A Modeling Framework to Estimate Airport Runway Capacity

in the National Airspace System

Yueh-Ting Chen

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

In

Civil and Environmental Engineering

Committee Members:

Dr. Antonio Trani

Dr. Hojong Baik

Dr. Hesham Rakha

December 12, 2006

Blacksburg, VA

Keywords: airport runway capacity, Airfield Capacity Model

A Modeling Framework to Estimate Airport Runway Capacity

in the National Airspace System

Yueh-Ting Chen

Abstract

The objective of this study is to estimate the airport capacity in the National Airspace System (NAS). Previous studies have focused on the airport capacity of large commercial airports. This research study estimates the runway capacity for more than two thousand airports in the NAS in order to understand future tradeoffs between air transportation demand and supply. The study presented in this report includes capacity estimates for general aviation and commercial airports. To estimate airport runway capacity, the Federal Aviation Administration (FAA) Airfield Capacity Model (ACM) is used to assess the capacity at all candidate airports in a target airport set. This set includes all airports with potential Very Light Jet (VLJ) operations. The result of the study provides a broad view about the airport capacity in the future air transportation system, and could help decision makers with a modeling framework to identify congestion patterns in the system. Moreover, airport capacity is an important limiting factor in the growth of air transportation demand. The main motivation in our analyis is to include airport capacity constraints in forecasts of air transportation demand. The framework described in this report has been integrated into the Transportation Systems Analysis Model (TSAM). TSAM is a comprehensive intercity and multimode transportation planning tool to predict future air transportation demand.

Acknowledgements

I would like to thank Dr.Trani for the opportunity to work with him in the Air Transportation Systems Laboratory. His wisdom and patience guide me throughout this study, and the past two years are definitely one of the most invaluable learning experiences that I have ever had. I would also like to thank Dr.Baik, Dr.Rakha, Mr.Swingle and Nick for all of their advices and help throughout this research project, especially the great work of TSAM integration by Nick. Besides, I am really glad to have Senanu, Anand, Sohn, Jeff, Yue, and Nola as my colleagues. Fang-Yi, Chu-Yun, Chih-Yang, and Wen-Chun, I would like to thank all your encouragements. Last but not least, I have to thank the support from my parents and brother throughout my study in the Virginia Tech. Thank your all!

Table of Contents

List of Figures

List of Tables

CHAPTER 1 INTRODUCTION

The air transportation system has been an important mode of transportation for medium and long haul trips. According to the Federal Aviation Administration (FAA) Aerospace Forecast [FAA, 2006], U.S. and foreign flag carriers carry 138.7 million passengers between the U.S. and the rest of the world. The total number of enplanements of U.S. commercial air carriers for fiscal year 2005 was 628.5 million, a 6% increase compared to the previous year. In addition to commercial flights, general aviation plays an important role in the National Airspace System (NAS). The FAA Aerospace Forecast indicates that non-commercial aircraft activity at FAA and contract towers reached 37 million instrument flight operations in 2005. Of these 26.1 million were commercial flights. This implies that 10.9 million operations in the NAS are attributed to general aviation activity. The economic influence of the air transportation system is substantial. The report entitled "The Economic & Social Benefits of Air Transport" [Air Transport Action Group, 2005] indicates that the air transport industry in North America contributes nearly US$ 410 billion to the gross domestic product (GDP). In the future it is important to maintain an efficient and congestion-free air transportation system.

Congestion, which affects the efficiency of the air transport network, is always one of the most critical issues in the development of any air transportion system. The problem that air operations cannot depart/arrive on-time not only disrupts the travel experience of passengers, but also disrupts the airlines' operating efficiency and thus affects the national economy. Review of year 2000 data shows that 27 percent of the scheduled flights in the U.S. were delayed more than fifteen minutes or canceled [U.S. Senate, 2001].

To help develop a better, more efficient air transportation system, it is important to understand the capacity constraints of the system. This research effort attempts to predict the airport runway capacity of many airports comprising the NAS. Although there have been several studies addressing the capacity of large commercial airports, there have been few efforts to study the capacity of large numbers of airports in the NAS. With this study and demand forecast models, aviation planners and the Federal Government could identify airport capacity shortcomings in the current system and evaluate possible approaches to improve the air transportion system in the future. This research effort supports the development of the Next Generation Air Transportation System (NGATS) currently under study by National Aeronautics and Space Administration (NASA) and the FAA.

Section 1.1 Objectives

The objectives of this study are:

a) To estimate airport runway capacity for a large number of airports in the NAS. These airports will be candidates to receive very light jet aircraft (VLJ) operations in the future.
b) To identify future airport capacity shortcomings as the demand for air transportation increases in the future.

Section 1.2 Scope of Study

Previous capacity studies have been mostly limited to “Large Hub” commercial airports. According to the definition in the United States Code, Title 49, Chapter 471 [49 U.S.C. Chapter 471, 2006], a large hub airport is defined as an airport carrying one percent or more of all annual passenger boardings in the NAS. Our study includes all public airports in the continental U.S. capable of handling VLJ traffic in the future. The analysis includes the operational characteristics, geographical specifications, and related capacity of public airports with an effective runway length greater or equal to 915 meters (3,000 feet).

The emergence of Very Light Jets (VLJ) is one of the most important challenges in the future of NAS operations. The National Business Aviation Association defines a VLJ as “a jet aircraft weighing 10,000 pounds or less maximum certificated takeoff weight and certificated for single pilot operations” [NBAA, 2005]. The Virginia Tech Air Transportation Systems Laboratory (ATSL), NASA Langley Research Center and Swales Aerospace have conducted a series of studies to estimate the demand and impact of VLJ traffic in the United States. At the centerpiece of this analysis is the Transportation Systems Analysis Model (TSAM). TSAM is a large-scale, multimode intercity demand generation model [A.A. Trani et al., 2003]. These studies indicate that up to 4,100 VLJ aircraft could be flying in the NAS by 2017 [A.A. Trani et al., 2006]. Although the first VLJ in operation is expected in early

2007, it will take several years for the market to develop. Some media reports have stated possible congestion brought by VLJ to already congested airports and airspace [The Wall Street Journal, 2006]. One obvious research question is whether or not the airports in the NAS will have the capacity to handle these emerging flights.

To fill in the gap of previous studies and to help decision makers to understand the nationwide impacts of future air transportation demand, this research effort would assess airport capacity for thousands of airports that, in general, are ignored in NAS-wide studies.

Section 1.3 Characteristics of Airport Capacity

Airport capacity is the number of operations, either takeoffs or landings that can be performed in a unit of time, usually an hour, without violating aircraft safety regulations. If an airport cannot meet the demand function with its existing capacity, a degradation in the perceived level of service (as viewed by airlines and passengers) develops. This results in either a permanent adjustment to the schedule or flight delays.

Airport capacity can be discussed from two viewpoints: airside capacity and landside capacity. The airside capacity includes components like the runway, taxiway system, and adjacent airspace to the airport. Landside capacity includes the terminal, gate, and access roads. Table 1 lists the factors that influence airport capacity.

Table 1: Factors Influencing Airport Capacity.

Factor	Description
Air Traffic Control	Navigation aids, air traffic control rules and procedures
Runway System	Layout and number of runways
Taxiway System	Configuration of taxiways
Apron/Gate Facilities	Capability to accommodate aircraft in apron/gate area
Terminal Facilities	Landside facilities that passenger move through from curb to the loading bridge, like passenger waiting area, ticket counters, security screening points, customs and immigration etc.
Ground Transportation System	Landside access system for travelers such as access roads, parking facilities, and public transit.

Operating Restrictions	Regulations and rules to prevent the airport from operating at full capacity; such as curfews, special departure/arrival procedures, activities in the adjacent airspace
Meteorological Conditions	Winds, visibility, ceiling, and precipitation

[A.T. Wells, 2000]

Traditionally, the runway system is one of the most critical components driving airport capacity. This is the focus of this research. Factors influencing airport runway capacity are listed in Table 2.

Table 2: Runway Capacity Influencing Factors.

Factor	**Description**
Number of Runways and Configuration	The layout and number of runway(s)
Runway Operating Strategy	The way runways are used. For example, runway for arrival only/departure only or mix operations
Runway Occupancy Time (ROT)	The time an aircraft occupies the runway. For arrivals it starts when aircraft passes runway threshold and ends when aircraft exit the runway. For departure it starts when aircraft enters the runway and ends when the aircraft passes the departure end of the runway
Aircraft Mix and Operating Sequence	The percentage of operations among all aircraft weight classes and their arrival/departure sequence
Common Approach Path Length	The distance aircraft fly in-trail during the approach stage
Weather	Visibility and ceiling
Separation Requirements	The required minimum distance/time between leading and trailing aircraft
Approach Speed	The speed when aircraft passes through arrival end of runway
Air Traffic Control Related	Availability of tower, radar coverage

This analysis employs data available from various aviation databases to:

a) identify aircraft operating at each airport, and

b) to derive existing operational conditions at each airport.

A detailed process and methodology to estimate runway capacity is described in Chapter 3 of this report.

Section 1.4 Summary

Airport runway capacity estimation procedures have been developed for major commercial airports. Whereas some of these procedures can be used at towered airports, there are thousands of airports where such procedures do not provide realistic runway saturation capacity estimates. To address this problem, this study integrates several aviation databases and basic runway capacity estimation methods to estimate the capacity for a large number of airports in the NAS. The results could provide Federal State and local authorities with a powerful tool to allocate and optimize limited funding resources throughout the NAS. This in turn will benefit air transportation system users. Morevover, knowledge of airport capacity is a requirement to understand the balance between air transportaton demand and supply in the future NAS.

CHAPTER 2 LITERATURE REVIEW

This chapter reviews previous studies on the subject of airport capacity analysis. The review of the existing literature helps us to understand various perspectives and approaches to estimate airport runway capacity. The chapter discusses various methodologies to estimate airport runway capacity analysis, including the merits and drawbacks of each method. This review process can help define the scope and focus of this research. The last section of the chapter concentrates on past studies on the subject matter with emphasis on runway capacity-related factors.

Section 2.1 Airport Capacity Analysis Methodologies

There are two broad categories of airport capacity estimation models:

a) analytic models, and
b) simulation-based models

Analytic models consist of a series of close-form equations that compute hourly airport capacity with known input parameters. Simulation models emulate the movement of aircraft by using discrete-event or fixed-timed techniques. These models can use statistical sampling techniques, and then infer the airport capacity with the data generated during the simulation [FAA, 1998]. Airport simulation models do not estimate directly airport capacity. Capacity is inferred from values of delay calculated by such models.

