Optimal Updating of Forecasts for the Timing of Future Events

Robert A. Shumsky
William E. Simon Graduate School of Business Administration, University of Rochester, Rochester, New York 14627

A major problem in forecasting is estimating the time of some future event. Traditionally, forecasts are designed to minimize an error cost function that is evaluated once, possibly when the event occurs and forecast accuracy can be determined. However, in many applications forecast error costs accumulate over time, and the forecasts themselves may be updated with information that is collected as the expected time of the event approaches. This paper examines one such application, in which flow control managers in the U.S. air traffic system depend on forecasts of aircraft departure times to predict and alleviate potential congestion. These forecasts are periodically updated until take-off occurs, although the number of updates may be limited by the cost of collecting, processing, and distributing information. The procedures developed in this paper balance the costs of accumulated forecast errors and the costs of forecast updates. The procedures are applied to the aircraft departure forecasting problem and are compared with methods currently used by the air traffic management system. Numerical examples demonstrate that the procedures increase forecast accuracy while reducing the costs associated with frequent forecast updates.

(*Forecasting*; *Dynamic Programming Applications*; *Air Transportation*)

1. Introduction

This article considers forecasting problems in which forecasts of the time of some future event are used as input to a series of decisions, so that the consequences of forecast errors accumulate until the event occurs. Forecasts may be updated as the event approaches, and the relative importance of forecast errors may depend on the timing of the forecast relative to the actual event. In the application examined in this paper, forecasts of aircraft take-off times inform the decisions of air traffic managers. The take-off time forecasts are updated frequently, and the benefits of accurate forecasts to the managers vary over time. For example, local traffic controllers at major airports rely on take-off time predictions over forecast horizons measured in minutes, and they typically disregard forecasts that are produced hours in advance. Tactical traffic managers, on the other hand, rely on forecasts over four- or five-hour horizons.

The methods developed in the article use real-time data to update forecasts of the event time, given forecast error costs that materialize before the event occurs. This problem differs from the traditional forecasting problem, in which the optimal predicted time minimizes a single expected cost, such as the squared deviation between the forecast and the actual time of the event. This traditional optimality criterion implies that the cost is evaluated once, possibly when the event occurs and the consequences of forecast inaccuracy are seen. In practice, a sequence of forecasts may be produced, and a complete forecasting procedure must specify an *update schedule*, the times at which the forecasts are generated, as well as the forecasts themselves. In addition, each forecast is used many times, and forecast error costs must be evaluated over the entire period of forecast use.

The notion of an update schedule can also be found in the literature on reliability and machine inspection. Barlow et al. (1963) designed inspection schedules for systems with random times until failure. Later work produced schedules for systems which travel through three states: healthy, failed but asymptomatic, and

0025-1909/98/4403/0321$05.00

failed with symptoms (see, for example, Sengupta 1980, Parmigiani 1993, and Zelen 1993). The inspection schedules are designed to minimize a function of both inspection costs and the expected lapsed time between system failure and the first subsequent inspection. While the forecast update problem presented in this article also involves the design of a sequence of inspections and updates, both the cost function and the solution structure are fundamentally different. For the forecast update problem, forecast error costs can be assessed throughout the time up to the event, rather than during the time between the event and its discovery. In addition, the procedures derived here determine both a schedule of forecast updates and a sequence of updated forecasts of the remaining time until the anticipated event.

While forecast updates may reduce expected errors, each update may incur costs for data collection, forecast generation, and forecast distribution. Forecast updates can also be disruptive, diverting human attention away from other tasks. Therefore, the methods developed in this article balance the costs associated with updating forecasts and the costs associated with inaccurate forecasts.

The particular formulation presented here is motivated by a problem in air traffic management, the prediction of aircraft take-off times. In the U.S. air traffic control system, take-off time forecasts are generated by an automated air traffic management system operated by the Federal Aviation Administration (FAA). The primary components of the system are located in Atlantic City, New Jersey and Cambridge, Massachusetts, but forecasts generated by the system are distributed to air traffic managers throughout the United States. The forecasts are used by the managers to anticipate and prevent congestion at airports and in airspace sectors. In the current management system, updates to take-off time forecasts may be generated and distributed every five minutes. Only the most rudimentary information is available for updating: the fact that the departure has not yet occurred and the aircraft is still on the ground. While this article will focus on the take-off time forecasting problem (in this article the word ‘‘forecast’’ will be interchangeable with the phrase ‘‘take-off time forecast’’), the concepts and procedures developed for the air traffic control application are applicable to the general forecasting problem in which information arrives over time while decisions must be made from forecasts which are repeatedly updated.

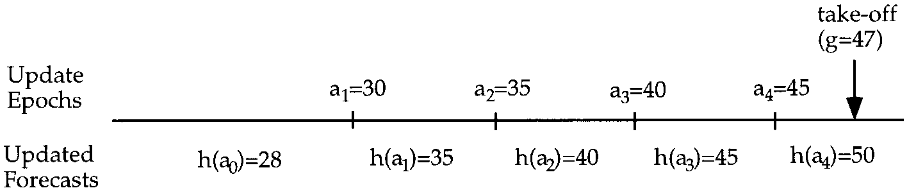
In this article §2 describes both the aircraft departure update problem and a few heuristics for its solution. Section 3 introduces a cost function that includes penalties for both forecast errors accumulated over time and update costs. Section 4 describes an optimization procedure for finding forecasts, given an update schedule, as well as a dynamic program to find an optimal update schedule, given a sequence of forecasts. A heuristic combines these procedures to produce a combination of update schedules and forecasts. In §5 the procedures are tested with numerical examples that approximate air traffic management scenarios. The procedures developed here are compared to the procedure used by the present air traffic control system. The examples demonstrate that when prior uncertainties are large, the optimization procedures have significantly greater forecast accuracy and fewer forecast updates than the current system. For moderate levels of uncertainty, simple heuristics provide benefits close to those of the optimization procedures.

2. The Aircraft Departure Update Problem

The FAA’s automated air traffic management system distributes aircraft take-off time forecasts to air traffic managers throughout the country, and the managers use the forecasts to inform traffic flow control decisions. When a flight takes off, the traffic management system is notified almost immediately. Therefore, the *absence* of a take-off signal also conveys information about the flight. Every five minutes, the system has the opportunity to distribute a revised take-off time forecast based on this information.

The current system uses a simple procedure for calculating these revisions. If the initial forecast time has passed, and if the flight has not taken off by the end of the next five-minute period, then the system predicts that the flight will take off five minutes later. If the aircraft is still on the ground when the new forecast expires, five additional minutes are added to the forecast. These five-minute updates repeat, as shown in Figure 1.

Figure 1 Constant Interval Updating with an Original Forecast of 28 Minutes



The following notation will help to describe this procedure and its alternatives. Let random variable g represent the difference between an aircraft's actual take-off time and some arbitrary time origin, such as the scheduled gate departure time. Long before a scheduled aircraft departure, the traffic management system generates a forecast of g which is derived from the weather forecast, the airport of departure, and other factors. Later, information about the flight's status will be used to update the forecast.

Each flight has a predefined update schedule, a schedule of times when forecasts are updated, given that a take-off has not been observed. The update schedule is followed until actual take-off occurs. Define a forecast update at time a_k to be an *update epoch* and the ordered sequence of update epochs $\mathbf{a} = \{a_0, a_1, \ldots, a_n\}$ is an update schedule. The update epoch a_0 is the time for production of the initial forecast of g. This forecast will be updated at a_1 as long as a take-off has not yet been observed. A final update is performed whenever a take-off does occur.

