Working Paper No. RHS-06-034

December 2005

Estimating Flight Departure Delay Distributions —A Statistical Approach With Long-term Trend and Short-term Pattern

Yufeng Tu

Michael Ball

Wolfgang Jank

Estimating Flight Departure Delay Distributions —A Statistical Approach With Long-term Trend and Short-term Pattern

Yufeng Tu†‡, Michael Ball†‡, Wolfgang Jank†*

†Department of Decision and Information Technologies
Robert H. Smith School of Business
and
‡The Institute for Systems Research

University of Maryland
College Park, MD, 20742

May 12, 2006

Abstract

In this paper, we develop a model for estimating flight departure delay distributions required by air traffic congestion prediction models. We identify and study major factors influencing flight departure delays, and develop a strategic departure delay prediction model. This model employs nonparametric methods for daily and seasonal trends. In addition, the model uses a mixture distribution to estimate the residual errors. In order to overcome problems with local optima in the mixture distribution, we develop a global optimization version of the Expectation Maximization algorithm, borrowing ideas from Genetic Algorithms. The model demonstrates reasonable goodness of fit, robustness to the choice of the model parameters, and good predictive capabilities. We use flight data from United Airlines and Denver International Airport from the years 2000/01 to train and validate our model.

Keywords: Smoothing spline, mixture model, Expectation Maximization (EM), Genetic Algorithm (GA), airline delay, airspace congestion, delay distribution.

*Corresponding author.

1 Introduction

The U.S. National Airspace System (NAS) is inherently highly stochastic. Yet, many existing decision support tools for air traffic flow management take a deterministic approach to problem solving. For example, to predict when an airspace sector will become overloaded, the Federal Aviation Administration (FAA) employs a module called *Monitor Alert*. This tool predicts airspace traffic levels by projecting, for each planned flight, time/space epochs through the airspace based on a single flight plan (route) and a single estimated departure time. The estimated departure time used is typically the flight's *scheduled* departure time. This deterministic approach fails to capture three important stochastic factors: i) the uncertainty in a flight's departure time (including the possibility of flight cancellation), ii) changes in a flight's route immediately before takeoff or after the flight is airborne, and iii) airspace queueing effects. On-going research and development efforts are seeking to develop stochastic models to replace this deterministic system (see Chandran (2002) for preliminary work and Wanke et al. (2005) for an alternate approach). This paper represents one component of these research efforts that addresses factor i). That is, in this paper we describe a model for estimating flight departure delay distributions. We emphasize that a major objective is to produce not just point estimates but estimates of the entire distribution since the congestion estimation models envisioned require delay distribution functions, e.g. to produce expected traffic levels for arbitrary time intervals. It is perhaps unnecessary to emphasize the potential benefits of reducing airspace congestion and delays. As an example, delays directly attributed to air traffic control actions are estimated to cost airlines 2.9 billion dollars in 1998 in addition to the cost of delays borne by passengers (ATA, 1999).

The Bureau of Transportation Statistics (BTS) releases summary statistics and basic analysis on airline performance each month. Most of its delay analysis focuses on arrival delays rather than the departure delays since arrival delays are more closely related to ultimate passenger satisfaction. On the other hand, when trying to understand the source of arrival delays and airspace congestion in general, study of departure delays becomes quite relevant. We should also note that the BTS analysis and most prior studies of airspace delays typically only provide average delay statistics and do not focus on estimates of distribution functions. Probably the most typical approach to estimating distributions for aviation analysis involves the generation of histograms from historical data. In Inniss and Ball (2004), such an approach is used to estimate airport departure capacity distributions. The estimates vary by hypothetical "seasons", which are determined through an optimization model. This approach to characterizing seasonal variation jumps from one estimated distribution to another at discrete points in time. The approach developed in this paper employs smoothing methods to allow for continuous variations in estimates over time, which is much more consistent with the underlying physical system. SimAir, a modular airline simulation tool developed in the year 2000, employs raw historical aggregate distributions (Rosenberger et al., 2000). Although raw historical distributions are a simple way to capture departure delays, they can potentially be too sensitive to specific random variation in the data. In our analysis, we attempt to separate random variation from observable patterns in the data. Specifically, we characterize the underlying mechanisms behind delay, then model and regenerate delay using functional characterizations. In that sense, our method could be used as input into simulation tools such as SimAir. One of the distinctive features of our model is the characterization of seasonal and daily delay patterns. This is one aspect in which it distinguishes itself from other work on modeling delay distributions (e.g. Mueller and Chatterji, 2002). Also, we consider a flexible continuous probability model for the error distribution while Mueller and Chatterji (2002) assume a discrete Poisson model. While the authors consider data across several different airports and airlines, we focus here on one particular airport/airline combination, and a longer time span, with the goal of extracting airport/airline specific patterns. We want to point out that our method is flexible and can be readily adapted to other airline/airport combinations.

The specific delay value we consider is the push-back delay, which measures the discrepancy between the scheduled departure time and the *actual* departure time from the gate (push-back time). Other delays, such as taxi-out delay, delay in the air, taxi-in delay, and arrival delay, are all generated after the flight leaves the gate. There is a body of related, prior research that uses models to estimate departure delays or employs departure delay estimates within broader models. These models typically address problems involving airport surface congestion. For example, Odoni et al. (1994) develop a non-homogeneous queueing model to analyze congested airports. Shumsky (1997) extends this model and estimate take-off times under non-steady state conditions. Idris et al. (2002) develop a queueing model for taxi-out time estimation. The result of our work could potentially be used as inputs into any of these models.

A key component of our model is the estimation of the delay propagation effect. Delay built-up from previous flights is known as the delay propagation and its effects on delays have been studied in several prior papers (see for example Beatty (1998), Schaefer and Millner (2001) and Wang et al. (2003)). Our work provides a functional characterization of this effect at a single airport and uses the underlying function as input into departure delay estimates.

In addition to the daily propagation effect, many other factors influence departure delay, such as weather conditions, holiday demand surges, luggage problems, mechanical problems, airline policies, airport congestion, etc. Instead of studying the impact of each individual factor alone, we group factors into three major categories: seasonal trend, daily propagation pattern and random residuals. Our model uses each of these three categories as an individual building block. To estimate the seasonal trend and the daily propagation pattern, we employ a smoothing spline model. Its nonparametric nature eliminates the need for assuming a rigid (and possibly incorrect) form for the dependence of response and predictors (Hastie and Tibshirani, 1990). In our analysis, we do not have prior knowledge of the form of the seasonal trend nor of the daily propagation pattern. In addition, by using a smoothing spline, we can treat time as a continuous factor, which is appropriate since the delay at the end of one month will not vary significantly from the delay at the beginning of the next month; there is a similar smooth fluctuation in delay over the course of one entire day, making the smoothing spline also an advantageous approach for addressing the daily propagation effect. Finally, we assume a mixture model for the residuals and estimate the mixture-components using the EM (Expectation Maximization) algorithm. The EM algorithm is known for its fast convergence, stability and convenient implementation in mixture problems (Bilmes, 1998). One drawback of EM is that it typically converges only to a *local* optimum of the likelihood function. The mixture model likelihood, however, is known to have many local, sub-optimal solutions, especially when the data-dimensionality and/or mixture-number are large (McLachlan and Peel, 2000). This means that EM can get trapped in a solution far away from the *global optimum* (see e.g. Jank, 2006a,b).

In an effort to find the global optimum, we develop a global optimization version of EM by combining EM with the ideas of the Genetic Algorithm (GA). GAs were first introduced by Holland (1975) based on the principles of natural selection or "survival of the fittest" in the evolution of species. The GA approach has been applied to many areas including marketing, biology, engineering, etc. In this paper, we use the principles of GA to overcome local maxima in mixture distributions within the framework of the EM algorithm. We want to point out that there exists, to date, only little research on making EM suitable for solving global optimization problems. Some very recent efforts into that direction include Heath et al. (2006), Jank (2006a) or Pernkopf and Bouchaffra (2005).

To illustrate the performance of our model, we select Denver International Airport, a hub for United Airlines (UA), as our case study. Our model shows promising results for estimation and prediction of departure delays. Although the case study is for Denver International Airport and UA only, our model can be readily generalized to other airports and airlines as well.

The paper is organized as follows. Section 2 introduces the model structure and assumptions. Section 3 proposes a Genetic Algorithm version of the EM algorithm. In Section 4, we present the case study, describe our data and discuss computational results including model robustness and validation. Section 5 describes possible application of our work within the context of air traffic management. In that section we also describe a way of dynamically updating our model in real-time as new delay information becomes available. Section 6 summarizes our findings and points out areas for further research.

2 The Model Structure

Our model takes into account two types of delay structures: seasonal trend and daily propagation pattern. Every day, delay builds up according to the daily propagation pattern while, at the same time, it pursues a seasonal trend throughout the year. Random residuals capture the additional variation not accounted for by these two structures (see Figure 1).

