 = \frac{c_{d,5}^2}{(1 - \pi_{1,1})^2} - \frac{c_{1,2}^2}{(1 - \pi_{1,1})^2} \left( \frac{\lambda_{d,5}}{\lambda_{1,2}} \right) \left( \frac{2\pi_{1,1}}{1 + c_{1,2}^2} \right), \] (28)
  \[ c_{1,2}^2 = \frac{c_{d,6}^2}{(1 - \pi_{1,2})^2} - \frac{c_{1,1}^2}{(1 - \pi_{1,2})^2} \left( \frac{\lambda_{d,6}}{\lambda_{1,1}} \right) \left( \frac{2\pi_{1,2}}{1 + c_{1,1}^2} \right). \] (29)
In Eqs. (32)–(35), \( \pi_{1,1} \) (\( \pi_{1,2} \)) denotes the probability that the process of arrival at queue \( B_{1,1} \) (\( B_{1,2} \)) shuts down. These probabilities are determined using the following constraints. First, flow conservation implies that the throughput \( \lambda_{D,1} \) should equal \( \lambda_{d,5} (= \lambda_{d,6}) \). To this end we characterize the join station \( J_1 \) with the 6-tuple \((\lambda_{1,1}, c_{1,1}^2, N, \lambda_{1,2}, c_{1,2}^2, N)\). Second, there must be an equal number of customers in each of the clone paths, i.e., if \( L_i, i = 3, \ldots, 6 \), denotes the mean number of customers at station \( i \), and \( L_{1,1} \) and \( L_{1,2} \) denote the mean numbers of customers at buffers \( B_{1,1} \) and \( B_{1,2} \) respectively, then

\[
L_3 + L_6 + L_{1,2} = L_4 + L_5 + L_{1,1}.
\]

Algorithm 3 describes a bisection search algorithm that helps to determine \( \pi_{1,j}, \lambda_{1,j}, \) and \( c_{1,j}^2 \), for \( j = 1, 2 \).

**Algorithm 3 Bisection search for \( \pi_{1,1} \) and \( \pi_{1,2} \)**

1. **Given:** \( c_{d,5}^2, c_{d,6}^2, \lambda_{d,5} = \lambda_{d,6}, N, L_i, i = 3, \ldots, 6 \)
2. **Initialize:** \( LOW1 = 0, HIGH1 = 1, LOW2 = 0, HIGH2 = 1, \epsilon = 0.001 \)
3. **Begin Loop 1**
4. **for** \( j = 1, \ldots, 10 \) **do**
5. Set \( \pi_{1,2}^{(j)} = 0.5(LOW2 + HIGH2) \)
6. **Begin Loop 2**
7. **for** \( k = 1, \ldots, 10 \) **do**
8. Set \( \pi_{1,1}^{(k)} = 0.5(LOW1 + HIGH1) \)
9. Compute \( \lambda_{1,1}, \lambda_{1,2}, c_{1,1}^2, \) and \( c_{1,2}^2 \) with equations 32 to 35
10. Compute \( \lambda_{D,1} \) using equations 6 and 9, and set \( \lambda_{D,1} = \lambda_{D,1} \)
11. Compute \( \delta_1 = \lambda_{d,5} - \lambda_{D,1} \)
12. **if** \( \delta_1 < -\epsilon \) **then**
13. Set \( LOW1 = \pi_{1,1}^{(k)} \)
14. **else if** \( \delta_1 > \epsilon \) **then**
15. Set \( HIGH1 = \pi_{1,1}^{(k)} \)
16. **else if** \( |\delta_1| \leq \epsilon \) **then**
17. Stop
18. **end if**
19. **end for**
20. **End Loop 2**
21. Compute \( L_{1,1} \) and \( L_{1,2} \) using equations 7, 10, 8 and 11
22. Compute \( \delta_2 = (L_4 + L_5 + L_{1,1}) - (L_3 + L_6 + L_{1,2}) \)
23. **if** \( \delta_2 < -\epsilon \) **then**
24. Set \( LOW2 = \pi_{1,2}^{(j)} \)
25. **else if** \( \delta_2 > \epsilon \) **then**
26. Set \( HIGH2 = \pi_{1,2}^{(j)} \)
27. **else if** \( |\delta_2| \leq \epsilon \) **then**
28. Stop
29. **end if**
30. **end for**
31. **End Loop 1**