Lee et al. compared several airport capacity models developed over the past three decades [Lee et al., 1997]. In this study, the authors comment that simulation-based models emulate detailed maneuvers of airport operations. However, these models require extensive data and long computing times to generate results. For this reason, simulation models can only be applied to a limited number of airports or airport configurations considering practical resources and data input requirements. Another issue to consider is that large data inputs and outputs take longer time to collect and analyze. In contrast, analytical models require fewer parameters and many times produce adequate capacity estimations. These models are preferred for NAS-wide analyses. Table 3 summaries the characteristics of analytic and simulation-based models.

Table 3: Comparison of Analytic and Simulation Models.

	Analytic Model	Simulation Model
Complexity	Easier, faster to run	Complex, require more computing power
Accuracy	Adequate results	More accurate
Applicability	Preliminary planning and analysis	Detailed and microscopic analysis
Methodologies Used	• Time-space analysis • Queuing models	• Monte Carlo Simulation • Discrete-event simulation model
Example	FAA Airfield Capacity Model (ACM)	Airport and Airspace Simulation Model (SIMMOD)

Section 2.2 Review of Airport Capacity Models

The goal of this research project is to estimate airport runway capacity in the NAS for a large number of airports. This section reviews previous airport capacity studies. The review of the literature reveals that none of the previous studies addresses the scope and range of airports needed in our analysis.

The "Airport Capacity Benchmark Report 2004" [FAA, 2004], is an official airport capacity guideline produced by the FAA and the MITRE Corporation. This study covers the 35 busiest commercial airports in the U.S. and provides capacity information for three different weather conditions:

a) optimum (or visual approaches),
b) marginal (weather better than instrument conditions but without meeting criteria for visual approaches), and
c) IFR instrument conditions

The capacity numbers in the Benchmark Report do not consider any limitations on airport traffic flow, such as slot controls, taxiway and gate congestion. The FAA/MITRE team obtained some of the airport capacity numbers using the FAA Airfield Capacity Model (ACM). The same model is used in this research project for capacity calculations.

To validate the results generated by ACM and also to compare them with practical airport capacity numbers, the FAA adjusts the capacity numbers estimated by ACM as follows:

a) Experience of ATC personnel, like control tower and terminal radar control (TRACON) team, and
b) historical traffic data for arrivals and departures

The Benchmark Report 2004 provides practical measures of airport capacity with corrections made by ATC specialists. A shortcoming of this analysis is its limited scope to only large airports in the system. The FAA is expanding the benchmark study to 102 airports in the system. The coverage is still far from complete if airports with significant general aviation traffic are to be included. The NAS is an interactive system, which means congestion at key airports can trigger a ripple effect across the country. Future demands for air transportation will not necessarily be limited to the existing set of airports. For example, reliever airports in large metropolitant areas with little commercial traffic today might be heavily used in 20-30 years in the future to satisfy growing demands for air transportation regionally. These demands will shift to secondary and reliever airports as many existing commercial airports have little room to expand. Therefore, there is a need for NAS-wide airport capacity estimates to better predict future air transportation demand.

Blumstein [Blumstein, 1959] developed a simple mathematical model to estimate capacity for an arrival-only runway. The assumptions made by Blumstein are:

1. Aircraft land in the order in which they arrive at the entry gate (see Figure 1).
2. Aircraft arrive at the gate independently and in random sequence.
3. Aircraft must maintain a minimum distance separation at the gate and a minimum time separation on the runway.
4. The runway is used only for landings and is operating to capacity, i.e., aircraft operate as close to each other as minimum separation permits.

5. Aircraft maintain constant velocity from the time they enter the entry gate until they reach the runway.

The model can be presented as following:

Vx velocity for aircraft x

V_2^* the critical velocity of the trailing aircraft so that the separation at landing will be exactly t_0

S_0 minimum required separation in distance at gate

T separation expressed in time

T_{ij} the time separation between arrivals of leading aircraft i and trailing aircraft j

t_0 minimum landing separation time

m common approach path

n common approach path plus separation between leading and trailing aircraft

τ expected value of headway between consecutive arrivals

P_iP_j probability of landing sequence (i,j)

$$T(V_1, V_2) = \begin{cases} n/V_2 - m/V_1 & for \quad V_2 < V_2^* \\ t_0 \quad for & V_2 \geq V_2^* \end{cases} \tag{1}$$

$$\tau = \sum_{j=1}^{j=n} \sum_{i=1}^{i=n} T_{ij} P_i P_j \tag{2}$$

$$\text{Capacity} = \frac{1}{\tau} \tag{3}$$

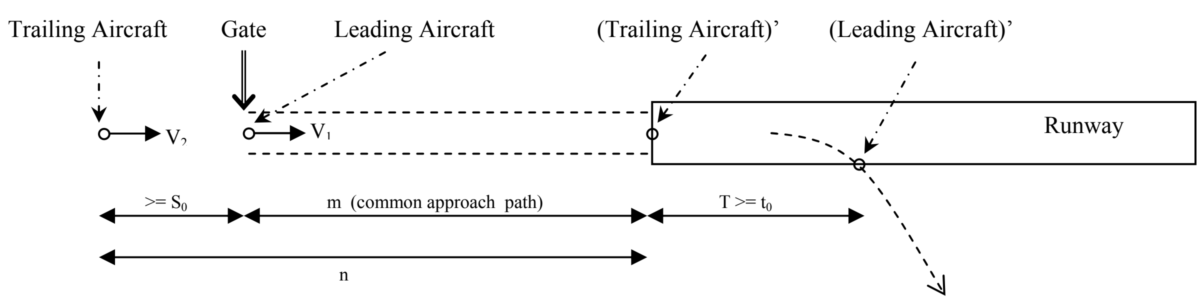

Figure 1: Problem Situation.

The mathematical model above can be used to explain the following time-space analysis.

Parameters:

ROT_i Runway Occupancy Time for aircraft i (seconds)

V_i speed of leading aircraft (knots)

V_j speed of trailing aircraft (knots)

δ_{ij} minimum separation required (nautical miles)

δ minimum arrival-departure separation (nautical miles)

γ common approach path (nautical miles)

T_{ij} headway between successive arriving aircraft (seconds)

B_{ij} buffer time between successive aircraft (seconds)

σ_0 standard deviation of the in-trail delivery error (seconds)

TD_k departure runway occupancy time (seconds)

P_v the probability of violating the minimum separation criteria between aircraft

q_v value of the cumulative standard normal at probability of violation Pv

τ time delay for departure operations (seconds)

εij minimum departure-departure headway

D_{L-T} number of hourly departures for arrival pair of leading aircraft L and trailing aircraft T

TG number of total arrival gaps per hour

P_{L-T} probability of the arrival pair of leading aircraft L and trailing aircraft T

DG_{L-T} number of departures performed in the gap of arrival pair with leading aircraft L and trailing aircraft T

Since aircraft speeds in the final approach path vary, two scenarios are studied (called closing and opening cases). These are depicted graphically in Figure 2 and 3, respectively.

The error free headway is calculated as T_{ij} = Ti – Tj (4)

$$T_{ij} = \frac{\delta_{ij}}{V_j} \tag{5}$$

The position error buffer time is given by,

$$B_{ij} = \sigma_0 q_v; \tag{6}$$

The error free headway is calculated as T_{ij} = Ti – Tj (7)

$$T_{ij} = \frac{\delta_{ij}}{V_j} + \gamma\left(\frac{1}{V_j} - \frac{1}{V_i}\right) \tag{8}$$

The position error buffer time is given by,

$$B_{ij} = \sigma_0 q_v + \delta_{ij}\left(\frac{1}{V_j} - \frac{1}{V_i}\right); \text{ if Bij <0, then Bij = 0} \tag{9}$$

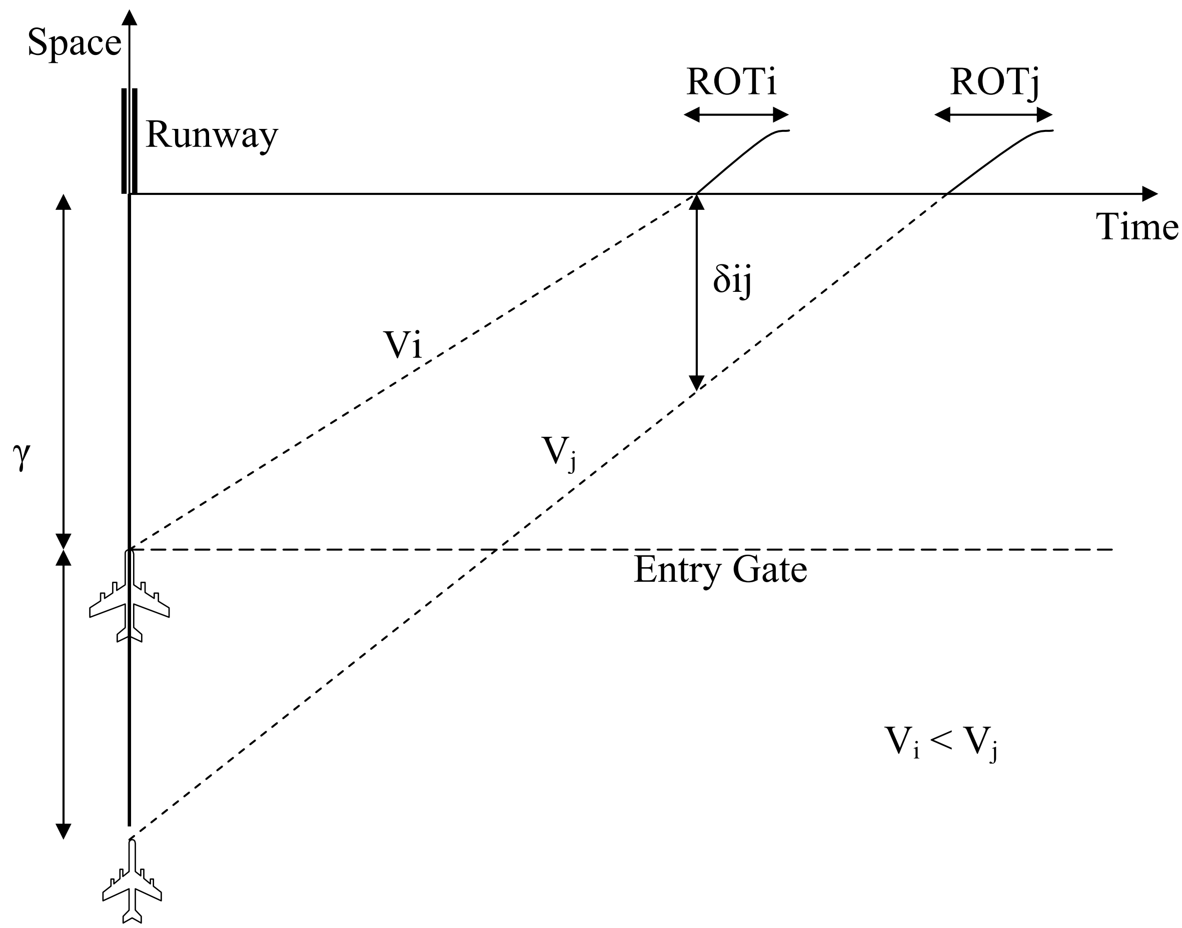

Figure 2: Final Approach and Landing Processes; Arrival-Only Scenario; Vi < Vj

The arrival saturation capacity is,

$$\text{Carr} = \frac{1}{E(Tij + Bij)} \tag{10}$$

Similarly, departure-only capacity is equal to the reciprocal of departure headway.