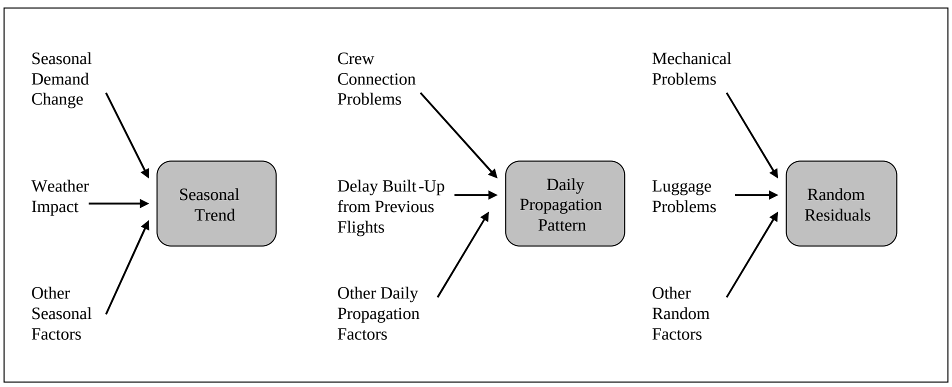

Figure 1: *Factors Influencing Departure Delay*

Instead of attempting to explicitly account for all the different factors depicted on the left hand side of the arrows in Figure 1, we use the much simpler structures on the right hand side. Therefore, the departure delay for each individual flight can be decomposed into three major parts: a main effect due to seasonal variation, a main effect due to daily delay propagation, plus random errors.

The model formulation is thus as follows: Let $y_i(s,t)$ be the departure delay for flight i scheduled to depart on day s at time t. Let $f(s)$ be the seasonal trend, $\varphi(t)$ be the daily delay pattern, and ϵ_i denote the random error. We propose an additive model of the form

$$y_i(s,t) = f(s) + \varphi(t) + \epsilon_i \tag{1}$$

where the seasonal trend is a function of only day s and the daily delay pattern is function of only time t. We further assume that the random error is independent of both s and t. In that sense, $y_i(s,t)$ denotes the delay of any flight scheduled at day s and time t; if i and i' were two flights scheduled at the *same* day and time, then their only delay-difference would be due to random error ϵ_i.

Note that in this model we assume the effects of season and day to be additive. That is, controlling for the seasonal trend in the data, one day does not impact another. Moreover, controlling for season and day, the residuals are iid (identically & independently distributed). While this model may appear simplistic, our results show high predictive accuracy. In addition, the simplicity of the model allows for easy implementation, maintenance and updating, and results in robustness with respect to the choice of the model parameters.

2.1 Seasonal Trend

We model the seasonal trend using smoothing splines. This nonparametric approach allows us to trace the seasonal trend without assuming a rigid (and possibly incorrect) functional form for the dependence of response and predictors. Smoothing splines are also known to provide good fit to the data without exhibiting excessive local variability (Green and Silverman, 1994).

Let $\Pi = \{\pi_1, \ldots, \pi_V\}$ be a set of knots (i.e. the break points of the piecewise-defined spline), then a polynomial spline of order d is given by

$$f(s) = \beta_0 + \beta_1 s + \beta_2 s^2 + \cdots + \beta_d s^d + \sum_{v=1}^{V} \beta_{dv}(s - \pi_v)_+^d \tag{2}$$

where $a_+ = aI_{[a \geq 0]}$ denotes the positive part of the function a. Let $\boldsymbol{\beta}= (\beta_0, ...\beta_d,\ \beta_{d1}, ..., \beta_{dV})'$ be the vector of coefficients in (2). The choice of V and d strongly influences the local variability of the function f. One can measure the degree of departure from a straight line by defining a roughness penalty

$$PEN_m = \int (D^m f(s))^2 ds \tag{3}$$

where D^m, $m = 1, 2, ...$, denotes the mth derivative of the function f. Using $m = 2$ and $d = 3$ leads to the popular cubic smoothing spline. We find $f(s)$ by minimizing the penalized residual sum of squares (PENSSE):

$$PENSSE_{m=2} = \sum_{s=1}^{365} (\bar{y}_s - f(s))^2 + \lambda_S \int_1^{365} (f''(s))^2 ds, \tag{4}$$

where λ_S is the smoothing parameter. (The subscript S distinguishes it from the subsequent smoothing parameter for the daily propagation pattern λ_D.) $\bar{y}_s$ denotes the average daily delay and is calculated via

$$\bar{y}_s = \frac{\sum_i \sum_t y_i(s,t)}{\sum_t n_{st}} \qquad s = 1, 2, 3, \ldots, 365, \tag{5}$$

where n_{st} refers to the number of flights on day s at time t.

The parameter λ_S controls the smoothness of the spline. Large values of λ_S produce smoother curves while smaller values produce locally more variable curves. In our study, we balance data-fit and smoothness by choosing an equilibrium value for λ_S (see Section 4.3). As to the number and placement of the knots π_v, we set them to the unique values of $\bar{y}_s$ (e.g. Reinsch, 1967; de Boor, 1978).

2.2 Daily Propagation Pattern

Since the airline operating resources are linked together, delaying one flight can affect other flights. Among the inter-connected resources affected by delayed flight operations are crews, aircrafts, passengers, and gate spaces. Because of this connectivity, airline departures are quite sensitive to delays earlier in the day—the delay of one flight tends to propagate in time to many others.

The same smoothing approach as earlier is employed to model the daily propagation pattern. We define the daily propagation function $\varphi(t)$ to be one that minimizes the penalized residual sum of squares:

$$PENSSE_{m=2} = \sum_{t=00:00}^{24:00} (\bar{y}_t - \varphi(t))^2 + \lambda_D \int_{00:00}^{24:00} (\varphi''(t))^2 dt \tag{6}$$

where λ_D is again the smoothing parameter and $\bar{y}_t$ denotes the average desesonalized delay at time t. We calculate $\bar{y}_t$ as follows. Let $y_i^{'}(s,t)$ denote the delay after removing the seasonal trend,

$$y_i^{'}(s,t) = y_i(s,t) - \hat{f}(s) \qquad \forall s, t, i \tag{7}$$

Then, we calculate $\bar{y}_t$ as

$$\bar{y}_t = \frac{\sum_i \sum_{s=1}^{365} \sum_t^{t+T} y_i^{'}(s,t)}{\sum_{s=1}^{365} \sum_t^{t+T} n_{st}} \qquad t = 00:00, T, 2T, \ldots, 24:00, \tag{8}$$

where T denotes a very short time interval ($T = 5$ minutes in our study). We choose λ_D and π_v in a similar manner as before (see also Section 4.5).

2.3 Finite Mixture Distribution for Residuals

The residuals are defined as the errors remaining after accounting for seasonal trend and daily propagation delay. Residuals originate from many unpredictable factors such as customers running late, mechanical problems, extreme weather conditions, etc. To capture the residual delay distribution, we employ a finite mixture model with several components. Many of the underlying mechanism of delay suggest the use of an error model comprised of different components: A few flights depart earlier than the scheduled departure time; this calls for a component that accounts for early-departers. Another component may account for the majority of flights; i.e. the majority of flights that depart right around the scheduled time. And yet, there may be another component (or two) that account for those flights having extremely long delays.

We thus model the residual distribution as a function of a J-component mixture in $\Re^1$. The random residuals ϵ_i are calculated by removing the daily propagation pattern and the seasonal trend from the original data,

$$\epsilon_i = y_i(s,t) - \hat{f}(s) - \hat{\varphi}(t). \tag{9}$$

The mixture density of the ith residual ($i = 1, ..., n$) is then given by

$$g(\epsilon_i|\boldsymbol{\theta}) = \sum_{j=1}^{J} p_j \psi_j(\epsilon_i|\boldsymbol{\alpha}_j) \tag{10}$$

where p_j ($p_j \in [0,1], \sum_{j=1}^{J} p_j = 1$) is the mixing proportion and $\psi_j(\boldsymbol{\epsilon}|\boldsymbol{\alpha}_j)$ is a density function in the parameter $\boldsymbol{\alpha}_j$. Collecting all parameters into one vector, we write $\boldsymbol{\theta} = (p_1, ..., p_J, \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_J)$. Moreover, assuming normal group-conditional densities we can write

$$\psi_j(\epsilon_i|\boldsymbol{\alpha}_j) = \psi_j(\epsilon_i|\mu_j, \sigma_j), \boldsymbol{\alpha}_j = (\mu_j, \sigma_j) \tag{11}$$

where μ denotes the mean and σ denotes the variance, respectively. The log-likelihood is then

$$\log L(\boldsymbol{\theta}|\boldsymbol{\epsilon}) = \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{J} p_j \psi_j(\epsilon_i|\boldsymbol{\alpha}_j) \right\}. \tag{12}$$

One can maximize above log-likelihood by appealing to the missing information principle which makes the mixture likelihood very appealing for the use of the EM algorithm. Specifically, we assume that ϵ_i arises from one of the J groups. Let $\boldsymbol{z}_i = (z_{i1}, ..., z_{iJ})$ be the corresponding J-dimensional group indicator vector; that is, $z_{ij} = 1$ if and only if ϵ_i belongs to group j; otherwise it equals zero. Notice that $\boldsymbol{z}_i$ is unobserved (or missing). By writing $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_n)$ for the observed data and $\boldsymbol{Z} = (\boldsymbol{z}_1, ..., \boldsymbol{z}_n)$ for the unobserved data we get the complete data as $\boldsymbol{\Omega} = (\boldsymbol{\epsilon}, \boldsymbol{Z})$. The log-likelihood of the complete data can then be written as

$$\log L_c(\boldsymbol{\theta}|\boldsymbol{\Omega}) = \sum_{i=1}^{n} \sum_{j=1}^{J} z_{ij} \left\{ \log p_j + \log \psi_j(\epsilon_i|\boldsymbol{\alpha}_j) \right\}. \tag{13}$$

2.4 Mixtures and Local Optima

One of the biggest challenges for the EM algorithm is that it only guarantees convergence to a *local* solution. The EM algorithm is a greedy method in the sense that it is attracted to the locally optimal solution closest to its starting value which can be a problem when several locally optimal solutions exist. This problem frequently occurs in the mixture model.