Associated with each update schedule is an ordered sequence of updated forecasts, $\mathbf{h}(\mathbf{a})$. The forecast $h(a_i)$ is the forecast of g that is in effect from a_i until the next update at a_{i+1}, so that $\mathbf{h}(\mathbf{a}) = \{h(a_0), h(a_1), \ldots, h(a_n)\}$. This notation is now applied to the procedure used by the current system.

Constant Interval Updating. The current implementation of the air traffic management system has a cycle time of five minutes; every five minutes the system refreshes the video screens of air traffic managers throughout the country. After making an initial prediction, $h(a_0)$, the system does not update the forecast of any particular flight until the beginning of the first five-minute cycle *after* $h(a_0)$. Thereafter it repeatedly adds five minutes to the forecast until the flight departs. Therefore, update epoch $a_1 = \lceil h(a_0) \rceil^5$, where $\lceil x \rceil^5$ is the beginning of the first five-minute interval after x. The updated forecast $h(a_1) = a_1 + 5$ and the next update epoch $a_2 = h(a_1)$. Similarly, $h(a_2) = a_2 + 5$, $a_3 = h(a_2)$, etc. If an hour has passed and the flight is still on the ground, the flight is marked as canceled. If the flight eventually does take off, it is immediately reactivated but is seen as a surprise arrival to the users of the system.

Figure 1 illustrates an example of this update procedure. The initial forecast $h(a_0) = 28$ min. The beginning of the first cycle after 28 min. is 30 min. An update occurs at 30 min., producing a new forecast of 35 min. The update at 35 min. produces a new forecast of 40 min., and this continues until the flight departs at 47 min. There is a total of six updates in this example: our initial forecast $h(a_0)$, updates at 30, 35, 40, and 45 minutes, and a final update when the flight does depart.

The number of *scheduled* updates is fixed: one every five minutes until an hour passes. However, the number of updates that are actually performed is a random variable with a minimum of two: one "update" for the initial forecast and one at the true time of take-off. Note that the final update at take-off occurs instantly and is not restricted to be at any scheduled update epoch a_k, since a departing aircraft immediately appears on the radar screens of the local air traffic managers and they need not wait for an update from the national air traffic management system.

The constant interval procedure has two fundamental flaws. The first is that the updated forecasts ignore distributional information about g. While it is simple to add five minutes to each forecast, there is no reason to believe that such a rule minimizes a relevant measure of expected forecast error. The second flaw lies in the timing of the update epochs, which is also unlikely to be optimal. The system may schedule too many updates, and each update requires significant computer processing to alter the large data structures that store flight information. In addition, frequent updates can confuse air traffic managers and can lead to distrust in a system that cannot ‘‘make up its mind.’’ Thus, what little is gained in accuracy can be lost in greater expense and frustration.

Now consider two alternative procedures for generating update schedules and forecast sequences of aircraft departure times. The first is driven by the conditional means of the distribution of g. The second is a continuous updating scheme which should outperform any discrete procedure in terms of forecast accuracy.

Discrete Conditional Updating. The update epochs are determined in the same manner as the constant interval epochs. Once a forecast has expired, the next update is scheduled at the beginning of the following cycle. However, each updated forecast $h(a_k)$ is the *conditional expectation* of g given that $g > a_k$. As is well-known in decision analysis, the conditional expected value minimizes the mean squared error of the forecast. Other forecasts, such as the median, minimize other loss functions.

As in our previous numerical example, the initial forecast is 28 min. and the next update epoch a_1 occurs at the beginning of the following cycle, at 30 min. (see Figure 2). Suppose at time 30 we determine that $E(g \mid g > 30) = 39$ min., so that the first updated forecast is $h(a_1) = 39$ min. The next update occurs at 40 min., the beginning of the following cycle. Finally suppose that the second conditional forecast, $h(a_2)$, is 48 minutes, and the flight departs before the next update can occur.

In this example, there were four updates. We will see that when compared with constant interval updating, discrete conditional updating reduces the number of expected updates while producing forecasts that reduce the expected forecast error.

Continuous Conditional Updating. Suppose that the traffic management system has an infinitesimal cycle time, so that forecast updates may be released in an essentially continuous manner. Such a system would allow continuous conditional updating, so that $h(t) = E(g \mid g > t)$ for all t. Since the system would always produce the most accurate forecast possible, it would outperform its discrete counterparts but would require an essentially infinite number of updates for each flight. For certain cost functions, such a system establishes a *lower bound* on forecast error.

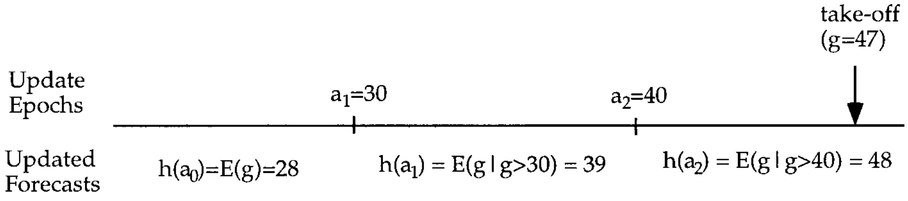
As these examples suggest, there is a tradeoff between forecast accuracy and update frequency. To measure the utility of each update procedure, the next section defines the costs associated with update schedules and forecasts. Later sections derive schedules and forecasts that minimize these costs.

3. Cost Function for a Sequence of Forecasts

Consider an air traffic manager responsible for an airspace sector (a distinct region of controlled airspace). During the next few hours a large number of aircraft are scheduled to fly through the sector, and the traffic manager is concerned that the sector will become dangerously crowded. If the manager is warned of the congestion in advance, then it is possible to delay the departures of aircraft still on the ground or to divert aircraft en-route.

The manager must make these decisions before many of the aircraft depart and therefore relies on congestion forecasts for the airspace sector. These forecasts are derived from take-off time predictions for the aircraft that are still on the ground. Recent studies of the air traffic management system have found that these take-off time forecasts can be extremely inaccurate (Goranson 1992 and Shumsky 1995). The inaccurate take-off time forecasts lead to unreliable congestion forecasts at downstream sectors.

Figure 2 Discrete Conditional Updating with an Original Forecast of 28 Minutes



Now consider the cost of take-off time forecast inaccuracy to the air traffic manager. For a manager with a decision horizon measured in hours, aircraft arrivals a few minutes before or after the forecasted times are unlikely to change the overall traffic density substantially. Larger forecast errors have more serious consequences, for air traffic managers may be surprised by the early appearance of numerous flights. Given large forecast errors, managers may also anticipate congestion which is later found to be a statistical illusion. Therefore, we represent the cost of a single forecast error as $C(g_0 - h(t))$, a function of the difference between an actual take-off time g_0 and $h(t)$, the forecast at time t. The specific form of this cost function is determined by the relationship between forecast error size and traffic management effectiveness. For example, the traditional squared error function, $(g_0 - h(t))^2$, would be reasonable if the largest errors are much more likely to lead to unexpected congestion or unnecessary and expensive control activities. Finally, note that the forecast error cost is not necessarily a monetary cost. Just as the variance of a random variable is the expected squared deviation from the mean, the cost function defined here is the expected total forecast error that has a significant impact on traffic management decisions.

The function $C(g_0 - h(t))$ assigns an error cost to a single forecast but does not capture the dynamic nature of the traffic manager's forecast error costs. The manager makes multiple decisions over time, and we wish to assess the accuracy of an update schedule which specifies a number of forecasts over a number of time periods. The cost function proposed here accumulates forecast error over time while weighting these errors according to a time-varying function, $w(t)$. For example, errors made far before the proposed departure time of an aircraft may be given less weight than those made close to the proposed departure, when the traffic manager may be most interested in the forecast.