$$C_{\text{dep-only}} = \frac{1}{E(\varepsilon_{\text{ij}})} \tag{11}$$

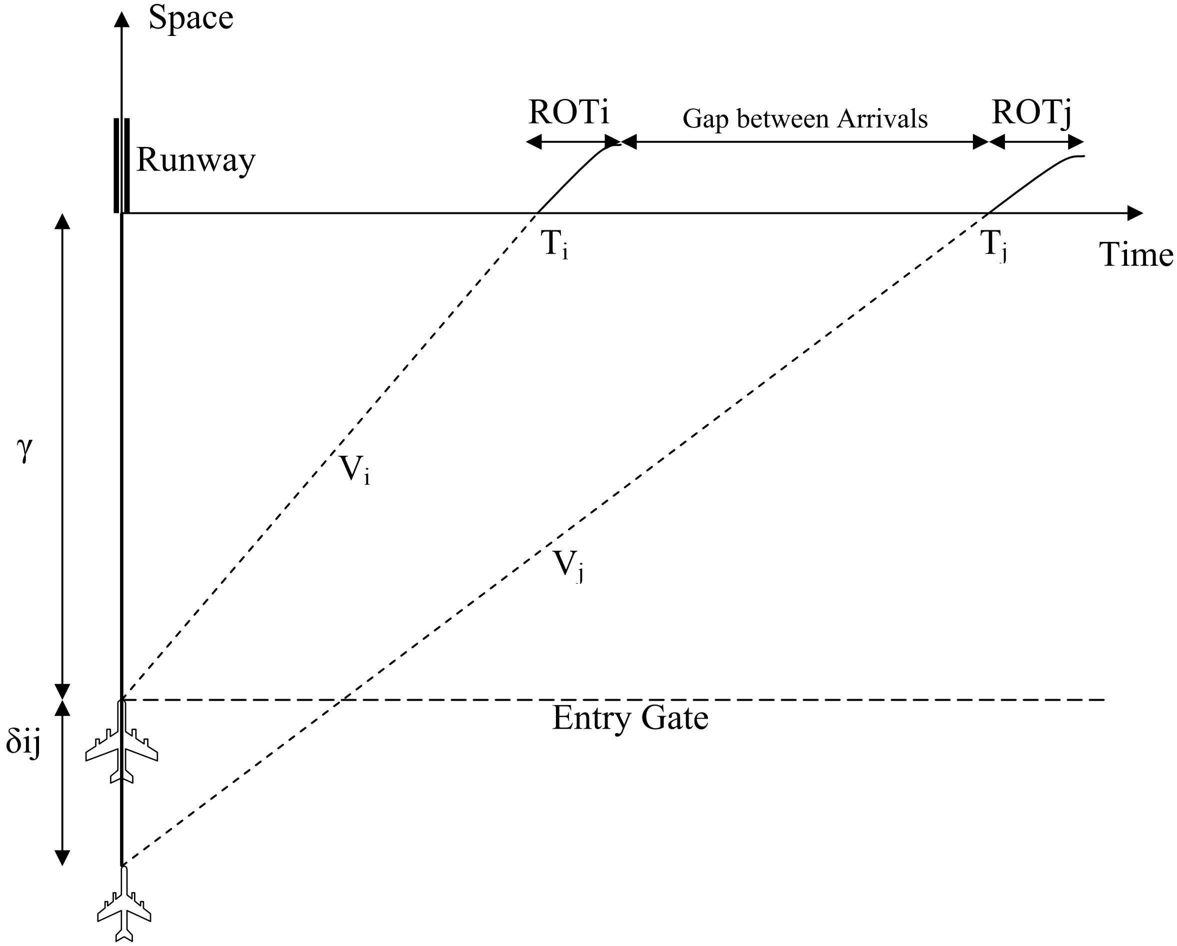

Figure 3: Final Approach and Landing Processes; Arrival-Only Scenario; Vi > Vj
[Trani, A.A., 2005]

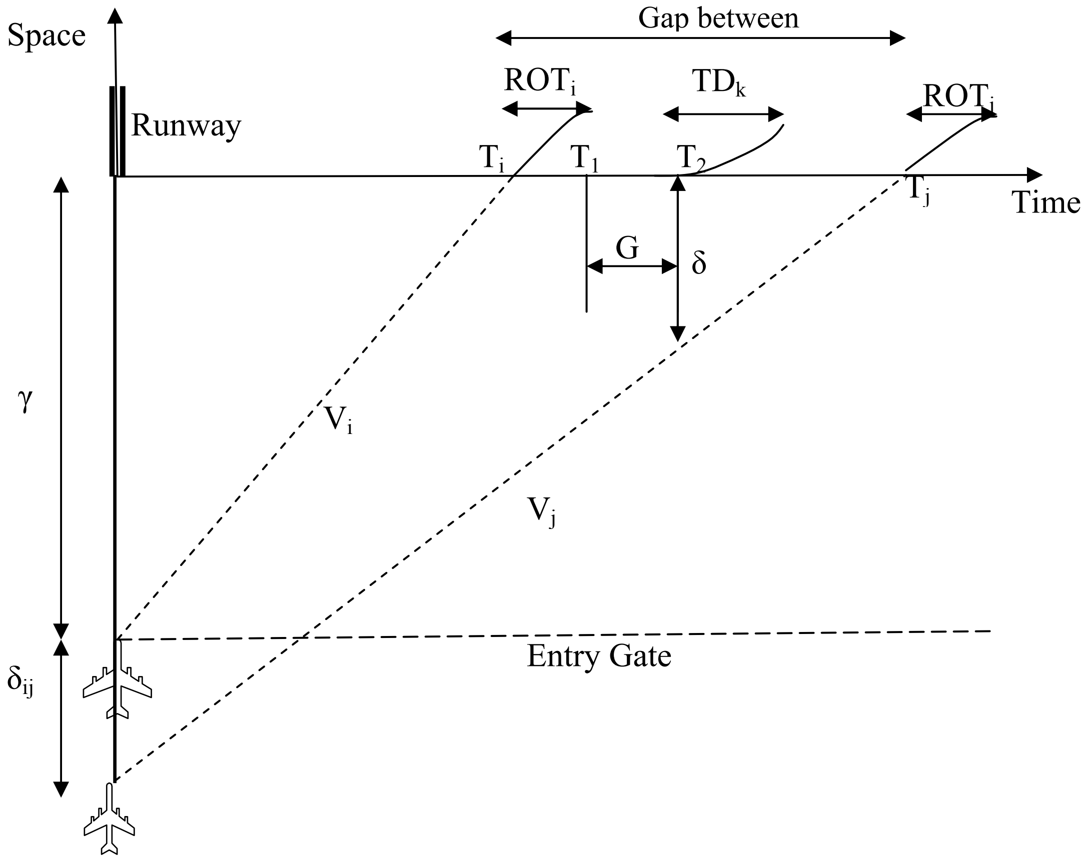

Figure 4: Final Approach and Landing Processes, Mix Operation Scenario. [Trani, A.A., 2005]

Under saturation conditions, aircraft operate as close to each other as the minimum separation permit. However, in most instances successive arrivals will leave gaps that can be exploited to process departures. This is illustrated in the Figure 4.

$$T_1 = T_i + ROTi \tag{12}$$

$$T_2 = T_j - \frac{\delta}{Vj} \tag{13}$$

$$G = T_2 - T_1 \tag{14}$$

then

$$G = T_j - \frac{\delta}{Vj} - T_i - ROT_I \tag{15}$$

For n departures in gap K, we use the following expression to estimate whether departures can be processed.

$$Tij + Bij = \frac{\delta}{Vj} + ROTi + (n-1)TDk + \tau \quad (16)$$

The value of *n* can be substituted by any positive integer to examine whether the gap between arrivals can accommodate n departures. If the right hand side of the equation is smaller than left side, then (n+1) departures should be examined to see whether more departures can be performed in each gap. This process will continue until the right hand side of the equation is larger than the arrival gap, i.e., ($Tij + Bij$). The critical number of departures per gap is the maximum number of departures processed in that gap.

$$D_{L\text{-}T} = TG\ (P_{L\text{-}T})\ (DG_{L\text{-}T}) \quad (17)$$

Repeating this process for all aircraft type pairs the expected number of departures in one hour can be calculated. The total number of departures is,

$$\sum D_{L\text{-}T} = \sum \{TG\ (P_{L\text{-}T})\ (DG_{L\text{-}T})\} \quad (18)$$

The total operations are the sum of total arrivals and total departures.

Gilbo [Gilbo, 1993] used linear programming techniques to optimize the number of operations at airports. Using historical operational data at Chicago O'Hare International Airport, Gilbo creates capacity curves and operational limits for various airport conditions. Given operational constraints and demands for departures and arrivals over time, a mathematical programming model can allocate slot resources to meet the demand. Although linear programming provides guidance to optimize limited airport capacity resources, the modeling framework developed still requires historical operational data and enough samplings to avoid errors. This procedure is difficult to apply to small airports in the NAS, because such airports do not have good historical operational data.

The Macroscopic Capacity Model [Donohue, 1999] developed at George Mason University also addresses capacity issues. The study develops a simple aggregate model based on airport capacity and delay simulation models. The result is a simplified model which can be written as:

$$\text{Cmax} = \sum_{i=1}^{m} AC\max_i - \sum_{j=1}^{n} CL_j \quad (19)$$

$ACmax_i$ - Maximum airport capacities for airport i

CL_j - Air space capacity loss for segment j

The study estimates that 57% of the maximum NAS-wide capacity is used in the year 1999. According to the same study, the system will be operating at 70% of the national capacity by 2010. The study also found that adoption of new technology and operational concepts have a better impact on capacity growth than construction of new runways. One problem with the Macroscopic Capacity Model is that it neglects differences between airports, such as aircraft mix and the number of runways at each airport. It is suitable for regional/system-wide capacity analyses. However, the lack of information on individual airfield makes this model unsuitable for this study.