Consider Figure 2. The top panel of Figure 2 shows 40 observations, $y_1, \ldots, y_{40}$, simulated according to a mixture of two univariate normal distributions, $y_i \overset{iid}{\sim} [p_1 N(\mu_1, \sigma_1^2) + p_2 N(\mu_2, \sigma_2^2)]$, with $p_1 = p_2 = 0.5$, $\mu_1 = -1$, $\mu_2 = 2$, $\sigma_1^2 = 0.001$ and $\sigma_2^2 = 0.5$. Notice that this is a special case

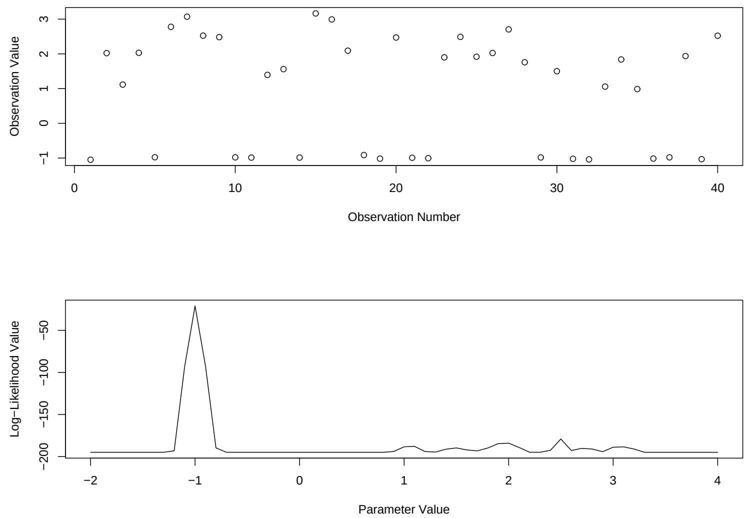

Figure 2: Log-likelihood function for a simple two-component mixture problem. The top panel shows the simulated data. The bottom panel shows the log-likelihood function for μ_1, the mean of the first likelihood component, holding all other parameters constant at their true values.

of the normal mixture model in (10) with $J = 2$. Notice also that the first mixture component has almost all its mass centered around its mean $\mu_1 = -1$. This results in a log-likelihood for μ_1 depicted in the bottom panel of Figure 2. We can see that, as expected, the global optimum of this log-likelihood is achieved at $\mu_1 = -1$. However, we can also see at least five local optima, located around the values $\mu_1 = 1, 1.5, 2, 2.5$ and 3. Clearly, depending on where we start EM, it may be trapped very far away from the global (and true) parameter value. In the following, we propose a new version of EM that, by borrowing ideas from the Genetic Algorithm, can overcome this problem.

3 A Genetic Algorithm Version of EM

The EM algorithm is an iterative procedure which alternates between two steps: an E-step and an M-step. The E-step computes the conditional expectation of the complete data log likelihood, conditional on the observed data (flight departure delays in our case) and the current parameter values. Let

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k-1)}) = E[\log L_c(\boldsymbol{\theta}|\boldsymbol{\Omega})|\boldsymbol{\epsilon}; \boldsymbol{\theta}^{(k-1)}] \quad (14)$$

where k denotes the kth iteration. Using equation (13), (14) can be simplified to

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k-1)}) = \sum_{i=1}^{n}\sum_{j=1}^{J} \eta_{ij}^{(k-1)} \left\{\log p_j + \log \psi_j(\epsilon_i|\boldsymbol{\alpha}_j)\right\} \tag{15}$$

where $\eta_{ij}^{(k-1)} = E(z_{ij}|\epsilon_i;\boldsymbol{\theta}^{(k-1)})$ is the posterior probability that ϵ_i belongs to the jth component in the mixture. The M-step maximizes $Q(\cdot|\boldsymbol{\theta}^{(k-1)})$. That is, the kth M-step finds the value $\boldsymbol{\theta}^{(k)}$ which satisfies

$$Q(\boldsymbol{\theta}^{(k)}|\boldsymbol{\theta}^{(k-1)}) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k-1)}) \tag{16}$$

for all $\boldsymbol{\theta}$ in the parameter space.

One appeal of assuming a normal mixture distribution (11) is that we obtain closed-form updates for the E- and M-steps (McLachlan and Peel, 2000):

- E-step : For $i = 1, ..., n$ and $j = 1, ..., J$ we compute

$$\eta_{ij}(\boldsymbol{\theta}^{(k)}) = \frac{p_j^{(k)}\psi(\epsilon_i|\mu_j^{(k)}, \sigma_j^{(k)})}{\sum_{j=1}^{J} p_j^{(k)}\psi(\epsilon_i|\mu_j^{(k)}, \sigma_j^{(k)})}. \tag{17}$$

- M-step : Write $\boldsymbol{\theta}^{(k+1)} = (p_1^{(k+1)}, ..., p_J^{(k+1)}, \mu_1^{(k+1)}, ..., \mu_J^{(k+1)}, \sigma_1^{(k+1)}, ..., \sigma_J^{(k+1)})$ for the parameter update where its components are given by

$$\begin{aligned} p_j^{(k+1)} &= \frac{1}{n}\sum_{i=1}^{n} \eta_{ij}(\boldsymbol{\theta}^k) && (18)\\ \mu_j^{(k+1)} &= \frac{\sum_{i=1}^{n} \eta_{ij}(\boldsymbol{\theta}^k)\epsilon_i}{\sum_{i=1}^{n} \eta_{ij}(\boldsymbol{\theta}^k)} && (19)\\ \sigma_j^{(k+1)} &= \frac{\sum_{i=1}^{n} \eta_{ij}(\boldsymbol{\theta}^k)(\epsilon_i - \mu_j^{(k+1)})(\epsilon_i - \mu_j^{(k+1)})^T}{\sum_{i=1}^{n} \eta_{ij}(\boldsymbol{\theta}^k)}. && (20) \end{aligned}$$

The E-step and M-step are repeated until convergence. Convergence is often assessed by monitoring the improvements in the parameter estimates and/or the improvements in the log-likelihood function.

As pointed out earlier, one of the biggest challenges for EM is that it only guarantees convergence to a local optimum and thus, especially in the mixture model, can get trapped in a sub-optimal solution, possibly far away from the global optimum. In the following we propose a new variant of EM that can overcome this challenge. To do so, we borrow ideas from the literature on global optimization and in particular from the Genetic Algorithm.

The Genetic Algorithm (GA) was first proposed by Holland (1975). It has been applied to many functional areas including marketing, biology, and engineering (Goldberg, 1989). The basis for the algorithm comes from the observation that a combination of sexual reproduction and natural selection allows nature to develop living species that are highly adaptive to the natural environment. In the following we propose a Genetic Algorithm version of EM. Our algorithm shares similarities with other efforts on the same topic (Heath et al., 2006; Jank, 2006a; Pernkopf and Bouchaffra, 2005).

The Genetic Algorithm begins with an initial population of chromosomes. One evaluates their structure and allocates reproductive opportunities in such a way that those chromosomes which represent better solutions to the target problem are given a better chance to produce offspring. The expectation is that some members of the resulting offspring population acquire the best characteristics of both parents and, as a consequence, can better adapt to the environmental conditions, providing an improved solution to the problem.

For this problem we can think of each parameter-component $p_1, ..., p_J, \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_J$ as one individual gene. Then the vector $\boldsymbol{\theta} = (p_1, ..., p_J, \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_J)$ is a string of parameters just as a chromosome consists of a string of genes. The fitness function is the likelihood function (12). The resulting EM-based Genetic Algorithm can then be implemented as follows:

Step1 Initialization: Randomly generate an initial population of l chromosomes, which serves as the pool of parents. Initial parent pool $= \left\{\boldsymbol{\theta}_1^p, ...\boldsymbol{\theta}_l^p\right\}$.

Step2 Evaluation: Evaluate the fitness of each chromosome by calculating $\max_{\boldsymbol{\theta}}\{\log L(\boldsymbol{\theta}|\boldsymbol{\epsilon})\}$ in (12) via the EM algorithm using $\boldsymbol{\theta}_k^p, k = 1, \ldots, l$, as the starting value. Record the corresponding maximum likelihood value $\mathrm{MLK}^p = \left\{\mathrm{MLK}_1^p, \ldots, \mathrm{MLK}_l^p\right\}$

Step3 Crossover: Randomly choose a pair of parents $\boldsymbol{\theta}_k^p$ and $\boldsymbol{\theta}_{k'}^p$ from the initial pool, and exchange their genes at random positions to generate a pair of children. Specifically, crossover the p_j's or $\boldsymbol{\alpha}_j$'s between two parents randomly. Repeat this step until we get l children. Children pool $= \{\boldsymbol{\theta}_1^c, ...\boldsymbol{\theta}_l^c\}$.

Step4 Mutation: Specify a fixed and small probability of mutation p_m. Draw a random number between 0 and 1; if that number is smaller than p_m, then the new child chromosome is randomly mutated, which means p_j or $\boldsymbol{\alpha}_j$ are changed at random.