- Equations linking station \( J_1 \) to station 7:
  \[
  \lambda_{a,7} = \lambda_{D,1}, \quad \lambda_{d,7} = \lambda_{D,1},
  \]
  \[
  c_{a,7}^2 = c_{D,1}^2; \quad c_{d,7}^2 = c_{D,1}^2;
  \]

- Equations linking station 7 to station 8:
  \[
  \lambda_{a,8} = \lambda_{d,7}, \quad \lambda_{d,8} = \lambda_{d,7},
  \]
  \[
  c_{a,8}^2 = c_{d,7}^2; \quad c_{d,8}^2 = c_{d,7}^2;
  \]

(37) Through (40)
• Equations linking station 8 to station $J_0$:

$$\lambda_{0,1} = \frac{\lambda_{d,8}}{1 - \pi_{0,1}},$$

(41)

$$c_{0,1}^2 = \frac{c_{d,8}^2}{(1 - \pi_{0,1})^2} - \frac{c_{0,2}^2}{(1 - \pi_{0,1})^2} \left( \frac{\lambda_{d,8}}{\lambda_{0,2}} \right) \left( \frac{2\pi_{0,1}}{1 + c_{0,2}^2} \right).$$

(42)

In Eqs. (41) and (42), $\pi_{0,1}$ denotes the probability that the arrivals at queue $B_{0,1}$ shut down. This probability is determined using Algorithm 1.

The linking equations above yield closed-form expressions that relate the means and SCVs of the different stations in the network. Finally, in the solution step we apply Little’s law to the entire network to ensure that the mean queue lengths of the different stations in the network add up to the network capacity $N$. This results in a set of nonlinear equations in the set of unknown parameters defining the traffic processes in the closed queueing network. The solution step solves this set of equations for the unknowns using an iterative algorithm. The algorithm is initialized with an estimate of the parameters characterizing the process of departure from one of the stations: i.e., if station $J_{0,1}$ is chosen, the algorithm begins with an initial estimate of $(\lambda_{d,0}, c_{0,2}^2, D_{0})$. Starting with this initial estimate, the iterative procedure progressively updates the estimates of the traffic process parameters until they converge and are consistent with the input parameters. It is described in Algorithm 4.

5. Numerical results and example application

This section summarizes the results of experiments performed to test the numerical accuracy of the algorithms proposed in the previous sections. Subsequently, we demonstrate how the analytical method can be used to derive managerial insights and make decisions to improve airfield operations.

5.1. Numerical results

To test the numerical accuracy of the analytical method a three-level factorial design of experiments was performed. The factors that were varied were (i) the capacity of the network, $N$, (ii) the SCVs of the service times of the various operations in the airfield, $c_{s,i}^2$, $i = 1, \ldots, 8$, and (iii) external rate of arrival of aircraft at the network, $\lambda_{0,2}$.

For the capacity of the network, the values chosen were $N = 4, 8, 12$, corresponding to low, medium, and high levels of network capacity. In all the experiments, the mean service rates at the different stations in the airfield network were kept constant. Specifically, the landing and takeoff rates, $\mu_{s,i}$, $i = 1, 8$, were set equal to 30 per hour, the parking and taxiing rates, $\mu_{s,i}$, $i = 2, 7$, were set equal to 8 per hour, and the mean processing rates of cargo loading, refuelling, liquid oxygen servicing, and maintenance, $\mu_{s,i}$, $i = 3, \ldots, 6$, were set equal to 1 per hour. These values might not correspond to actual data at any particular airfield, but were chosen as reasonable estimates for the purpose of experimentation.