The assignment of weights to forecast errors over time requires an understanding of the relative importance of forecast accuracy as the take-off time approaches. The assignment of weights may be accomplished by monitoring when, and how, traffic managers use forecasts. The managers themselves can report forecast usage over time. A simple numerical example will be explored in section 4 in which forecasts are ignored until flights are scheduled to depart from the gate. Therefore, the weighting function $w(t)$ rises from zero to one at the scheduled gate departure time. This weighting function would be appropriate for local managers making relatively short-term decisions.

The total forecast error cost is found by accumulating the weighted forecast errors up to the actual take-off time. In addition to the costs associated with forecast errors, a penalty K_u is assessed for each forecast update. If forecast accuracy is the primary concern, then K_u may be adjusted downward. However, an analyst will be able to examine this trade-off between the expected costs associated with forecast errors and costs associated with forecast updates by varying K_u and repeatedly solving for the optimal update schedule and associated forecast error costs.

3.1. Cost Function for a Given Departure Time

Many well-established methods exist for assessing forecast accuracy and for designing optimal forecasts. For a single forecast we will adopt the general framework proposed by Seidmann and Smith (1981), which allows the cost of an inaccurate forecast to assume a variety of functional forms. Given a forecast $h(t)$ at time t and an actual departure time, g_0, specify a cost function C on the forecast error $g_0 - h(t)$. Note that $C(g_0 - h(t))$ is the forecast error cost given g_0, a realization of random variable g. In the next section we will derive the expected forecast error costs, given the distribution function of g.

In order to include many possible functional forms, define two cost functions for each forecast. One cost is assessed if the flight departs before the forecast and another if the flight departs after the forecast:

$$C(g_0 - h(t)) = \begin{cases} C_r(h(t) - g_0) & \text{if } g_0 < h(t), \\ C_l(g_0 - h(t)) & \text{if } g_0 > h(t), \\ 0 & \text{if } g_0 = h(t). \end{cases} \quad (1)$$

Cost function C_r applies when there is an early departure before the predicted departure time while C_l applies when there is a late departure after the predicted time. Throughout this discussion, assume that the functions C_r and C_l are monotone increasing, strictly convex, twice differentiable, and vanish at the origin. This includes many reasonable cost functions and enables the derivation of simple expressions for the optimal forecasts.

The cost function C is a static one, for it does not take into account the fact that we may update our forecast as time progresses. Therefore, define the forecast error cost for an actual departure at time g_0 to be:

$$C_F(g_0;\, \mathbf{a}, \mathbf{h}(\mathbf{a})) = \int_{-\infty}^{g_0} w(t)C(g_0 - h(t))dt, \qquad (2)$$

where

$$w(t) \geq 0.$$

The weighting function $w(t)$ reflects the relative importance of forecast error cost $C(g_0 - h(t))$ at time t. The forecast $h(t)$ is determined by the update schedule and is revised at update epochs prior to g_0. This cost function weights the forecast error by $w(t)$ and accumulates this error until the departure of the flight. Besides the requirement that $w(t)$ be nonnegative, we will have no restrictions on its functional form. The function $w(t)$ may rise or fall to give greater emphasis to those time periods when forecast errors could be especially damaging.

The cost $C_F(g_0;\, \mathbf{a}, \mathbf{h}(\mathbf{a}))$ will depend on both the update schedule $\mathbf{a}$ and updated forecasts $\mathbf{h}(\mathbf{a})$. In order to make this dependence explicit, rewrite the cost function as the sum of costs over each interval $[a_k, a_{k+1})$. The value of each term in the sum will depend only on the forecast produced for that interval,

$$h(t) = h(a_k) \quad \text{for } a_k \leq t < a_{k+1}, \qquad (3)$$

so that Equation (2) may be rewritten as:

$$\begin{aligned} C_F(g_0;\, \mathbf{a}, \mathbf{h}(\mathbf{a})) = \sum_{i=0}^{k-1} W(a_i, a_{i+1})C(g_0 - h(a_i)) \\ + W(a_k, g_0)C(g_0 - h(a_k)) \qquad (4) \\ \forall\, a_k \leq g_0 < a_{k+1}, \end{aligned}$$

where

$$W(a, b) = \int_a^b w(t)dt. \qquad (5)$$

In addition to this forecast cost, a cost K_u is assessed on each update, where K_u is expressed in units that are equivalent to the units used for the forecast error cost. The minimum cost of updating for any flight is $2K_u$: a charge of K_u for the initial forecast at a_0 and another for the update when the aircraft takes off at g_0.

3.2. Expected Cost of a Sequence of Forecasts

Let F_g be the cumulative distribution function of g and let $F(\mathbf{a}, \mathbf{h}(\mathbf{a}))$ be the expected forecast error cost under update schedule $\mathbf{a}$ and forecast sequence $\mathbf{h}(\mathbf{a})$. Therefore,

$$F(\mathbf{a}, \mathbf{h}(\mathbf{a})) = \int_{-\infty}^{\infty} C_F(g_0;\, \mathbf{a}, \mathbf{h}(\mathbf{a}))dF_g(g_0). \qquad (6)$$

Using Equation (4) and rearranging terms, we find:

$$F(\mathbf{a}, \mathbf{h}(\mathbf{a})) = \sum_{k=0}^{n} M_k(h(a_k)), \qquad (7)$$

where n is the number of scheduled updates. For $k = 0 \cdots n - 1$,

$$\begin{aligned} M_k(h(a_k)) = \int_{a_k}^{a_{k+1}} W(a_k, g_0)C(g_0 - h(a_k))dF_g(g_0) \\ + W(a_k, a_{k+1}) \int_{a_{k+1}}^{\infty} C(g_0 - h(a_k))dF_g(g_0), \end{aligned} \qquad (8)$$

and for $k = n$,

$$M_n(h(a_n)) = \int_{a_n}^{\infty} W(a_n, g_0)C(g_0 - h(a_n))dF_g(g_0). \qquad (9)$$

In this sum, each term $M_k(h(a_k))$ represents the prior cost, which we expect to accumulate during the interval $[a_k, a_{k+1})$. Throughout this interval, $h(a_k)$ is used as the forecast.

Given update schedule $\mathbf{a}$, the expected number of update epochs is:

$$U(\mathbf{a}) = \sum_{k=0}^{n} (1 - F_g(a_k)) + 1. \qquad (10)$$

The single update added to this total represents the final update when the flight departs. The total expected cost for update schedule $\mathbf{a}$ and the sequence of forecasts $\mathbf{h}(\mathbf{a})$ is $F(\mathbf{a}, \mathbf{h}(\mathbf{a})) + K_uU(\mathbf{a})$.

When the range of random variable g is not bounded, some updating procedures, such as the discrete conditional procedure, generate a schedule with an infinite number of updates. For the forecast distributions and cost functions tested in this article, it is always possible to choose a sufficiently large n such that both $(1 - F_g(a_n))$ and $M_n(h(a_n))$ are arbitrarily close to zero. Therefore, an infinite update schedule can be approximated closely by a finite schedule.

Now consider the minimization problem:

$$\min_{\mathbf{a},\mathbf{h}(\mathbf{a})} F(\mathbf{a},\mathbf{h}(\mathbf{a})) + K_u U(\mathbf{a}) \tag{11}$$

$$= \min_{\mathbf{a},\mathbf{h}(\mathbf{a})} \sum_{k=0}^{n} \{M_k(h(a_k)) + K_u(1 - F_g(a_k))\} + K_u. \tag{12}$$

The next section develops procedures for finding optimal update epochs and forecasts.