Step5 Update: Take the fitness of all parents $\mathrm{MLK}^p = \left\{\mathrm{MLK}_1^p, ...\mathrm{MLK}_l^p\right\}$. Similarly, compute and record the fitness of all children $\mathrm{MLK}^c = \{\mathrm{MLK}_1^c, ..., \mathrm{MLK}_l^c\}$. Choose the best l chromosomes from the combined parents and children to remain in the gene pool. Update MLK from $\{\mathrm{MLK}^p \cup \mathrm{MLK}^c\}$; update the gene pool correspondingly.

Step6 Iteration: Repeat Step 2-4 until the Nth generation is produced. N is typically a number fixed in advance.

We refer to our Genetic Algorithm version of EM as the GA-EM algorithm. Practical implementation of a GA-EM requires the selection of several algorithm parameters such as the population size l, the number of generations N, and the mutation rate p_m. In our application, we chose these parameters as $l = 100$, $N = 100$ and p_m= 1/(number of parameters+1) (see e.g. Willighagen, 2005). However, the algorithm performance is very robust to different choices (see Section 4.5).

4 Case Study

We select Denver International Airport and United Airlines for our case study. We train and validate our model on data from the year 2000. We also investigate its forecasting capabilities for data from 2001 (see Section 5). Notice that our method is general and can also be applied to data from other airports/airlines/time-ranges.

4.1 Data

The data used in this study is based on Airline Service Quality Performance (ASQP) data, which are collected by DOT (US Department of Transportation) under authority of 14 Code of Federal Regulations (CFR). Any airline with more than 1 percent of total domestic enplanements is required to report performance data to DOT.

In the year 2000, 10 carriers met the reporting requirement threshold. Among them, American, Northwest, United, and US Airways use ACARS (Aircraft Communications Addressing and Reporting System) exclusively; Continental, Delta, and Trans World Airlines use a combination of ACARS and manual reporting system; and America West, Southwest, and Alaska Airlines rely solely on their pilots, gate agents and/or ground crews to record arrival times manually (FAA, 2002).

We choose the year 2000 to avoid the September 11th terrorist attacks and their consequential impact on airline performance. We split our data into a training and a validation set: we estimate our model on 70% of the data; the remaining 30% are used for model validation.

4.2 Data Preparation

In the year 2000, a total of 92,865 UA flights departed from Denver International Airport, which equals an average of about 254 flights per day. Delay considered in this study is the pushback delay which measures the difference between the actual and scheduled departure time. Let t^i_{dep} denote the actual departure time and let t^i_{sch} be the scheduled departure time for flight i. Pushback delay $y_i(s,t)$ in equation (5) is defined as $y_i(s,t) := t^i_{dep} - t^i_{sch}$. Descriptive statistics for pushback delays

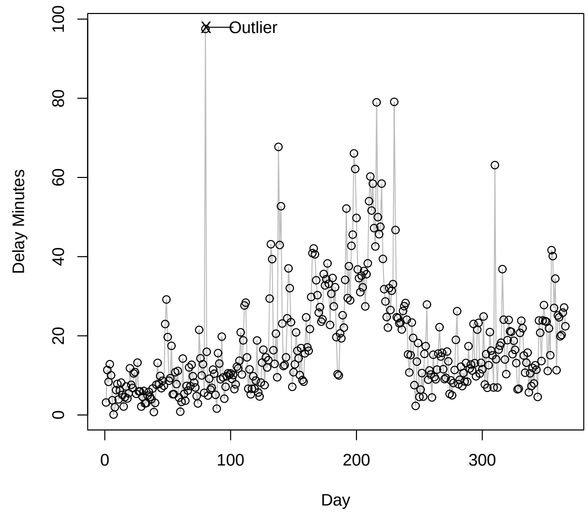

Figure 3: *Average Daily Delay in Year 2000*

Table 1: *Summary Statistics of the Pushback Delay (minutes)*

Min	1st Quartile	Median	Mean	3rd Quartile	Max	Std.
-18.00	-1.00	3.00	18.16	20.00	802.00	37.16

(in minutes) are shown in Table 1. We notice that the mean is much larger than the median, suggesting that delay is heavily right-skewed.

Figure 3 shows a time-series plot of average daily delay over the 366 day period under study. We identify an extreme value around observation 90 (March 20th). On that day, average delay is significantly larger than on any other day. The following excerpt from the NCAR (the National Center for Atmospheric Research) news release explains what happened on that particular day (see NCAR, 2002) :

> *Cancellations and delays due to icy weather can cost airlines millions of dollars in a single day. On March 20, 2000, icing conditions at Denver International Airport forced Air Wisconsin to cancel 152 flights. United canceled 159 outbound and 140 inbound flights the same day, most because of weather.*

March 20th is a special case with extreme icing condition. Politovich et al. (2002), describe the results of a survey sent out to pilots that flew in and out of Denver. On one of the question “Was March 20th an extremely unusual event for DEN?”, 23 out of 26 pilots answered *Yes*. Therefore we consider that observation an outlier and exclude March 20th from our study.

4.3 Estimating the Seasonal Trend

Note that the year 2000 has 366 days. Since we exclude March 20th, we remain 365 days in our dataset. A smoothing spline fit to these 365 daily delays is depicted in Figure 4(a). The vertical axis gives the average delay in minutes and the horizontal axis shows the day of the year. Delays are high in summer and winter but low in spring and fall, which suggests a strong seasonal pattern. The solid line corresponds to a cubic smoothing spline for the seasonal trend $\hat{f}(s)$, using $\lambda_S = 1.03$.

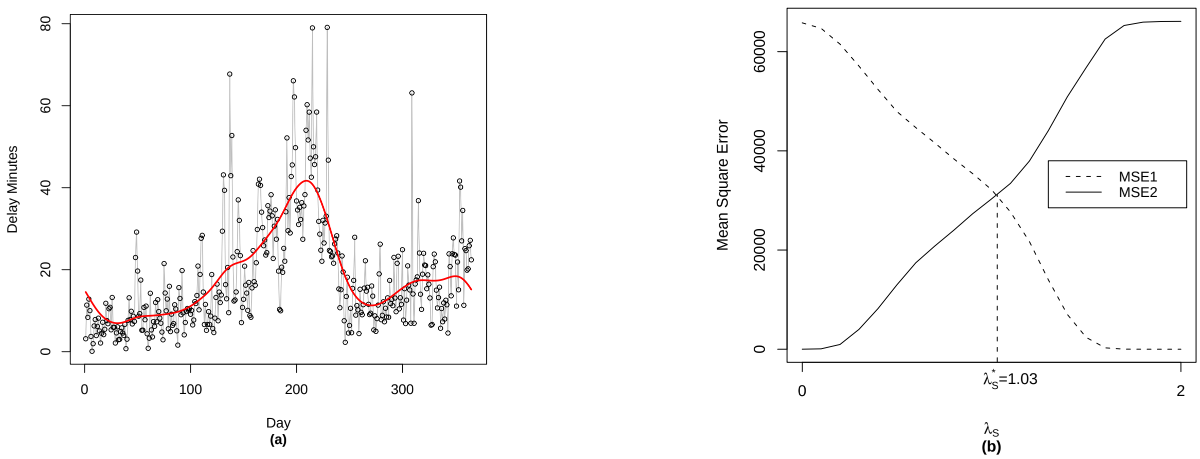

Figure 4: *Estimating the Seasonal Trend:* **(a)** *A fitted smoothing spline that represents the seasonal trend;* **(b)** *The compromise between goodness of fit and fluctuation for the smoothing parameter.*

Balancing data fit and smoothness, we choose λ_S in the following way. For different values of λ_S, we calculate the mean squared error (MSE) between the fitted spline and a simple straight-line regression through the data. We can think of this as a measure of local variation since a straight-line regression provides the smoothest data fit. We also calculate the MSE between the spline and the observed data as a measure of goodness of fit. Figure 4(b) shows the resulting two MSE measures as a function of different λ-values.

MSE1 measures local variation (or departure from smoothness); local variation decreases (i.e. smoothness increases) as λ_S increases. MSE2 measures data fit. As λ_S increases, MSE1 decreases (i.e. smoothness increases) while MSE2 increases (i.e. data fit decreases). Figure 4(b) shows that we achieve a good balance between local variation and data fit by choosing $\lambda_S = 1.03$ (i.e. the point where MSE1 and MSE2 intersect). We also explore a range of alternative values for λ_S in Section 4.6 and find that our model is very robust to changes in the smoothing parameter λ_S.

4.4 Estimating the Daily Propagation Pattern

After removing the seasonal trend, we use a similar smoothing approach for estimating the daily propagation pattern. Figure 5 shows the resulting smoothing spline $\hat{\varphi}(t)$. The horizontal axis corresponds to the scheduled departure time (from 00:00 to 24:00 calculated in minutes), and the vertical axis shows the delay in minutes. Note that no flight is scheduled to depart before 6:00 or after 24:00. As a result, the horizontal axis covers only part of an entire day. We can see that delay gradually builds up as the day goes on and decreases only deep into the night. The roughness penalty λ_D is set at 0.44 using a similar rationale as before (see Figure 5(b)).