To study the impact of variability of operation times on airfield performance, three values of the SCVs of the service times at each station were considered, namely $c_{s,i}^2 = 0.7, 1.0, 2.0$, for $i = 1, \ldots, 8$, corresponding to low, medium, and high values of variability. We use the notation $c_{s}^2 = c_{s,i}^2$, $i = 1, \ldots, 8$, to account for the fact that we vary the service SCVs of all the stations at the same time.

The effect of external loads on performance was studied by considering three values of external arrival rate of aircraft, namely, $\lambda_{0,2} = 0.8, 0.9$, and 1.0 corresponding to low, medium, and heavy loads. In all the experiments it was assumed that the air traffic control would not permit more than five aircraft to wait for permission to land at the airfield at any time, i.e., $K_{0,2} = 5$. Also, the SCV of the inter-arrival times of aircraft for the network, $c_{0,2}^2$, was set at 1.0 in all the experiments. These sets of parameters yield 27 experiments for each of the two networks represented in Figs. 1 and 2. Numerical values for those design parameters are summarized in Table 5.

The algorithms used in the analytical method were implemented on a Pentium III PC and solved using MATLAB, a commercial software package for mathematical computation [17]. The performance measures computed were throughput together with the utilization and mean queue length at each station in the network. The numerical accuracy of the results obtained from the analytical method was compared against estimates obtained from simulation experiments conducted using ProModel, a commercial software package for discrete-event simulation [18]. For each
Algorithm 4 Solution algorithm for a network with concurrent operations

1: **Input:** $\mu_{s,i}^2, c_{s,i}^2, i = 1, \ldots, 8, \lambda_{0,2}, c_{0,2}^2, K_{0,2}, N$

2: **Initialize:** $LOW = 0$, $HIGH = \min(\mu_{s,i}, i = 1, \ldots, 8), \epsilon = 0.001$

3: **Begin Loop 1**

4: for $j = 1, \ldots, 10$ do

5: \[ \lambda_{D,0}^{(j)} = 0.5(LOW + HIGH) \] and $c_{D,0}^2(j) = 1$

6: **Begin Loop 2**

7: while $|\delta_2| > \epsilon$ do

8: **For Station** $i = 1$

9: Compute $\lambda_{a,i}$ and $c_{a,i}^2$ using linking equations 22 and 23

10: Compute $\lambda_{d,i}, c_{d,i}^2$ and $L_{S,i}$ using characterization equations 1, 2 and 3

11: **For Station** $i = 2, \ldots, 6$

12: Compute $\lambda_{a,i}$ and $c_{a,i}^2$ using linking equations 24 to 31

13: Compute $\lambda_{d,i}, c_{d,i}^2$ and $L_{S,i}$ using characterization equations 1, 2 and 3

14: **For Station** $J_1$

15: Solve for $\pi_{1,1}, \pi_{1,2}, \lambda_{1,1}, \lambda_{1,2}, c_{1,1}^2$, and $c_{1,2}^2$ using Algorithm 3 and current estimates of $\lambda_{d,5}, c_{d,5}^2, \lambda_{d,6}$, and $c_{d,6}^2$

16: Compute $\lambda_{D,1}, c_{D,1}^2, L_{1,1}$, and $L_{1,2}$ using characterization equations 6 to 12

17: **For Station** $i = 7, 8$

18: Compute $\lambda_{a,i}$ and $c_{a,i}^2$ using linking equations 39 and 40

19: Compute $\lambda_{d,i}, c_{d,i}^2$ and $L_{S,i}$ using characterization equations 1, 2 and 3

20: **For Station** $J_0$

21: Solve for $\pi_{0,1}, \lambda_{0,1}$, and $c_{0,1}^2$ using Algorithm 1 and current estimates of $\lambda_{d,8}, c_{d,8}^2$

22: Compute $\lambda_{D,0}, c_{D,0}^2, L_{0,1}$, and $L_{0,2}$ using characterization equations 6 to 12

23: Compute $\delta_2 = |c_{D,0}^2 - c_{D,0}^2(j)|$. Set $c_{D,0}^2(j) = c_{D,0}^2$