4. Optimal Update Schedules and Forecasts

An optimal solution to the forecast cost minimization problem requires that expression (11) be minimized over both the update schedule **a** and the sequence of forecasts **h**(**a**). While an exact solution may be a subject for further study, the decomposition heuristic presented here produces near-optimal results when applied to the numerical examples in the next section. The heuristic divides the problem into two steps, first finding an optimal update schedule under a given method for deriving forecasts and then revising the forecasts to be optimal for the update schedule derived in the first step. The steps are shown in the flow chart, Figure 3. Each of these two steps may be used independently. A system may be required to follow a predetermined update schedule, and the second step determines an optimal sequence of forecasts under this constraint. Alternatively, a particular class of forecasts, such as the simple conditional mean, may be preferred at each update epoch, and the first step finds the optimal times at which these forecasts should be produced.

The next subsection discusses the production of forecasts and derives methods which will be used in the first and second steps of the heuristic. Two classes of forecasts are presented: one-time forecasts and sequential forecasts. A *one-time forecast*, $h^o(t)$, minimizes the expected value of the cost function $C(g - h(t))$ and is optimal in the traditional sense: the cost is evaluated at only one time, i.e., when the aircraft departs. *Sequential forecasts* minimize the expected cumulative cost defined in Equation (6) for a given update schedule. Step 2 in the heuristic solves for the sequential forecasts. Section 4.2 describes the dynamic program used in step 1 to determine the optimal update schedule, given a particular forecasting method.

4.1. Optimal One-time and Sequential Forecasts

Assume that at a potential update epoch t_k we may make a single forecast and that we cannot change this forecast until the flight departs. In the present air traffic management system, a potential update epoch t_k might be the beginning of a five minute cycle when the system may generate and distribute a new forecast. Given the information available in the aircraft departure update problem, the optimal one-time forecast, $h^o(t_k)$, minimizes:

$$E[C(g - h(t_k)) | g > t_k)]$$

$$= (1 - F_g(t_k))^{-1} \int_{t_k}^{\infty} C(g_0 - h(t_k))dF_g(g_0). \tag{13}$$

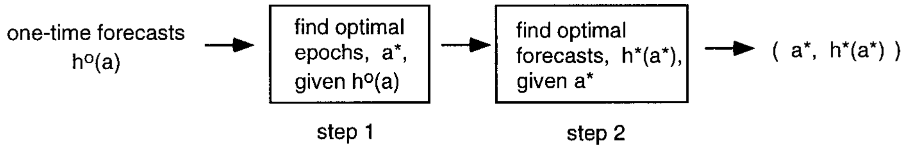
For a linear or quadratic cost function, $h^o(t_k)$ is the conditional median, or mean, respectively.

For a particular update schedule **a**, the one-time forecasts will not necessarily minimize $F(\mathbf{a}, \mathbf{h}(\mathbf{a}))$. Let $\mathbf{h}^*(\mathbf{a})$ be the optimal sequential forecasts, so that $\mathbf{h}^*(\mathbf{a})$ is the sequence $\{h^*(a_0), h^*(a_1), \ldots, h^*(a_n)\}$ which minimizes $F(\mathbf{a}, \mathbf{h}(\mathbf{a}))$, given **a**. From Equation (7), the kth term in the sum of forecast error costs depends only on $h(a_k)$. Therefore, each forecast $h^*(a_k)$ need only minimize the term $M_k(h(a_k))$. The appendix describes a procedure for finding the optimal sequential forecast $h^*(a_k)$.

4.2. Optimal Update Schedules Given Forecasts

Assume that we have a finite number of potential update epochs, $t_0, t_1, \ldots, t_m$. In the following optimization

Figure 3 Two-step Heuristic for Finding Update Epochs and Forecasts



problem we will choose these update epochs to form a large "window" around the proposed departure time. Also assume that we are given forecast $h(t_k)$ at each potential update epoch t_k, $k = 0, 1, \ldots, m$ and that no forecast updates are possible after t_m. The time of this last potential update epoch can be chosen so that $(1 - F_g(t_m))$ is sufficiently close to zero, so that any updating after t_m does not significantly change expected costs.

In order to simplify the notation, define $t_0 \equiv -\infty$ and set $h(t_0)$ to be our initial, prior forecast. Setting $t_0 \equiv -\infty$ indicates that the time of the initial forecast is fixed to be far in the past, before information is available which would alter the prior probability distribution. This is a convenient and accurate shorthand for current practice in the traffic management system. The initial forecast for a flight is often produced during the previous night, when the system's computers are underutilized.

The following dynamic program finds the optimal update schedule:

Indices: $k = 0, 1, \ldots, m$, the index for each potential update epoch.

Potential update epochs: t_k.

State variables:

x_k = index of the most recent update epoch

$$(x_k = 0, 1, \cdots k - 1).$$

$$d_k = \begin{cases} 0 & \text{if the aircraft has not departed by time } t_k, \\ 1 & \text{if the aircraft has departed by time } t_k. \end{cases}$$

Control variables: The zero-one variable u_k represents the decision to update at epoch t_k:

$$u_k = \begin{cases} 0 & \text{if no update occurs at } t_k, \\ 1 & \text{if an update occurs at } t_k, \end{cases} \tag{14}$$

for $1 \le k \le m$. Let $u_0 = 0$.

Transition Probabilities: The state variable d_k is stochastic with transition probabilities:

$$p(d_k = 0 \mid d_{k-1} = 0) = \frac{1 - F_g(t_k)}{1 - F_g(t_{k-1})},$$

$$p(d_k = 1 \mid d_{k-1} = 0) = \frac{F_g(t_k) - F_g(t_{k-1})}{1 - F_g(t_{k-1})},$$

$$p(d_k = 0 \mid d_{k-1} = 1) = 0,$$

$$p(d_k = 1 \mid d_{k-1} = 1) = 1.$$

The state variable x_k is deterministic. Its value is determined by the value of the control variable u_k:

$$x_{k+1} = \begin{cases} x_k & \text{when } u_k = 0, \\ k & \text{when } u_k = 1. \end{cases} \tag{15}$$

Costs: Let $J_k(x_k, d_k)$, $k = 0, \ldots, m$, be the cost-to-go function at time t_k, given d_k and the most recent update at time t_{x_k}. At time t_k an update may occur and the value of x_{k+1} is determined according to Equation (15). The forecast $h(t_{x_{k+1}})$ applies during the period $[t_k, t_{k+1})$, since the update occurs at the beginning of the period. For $k = 0, \ldots, m - 1$,

$$J_k(x_k, 0) = \min_{u_k} \left\{ \frac{M_k(h(t_{x_{k+1}}))}{1 - F_g(t_k)} + u_k K_u + \mathrm{E}_{d_{k+1}} [J_{k+1}(x_{k+1}, d_{k+1})] \right\}, \tag{16}$$

$$J_k(x_k, 1) = K_u, \tag{17}$$

and the terminal cost is $J_{m+1}(x_{m+1}, d_{m+1}) = K_u$. If $d_k = 0$, the expected remaining cost is the sum of the expected forecast error cost from t_k to t_{k+1} using forecast $h(t_{x_{k+1}})$, the update cost at t_k if there is an update, and the expected costs accumulated after t_{k+1}. The quantity $M_k(h(t_{x_{k+1}}))$ is the unconditional expected forecast error cost accumulated between t_k and t_{k+1} when $h(t_{x_{k+1}})$ is used as a forecast (see Equation (8)). If $d_k = 1$, the procedure ends with a final update with cost K_u. It can be verified that the total cost $J_0(0, 0)$ is equal to the expected total cost of Equation (12), with the $h(t_k)$ replaced by $h(t_{x_{k+1}})$ and the control variable u_k multiplying the potential update costs.