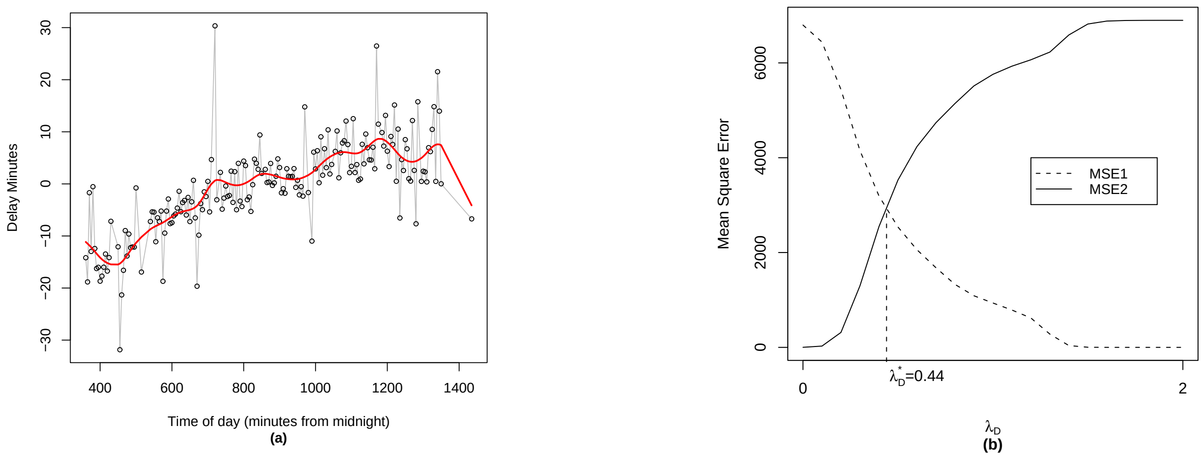

Figure 5: *Estimating the Daily Propagation Pattern:* **(a)** *A fitted smoothing spline that represents the daily propagation pattern;* **(b)** *The compromise between goodness of fit and fluctuation for the smoothing parameter.*

We want to point out that the daily propagation pattern in Figure 5 is not really “daily” in the true sense of the word. In fact, the propagation effect takes place over two consecutive days. The break point between two “days” is in the early morning hours around 5:00 am or 6:00 am, when the airport finally consumes all delays and no more flights depart.

Figure 6 shows the scatter of the average delay against the *actual* departure time. We notice a very distinct spiky pattern: delay increases sharply within constant time intervals and then drops at the interval-end. We can also see that the delay is extremely high in the very early morning. The reason for this is that the horizontal axis is the *actual* departure time. Since no flight is scheduled to depart in the very early morning hours, a flight that actually does depart at that time indicates a flight that has been delayed for an extremely long amount of time (i.e. from the previous day). When randomly sub-sampling 30% of the data, we notice that the pattern persists (Figure 6(b)). This suggests that, surprisingly, it does not depend on only a few extreme values.

Airline scheduling and National Air Space (NAS) queueing effects may contribute to the spiky pattern in Figure 6. When many flights are scheduled to depart in a very short time interval, limitations on the airport departure rate result in long queues. Figure 6(c) shows the distribution of flights scheduled to depart over the course of one day. Each bar corresponds to the number of flights scheduled within a 2-minute interval. We see several spikes above 1,500 (i.e. more than 1,500 aggregated flights were scheduled to depart during several 2-minute intervals). However, less than 800 flights actually did depart during these intervals (see Figure 6(d)). This difference between scheduled and actual departures translates into delay which propagates itself over the day.

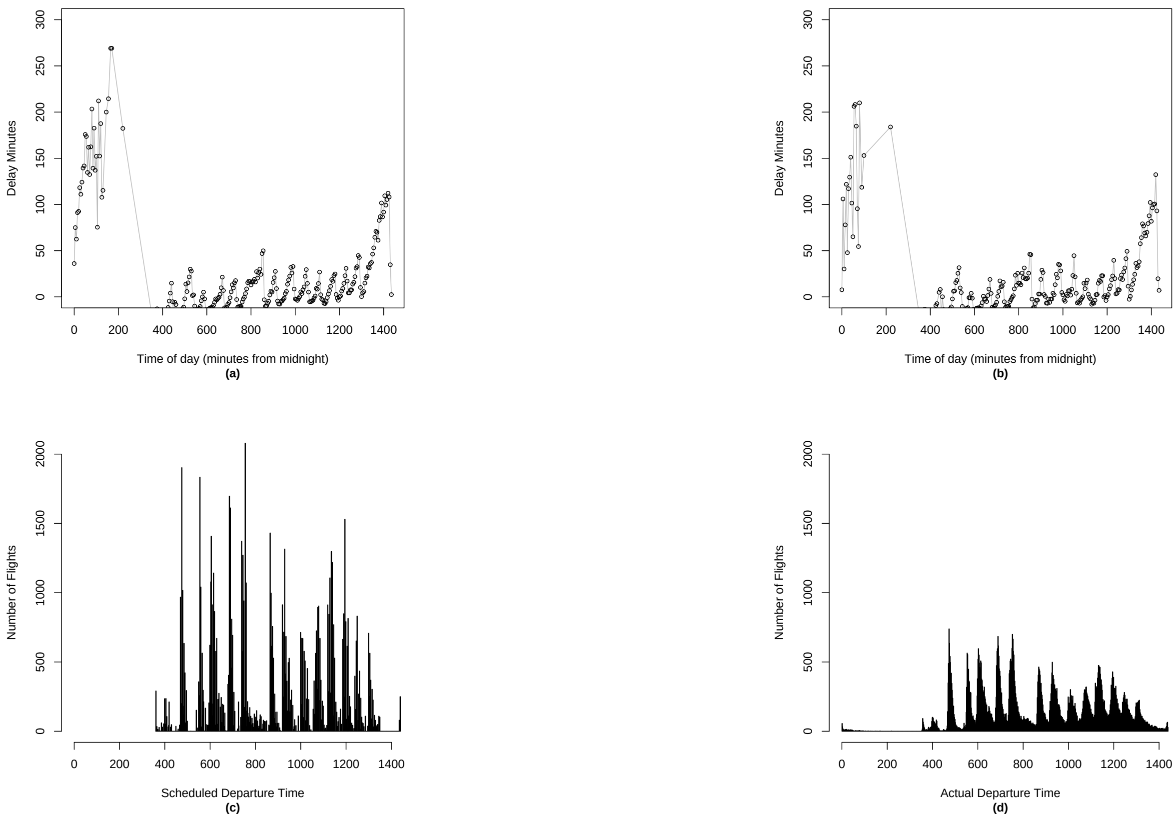

Figure 6: *Pattern in Delays vs Actual Departure Times:* **(a)** *Delay vs. actual departure time* **(b)** *Delay vs. actual departure time for a random sample of only 30% of the data* **(c)** *Distribution of number of flights scheduled to depart* **(d)** *Distribution of number of flights that actually departed*

Queueing effects and "flight banks" in scheduling are well known in airline studies. However, it is quite surprising to see the well shaped pattern in Figure 6 to persist even when we aggregate over the entire year since one may expect queueing delays on different days to cancel each other out.

4.5 Mixture Estimation using GA-EM

After removing both the seasonal trend and the daily propagation pattern, we estimate the mixture distribution for the residuals. As pointed out earlier, we use our Genetic Algorithm version of EM in the search for the global optimum.

We apply our GA-EM algorithm using $l = 100$ parents and $N = 100$ generations. Random

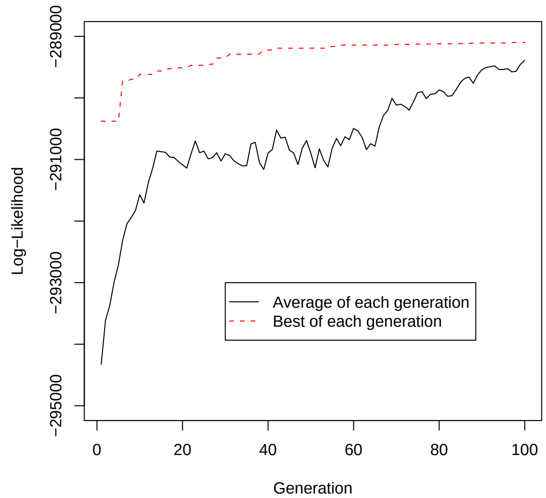

Figure 7: *Finding the Global Maximum via Genetic Algorithm*

starting values were generated to form the pool of parents/chromosomes. The mutation rate was set at p_m= 1/(number of parameters+1) (see e.g. Willighagen, 2005). This results in the generation-history shown in Figure 7: as more and more generations are calculated, the overall fitness improves. Moreover, the convergence rate is fast since both average fitness per generation (solid line) and best fitness per generation (dashed line) increase quickly and join (at least almost) at generation 100. The roughness of the average fitness stems from the fact that mutation inflicts shocks into the evolution process which may cause the method to temporarily seek worse solutions. In effect, this allows the method to overcome local solutions and, eventually, visit the global optimum.

The performance of the GA-EM algorithm may depend on the choice of the algorithm parameters l, N and p_m. Figure 8 shows the performance of the method when we vary these parameters. We can see that, regardless of the mutation rate, the population size or the number of generations, GA-EM converges to the same likelihood value after about 100 generations. We also investigated the method's dependence on its inherent randomness (e.g. due to the choice of the starting values etc.), and, similarly, found that the method converges to the same likelihood value after about 100 generations. We take this as evidence that 100 generations is a reasonable generation-size for this application.