24: **end while**

25: **End Loop 2**

26: Compute $\delta_1 = (L_1 + L_2) + (L_4 + L_5 + L_{1,1}) + (L_7 + L_8 + L_{0,1}) - N$

27: if $\delta_1 < -\epsilon$ then

28: Set $LOW = \lambda_{D,0}^{(j)}$

29: else if $\delta_1 > \epsilon$ then

30: Set $HIGH = \lambda_{D,0}^{(j)}$

31: else if $|\delta_1| \leq \epsilon$ then

32: Stop

33: **end if**

34: **end for**

35: **End Loop 1**

---

Table 5

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{0,2}$</td>
<td>5</td>
</tr>
<tr>
<td>$\lambda_{0,2}$</td>
<td>0.8, 0.9, 1.0</td>
</tr>
<tr>
<td>$c_{0,2}^2$</td>
<td>1.0</td>
</tr>
<tr>
<td>$N$</td>
<td>4, 8, 12</td>
</tr>
<tr>
<td>$\mu_{s,1}, \mu_{s,8}$</td>
<td>30</td>
</tr>
<tr>
<td>$\mu_{s,2}, \mu_{s,7}$</td>
<td>8</td>
</tr>
<tr>
<td>$\mu_{s,i}, i = 3, \ldots, 6$</td>
<td>1</td>
</tr>
<tr>
<td>$c_{s}^2(= c_{s,i}^2, i = 1, \ldots, 8)$</td>
<td>0.7, 1.0, 2.0</td>
</tr>
</tbody>
</table>
parameter setting, the simulation results were recorded from five independent runs, each of which corresponded to 100,000 departures from the airfield. Of these, data corresponding to the initial 10,000 departures were discarded to account for transient start-up effects. For each experimental observation, the 95% confidence intervals were also recorded. Tables 6–11 summarize the results from these experiments. The computational time required by the analytical method to obtain these estimates is low (around 15 s on average and always less than 1 min) when compared to that needed for the simulation runs (from 10 to 15 min).

Table 6 presents sample output comparing analytical and simulation results for one of the 27 experiments for a network with no concurrent operations.

Table 7 presents the corresponding results for an airfield with concurrent operation. The tables report the percentage error in analytical estimates as compared to simulation. For the station utilization, the percentage error is computed as

\[
\% \text{ Error (Utilization)} = 100 \times \frac{\text{Analytical} - \text{Simulation}}{\text{Simulation}}.
\]

The percentage error in mean queue length was computed as a percentage of the capacity of the network \(N\) to avoid potential problems that might arise when the mean queue length itself is small. That is,

\[
\% \text{ Error (Queue)} = 100 \times \frac{\text{Analytical} - \text{Simulation}}{N}.
\]

Tables 8–11 use the following abbreviations to reduce the sizes of the headings:

- \(\text{Q}(\text{Cargo})\) for Queue at Cargo,
- \(\text{U}(\text{Cargo})\) for Utilization at Cargo,
- \(\text{T}(\text{Ground})\) for Time on Ground,
- \(\text{Q}(B_{0,2})\) for Queue at Buffer \(B_{0,2}\),
- \((\text{S})\) for simulation method,
- \((\text{A})\) for analytical method.

Table 8 compares the output from the simulation model with the output from the analytical model, for some key performance measures and for all 27 experiments in a network without concurrent operations. Table 9 presents a
similar comparison for the experiments conducted on an airfield with concurrent operations. Tables 8 and 9 show that in most cases the estimates from the analytical model provide a lower bound for the estimate from the simulation model.

The results in Tables 6–9 indicate that the percentage errors in estimates of performance measures are reasonable: less than 5% for throughput, less than 9% (6%) for mean queue lengths (utilization) at the bottleneck station. This accuracy in performance estimates seems very encouraging, especially when considered in the context of the wide range of networks that the algorithm is capable of analyzing. This accuracy, coupled with the short execution times for the analytical method, makes this method an attractive tool for understanding performance tradeoffs, conducting what-if analyses, and identifying opportunities to improve network performance. Next we demonstrate the use of the analytical method to make decisions that would help improve airfield performance.