Given m potential update epochs, the dynamic program must calculate and compare costs along pairs of arcs from $m(m + 1)/2$ nodes. Therefore, the algorithms has a running time of $O(m^2)$. Each step requires the calculation of $M_k(h(t_{k+1}))$, and the computational requirements of this calculation depend on the form of $w(t)$, $F_g(g_0)$, and the cost function for forecast errors. Algebraic expressions for M_k may be derived when the $w(t)$ is constant within each time period and $F_g(g_0)$ is a normal or gamma distribution (Shumsky 1995). This enables efficient computation of M_k when solving the dynamic program.

The dynamic program is used in Step 1 of the heuristic to find an update schedule, given the one-time forecasts $h^o(t_k)$ at each potential update epoch. In Step 2 we derive the optimal sequential forecasts for the update schedule generated in Step 1. An alternate to the dynamic program for specifying update epochs, discrete conditional updating, was described earlier. Whether a schedule is generated with the dynamic program or by discrete conditional updating, forecast accuracy can be improved by resolving for the optimal forecasts.

5. Examples and Numerical Results

In the following numerical examples, six distinct procedures are used to generate forecast sequences. For the first example, the prior take-off time distribution is similar to the distribution seen during poor weather at Atlanta Hartsfield International Airport (ATL). Additional experiments test the sensitivity of the procedures to the objective function parameters and the shape of the prior distribution.

Five of the six update procedures are restricted to potential update epochs at five-minute intervals, the cycle time of the present air traffic management system. The sixth technique, continuous conditional updating, assumes that the cycle time is zero and that updates occur at all times. This establishes a lower bound on expected forecast error costs. The six procedures are:

1. *Constant interval updating* with five-minute intervals. This is the procedure used in the current air traffic management system.

2. *Discrete conditional updating using one-time forecasts*. Use discrete conditional updating to set the timing of the update epochs. A one-time forecast, $h^o(t)$, is produced at each update epoch.

3. *Dynamic programming using one-time forecasts*. This is Step 1 in the heuristic, with the dynamic program (DP) generating the update schedule.

4. *Discrete conditional updating using sequential forecasts*. Discrete conditional updating determines the update schedule and then optimal forecasts are generated for that schedule.

5. *Dynamic programming using sequential forecasts*. The DP determines the update schedule and then optimal forecasts are found, given this schedule. This corresponds to Steps 1 and 2 in the heuristic.

6. *Continuous conditional updating*. This procedure was described earlier. Since we will be specifying a piecewise linear cost function, the optimal forecast used at all times will be the conditional median.

The next section describes the parameters of the numerical examples. This is followed by comparisons of the update schedules, forecasts, and expected costs produced by each procedure.

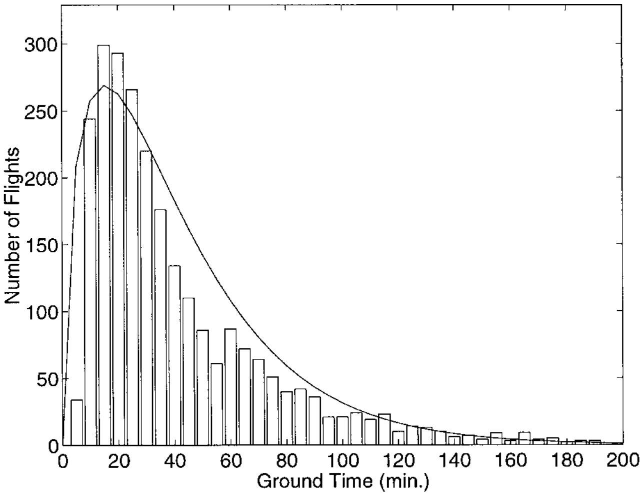
5.1. Parameters for the Numerical Examples

Parameters for the first experiment are derived from historical data collected during poor weather at ATL. The data include the *ground times* for departures by the major airlines, where we define the ground time as the time between the flight's scheduled departure from the gate and the actual take-off. The average ground time at ATL during July, 1995 was 30 minutes, with a standard deviation of 26.7 minutes. Figure 4 is a histogram of 2,551 ground times from ATL during four poor-weather days in July, 1995.[1] The mean of this bad-weather sample is 41 minutes and the standard deviation is 33 min. While this sample is not representative of all departures from ATL, accurate departure time forecasts are particularly important for these flights since poor weather reduces capacity and increases traffic congestion.

Superimposed on the histogram of Figure 4 is a gamma distribution with parameters $a = 1.58$ and $b = 26.2$. The parameters were determined by the method of moments, although a maximum likelihood method suggested by Bonvik (1994) produces nearly identical parameters from the data. The fitted distribution is similar in shape to the histogram but is not sufficiently "peaked." Indeed, a Chi-square test resoundingly rejects the hypothesis that the data were generated from this distribution. However, the gamma does capture the general shape of the histogram more accurately than a beta distribution, and fits much more closely than a symmetric distribution such as the normal. Another advantage of the gamma distribution is that the cost function may be expressed algebraically in terms of its cumulative density function. Therefore, the following numerical experiments will assume a gamma prior.

[1] Approximately 1% of the flights had ground times over 200 minutes, and one unfortunate flight was delayed over six hours at the gate. These flights were treated as outliers and removed from the sample. Cancelled flights were also ignored.

Figure 4 Ground Times During Poor Weather at Atlanta Hartsfield in July 1995 (Superimposed Gamma Distribution with Parameters $a = 1.58$ and $b = 26.2$)



Also assume that at the flight's scheduled gate departure time, regional air traffic managers begin to include the flight in their sector congestion forecasts. Given the distribution shown in Figure 4, this implies that forecasts are relevant about 41 minutes before the expected take-off time. Therefore, define the forecast error weighting function $w(t)$ as a step function that rises from 0 to 1 at time 0.

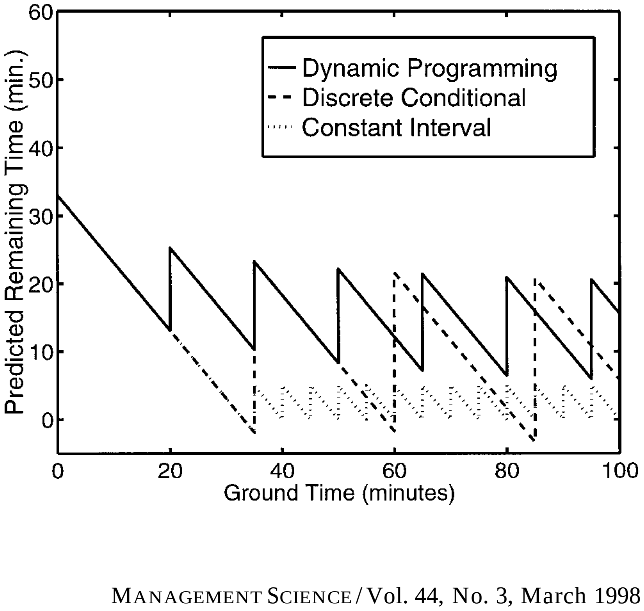
To specify the instantaneous forecast error cost function, $C(g_0 - h(t))$, assume that the immediate impact of a forecast error is proportional to its size. It is reasonable to place a heavier penalty on larger errors, since large forecast errors are more likely to lead to unexpected congestion or unnecessary and expensive traffic management actions. Assume a piecewise linear cost function, $C(g_0 - h(t)) = |g_0 - h(t)|$, although a quadratic cost function may be implemented if even heavier penalties should be placed on larger errors. Set $K_u = 25$, so that a penalty of 25 units is assessed for each update. Section 5.3 will test the sensitivity of the dynamic programming procedure to this parameter. The following numerical tests derive forecasts from the gamma distribution shown in Figure 4. By varying the parameters of the prior distribution, we will test the sensitivity of the procedures to the shape of the prior take-off time distribution.