The computing effort of our method is reasonable. Each EM-step takes about 0.25 seconds and it takes on average 10 iterations for EM to converge. Thus, one run of GA-EM with 100 parents and 100 generations takes about $0.25 \times 10 \times 100 \times 100 = 25,000$ seconds or 6.94 hours. This is the time-investment necessary for one data set. In practice, we may have to update the parameters

Table 2: *Values of the Parameters in Mixture Density Fitting*

	p_1, p_2, p_3, p_4	$\mu_1, \mu_2, \mu_3, \mu_4$	$\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2$
Solution	0.34,0.41,0.18,0.07	-17.05,-8.69,19.20,92.69	108.49,84.92,721.27,4184.54

occasionally because of newly arriving data. This can be done in a computationally efficient manner as we discuss in Section 5.

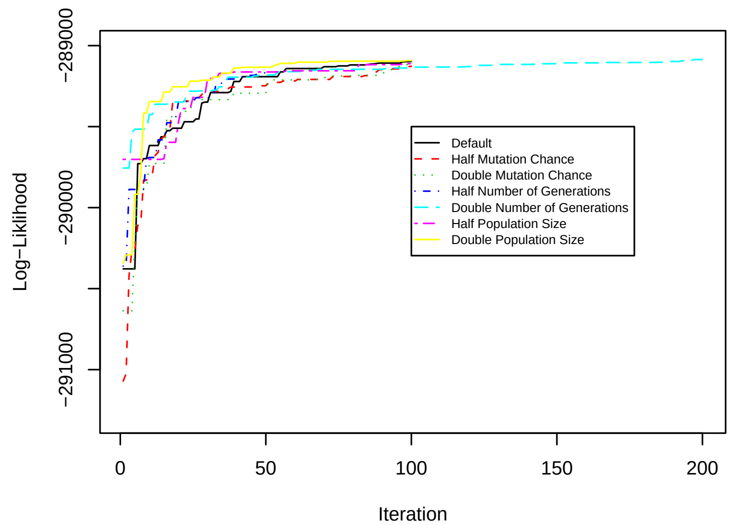

Figure 8: *Robustness of GA-EM to algorithm parameters*

One important decision in mixture-modeling is to choose the number of mixture components J. As J increases we typically get a better data fit, however we also run the risk of over-fitting. Moreover, model-parsimony considerations suggest the lowest possible value of J. From a global optimization point of view, the optimization problem becomes harder with increasing J since the solution space becomes more and more complex, showing more and more locally sub-optimal solutions. Thus, the chances of finding the global optimum decrease with increasing J. Figure 9 shows the trade-off between J and the best solution found by GA-EM. Notice that for $J = 2$ we have to determine 2*3-1 = 5 parameter components; however, for $J = 8$ this increases to 8*3-1 = 23 components. Unsurprisingly, Figure 9 suggests that J should not be chosen too large. In fact, $J = 4$ mixture components provide a good balance between data fit, model parsimony and problem complexity. We will therefore use this value throughout the remainder of this study.

Table 2 shows the parameter values of our best solution. In Figure 10 we compare the distribution of the true residuals (left panel) versus the estimated distribution based on our mixture model using the parameters in Table 2 (right panel). Notice that our model provides a very good fit: the

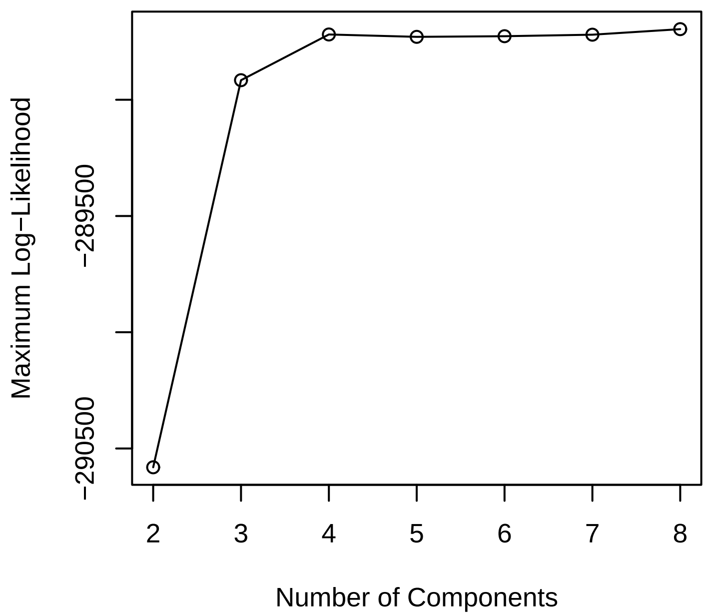

Figure 9: *Exploring the Number of Components in GA-EM Algorithm*

Table 3: *Quantiles of the true and the fitted distribution*

Percentile	10%	20%	30%	40%	50%	60%	70%	80%	90%
Original Residuals	-24.86	-19.52	-15.72	-12.32	-8.97	-5.35	-0.75	7.99	35.17
Fitted Residuals	-25.19	-19.77	-15.78	-12.28	-8.89	-5.31	-0.73	7.49	36.32

distribution in the right panel is almost indiscernible (at least visually) from the one in the left panel. Notice also the negative values in the left half of each distribution. These negative values indicate flights that have shorter delays compared with the seasonal and daily average.

Our mixture model has four mixture components. The individual components are overlaid in Figure 10(b). We notice that two components form the center of the distribution, accounting for the most "typical" delay. The third component captures medium delays while the fourth one accounts for the extremely large delays.

As pointed out earlier, the true and fitted delay distributions are very similar (at least visually). A more objective way of gauging their differences is via comparing their quantiles (see Table 3). We notice that 8 out of the 9 quantile-pairs have differences less than 1 minute. Only the largest quantile (i.e. the right tail of the distribution) has a slightly larger difference. We will investigate the tail-behavior in more detail below.

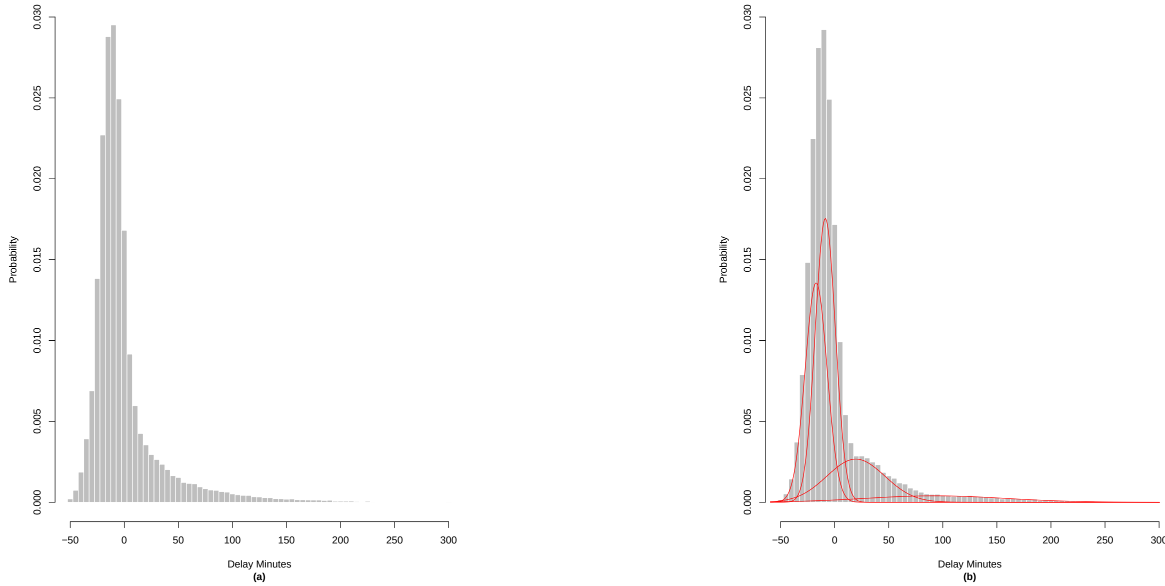

Figure 10: *Fitting the Residuals:* **(a)** *Density distribution of the original residuals* **(b)** *The fitted distribution with its four components*

4.6 Model Validation

In this section, we validate our model by checking its predictive ability on the holdout sample (30% of the data). Notice that the holdout sample stems from the same time period as the training sample; that is, both samples are from the year 2000. This approach provides a measure of performance when the model is viewed as a static model for characterizing delays in the same year. Of course, many interesting applications focus on predictions for *future* time periods. In Section 5, we describe the Monitor Alert application in detail. Since this application requires predictions of future delays, we suggest an approach for using our model to predict future delays by iteratively executing it in a rolling horizon mode. In that section, we also provide additional validation for its predictive capabilities.