5.2. Example application

Consider an airfield network with a capacity of $N = 12$, wherein the operations have mean service times as stated in the previous subsection, but have SCVs equal to 2.0. Assume further that the external arrivals of aircraft at the airfield are at the rate of 0.9 per hour. Numerical results from the analytical method as well as simulation indicate that under these conditions, the time on the ground exceeds 16 h for an aircraft and the throughput of the network is below 0.7 aircraft per hour (see the section for $\lambda_{0,2} = 0.9$, $c^2 = 2.0$, $N = 12$ of Table 8). Consider a situation where an airfield operations manager is interested in reducing the time on the ground to less than 14 h, while at the same time increasing the throughput to 0.75 aircraft per hour. The numerical results in Table 8 indicate that without reducing the mean time or SCV of airfield operations, these objectives might be hard to achieve for an airfield without concurrent operation (see entries for $N = 12$ in Table 8). For instance, although the entries in Table 8 indicate that throughput values greater than 0.75 can be achieved with $N = 12$, they are possible only in the scenarios where $c^2 < 2.0$ (for any value of $\lambda_{0,2}$ from 0.8 to 1.0). When $N = 12$ and $\lambda_{0,2} = 0.9$, an average time on the ground of less than 14 h is achieved only when $c^2 < 1.0$. By contrast, the corresponding section of Table 9 indicates that by redesigning the airfield so that some of the operations are done concurrently, the desired managerial objective can be met (entries for $N = 12$, $\lambda_{0,2} = 0.9$, and $c^2 = 2.0$ in Table 9). However, such reorganizations can be expensive.
Furthermore, our analytical method enables us to examine the impact on performance of variability of airfield operations. It provides us with alternative means of achieving the desired managerial objective. We continue to consider the situation of the preceding paragraph ($N = 12$). The results in Table 8 indicate that if $c_2^2$ were reduced from 2.0 to 0.7, a throughput of more than 0.8 aircraft per hour and a time on the ground of less than 14 h can be simultaneously achieved in a network with no concurrent operation. At this point, we note that such reduction in variability of airfield operations could usually be achieved with much less investment than that required for redesigning airfields to permit concurrent operations.

Table 8 also indicates that reduction in variability results in smaller queues at the bottleneck resource (Cargo), as well as a reduction in the mean number of aircraft that await permission to land ($Q(B_{0,2})$). The numerical results in Table 8 indicate that subsequent to reduction in variability, the airfield might be able to handle a higher air traffic load, and even under such an increased load, the managerial targets of throughput and time on the ground could be achieved without increasing average queues at the bottleneck operations beyond their current levels.

The results summarized in Table 9 allow us to understand the combined impact of reduction in variability and concurrent operations. If investments in concurrent operations were made subsequent to reduction in variability levels, further improvements in airfield operations could be obtained. For instance, Table 9 indicates that if the airfield operations have a SCV of 0.7, then airfield could yield a throughput greater than that of the corresponding network with no concurrent operation. These networks also yield result in a reduction in the average time on the ground for an aircraft.

Clearly, reduction in variability offers substantial improvements in airfield operations. These opportunities are often ignored by deterministic optimization models that attempt to improve airfield operations. Variability in operations...
Table 9
27 Experiments for the network of Fig. 2