5.2. Comparison of Forecast Sequences and Costs

Figure 5 displays the update schedules and forecasts produced by the first three procedures described above. The vertical axis of the figure is the predicted time remaining for each sequence of forecasts, so that if $h(t)$ is the forecast at time t, then the graph shows $h(t) - t$ plotted against t. Update epochs are visible as vertical jumps in the predicted remaining time. The figure displays update *schedules*, and for any particular flight the sequence of forecasts is likely to be cut short by the actual departure. The figure displays only the first 100 minutes of the schedule, although the forecast sequence extends to three hours after the scheduled departure time. With the given distribution of g, flights have a probability of 0.004 of departing after 3 hours.

All three procedures begin with a forecast equal to the prior median, 33 min. This initial one-time forecast minimizes the expected instantaneous cost, which is a linear function of the forecast error. The first update under the constant interval procedure occurs at 35 min., the beginning of the first cycle after the forecast has expired. Updates are produced at five-minute intervals thereafter. Discrete conditional updating (DC) first revises its forecast at 35 min. The revised forecast is 58 minutes, the conditional median given that $g > 33$. Therefore, at $t = 35$ the predicted time remaining is 23

Figure 5 Predicted Time Remaining for Three Forecast Sequences with $g \sim$ Gamma, Parameters $a = 1.58$ and $b = 26.2$, $w(t) = \mu_1(0)$, $K_r = K_l = 1$, and $K_\mu = 25$



minutes. The next update under the discrete conditional procedure occurs at 60 minutes.

The dynamic program (DP) also has an initial forecast of 33 minutes, but prescribes an initial update at 20 min. By this time the initial prediction of 33 minutes is sufficiently obsolete so that the update penalty is lower than the potential benefits of increased forecast accuracy. The dynamic program schedules subsequent updates at 15-minute intervals.

The forecasts produced by these three procedures are one-time forecasts, which are not optimal for the given update schedules. Figure 6 compares the one-time forecasts with the optimal sequential forecasts produced for the discrete conditional schedule. The optimal sequential forecasts are consistently higher than the one-time forecasts. The forecasts shown are about 10 minutes larger, while the optimal forecasts for the dynamic programming schedule are about 5 minutes larger. This is a consequence of the cumulative nature of the forecast cost function (2). The optimal sequential forecast $h^*(a_k)$ produced at update epoch k must minimize the expected cumulative forecast cost for departures after time a_k. Flights that depart immediately after a_k will have little time to accumulate error before take-off, and therefore receive less weight in the function to be minimized. Flights which depart after a_{k+1}, the next update epoch, will accumulate forecast error over the entire period $[a_k, a_{k+1})$, and therefore will have a relatively large impact on the cost function. This shifts the forecast $h^*(a_k)$ towards later flights. However, the difference between one-time and optimal sequential forecasts decreases as the time between update epochs decreases. As the time between update epochs approaches zero, as in continuous conditional updating, the one-time and optimal sequential forecasts converge.

Figure 6 **Predicted Time Remaining for Forecast Sequences Generated by the Discrete Conditional Procedure with Optimal Sequential and One-time Forecasts**

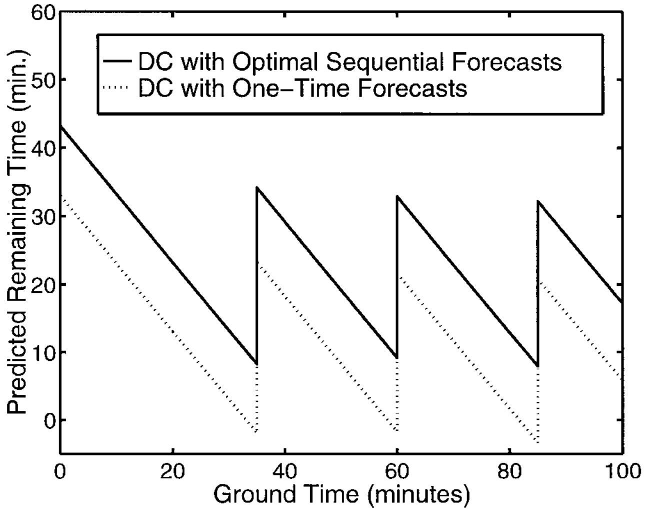


Table 1 **Expected Forecast Error and Update Costs for Six Update Schedules**

Update Procedure	Expected Forecast Error Cost	Expected Number of Updates	Expected Total Cost
Constant Interval	1,097	5.3	1,229
Discrete Conditional (DC) with One-time Forecasts	1,029	2.9	1,101
Dynamic Program (DP) with One-time Forecasts	952	4.0	1,052
Discrete Conditional (DC) with Optimal Forecasts	951	2.9	1,023
Dynamic Program (DP) with Optimal Forecasts	927	4.0	1,027
Continuous Updating	895	–	–

Table 1 contains the expected forecast error costs $F(\mathbf{a}, \mathbf{h}(\mathbf{a}))$, expected number of updates $U(\mathbf{a})$, and expected total costs for each forecast sequence. The expected forecast error cost for continuous updating is a lower bound. Constant interval updating has the largest expected forecast and update costs because of its frequent updates and uninformative forecasts. The discrete conditional procedure is relatively ‘‘stingy’’ with its updates and expects only 2.9 updates to the dynamic program’s 4.0. However, when using the one-time forecasts, the dynamic program achieves a significantly lower expected forecast error cost than discrete conditional updating. The extra updates pay for themselves with increased forecast accuracy.

When optimal sequential forecasts are used with each update schedule, the procedures come closer to the lower bound on expected forecast error cost and continue to improve upon constant interval updating. Both the discrete conditional and dynamic programming schedules, when combined with optimal forecasts, have an expected total cost 17% lower than the total cost of the constant interval updating procedure. The discrete conditional schedule and its small number of expected updates produces the lowest overall cost. The dynamic program's schedule, when combined with the "optimal" forecasts, need not be a global optimum.

Finally, it is interesting to note that the conditional *means* are consistently closer to the optimal forecasts than the conditional medians. For example, the prior mean of 41 minutes is close to the initial optimal sequential forecast of 43 minutes shown in Figure 6. Recall that the median is the optimal forecast for the instantaneous cost function but not for the overall cost function. This suggests that use of the conditional mean may produce lower forecast error costs than use of the conditional median. Numerical tests find that when using conditional means instead of conditional medians, total costs are reduced by about 1%.

5.3. Sensitivity to Prior Distribution and Update Costs

The following experiments compare the performances of the forecasting procedures as the parameters of the prior distribution and cost function vary. We first vary parameter *b* of the Gamma distribution while holding parameter *a* equal to the original value of 1.58. Both the mean and the standard deviation of the gamma distribution increase in direct proportion to the parameter *b*. Varying *b* produces prior means ranging from 10 to 60 minutes.