We check predictive performance on the holdout sample by investigating our model's ability to predict the probability of a delay. To that end, we investigate its predictive performance around the center of the distribution and in its tail. Specifically, let $C_p = [a, b]$, where $[a, b]$ is the interval centered around the mean of the distribution of X such that $P(X \in C_p) = p$, i.e. C_p denotes "middle" p-percent of the distribution. As an example, $C_{80\%}$ denotes the middle 80% of the distribution (i.e. b is the 90th percentile and a is the 10th percentile); and similar for $T_p = [a, +\infty)$ where a is defined so that $P(X \in T_p) = p$, i.e. a denotes the (1-p)th percentile. To check the performance of our model, we first compute intervals C_p or T_p from our model for given values of p. We then compare p with the corresponding empirically-computed probabilities $\hat{p}$ calculated from

Table 4: *Model Robustness With Different Smoothing Penalties: Parameter Sensitivity Test*

λ_S	λ_D	$C_{80\%}$	$C_{90\%}$	$T_{3.00\%}$
1.03	0.41	81.07%	90.11%	2.60%
1.03	0.47	81.15%	90.12%	2.60%
1.03	**0.44**	81.11%	90.13%	2.60%
1.00	**0.44**	81.12%	90.08%	2.60%
1.06	**0.44**	81.04%	90.04%	2.59%

the observed data.

Table 4 illustrates the predictive capability of our model using $C_{80\%}, C_{90\%}$ and $T_{3.00\%}$. For instance, the value 81.07% in the first row implies that the interval associated with the middle 80% of our predicted distribution contains 81.07% of the true data. Similarly, the value 2.60% implies that the predicted upper 3.00% tail holds 2.60% of the true data. Thus, our model predicts well in the center of the distribution and in the tail.

Table 4 shows the performance for different values of the smoothing parameters λ_S and λ_D. The third row shows the results for the values we use in this study; the remaining rows illustrate the robustness of our results to varying values of λ_S and λ_D. We can see that our model manages to predict the middle of the distribution and its tail with only little error. Also, the predictive capabilities do not vary by much for slight changes in the smoothing parameters.

5 Application

We now describe how our model can be applied to improve congestion prediction within the National Airspace System (NAS). Our long-term research objective is a fairly complete overhaul of the current mechanism for predicting airspace congestion. Here, we show that our model in its current form can be used to improve the current process. In Section 5.1, we describe a basic approach to generating new congestion predictions. When used in this setting, our model would need to predict future delays. In Section 5.2, we show how this can be accomplished using a rolling horizon execution mode.

5.1 Improving Monitor Alert Predictions

In order to manage air traffic flows within the U.S., the Federal Aviation Administration (FAA) has contracted with the Volpe National Transportation Systems Center to operate the enhanced traffic management system (ETMS). Airspace sectors are three-dimensional volumes of airspace managed by a single team of controllers. Safety concerns dictate that controller workload should be kept within certain bounds and limits are placed on the number of aircrafts that can simultaneously occupy a sector. The *Monitor Alert* function within ETMS provides predictions when such overloads will occur (VNTSC, 2003). The goal of our work is to replace the current deterministic model for providing such predictions with a stochastic one.

We now provide a slightly simplified version of how *Monitor Alert* operates and then describe our approach to enhance it. We start by defining a set of variables defining future states, which we initially assume are deterministic. Later we will relax this assumption, by treating them as random variables.

$$\begin{aligned} F &= \text{ set of flights under consideration} \\ I_i(w,t) &= 1 \text{ if flight } i \text{ occupies sector } w \text{ at time } t \\ & \quad\ 0 \text{ otherwise} \\ N(w,t) &= \text{ the number of flights occupying sector } w \text{ at time } t \end{aligned}$$

ETMS continuously updates estimates of $N(w,t)$. The monitor alert function then compares these with sector capacities so as to determine if an alert is necessary. Since $N(w,t) = \sum_{i \in F} I_i(w,t)$, the process of computing $N(w,t)$ can be reduced to computing $I_i(w,t)$ for each flight i. ETMS maintains a prediction of the *flight plan* for each flight. Given an estimate of flight i's departure time, t^i_{dep}, the flight plan provides a deterministic prediction of the times at which the flight will pass through a series of airspace locations along its planned route. Specifically, it predicts the time at which the flight will pass over sector boundaries, and thus determines $I_i(w,t)$. Let τ denote the present time and t^i_{sch} the scheduled departure time of flight i, then ETMS and monitor alert operate as follows: if $\tau \leq t^i_{sch}$, t^i_{dep} is set equal to t^i_{sch} and if the flight has not departed but $\tau > t^i_{sch}$, t^i_{dep} is set equal to τ. Once the flight has departed, its airspace position and flight plan are dynamically updated based on current information.

There are many stochastic elements to this problem—our goal here is to address one of them, namely the possible variation in the flight's departure time. Specifically, for the case where $\tau \leq t^i_{sch}$, we treat t^i_{dep} as a random variable, which implies that $I_i(w,t)$ and $N(w,t)$ are also random variables. Then, in the above procedure we can use $E[N(w,t)] = \sum_{i \in F} E[I_i(w,t)]$. We note that generally flights have three states: on ground when $\tau \leq t^i_{sch}$, on ground when $\tau > t^i_{sch}$, and airborne. Our modifications only apply to flights in the first category. For these flights, since $E[I_i(w,t)] = Pr[I_i(w,t) = 1]$, we need to consider the problem of computing the probability that a flight is in a sector at a given time. Now, let $t^{i,w}_{in}$ be the time required for flight i to reach the sector boundary of w under the current flight plan estimate and $t^{i,w}_{pass}$ be the time required for fight

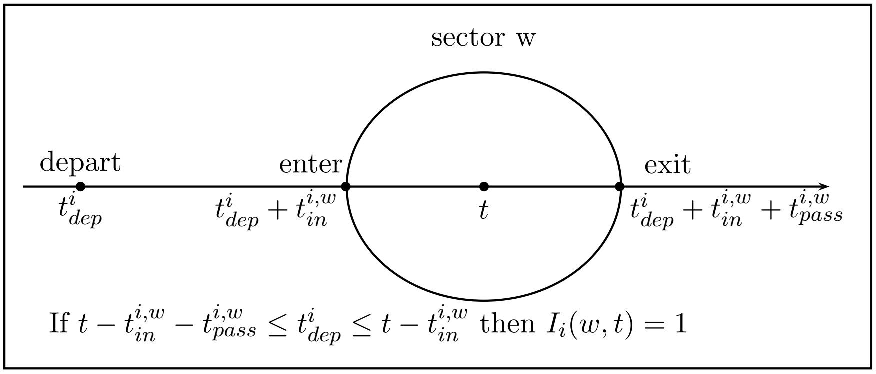

Figure 11: *A Typical Flight Path Through Sector* w

i to pass through sector w under the current flight plan (see Figure 11). Then,

$$Pr[I_i(w,t) = 1] = Pr(t - t^{i,w}_{in} - t^{i,w}_{pass} \leq t^i_{dep} \leq t - t^{i,w}_{in})$$

In previous sections, we provide methods for estimating the departure delay, which measures the discrepancy between the actual departure time and the scheduled departure time. The departure time t^i_{dep} is just the summation of the departure delay and the scheduled departure time.

For example, suppose a flight i is scheduled to depart at 9:50 am (t^i_{sch}=9:50) on Jan 10th. Let $t^{i,w}_{in}$=9 min and $t^{i,w}_{pass}$=15 min. Given the observation time t at 10:10 am, $t - t^{i,w}_{in} - t^{i,w}_{pass}$=9:46 and $t - t^{i,w}_{in}$=10:01. That is, in a deterministic model, since the scheduled departure time falls within this time interval, the flight will be predicted, with probability one, to be in sector w at 10:10 am. However, because of the possibility of delays, this may or may not be the true. Our model provides a way to calculate the actual probability of this event,

$$\begin{aligned} & Pr(9:46 <= t^i_{dep} <= 10:01) \\ = \quad & Pr(9:46 <= t^i_{sch} + y_i(Jan10th, 9:50am) <= 10:01) \\ = \quad & Pr(9:46 <= t^i_{sch} + f(Jan10th) + \varphi(9:50am) + \epsilon_i <= 10:01) \end{aligned} \tag{21}$$

where the seasonal delay $f(Jan10th)$ equals 10.7 minutes and the daily propagation delay $\varphi(9:50am)$ equals 4.56 minutes, as predicted by our model.

It is easily demonstrated that equation (21) can be written as

$$\begin{aligned} & Pr(-19.27min <= \epsilon_i <= -4.27min) \\ = \quad & Pr(\epsilon_i <= -4.27min) - Pr(\epsilon_i <= -19.27min) \\ = \quad & 0.628 - 0.205 = 0.42 \end{aligned}$$

Therefore the probability that flight i is in sector w at observation time t=10:10 am is 0.42.

By applying the same rationale to other flights, we can compute the expected number of flights in sector w at a specific time t.

5.2 Dynamic Model Updates and Future Predictive Performance

There are a number of factors that can cause substantial yearly shifts in air traffic delays. Air traffic levels (demand) can vary from year to year. For example, there was a substantial decrease in traffic from 2001 to 2002 and a corresponding decrease in delays. Also, the extent of adverse weather conditions can vary substantially from year to year to a degree that a noticeable impact on delay statistics is seen. Another factor is the relatively steady introduction of performance-improving technologies (e.g. new avionics) and infrastructure (e.g. new runways). We consider the problem of generating a model that adapts to such changes over time an interesting research topic and view the work in this paper as a fundamental basis on which to build such models. On the other hand, it is also the case that our model can be adapted in fairly simple ways to get quite reasonable results for this problem.