Welch et al. [Welch, 2001] used steady-state queuing theory to estimate airport system capacity using FAA operational delay data. The study used the Consolidated Operations and Delay Analysis System (CODAS) airborne delay data to calculate effective airport system capacity. The authors concluded that their model works best at airports that experience a steady demand. One limitation of the CODAS, is the availability of data for only 100 airports, which represent less than 2% of all public airports in the NAS.

T.R. Inniss and M.O. Ball developed a model that takes into consideration the stochastic nature of weather and models probabilistic airport capacity [Inniss and Ball, 2002]. The study generates input parameters required for the Ground Delay Program (GDP) and may be used to improve traffic flow management tools. The output of the model serves as an improved decision-support tool for GDP.

Mark Hansen et al., developed a model to assess the impact of expanding airfield capacity [Hansen, 1998]. The results of this study focus on Dallas Fort-Worth International Airport (DFW) and indicate that the impacts and adaptations of carriers and the airport capacity were more modest than expected. Though, airlines did operate at a higher concentration level than the expansion of airport capacity.

Bazargan et al. [Bazargan, Fleming, Subramanian, 2002] used the Total Airspace and Airport Modeler (TAAM) to simulate possible airport runway capacity for proposed expansion alternatives at Philadelphia International Airport (PHL). TAAM provides detail and realism to simulate the airport terminal area. The case study of PHL finds the diagonal runway concept layouts provide the best alternative for airport capacity at PHL.

In Europe, Pitfield and Jerrard [Pitfield and Jerrard, 1999] estimate the unconstrained runway capacity at Rome Fiumicino International Airport to examine the value of such methodology. The unconstrained runway capacity is the maximum runway throughput that can be achieved under ideal conditions, regardless of the level of service. Their study compares different runway operational scenarios and concludes that the concept of unconstrained capacity can help identify crucial operating parameters and optimize the use of the runway system.

Section 2.3 Review of Other Airport Capacity Research

For a more thorough understanding about the airport capacity analysis, it is necessary to assess the impacts of various operational factors that drive airport capacity. This section presents a review of studies about the subject matter.

MITRE proposed a phased approach to increase airport capacity through reduction of current wake turbulence separation standard [MITRE, 2003]. Wake vortex is a by-product of lift, the fundamental force that makes an aircraft fly. The pressure difference between air above and below a wing contributes to the generation of lift, but also produces wake vorticity behind the aircraft. Aircraft exposed to wake vortex turbulence can experience an induced roll that may not be easily corrected at slow approach speeds. To study the behavior and influence of wake vortices, the FAA and the NASA have spent millions of dollars on related wake vortex research. To make sure the turbulence has already decayed to acceptable safety level, air traffic controllers apply conservative spacings between leading and trailing aircraft. A study suggests current separation standard is too conservative for ninety-nine percent of the cases [U.S. House of Representatives, 1995]. MITRE tried to refine the standard to reflect the differences in wake vortex generated by different aircraft weight classes. The study found substantial airport capacity increments maybe possible by reducing separation, given while maintaining the same safety level. MITRE also suggested three development phases for implementing the improvement.

NASA developed the Aircraft Wake Vortex Spacing System (AVOSS) with the goal to use airspace and airport more efficiently [NASA, 2001]. AVOSS integrates a weather data subsystem, a wake vortex prediction model, and wake turbulence sensors to provide separation distance information for up to 30 minutes into the future. The field deployment of this system at Dallas Fort-Worth International Airport indicated that the system is successful and provided an average of 6% increase in throughput capacity.

Reynolds-Feighan and Button [Reynolds-Feighan and Button, 1999] conducted a research study to compare the capacity and congestion levels at European airports. The study collected operational statistical data for airports in Europe and inferred the traffic patterns at these airports. Options for improving management of airport capacity are also discussed in the paper. The research finds the traffic levels play the most important role among the causes of overall delay. Surprisingly, the study could not find an association between traffic growth rates and average delays. Hamzawi [Hamzawi, 1992] proposes four approaches for solving the issue of airport capacity deficit. These include: a) building or expanding facilities, b) reducing the demand by transferring portion of traffic to other modes of transportation or to other airports, c) spread the demands more evenly to alleviate congestion during peak hours, and d) application of new technology. The conclusions suggest that facility expansion is the best long-term solution to overcome the lack of airport capacity. However, other approaches can serve as quick and inexpensive short-term ways to mitigate the capacity problem.

Section 2.4 Summary

Analytic and simulation-based models are complimentary approaches to study airport runway capacity. Both modeling approaches have benefits and their use usually differs in the scope of analysis. For this study that requires the estimation of runway capacity for more than two thousand airports with different physical characteristics and operational strategies, the analytic model is better suited from a practical and computational viewpoint. The analytic model also provides the moderate accuracy needed for the study.

CHAPTER 3 METHODOLOGY

Section 3.1 Introduction

This chapter explains in detail the methodology and data sources used in the airport runway capacity analysis. The methodology is summarized in Figure 5. The study selects appropriate airports from the FAA Airport Master Record, on the basis of whether an airport is qualified to accommodate VLJ vehicles or not. Once the airport set is built, we employ several aviation databases to estimate the number of operations performed by each aircraft weight class (i.e. aircraft mix) for all the airports of interest. Other important parameters, such as aircraft separation matrices, aircraft approach speeds, and runway physical data, are also derived from various databases to estimate runway capacity. The last step in the analysis is to input each airport data into the FAA Airfield Capacity Model (ACM) to perform runway capacity analysis. To streamline the execution process for more than two thousand airports, a series of Matlab programs are written to run ACM in batch mode. These programs not only connect the source data and the ACM model, but also help filter the outputs of ACM. ACM produces multiple capacity data points in every run. The output of this process is the runway capacity for three weather scenarios at each airport (see Figure 5).

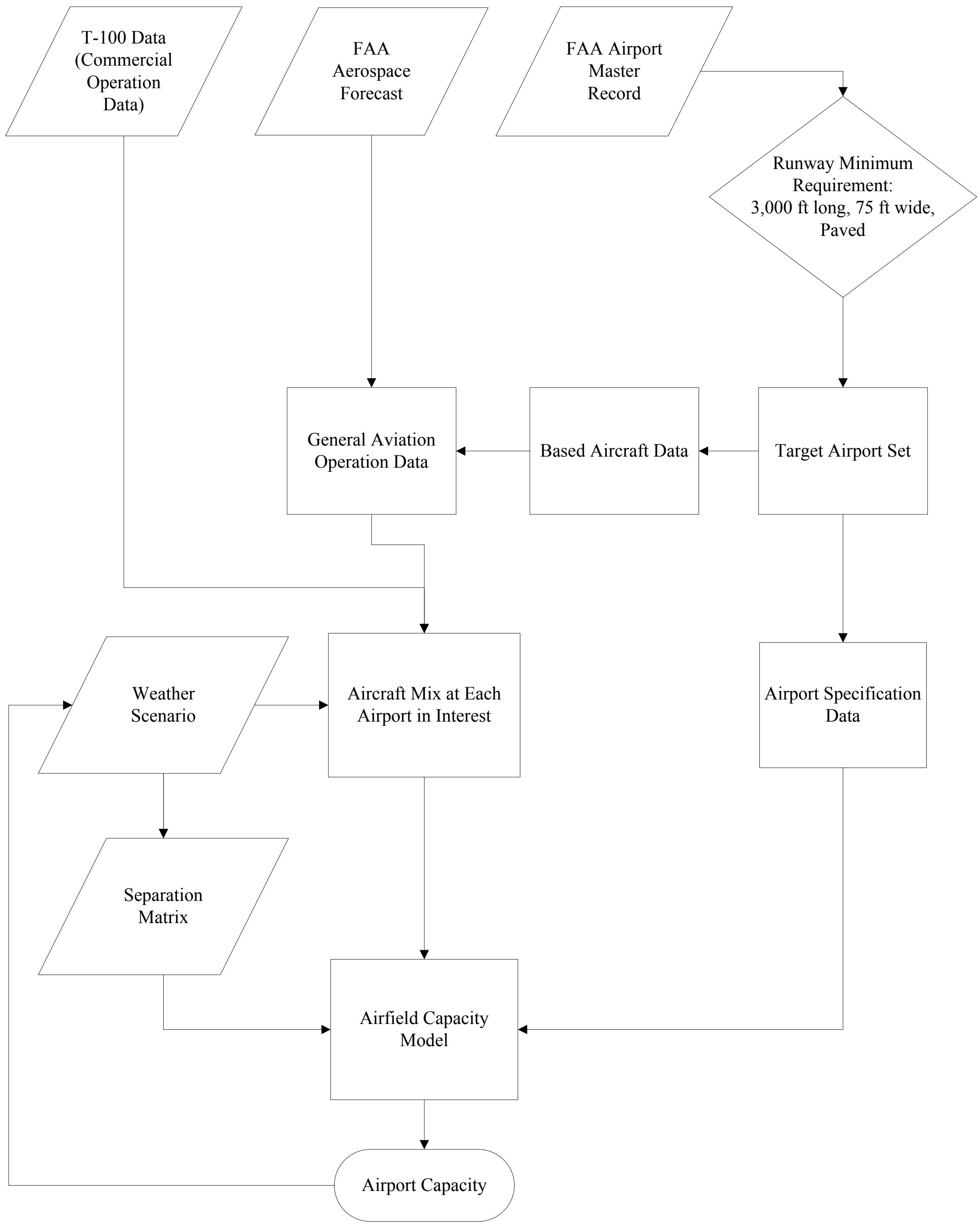

Figure 5: Flowchart of the Airport Runway Capacity Study.

Section 3.2 Methodology to Create Target Airport Set

Section 3.2.1 Overview and Scope of Target Airport Set

The first task of the airport runway capacity analysis is to construct a target airport set. The target airport set is the group of airports that will be considered in the study. In this analysis we are interested in the capacity of airports with commercial operations and airports with potential VLJ operations in the future. In other words, the study finds the minimum requirements of airports where VLJ operations can be conducted safely from a large set of airports in the NAS.

Section 3.2.2 Overview and Scope of FAA Airport Master Record

The study employs the FAA Airport Master Record (Form 5010) as the source of airfield data. The FAA Airport Master Record is the official landing facility database maintained by Office of Aeronautical Information Services of the FAA. It is updated every fifty-six days and contains a comprehensive aviation facility data. The FAA Airport Master Record is comprised of three tables:

a) Facilities Table, which contains the airport facility information,
b) Runways Table. This table provides data on airport runways, and
c) Remarks Table, which includes airport remarks information.