<table>
<thead>
<tr>
<th>$\lambda_{0,2}$</th>
<th>$c_2^2$</th>
<th>Throughput (S)</th>
<th>Q(Cargo) (S)</th>
<th>U(Cargo) (S)</th>
<th>T(Ground) (S)</th>
<th>Q($B_{0,2}$) (S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(A)</td>
<td>(A)</td>
<td>(A)</td>
<td>(A)</td>
<td>(A)</td>
</tr>
<tr>
<td>$N = 4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.7</td>
<td>0.70</td>
<td>0.66</td>
<td>1.32</td>
<td>1.02</td>
<td>70.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>66.40</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.67</td>
<td>0.64</td>
<td>1.33</td>
<td>1.07</td>
<td>66.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>64.10</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.58</td>
<td>0.58</td>
<td>1.28</td>
<td>1.17</td>
<td>57.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57.80</td>
</tr>
<tr>
<td>0.9</td>
<td>0.7</td>
<td>0.72</td>
<td>0.70</td>
<td>1.39</td>
<td>1.11</td>
<td>71.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>69.90</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.68</td>
<td>0.67</td>
<td>1.39</td>
<td>1.15</td>
<td>68.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>66.70</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.59</td>
<td>0.59</td>
<td>1.31</td>
<td>1.22</td>
<td>58.80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>58.90</td>
</tr>
<tr>
<td>1.0</td>
<td>0.7</td>
<td>0.74</td>
<td>0.72</td>
<td>1.43</td>
<td>1.17</td>
<td>73.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>72.10</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.68</td>
<td>0.68</td>
<td>1.41</td>
<td>1.20</td>
<td>68.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68.20</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.59</td>
<td>0.59</td>
<td>1.33</td>
<td>1.25</td>
<td>58.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>59.40</td>
</tr>
<tr>
<td>$N = 8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.7</td>
<td>0.77</td>
<td>0.74</td>
<td>2.17</td>
<td>1.54</td>
<td>76.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>74.00</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.75</td>
<td>0.73</td>
<td>2.28</td>
<td>1.69</td>
<td>75.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>72.60</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.68</td>
<td>0.68</td>
<td>2.25</td>
<td>2.05</td>
<td>67.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67.80</td>
</tr>
<tr>
<td>0.9</td>
<td>0.7</td>
<td>0.82</td>
<td>0.80</td>
<td>2.53</td>
<td>1.86</td>
<td>81.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>79.70</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.79</td>
<td>0.77</td>
<td>2.57</td>
<td>2.24</td>
<td>78.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>77.30</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.71</td>
<td>0.70</td>
<td>2.58</td>
<td>2.29</td>
<td>70.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70.40</td>
</tr>
<tr>
<td>1.0</td>
<td>0.7</td>
<td>0.85</td>
<td>0.84</td>
<td>2.78</td>
<td>2.12</td>
<td>84.47</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.70</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.81</td>
<td>0.80</td>
<td>2.85</td>
<td>2.23</td>
<td>81.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80.20</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.72</td>
<td>0.72</td>
<td>2.70</td>
<td>2.44</td>
<td>72.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>71.80</td>
</tr>
<tr>
<td>$N = 12$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.7</td>
<td>0.79</td>
<td>0.77</td>
<td>2.81</td>
<td>1.84</td>
<td>79.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>76.50</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.78</td>
<td>0.76</td>
<td>2.85</td>
<td>2.07</td>
<td>77.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>75.50</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.73</td>
<td>0.72</td>
<td>3.22</td>
<td>2.68</td>
<td>73.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>72.10</td>
</tr>
<tr>
<td>0.9</td>
<td>0.7</td>
<td>0.85</td>
<td>0.83</td>
<td>3.56</td>
<td>2.42</td>
<td>85.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>83.30</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.84</td>
<td>0.82</td>
<td>3.60</td>
<td>2.67</td>
<td>83.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>81.50</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.77</td>
<td>0.76</td>
<td>3.62</td>
<td>3.21</td>
<td>77.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>75.90</td>
</tr>
<tr>
<td>1.0</td>
<td>0.7</td>
<td>0.89</td>
<td>0.88</td>
<td>4.04</td>
<td>2.98</td>
<td>88.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>87.90</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.86</td>
<td>0.85</td>
<td>4.05</td>
<td>3.17</td>
<td>85.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85.10</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>0.78</td>
<td>0.78</td>
<td>3.92</td>
<td>3.55</td>
<td>78.60</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>77.80</td>
</tr>
</tbody>
</table>