Figure 7 Expected Total Cost Reduction from the Constant Interval Procedure (Range in Prior Mean is Produced by Varying Parameter *b*)

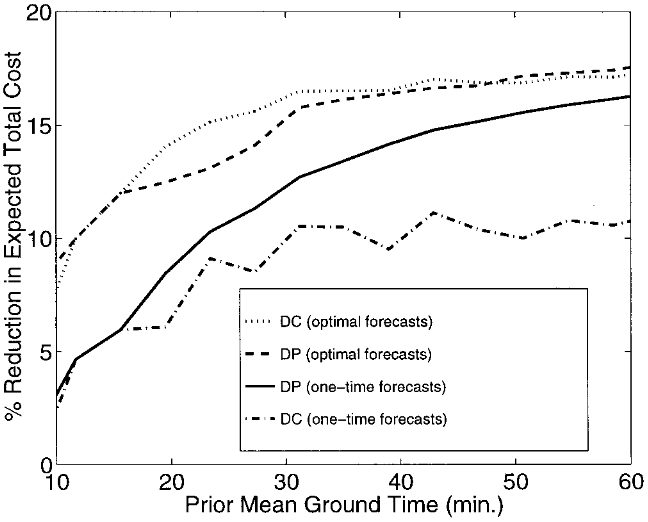


Figure 8 Expected Number of Updates as the Prior Mean Varies

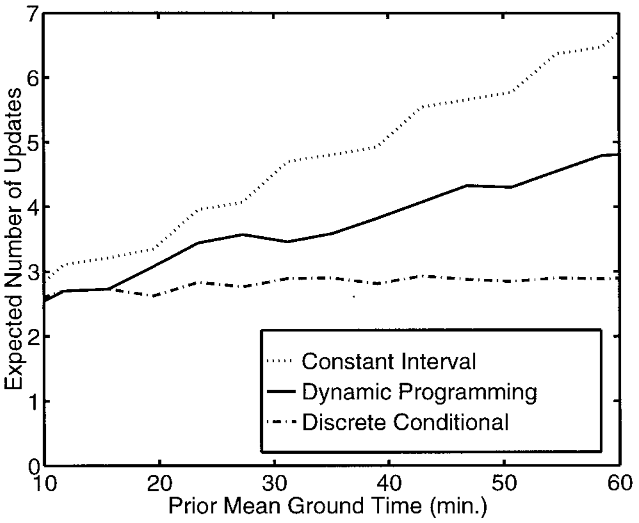


Figure 7 displays the percentage decrease in total costs from the constant interval procedure for each of the alternative procedures. The average ground time for all July 1995 flights from ATL was 30 minutes. Given a distribution with this prior mean, the savings vary from 10% for the discrete conditional procedure with one-time forecasts to 16% for the discrete conditional procedure with optimal forecasts. The largest cost reductions are achieved with the largest prior means. When the simple one-time forecasts are used, the dynamic program outperforms the discrete conditional procedure. When optimal forecasts are used, the discrete conditional and dynamic programming procedures generate similar expected costs.

These cost improvements are derived from both superior forecasts and reductions in the number of updates. Figure 8 displays the expected number of updates as the prior mean increases. Under the constant interval procedure the expected number of updates rises linearly, with a slope of approximately one update for every 15 minutes in the prior mean. Under the discrete conditional method the expected number of updates is nearly constant. The update frequency of the dynamic program falls between the two.

These experiments have held parameter $a = 1.58$, producing a prior distribution with a long right-hand tail. By varying parameter a we can assess the impact of the shape of the distribution on the relative advantages of the update procedures. With $a = 10$, for example, the gamma distribution is nearly symmetric. However, both the mean and the variance of the distribution are directly proportional to a. To control for changes in the mean, we set $b = 41/a$. This holds the mean to 41 minutes, the mean ground time in the original experiment described above. The variance is not constant, for as a rises the variance falls ($\sigma^2 = 41^2/a$).

In Figure 9 the relative advantages of the optimal update procedures decline quickly as a rises, the distribution becomes more symmetric, and the variance falls. In general, the optimal updating procedures are most effective when they limit the impact of a high prior variance and a long right tail.

One advantage of the dynamic programming procedure is that it allows managers to observe the trade-off between costs associated with forecast errors and costs associated with forecast updates. Figure 10 illustrates this trade-off for one-time forecasts (heuristic Step 1) and optimal sequential forecasts (heuristic Step 2). An analyst may use this curve to determine the update cost which produces a desired level of forecast accuracy for a reasonable number of expected updates. The graph also demonstrates that if optimal sequential forecasts are used, the forecast error cost is relatively insensitive to the number of updates for this distribution of g.

Figure 9 **Expected Total Cost Reduction from the Constant Interval Procedure as *a* Varies (Mean Held Constant at 41 minutes)**

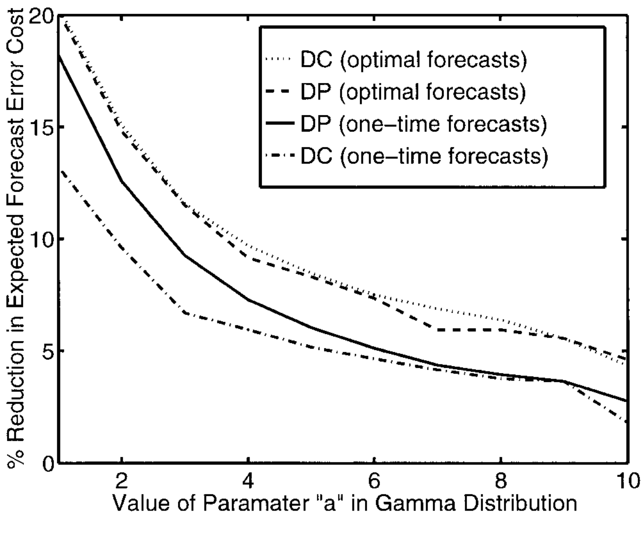
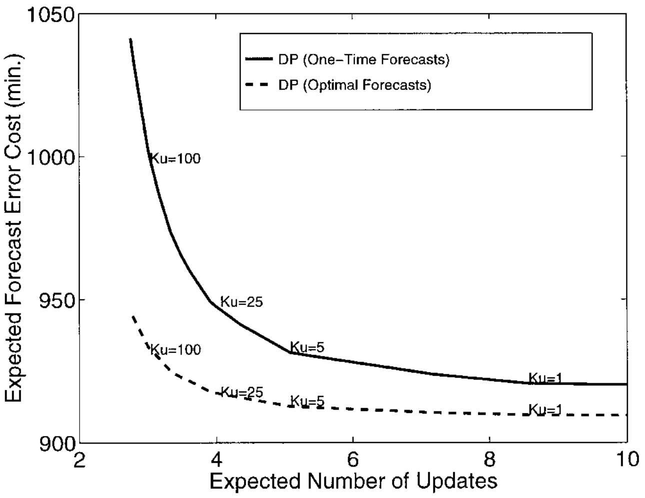


Figure 10 **Expected Number of Updates and Forecast Costs Produced by the DP over a Range of Update Costs (Parameters are** $a = 1.58$ **and** $b = 26.2$, $w(t) = \mu_1(0)$, $k_r = K_l = 1$**)**



These numerical experiments have assumed a particular cost function and prior distribution. Experiments with a normal prior distribution produce similar results (Shumsky 1995). Additional experiments find that the relative advantages of the updating procedures increase substantially when the cost function is assymetric, i.e., when $K_r \neq K_l$. Such a function would be reasonable if an unexpected, early arrival is more costly than an arrival after the predicted time.