For the needs of this study, the "Facilities" and "Runways" tables were jointed together to construct a single the airport database. The data set used in this study is derived from the Feb. 16th, 2006 version of the FAA Airport Master Record.

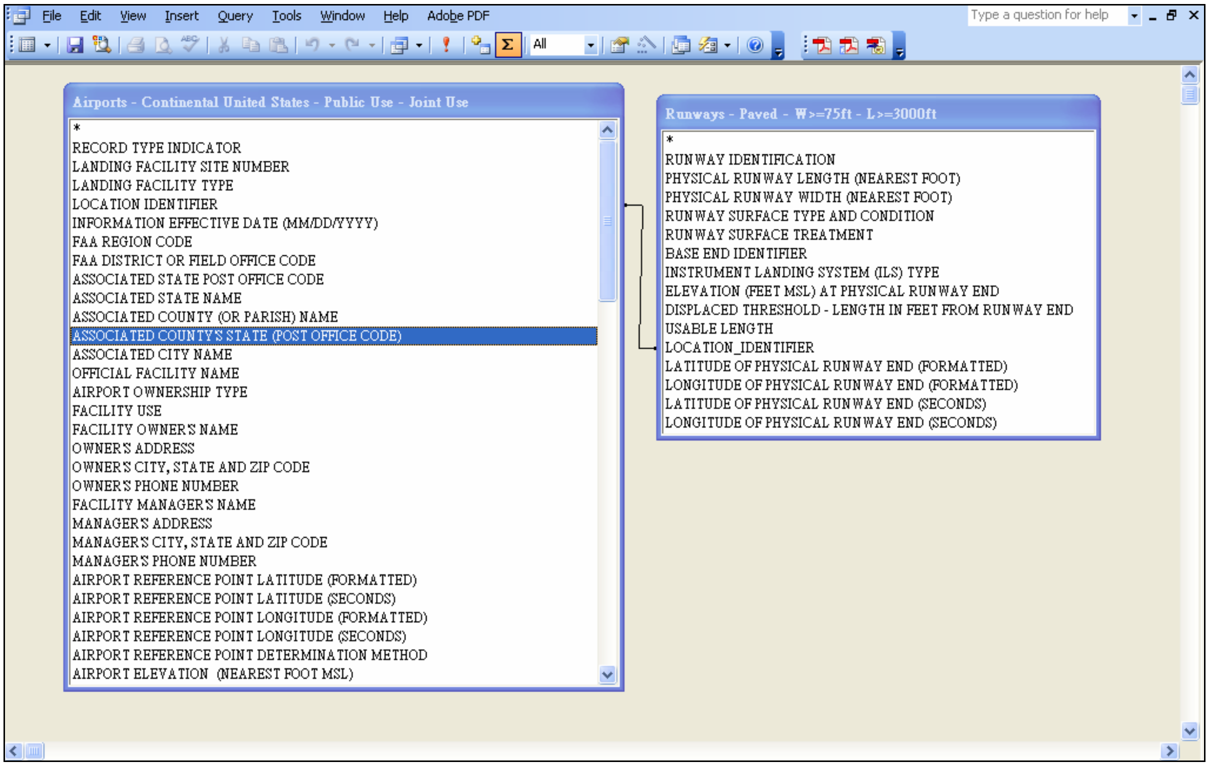

Figure 6: Screen Capture of FAA Airport Master Record Used in the Study.

Section 3.2.3 Procedure to Create Target Airport Set

According to VLJ manufacturer specifications, this group of aircraft requires an effective runway length of 915 meters (3,000 ft). Based on historical performance of light jet aircraft operating today, a realistic correction factor for runway length is to add 12% to the runway length requirement for every 320 meters (1,000 ft) of field elevation above sea level conditions. This correction factor considers temperature variations as well. VLJ are expected to be used in reliable air taxi operations under good and bad weather conditions. Therefore, the minimum runway width to comply with 3/4 of a mile runway visual range criteria is 75 ft.. Applying these criteria to the 2006 FAA Airport Master Record database produces a target airport set of 2,320 airports.

Section 3.3 Methodology to Estimate Aircraft Mix

Section 3.3.1 Overview of Aircraft Mix

One of the most important factors affecting the airport runway capacity is the aircraft mix. The performance specifications of aircraft, such as approach speed, separation requirements between successive aircraft, and runway occupancy time, all play important roles in airport runway capacity estimation. Considering several hundred aircraft types operating in the NAS, the research classifies aircraft by their maximum takeoff weight. The reason behind this is that wake vortex, a major driver in the separation between leading and trailing aircraft, is governed by the weight of the leading aircraft. Classifying aircraft by weight allow us to apply the same separation rules to vehicles in the same category. Hence, all vehicles operating in the NAS are classified into four aircraft classes: Small, Large, Boeing 757, and Heavy, according to their maximum takeoff weights. Table 4 shows the aircraft classes and their definitions used in this research.

Table 4: Description of Aircraft Weight Classes.

Class	Description
A380	Airbus A380 (hasn't yet introduced into commercial service)
Heavy	Aircraft with takeoff weights of more than 255,000 pounds.
B757	Boeing B-757
Large	Aircraft of more than 41,000 pounds, with a maximum certificated takeoff weight, up to 255,000 pounds.
Small	Aircraft of 41,000 pounds or less maximum certificated takeoff weight.

[Federal Aviation Administration Pilot/Controller Glossary (P/CG), 2005]

Aircraft mix is an index of aircraft with different performance characteristics [Pfleiderer, E.M., 2005]. In this study, operations at each airport are classified according to the vehicle weight. The aircraft mix represents the percentage of aircraft in each aircraft class: Small, Large, B757, and Heavy. Aircraft mix drives aircraft separation requirements and thus affects runway capacity. To perform the runway capacity calculations, we estimate aircraft mix at all airports in the Target Airport Set.

Section 3.3.2 Overview and Scope of Aircraft Mix Data Sources

To calculate aircraft mix, several sources are employed. These are the Bureau of Transportation Statistics (BTS) T-100 and the FAA Airport Master Record Based Aircraft Data. BTS T-100 is used to calculate commercial operations in Target Airport Set. The FAA Airport Master Record Based Aircraft Data is applied to find general aviation based aircraft (a precursor of general aviation activity at the airport). The flight counts from these two data sources are added up to estimate each of the aircraft classes operating at every airport in the Target Airport Set. These four numbers are transformed to derive percentage of each vehicle class, and ultimately the aircraft mix.

The comparison of all data sources is listed in the Table 5.

Table 5: Comparison of Aviation Industry Databases for Aircraft Mix Calculation.

Source	**Content**	**Coverage**	**Accuracy**	**Forecasts for future years**
BTS T-100	commercial operations	630 airports	O	N/A
Official Airlines Guide (OAG)	scheduled commercial operations	471 airports	O	N/A
Enhanced Traffic Management System (ETMS)	Instrument Flight Rule (IFR) operations		O	N/A
FAA Airport Master Record Based Aircraft Data	Aircraft based at each airport	20,429 landing facilities	O	N/A
Terminal Area Forecast (TAF)	All operations	3,580 airports	O (for tower airport) Δ (for non-towered airport)	O

O: Good, Δ: Fair, X: Poor

Table 5: Comparison of Aviation Industry Databases for Aircraft Mix Calculation. (Cont'd).

Source	Units	Advantages	Disadvantages
BTS T-100	Operations	Provide detailed commercial operation data	
Official Airlines Guide (OAG)	Operations		Doesn't have non-scheduled data
Enhanced Traffic Management System (ETMS)	Operations		Doesn't have Visual Flight Rule (VFR) data
FAA Airport Master Record Based Aircraft Data	Aircraft	Includes based aircraft information for all landing facilities	
Terminal Area Forecast (TAF)	Operations	Has forecast data	Integrated operation data; doesn't have number by type or weight class

O: Good, Δ: Fair, X: Poor

Section 3.3.3 Overview and Scope of BTS T-100 Data

The BTS T-100 Air Carrier Statistics database is a monthly commercial aviation traffic data reported by certificated U.S. and foreign air carriers. The main advantage of using the T-100 data is its reliability and coverage, which includes not only scheduled passenger flights, but also cargo and nonscheduled operations. This kind of completeness provides a clearer picture of the real composition of aircraft mix, especially for airports with freight operations or charter flights. The T-100's long term monthly traffic data begins in 1990 therefore it provides a wealth of data for our analysis.

There are four tables in the T-100 data bank, including domestic market table, domestic segment table, international market table, and international segment table. For the needs of this study, we use segment data, which includes both domestic and international non-stop segments operated by U.S. and foreign carriers, as a source of commercial operations. The T-100 data is used to estimate aircraft mix. Appendix C lists all the fields available in the T-100 segment data. Please refer to the BTS web site [www.bts.gov] for further details about the BTS T-100 data.

Section 3.3.4 Procedure to Calculate Commercial Operations Using BTS T-100 Air Carrier Statistics

We estimate departure and arrival operations at the airports of interest. As explained in the Section 3.2, we use the domestic T-100 data to construct a dataset with the appropriate records. The structure of this database is show in Figure 7.

The T-100 does not have aircraft weight class information. To address this issue, we assign the T-100 to an aircraft class using the FAA Aircraft Characteristics Database (see Figure 8) by inspecting the maximum certificated takeoff weight of each vehicle type. The completed operational data, which contains weight class information for all commercial operations at all target airports, is then sorted by weight classes so that a total number of flights in each class at each airport can be estimated. The results of this process are kept for future reference to calculate aircraft mix. Figure 9 illustrates the procedure to estimate commercial operations at airports.