Table 10
Network of Fig. 1 with $\lambda_{0.2} = 0.9$

<table>
<thead>
<tr>
<th>$N$</th>
<th>$c_2^2$</th>
<th>Throughput (A)</th>
<th>(S)</th>
<th>Q(Cargo) (A)</th>
<th>(S)</th>
<th>T(Ground) (A)</th>
<th>(S)</th>
<th>Q($B_{0.2}$) (A)</th>
<th>(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.7</td>
<td>0.60</td>
<td>0.59</td>
<td>0.89</td>
<td>0.94</td>
<td>6.40</td>
<td>6.69</td>
<td>3.31</td>
<td>3.62</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.56</td>
<td>0.56</td>
<td>0.90</td>
<td>0.95</td>
<td>6.92</td>
<td>7.13</td>
<td>3.52</td>
<td>3.71</td>
</tr>
<tr>
<td><strong>2.0</strong></td>
<td><strong>0.48</strong></td>
<td><strong>0.49</strong></td>
<td><strong>0.92</strong></td>
<td><strong>0.96</strong></td>
<td><strong>8.23</strong></td>
<td><strong>8.18</strong></td>
<td><strong>3.87</strong></td>
<td><strong>3.87</strong></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.7</td>
<td>0.75</td>
<td>0.75</td>
<td>1.60</td>
<td>1.82</td>
<td>9.30</td>
<td>10.10</td>
<td>2.11</td>
<td>2.41</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.71</td>
<td>0.71</td>
<td>1.66</td>
<td>1.87</td>
<td>10.29</td>
<td>10.86</td>
<td>2.46</td>
<td>2.69</td>
</tr>
<tr>
<td><strong>2.0</strong></td>
<td><strong>0.61</strong></td>
<td><strong>0.61</strong></td>
<td><strong>1.76</strong></td>
<td><strong>1.85</strong></td>
<td><strong>12.60</strong></td>
<td><strong>12.58</strong></td>
<td><strong>3.12</strong></td>
<td><strong>2.98</strong></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.7</td>
<td>0.80</td>
<td>0.81</td>
<td>2.19</td>
<td>2.57</td>
<td>13.41</td>
<td>13.06</td>
<td>1.46</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.77</td>
<td>0.77</td>
<td>2.31</td>
<td>2.60</td>
<td>13.26</td>
<td>14.09</td>
<td>1.83</td>
<td>1.92</td>
</tr>
<tr>
<td><strong>2.0</strong></td>
<td><strong>0.68</strong></td>
<td><strong>0.69</strong></td>
<td><strong>2.53</strong></td>
<td><strong>2.67</strong></td>
<td><strong>16.48</strong></td>
<td><strong>16.31</strong></td>
<td><strong>2.58</strong></td>
<td><strong>2.42</strong></td>
<td></td>
</tr>
</tbody>
</table>

plays an important role in system performance. As seen in Tables 8 and 9, for both networks with and without concurrent operations, an increase in variability of airport operations results in reduced throughput, increased queues at the bottleneck, and increased average time on the ground for aircraft. However, the degradation in performance due to increase in variability appears to be less for an airfield with concurrent operations.
Table 11
Network of Fig. 2 with $\lambda_{0.2} = 0.9$

<table>
<thead>
<tr>
<th>N</th>
<th>$c^2_2$</th>
<th>Throughput (Cargo)</th>
<th>T(Ground)</th>
<th>Q(B0.2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(A)</td>
<td>(S)</td>
<td>(A)</td>
</tr>
<tr>
<td>4</td>
<td>0.7</td>
<td>0.70</td>
<td>0.72</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.67</td>
<td>0.68</td>
<td>1.15</td>
</tr>
<tr>
<td><strong>2.0</strong></td>
<td><strong>0.59</strong></td>
<td><strong>0.59</strong></td>
<td><strong>1.22</strong></td>
<td><strong>1.31</strong></td>
</tr>
<tr>
<td>8</td>
<td>0.7</td>
<td>0.80</td>
<td>0.82</td>
<td>1.86</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.77</td>
<td>0.79</td>
<td>2.00</td>
</tr>
<tr>
<td><strong>2.0</strong></td>
<td><strong>0.70</strong></td>
<td><strong>0.71</strong></td>
<td><strong>2.29</strong></td>
<td><strong>2.58</strong></td>
</tr>
<tr>
<td>12</td>
<td>0.7</td>
<td>0.83</td>
<td>0.85</td>
<td>2.42</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.82</td>
<td>0.84</td>
<td>2.67</td>
</tr>
<tr>
<td><strong>2.0</strong></td>
<td><strong>0.76</strong></td>
<td><strong>0.77</strong></td>
<td><strong>3.21</strong></td>
<td><strong>3.62</strong></td>
</tr>
</tbody>
</table>