6. Conclusions and Extensions

The traditional forecasting problem has been reformulated to allow for the generation of a sequence of forecasts rather than a single forecast. By specifying a weighting function $w(t)$, a manager can indicate when forecast accuracy is most important. Methods were developed for finding optimal sequential forecasts, given a fixed update schedule. A dynamic program found the optimal update schedule, given a particular forecast technique. In the numerical examples the procedures obtained forecast errors that were close to the lower bound established by continuously updating the forecast.

The empirical results demonstrate increasing advantages for the optimal updating procedures as the mean take-off time rises and the level of prior uncertainty grows. When using optimal sequential forecasts, there is little difference between the performances of the discrete conditional and dynamic programming procedures. Figure 8 shows that when prior uncertainty is high, discrete conditional updating requires significantly fewer updates than the other procedures.

After conducting these experiments, we recommended to the FAA that the simple discrete conditional procedure be adopted to find the update schedules for flights departing from most airports. This procedure achieves high forecast accuracy while using fewer updates than the current system. The optimization procedures should be considered at airports with frequent, major disruptions to aircraft departures.

The numerical examples also demonstrate how these procedures may be implemented efficiently for the tens of thousands of flights which depart each day. While update schedules may be costly to generate, the procedures can calculate schedules in advance for large numbers of flights. First, use historical data to derive prior take-off time distributions for groups of flights with similar characteristics. In the numerical example of the previous section, we applied this procedure to ATL departures in poor weather. Then derive a generic update schedule from the take-off time distribution. Finally, apply the schedule to a specific flight by shifting the schedule's origin to align with the flight's scheduled departure time.

Alternative procedures may be used to generate forecasts. For example, a sophisticated model of consumer behavior may be adopted to predict purchasing decisions. In general, the computational effort required to generate update schedules will depend on the complexity of the forecasting model and the number of distinct event categories, but once the schedules are generated and installed in a database the update procedures require little more than a single table lookup and a bit of addition. In the current air traffic management system, using the derived schedules will be only slightly more computationally intensive than the constant interval procedure, and the schedules will reduce the costs associated with distributing frequent updates.

The forecast updates presented in this article were driven by the simplest information, the fact that the event of interest had not yet occurred. This formulation reflects the capabilities of the current air traffic management system, but for many systems it is possible to update with more complex information. More general approaches would take multiple scenarios into account and incorporate many sources of data into the forecasts. For example, service facilities such as hospitals and hotels may revise a forecast of a customer's exit time using real-time information about the customer's status. Optimal updating procedures would use this information to efficiently produce accurate and timely forecasts.[2]

Appendix

Optimal Sequential Forecasts for a Given Update Schedule

We wish to minimize $M_k(h(a_k))$ over $h(a_k)$. To simplify the notation slightly, consider minimizing $M_k(h)$ over forecast h, and let the optimal solution be h^*. Assume that $h \geq a_k$, since h is the forecast of g, conditioned on $g \geq a_k$. The functional form of $M_k(h)$ changes at $h = a_{k+1}$, so there are two cases to evaluate:

Case (i): $(a_k \leq h < a_{k+1})$.

$$\begin{aligned} M_k(h) = & \int_{a_k}^{h} W(a_k, g_0)C_r(h - g_0)f_g(g_0)dg_0 \\ & + \int_{h}^{a_{k+1}} W(a_k, g_0)C_l(g_0 - h)f_g(g_0)dg_0 \\ & + W(a_k, a_{k+1}) \int_{a_{k+1}}^{\infty} C_l(g_0 - h)f_g(g_0)dg_0. \end{aligned} \tag{18}$$

Using Leibnitz's rule and the fact that both $C_r(0)$ and $C_l(0)$ vanish at the origin,

$$\begin{aligned} M_k'(h) = & \int_{a_k}^{h} W(a_k, g_0)C_r'(h - g_0)f_g(g_0)dg_0 \\ & - \int_{h}^{a_{k+1}} W(a_k, g_0)C_l'(g_0 - h)f_g(g_0)dg_0 \\ & - W(a_k, a_{k+1}) \int_{a_{k+1}}^{\infty} C_l'(g_0 - h)f_g(g_0)dg_0. \end{aligned} \tag{19}$$

Since $M_k'(a_k) < 0$, there is no relative minimum at a_k. Since both $C_r(0)$ and $C_l(0)$ are strictly convex, $M_k''(h) > 0$ for $a_k \leq h < a_{k+1}$ and there is either a unique minimum for some h_1^*, $a_k \leq h_1^* < a_{k+1}$, or there is a local minimum as h approaches a_{k+1}.

Case (ii): $h \geq a_{k+1}$.

[2] The author is grateful to Arnold Barnett, Amedeo Odoni, and the anonymous referees for their helpful suggestions. The research was supported by a grant from the Federal Aviation Administration.

$$M_k(h) = \int_{a_k}^{a_{k+1}} W(a_k, g_0)C_r(h - g_0)f_g(g_0)dg_0 + W(a_k, a_{k+1}) \left\{ \int_{a_{k+1}}^{h} C_r(h - g_0)f_g(g_0)dg_0 + \int_{h}^{\infty} C_l(g_0 - h)f_g(g_0)dg_0 \right\}, \tag{20}$$

$$M_k'(h) = \int_{a_k}^{a_{k+1}} W(a_k, g_0)C_r'(h - g_0)f_g(g_0)dg_0 + W(a_k, a_{k+1}) \left\{ \int_{a_{k+1}}^{h} C_r'(h - g_0)f_g(g_0)dg_0 - \int_{h}^{\infty} C_l'(g_0 - h)f_g(g_0)dg_0 \right\}. \tag{21}$$

Since $M_k''(h) > 0$ for $h \ge a_{k+1}$, there is either a unique minimum for some $h_2^* \ge a_{k+1}$ or a local minimum at a_{k+1}.

In order to find the optimal solution, h^*, note that if $M_k'(a_{k+1}) > 0$, then $h_1^* < h_2^*$ and therefore $h^* = h_1^*$. If $M_k'(a_{k+1}) \le 0$, then $h_1^* \ge h_2^*$ and $h^* = h_2^*$. The solution is unique, and the function $M_k(h)$ is unimodal under both cases (i) and (ii), so numerical search techniques may efficiently find the optimal solution if no exact solution can be derived.

References

Barlow, R. E., F. Proschan, and L. Hunter, ‘‘Optimum Checking Procedures,’’ *SIAM*, 11 (1963), 1078–1095.

Bonvik, A. M., Estimating the Lead Time Distribution of Priority Lots in a Semiconductor Factory, Working Paper OR 293-94, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, May 1994.

Goranson, J., Looking for Trouble: How Well the FAA’s Enhanced Traffic Management System Predicts Aircraft Congestion, Master’s Thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, August 1993.

Parmigian, G., ‘‘On Optimal Screening Ages,’’ *J. American Statistical Association*, 88 (1993), 622–628.

Seidmann, A. and M. L. Smith, ‘‘Due Date Assignment for Production Systems,’’ *Management Sci*., 27 (1981), 571–581.

Sengupta, B., ‘‘Inspection Procedures when Failure Symptoms are Delayed,’’ *Oper. Res*., 28 (1980), 768–776.

Shumsky, R. A., Dynamic Statistical Models for the Prediction of Aircraft Take-off Times, Ph.D. Thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, June 1995.

Zelen, M., ‘‘Optimal Scheduling of Examinations for the Early Detection of Disease,’’ *Biometrika*, 80 (1993), 279–293.

Accepted by Edward H. Kaplan; received August 1995. *This paper has been with the author* $5\frac{1}{2}$ *months for* 2 *revisions.*