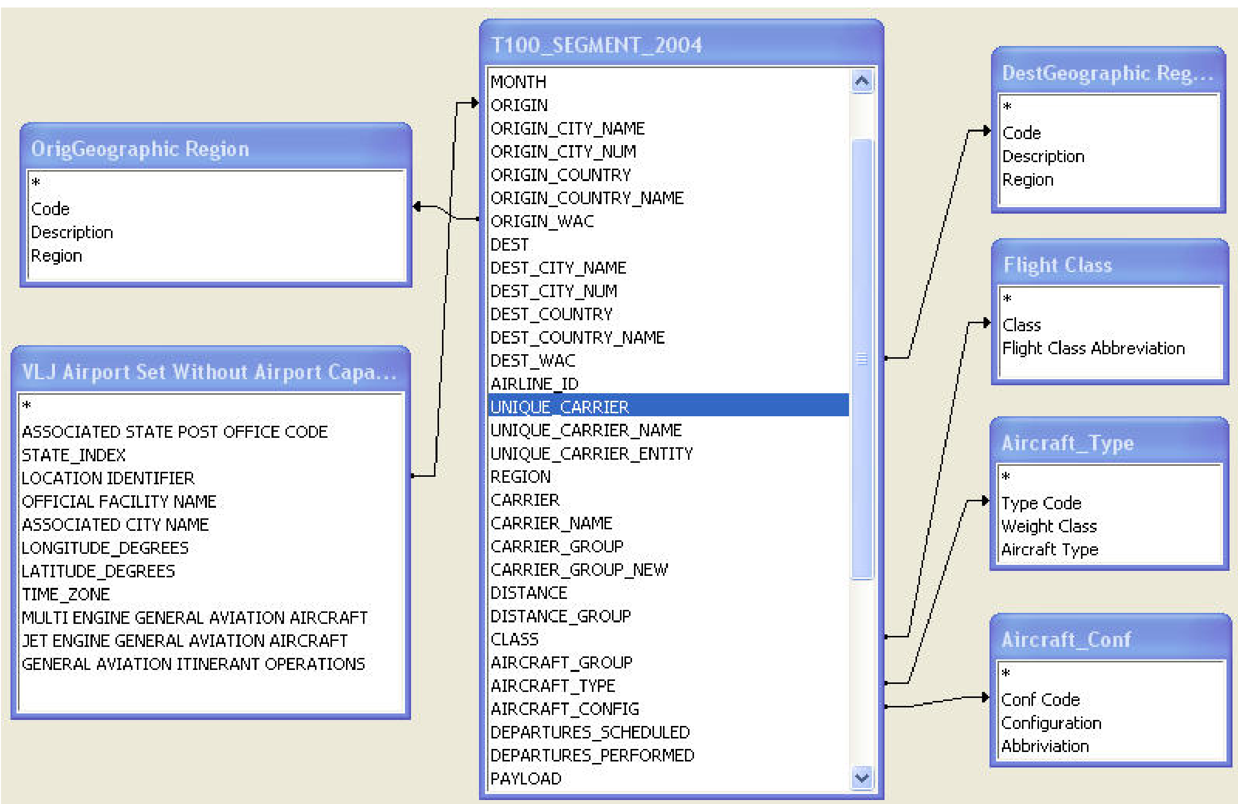

Figure 7: Database Structure for Filtering BTS T-100 Data.

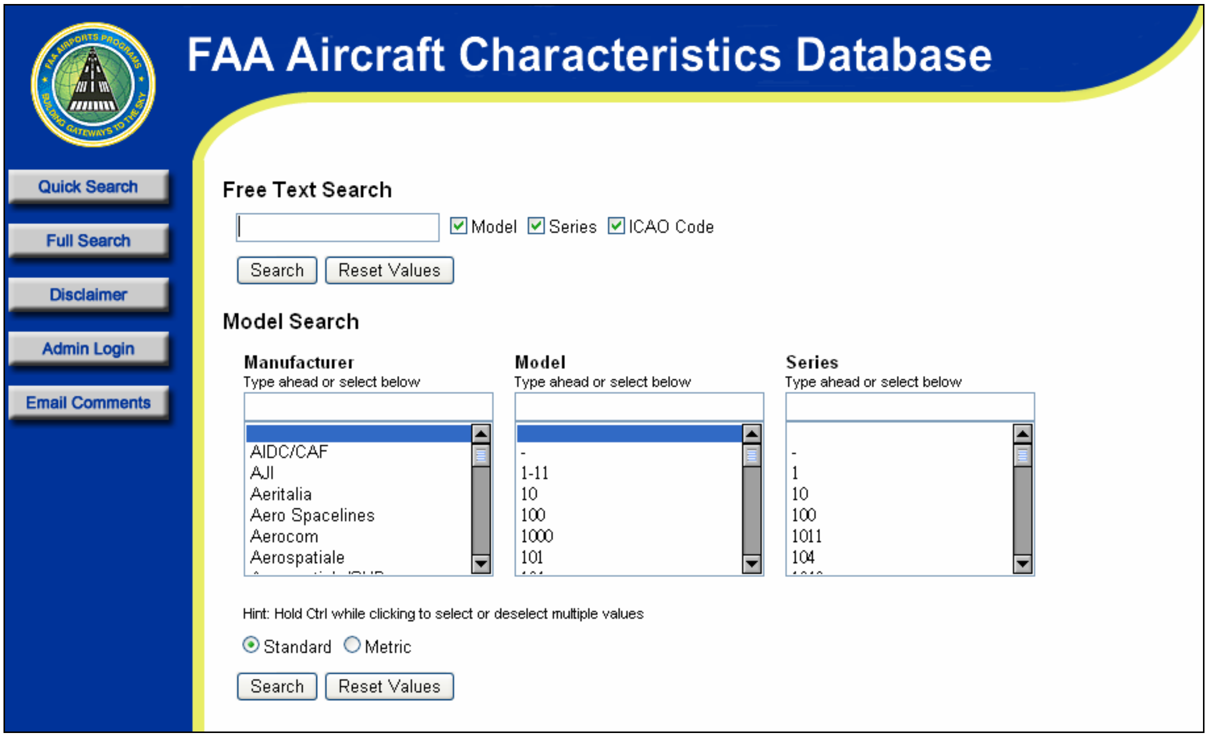

Figure 8: Screen Capture of the FAA Aircraft Characteristics Database.

(Source: FAA)

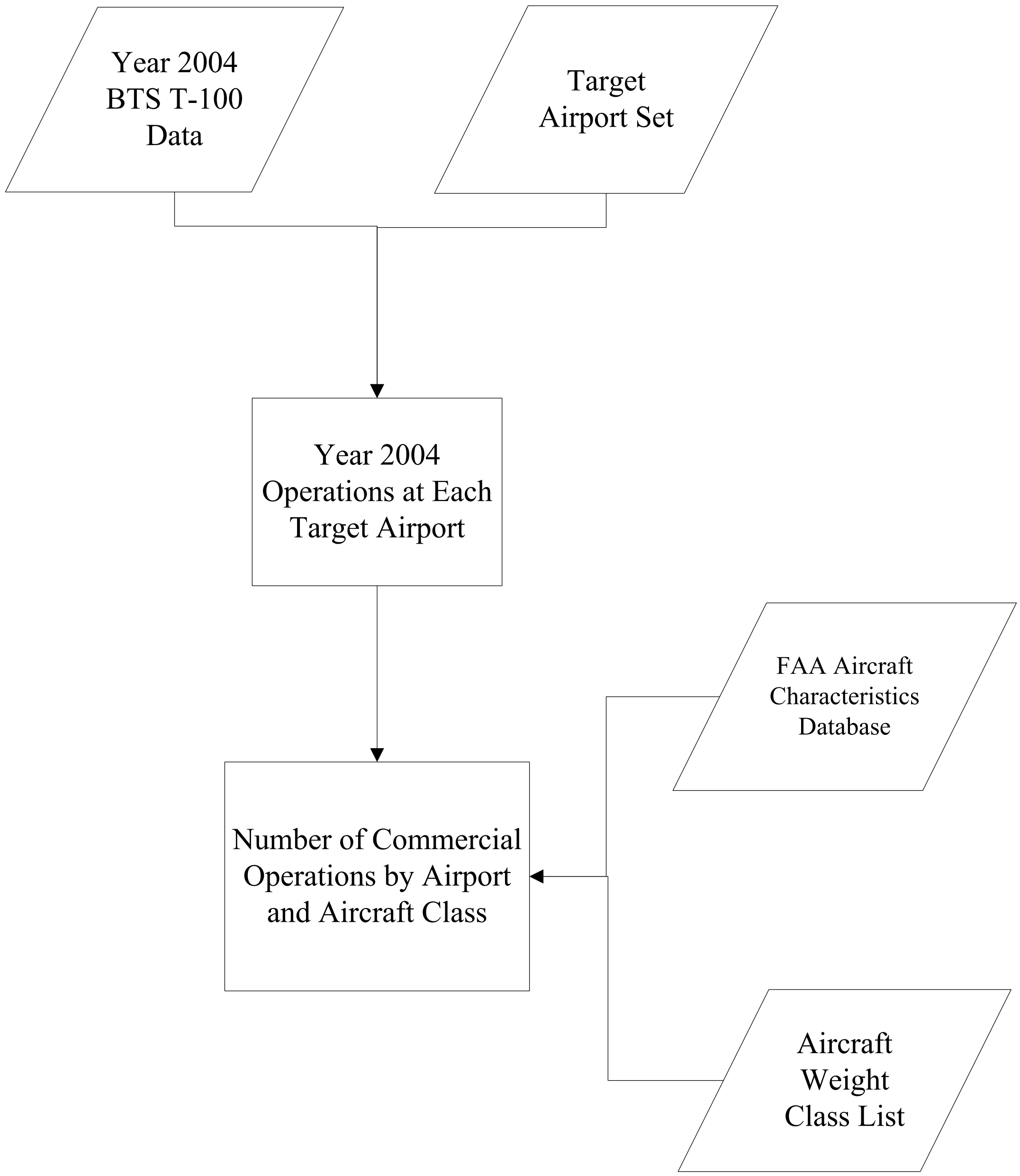

Figure 9: Flowchart of Calculating Commercial Operation at Target Airports.

Section 3.3.5 Overview and Scope of Based Aircraft Data

As stated in Chapter 1, the majority of aviation activities in the NAS today are the general aviation. However, there is no database like the T-100 to commercial operations that records every general aviation flight. The study uses the FAA Airport Master Record to infer general aviation operations at the airport level.

The FAA Airport Master Record provides facility physical information. It also provides the number of aircraft based at each airport by their engine types. In the FAA database there are three fields: "Single Engine General Aviation Aircraft (based)", "Multi Engine General Aviation Aircraft (based)", and "Jet Engine General Aviation Aircraft (based)". Compared to other databases which provide general aviation activity, the FAA Airport Master Record based aircraft information is believed to be more reliable and its coverage is also broader.

Section 3.3.6 Procedure to Estimate General Aviation Operations Using Based Aircraft Data

The based aircraft numbers are used to predict the number of general aviation operations at each airport. To accomplish this, we study the utilization rate for general aviation aircraft to project the annual general aviation operations at each airport. Table 6 shows the general aviation aircraft annual utilization derived from a previous study [Baik et al., 2006]. The result of this step gives the number of operations by aircraft engine type at each airport. In a follow-up step we find the distribution of weight classes among these general aviation aircraft. These distributions allow us to make a projection from "operations by engine type" to "operations by aircraft weight class".