Fig. 5. Network of Fig. 1 with $\lambda_{0.2} = 0.9$ (Table 10). In the legend, (A) refers to the analytical method and (S) to the simulation method, while the number in parentheses refers to the value of $c^2_2$. The legend is valid for the four plots.

Finally, we use the results summarized in Tables 10 and 11 to demonstrate the tradeoffs with respect to another key design parameter, namely the capacity of the network $N$. The graphs in Figs. 5 and 6 show as expected that the throughput of the airfield network increases with $N$. Although the relative increase in throughput is greater for a network without concurrent operation than for a network with concurrent operations (see the entries in bold in Tables 10 and 11), the network with concurrent operations always has higher throughput than a corresponding network.
without concurrent operations. The results in Tables 10 and 11 indicate that for the network under consideration, as the network capacity, \( N \), increases from 4 to 12, the average time on the ground almost doubles. This is true for networks with and without concurrent operations. However, the airfield network with concurrent operations has a higher throughput, a smaller queue of aircraft waiting to land on average, and a slightly larger mean queue length at the bottleneck. These results seem to indicate that networks with concurrent operations yield significant improvements in performance as compared to networks without concurrent operation. As observed previously, the SCVs of operations have a significant impact on system performance. Often operations managers have to determine whether to invest in additional airfield capacity (increase \( N \)) in order to achieve a desired performance target. It is not difficult to see that reduction in SCVs of airfield operations could be an attractive alternative to increasing \( N \). For instance, Tables 10 and 11 indicate that when the airfield operations have an SCV equal to 2.0 (see the entries in bold in Tables 10 and 11) the performance improvements that might be possible by increasing \( N \) from 8 to 12 might be achieved by decreasing \( c^2 \) from 2.0 to 0.7.

6. Conclusions

In this paper we have presented a method for rapidly improving the performance of stochastic networks by using queueing network techniques based on parametric decomposition and two-moment approximations. The method employs some recently devised developments of the approximations, especially in the context of closed queueing networks with synchronization stations. We presented several reasons for wanting to find fast methods of improving performance, and noted prior research that had made progress toward this goal.
We then presented two prototype models of airfield operations, and followed that by an overview of how our suggested method would proceed to approximate various performance measures in these models, in the case of processing stations as well as in that of fork stations or join stations. Next, we presented algorithms that provide estimates of the performance measures of the network. In Section 5 we presented computational results for the prototype networks, including managerially oriented analyses aimed at evaluating tradeoffs with respect to several strategies for performance improvement, such as introducing concurrent operations in the airfield, reducing variability, or increasing airfield capacity.

These results indicated that approximation methods appear to offer a very substantial advantage in speed. That fact suggests that this class of methods may hold promise for helping to improve the effectiveness of airfield operations and thus for contributing to greater effectiveness in air logistics systems.

Acknowledgements

The first author’s research reported here was sponsored in part by the Air Force Research Laboratory under agreement number F49620-01-1-0040, and in part by the U.S. Army Research Office under Grant DAAG19-01-1-0502. The U.S. Government has certain rights in this material, and is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the sponsoring agencies or the U.S. Government.

The second author’s research reported here was supported in part by Rensselaer Polytechnic Institute’s Faculty Research Initiation Grant.

The third author’s research reported here was sponsored in part by the Air Force Research Laboratory under agreements numbered F49620-01-1-0040 and FA9550-04-1-0192, and in part by the U.S. Army Research Office under Grant DAAG19-01-1-0502.

References