We propose an approach that can be viewed as a forward rolling horizon method. Consider our model as a method for generating delay distributions over a s-day time horizon (of course, as described in this paper, we use $s = 365$ and initiate the model on the January 1). Now consider the possibility of applying the model to predict delays on day $s + 1$. A seasonal trend value for day $s + 1$ can be obtained by functionally extending the seasonal trend for one additional day, i.e. $f(s + 1)$. A daily propagation value can be obtained via the daily propagation component $\varphi(t)$ estimated from the prior s days of data. This approach has appeal for several reasons since the daily propagation effect is based on the past s days of history as is the degree to which daily and seasonal effects are separated.

With this point of view it is quite natural to apply the model in a rolling horizon mode, where, in order to produce an estimate for a particular day, we create a model based on the previous s days. Over time we simply add the most recent day and delete the earliest day and update the model appropriately. For our particular application, we start by using all data from one year (say, year #1) to predict delays on the first day of the next year (say, January 1 of year #2). Once the true delay for January 1 of year #2 becomes available, we drop January 1 of year #1 (i.e. we drop the oldest observation in the data) and replace it for January 1 of year #2 (i.e. we replace it with the most recent observation). Based on this updated data set, we update the seasonal trend and daily propagation pattern and predict the next day, Jan 2 of year #2. We continue to iterate (or "roll") in this manner so that the predictions for any arbitrary day is always based on the data from the prior 365 days.

Table 5: *Forecasting performance on year 2001 data.*

$C_{80\%}$	$C_{90\%}$	$T_{3.00\%}$
82.93%	92.91%	2.65%

Notice that in the above rolling horizon method we update seasonal and daily trends with every incoming new observation. We could also update the error component in the same way; however, updating the error component is computationally much more expensive as pointed out earlier. Moreover, it may not even be necessary to update the error every day. It is unlikely that the error distribution changes by much over the period of a week or a month. Thus, rather than updating it with every new observation, we update it in blocks. In that form, the resulting GA-EM algorithm resembles the block-update EM algorithm proposed by Ng and McLachlan (2003). In the following we use the same error component for a period of three months. The resulting predictive performance of our model is strong, suggesting that the error component may not have to be updated too frequently in practice.

We apply the above approach to predict delays for each day of the first quarter (i.e. first three months) of the year 2001. Notice that this data is "new" in the sense that it has not been used in the building of our basic model which was based on year 2000 only. In that sense, it provides an estimate of the model's forecasting performance in rolling horizon mode. To assess its effectiveness, we use the same validation approach as in Section 4.6. That is, we compute the empirical probabilities of the intervals $C_{80\%}$, $C_{90\%}$ and $T_{3.00\%}$. The results, depicted in Table 5, show good forecasting performance, only slightly different from those in Table 4.

6 Conclusions and Future Research

Our approach to estimating flight departure delays has several distinctive (and new) features. First, we decompose observed delays into three components: seasonal trend, daily propagation pattern and random residuals, which provides a new perspective for understanding pushback delays. The additive model based on these three components is parsimonious, easy to implement and update, and robust; most importantly, it demonstrates good fit and strong predictive performance. Second, rather than providing only point estimates, we estimate the entire delay distribution. This distribution can be used to predict expected airspace congestion levels and lead to more accurate decisions. We also propose a new version of the EM algorithm that, by borrowing ideas from the Genetic Algorithms, can overcome local solutions associated with finite mixture models. Finally, we demonstrate a way to make our model dynamically adaptive, via a rolling horizon approach.

This approach shows promising forecasting abilities for delay of future time periods.

In this paper, we focus on United Airlines and Denver International Airport only. Our ultimate goal, of course, is to generate departure delay distributions for the entire NAS. Our model can be applied readily to other airline/airport combination. An interesting (and open) research problem is to combine individual airline/airport models into one general, NAS-wide model. As one step into that direction, one could try to extract, from individual airline/airport models, the effects that contribute to NAS-wide delay. Such an approach would provide more insight into the general structure of delays and also would be easier to maintain and update on a NAS-wide basis.

Another general area for further research is the development of dynamic models. Our rolling horizon approach represents a step into that direction. It could be augmented with other elements that allow real-time reaction to dynamically changing conditions such as weather, disruptive events, etc. Once a NAS-wide dynamic model is in place, it could be compared against Monitor Alert using real test scenarios over an extended period of time.

7 Acknowledgments

The authors would like to thank Aron Futur of the Volpe Transportation Systems Center and the two referees and associate editor for many helpful suggestions. The work of the first two authors was supported by the National Center of Excellence for Aviation Operations Research (NEXTOR) under FAA Research Grant 96-C-001 and Contract DFTA03-97-D00004. Any opinions expressed herein do not necessarily reflect those of the FAA or the U.S. Department of Transportation.

References

ATA (1999). *Approaching Gridlock Air Traffic Control Delays.* Washington DC: Air Transport Association, Departments of Air Traffic Management and Economics.

Beatty, R. (1998). Preliminary evaluation of flight delay propagation through an airline schedule. *2nd USA/EUROPE Air Traffic Management R&D Seminar*, Orlando, 1st–4th.

Bilmes, J. A. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *Technical Report TR-97-021*, Computer Science Division, Department of Electrical Engineering and Computer Science, U. C. Berkeley.

Chandran, B. G. (2002). *Predicting Airspace Congestion using Approximate Queueing Models.* University of Maryland, College Park: Master Thesis.

de Boor, C. (1978). *A Practical Guide to Splines.* New York: Springer-Verlag.

FAA (2002). *Performance Monitoring Analysis Capability V3.1 Addendum Document.* Washington DC: U.S. Department of Transportation.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley.

Green, P. J. and B. W. Silverman (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach.* London: Chapman and Hall.

Hastie, T. J. and R. J. Tibshirani (1990). *Generalized Additive Models.* London: Chapman and Hall.

Heath, J., M. Fu, and W. Jank (2006). Global optimization with MRAS, Cross Entropy and the EM algorithm. Working Paper, Smith School of Business, University of Maryland.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press.

Idris, H., J. P. Clarke, R. Bhuva, and L. Kang (2002). Queueing model for taxi-out time estimation. *Air Traffic Control Quarterly 10(1)*, 1–22.

Inniss, T. and M. Ball (2004). Estimating one-parameter airport arrival capacity distributions for air traffic flow management. *Air Traffic Control Quarterly 12*, 223–252.

Jank, W. S. (2006a). Ascent EM for fast and global model-based clustering: An application to curve-clustering of online auctions. *Computational Statistics and Data Analysis*, (forthcoming).

Jank, W. S. (2006b). The EM algorithm, its stochastic implementation and global optimization: Some challenges and opportunities for operations research (forthcoming). In F. Alt, M. Fu, and B. Golden (Eds.), *Topics in Modeling, Optimization, and Decision Technologies: Honoring Saul Gass' Contributions to Operations Research.* Springer Verlag, NY.

McLachlan, G. and D. Peel (2000). *Finite Mixture Models*. New York: John Wiley & Sons.

Mueller, E. and G. Chatterji (2002). Analysis of aircraft arrival and departure delay characteristics. In *Proceedngs of the AIAA Aircraft Technology, Integration, and Operations (ATIO) Conference, Los Angeles, CA, October 1-3, 2002*.

NCAR (2002). Airlines get new tools to avoid in-flight icing. *NCAR (National Center for Atmospheric Research) News Release*, March 26, Boulder, Colorado, available at http://www.ucar.edu/communications/newsreleases/2002/icing.html.

Ng, S.-K. and G. J. McLachlan (2003). On some variants of the EM Algorithm for fitting finite mixture models. *Australian Journal of Statistics 32*, 143–161.

Odoni, A. R., J. M. Rousseau, and N. H. M. Wilson (1994). Models in urban and air transportation. *Handbooks in Operations Research and Management Science 5*, 107–128.

Pernkopf, F. and D. Bouchaffra (2005). A genetic-based em algorithm for learning gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 1344–1348.

Politovich, M. K., B. C. Bernstein, J. Hopewell, T. Lindholm, L. Gaeurke, C. Knable, D. Hazen, and B. Martner (2002). An unusual icing case: 20 march 2000, denver, colorado. *9th AMS Conference on Aviation, Range and Aerospace Meteorology*, Portland, May.

Reinsch, C. (1967). Smoothing by spline functions. *Num. Math. 10*, 177–183.

Rosenberger, J. M., A. J. Schaefer, D. Goldsman, E. L. Johnson, A. Kleywegt, and G. L. Nemhauser (2000). Simair: A stochastic model of airline operations. *Proceedings of the 2000 Winter Simulation Conference*, Orlando, December.

Schaefer, L. and D. Millner (2001). Flight delay propagation analysis with the detailed policy assessment tool. *IEEE Systems, Man, and Cybernetics Conference*, Arizona, October.

Shumsky, R. A. (1997). Real-time forecast of aircraft departure queues. *Air Traffic Control Quarterly 5(4)*, 281–308.

VNTSC (2003). *Enhanced Traffic Management System, Functional Description, Version 7.6.* Cambridge, MA: Volpe National Transportation System Center, U.S. Department of Transportation.

Wang, P., L. Schaefer, and L. Wojcik (2003). Flight connections and their impact on delay propagation. *22nd Digital Avionics Systems Conference*, Indiana, October.

Wanke, C., L. Song, S. Zobell, D. Greenbaum, and S. Mulgund (2005). Probabilistic congestion management. In *Proceedings of 6th USA/Europe Air Traffic Management R&D Seminar, Baltimore, MD, June, 2005.*

Willighagen, E. L. (2005). *genalg - R based genetic algorithm.* version 0.0.6.