Table 6: General Aviation Aircraft Activity and Utilization. [Baik et al, 2006]

Aircraft Type	**Estimated Average Annual Utilization (hours)**	**Total Hours Flown (thousands)**	**Total Operations (thousands)**
Single Engine	128	6,980	8,318
Multi Engine	170	2,760	4,776
Jet Engine	320	1,210	2,302

To make sure the general aviation operations are projected into appropriate aircraft classes, we make the following assumptions:

1. All single engine and multi engine general aviation aircraft belong to the "Small" class category. The reason for this assumption is that the vast majority of propeller driven general aviation aircraft today have maximum takeoff weight of less than 41,000 pounds.

2. To split the jet engine general aviation aircraft, we use the runway length at the airport as a criterion. Previous research indicates that, in general, jet engine general aviation aircraft belonging to weight class “Large” require a minimum runway length of 5,000 feet for takeoff [Bonnefoy and Hansman, 2006]. Hence, this threshold is chosen to split jet engine aircraft operations. If an airport has runways less than 5,000 feet in length, all based jet engine general aviation aircraft are assigned to the “Small” weight category. If the airport has runway(s) with length of 5,000 feet or longer, we assign 10% of the jet engine general aviation operations to the “Large” class and 90% to the “Small” class. Statistics derived from BUCHair BizJets Database [BUCHair, 2004] indicates that jet engine aircraft in weight class “Large” accounts for about 10% of all registered business jets in the world.

The process to calculate the number of general aviation operations at each airport is illustrated in Figure 10.

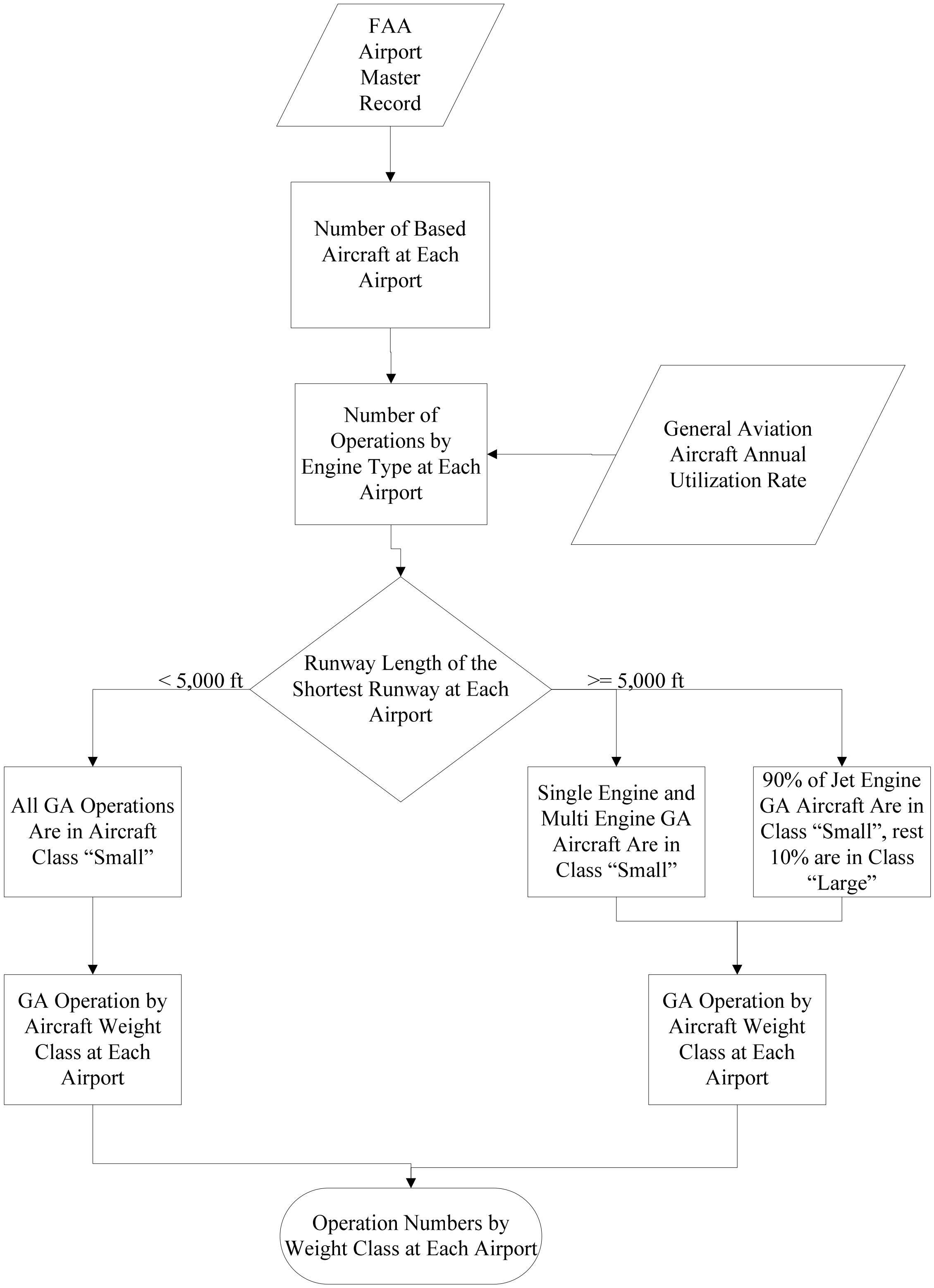

Figure 10: Flowchart to Estimate the Number of General Aviation Operation.

Section 3.3.7 Procedure to Calculate Aircraft Mix

Sections 3.3.4 and 3.3.6 described methods to calculate operations for commercial and general aviation aircraft. The last step is to synthesize the calculation of aircraft mix at all target airports. In this study, we add the number of flights in same aircraft weight class at each airport. Figure 11 is a flowchart of the process.

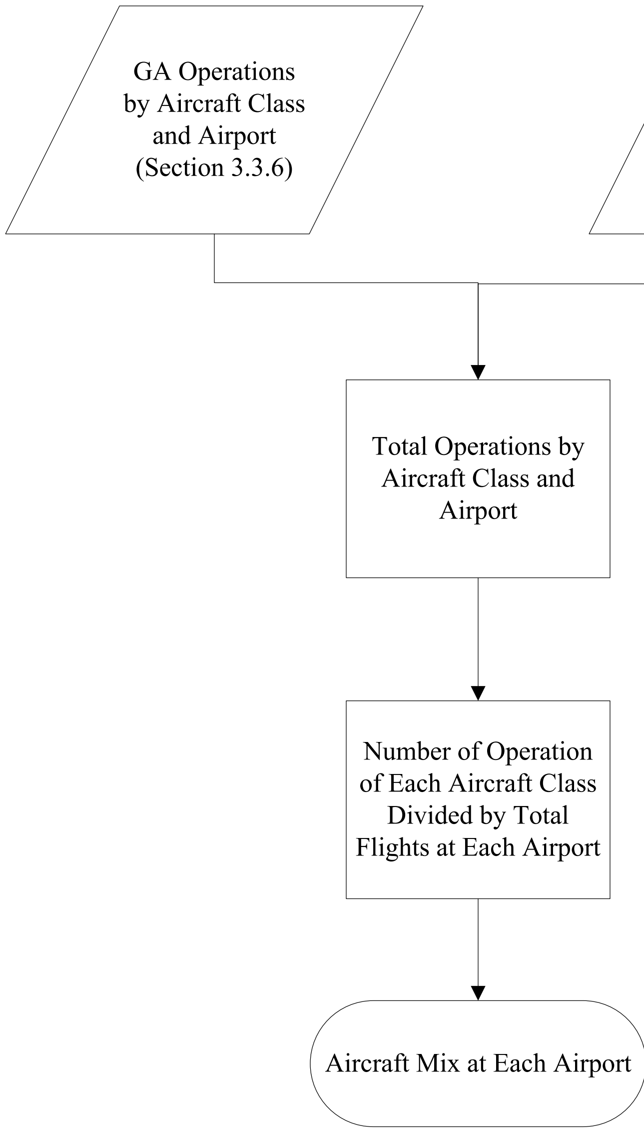

Figure 11: Flowchart of Calculating Aircraft Mix.

Section 3.4 Airfield Capacity Model Analysis

Section 3.4.1 Overview and Scope of ACM

The Airfield Capacity Model (ACM) is a computer program developed by the FAA and the MITRE Corporation [ACM Users Guide, 1981] to estimate airport capacity. The main output of the ACM model is to calculate the runway system maximum throughput capacity without consideration of delays. The capacity obtained from the ACM may be different from the actual throughput experienced at an airport, due to the aircraft mix uncertainties at a specific time period or to the assumed random sequence of aircraft arrivals. The theoretical capacity the ACM presents a good first order approximation of the runway capacity under various operational strategies and makes the procedure useful to compare between runway use alternatives.

The ACM uses the average separation time between arrivals to calculate arrivals-only capacity. The mix aircraft operation strategy estimates the probability of safely inserting departure(s) between consecutive arrivals. The number of departures is calculated and added to the arrival-only capacity. This capacity number is called "arrival-priority capacity". In addition to arrival-only capacity and arrival-priority capacity, ACM also calculates capacity with departures only. The capacity obtained is called "departures-only capacity". With "arrivals-only capacity", "arrivals-priority capacity", and "departures-only capacity", the ACM can estimate the capacity for any specific arrival-departure composition.

ACM can estimate several "Runway Use Configuration Geometries" for every airport. In our analysis, all multi-runway airports are classified into four runway configuration types:

a) parallel runways,
b) intersecting runways,
c) open-v runways, and
d) intersecting beyond threshold runways.

Using this classification, ACM calculates a limited number of runway use configurations reducing the computation time. To make sure the ACM generates accurate estimation, users have to choose an appropriate runway use model, based on the runway layout at each airport. Table 7: and Figure 12 have a complete list of all runway configuration models available in ACM.

Table 7: List of the ACM Runway Configuration Models.

Model No.	Geometry
1	Single Runway
2	Two Parallel Runways
3	three Parallel Runways
4	Four Parallel Runways
5	Two Open V Runways
6	Two Intersecting Runways
7	Three Intersecting Runways
10	Three Open V Runways
11	Four Open V Runways
12	Two Runways Intersecting Beyond Threshold
13	Three Runways Intersecting Beyond Threshold
14	Four Runways Intersecting Beyond Threshold
15	Four Intersecting Runways

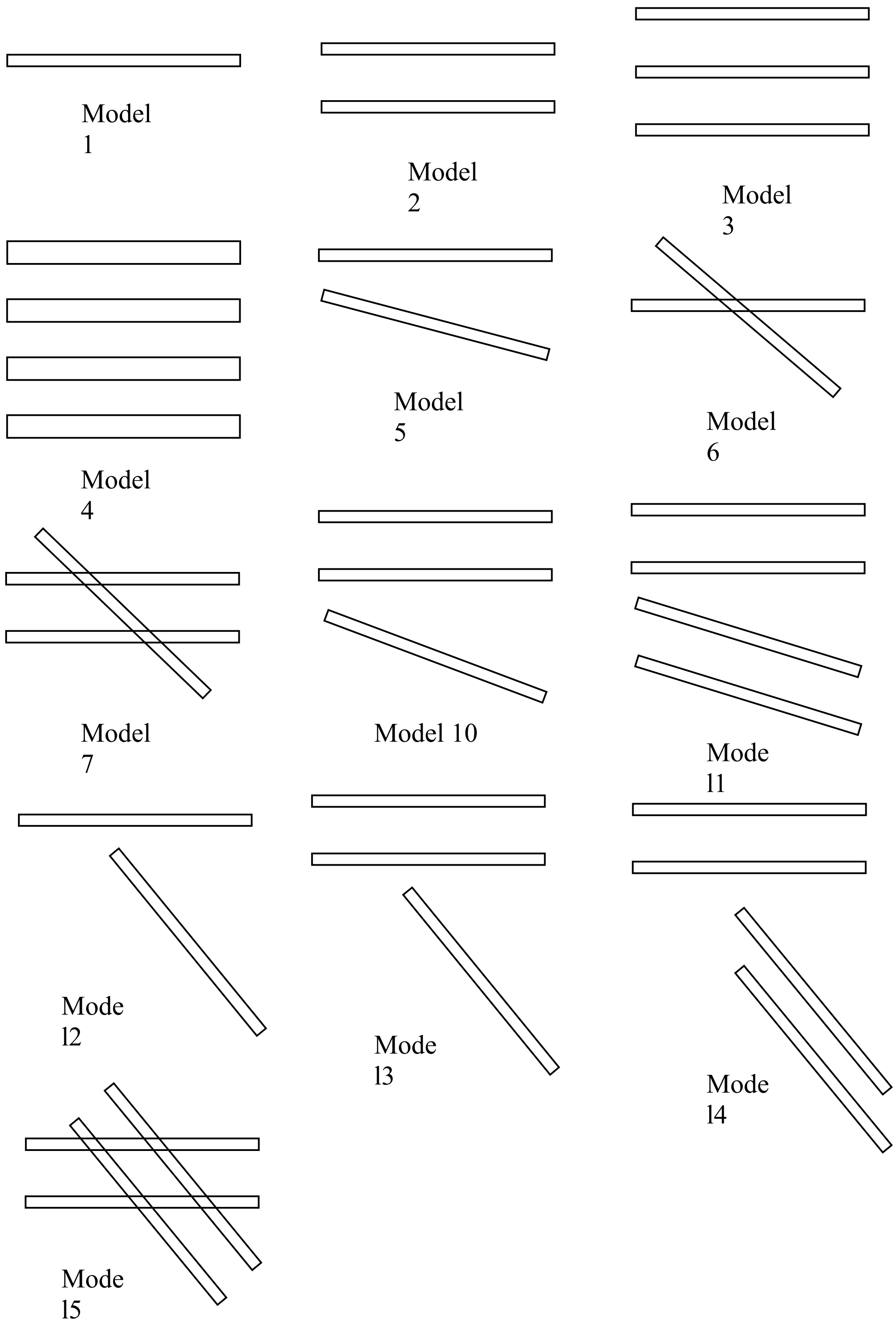

Figure 12: Layout of ACM Runway Configuration Models.

Beyond runway layout information, the runway operational strategy is also an important to estimate runway capacity. For example, a two-runway airport can use one runway for landings only and the other runway for takeoff operations. Alternatively, both runways could process landings and takeoffs. These variations lead to different runway capacity values. To take this issue into consideration, the ACM model presets operating scenarios and lets users to choose an appropriate runway use.

Since each airport has its own consideration for choosing a runway operational strategy, and considering that there is no nation-wide database on how airports use runways, our analysis calculates the capacity of all applicable runway operational strategies and compares the results. For more details about the ACM's calculation logic and methodology, please refer to the ACM Users Guide [ACM Users Guide, 1981].

Section 3.4.2 Assumptions about various ACM Parameters

To maintain consistencies between scenarios among the Target Airport Set, the following assumptions about ACM inputs are made:

1. Alternating arrivals are to be conducted.
2. Aircraft separation adopts reference values used in the NAS today. For non-towered airports, the separation between all aircraft classes is 18.6 NM under IFR conditions. Tables 8 through 11 show the separations used in the study.

Table 8: VFR Departure-Departure Separations for Tower and Non-Towered Airports.

Tower		Following	Following	Following	Following
	VFR	Small	Large	757	Heavy
Lead	Small	45	45	45	45
Lead	Large	60	60	60	60
Lead	757	120	120	120	120
Lead	Heavy	120	120	120	120

(seconds)

Non-Towered		Following	Following	Following	Following
	VFR	Small	Large	757	Heavy
Lead	Small	45	45	45	45
Lead	Large	60	60	60	60
Lead	757	120	120	120	120
Lead	Heavy	120	120	120	120

(seconds)

Table 9: IFR Departure-Departure Separations for Tower and Non-Towered Airports.

Tower		Following	Following	Following	Following
	IFR	Small	Large	757	Heavy
Lead	Small	45	90	90	90
Lead	Large	60	60	60	60
Lead	757	120	120	120	120
Lead	Heavy	120	120	120	120

(seconds)

Non-Towered		Following	Following	Following	Following
	IFR	Small	Large	757	Heavy
Lead	Small	240	240	240	240
Lead	Large	120	120	120	120
Lead	757	120	120	120	120
Lead	Heavy	120	120	120	120

(seconds)

Table 10: VFR Arrival-Arrival Separations for Tower and Non-Towered Airports.

Tower		Following	Following	Following	Following
	VFR	Small	Large	757	Heavy
Lead	Small	1.9	1.9	1.9	1.9
Lead	Large	2.7	1.9	1.9	1.9
Lead	757	3.5	3.0	3.0	2.7
Lead	Heavy	4.5	3.6	3.6	2.7

(Nautical Miles)

Non-Towered		Following	Following	Following	Following
	VFR	Small	Large	757	Heavy
Lead	Small	1.9	1.9	1.9	1.9
Lead	Large	2.7	1.9	1.9	1.9
Lead	757	3.5	3.0	3.0	2.7
Lead	Heavy	4.5	3.6	3.6	2.7

(Nautical Miles)

Table 11: IFR Arrival-Arrival Separations for Tower and Non-Towered Airports.

Tower		Following	Following	Following	Following
	IFR	Small	Large	757	Heavy
Lead	Small	3.0	3.0	3.0	3.0
Lead	Large	4.0	3.0	3.0	3.0
Lead	757	5.0	4.0	4.0	4.0
Lead	Heavy	6.0	5.0	5.0	4.0

(Nautical Miles)

Non-Towered		Following	Following	Following	Following
	IFR	Small	Large	757	Heavy
Lead	Small	18.6	18.6	18.6	18.6
Lead	Large	18.6	18.6	18.6	18.6
Lead	757	18.6	18.6	18.6	18.6
Lead	Heavy	18.6	18.6	18.6	18.6

(Nautical Miles)

3. Three preset weather scenarios:
    a. VMC- Visual Meteorological Conditions; in VMC, Visual Flight Rules (VFR) apply
    b. IMC- Instrument Meteorological Conditions; in IMC, Instrument Flight Rules (IFR) apply
    c. MMC- Marginal Meteorological Conditions; according to the ACM User's Manual, MMC only affects certain runway configurations. Including a mixed-operation single runway, or a close-spaced parallel pair with one runway for arrivals and another one for departures. MMC is basically an IFR environment, but with a better visibility so the 2 NM departure/arrival separation is superseded by visual separations.
4. Has no touch-and-go operation.
5. For intersecting runways, aircraft are not airborne at intersection.
6. There are 11 different arrival/departure composition scenarios: departure-only, 10% arrival, 20% arrival, 30% arrival, 40% arrival, 50% arrival, 60% arrival, 70% arrival, 80% arrival, 90% arrival, and arrival-priority.
7. For parameters not mentioned, we use the defaults of the program. Please refer to Appendix E-3 for detailed information.

Section 3.4.3 Procedure to Run ACM and Output

There are two modes a user can run ACM:

1. Interactive Mode- Under this mode, ACM prompts the user to enter input parameters through an interactive window. With the help of options and explanations the ACM shows on interactive window, this mode is useful for users who are not familiar with the ACM. However, since the ACM stops and waits for inputs at each step, it takes longer time to perform a single capacity analysis. This mode is only suitable for a single case study.
2. Batch Mode- Batch Mode is a more efficient way to perform the ACM capacity analysis. In this mode, all the input parameters are compiled in a specified format and saved in a text file. The ACM program then reads the input files and produces output files which contain capacity results. The advantage of the batch mode is its ability to

deal with numerous data in one step. For this study, the batch mode serves our purpose (i.e., more efficient).

To execute the analysis for the whole Target Airport Set in one step, the study uses Matlab scripts to connect the ACM and the data of the target airports. Four Matlab programs are written to serve different needs in the process. The steps and functions of these programs are described below:

1. Runway Classification- The purpose of this program is to calculate the distance and angle between runways. The information obtained is used to classify the airport configuration. Airports are classified by the number of runways (i.e., from one runway to five runways) and also by the runway configuration: parallel, intersecting, open-v, and intersecting beyond threshold runways. Airports with runway configurations that do not fit into any preset types are treated separately.
2. ACM Parser- A separate Matlab program is written to transform the data in spreadsheet file format into ACM input form. Once the input files are prepared, this program executes ACM and saves the capacity results in individual text files for each airport configuration tested.
3. Results Collection- The output produced by ACM is a file in text format. (Please refer to Appendix D for a sample ACM output file). To make the capacity results easier for future comparisons and further processing, the Matlab program here mines the text file for capacity numbers and save the figures in a spreadsheet.
4. Result Comparison- This program selects an appropriate capacity number from various operational strategies tested by ACM. The Matlab program is written here to find the range of maximum runway capacity. The largest number among capacity numbers is the upper-boundary of the runway capacity, and the smallest number among results of different strategies is the lower-boundary. This process defines the range of maximum capacity for each scenario at each airport runway configuration.

The procedures stated above generate two capacity streams: one for upper-boundary arrival-priority capacity, and a second one for lower-boundary arrival-priority capacity. This applies to runway system and each weather scenario. Both capacity streams are composed of capacity numbers from 11 different arrival/departure strategies calculated by ACM. The streams are presented in the form of departure-arrival capacity diagram. Figure 13 is an example of departure-arrival capacity diagram (i.e., pareto frontier diagram). Please see Appendix E for complete Matlab program listings.

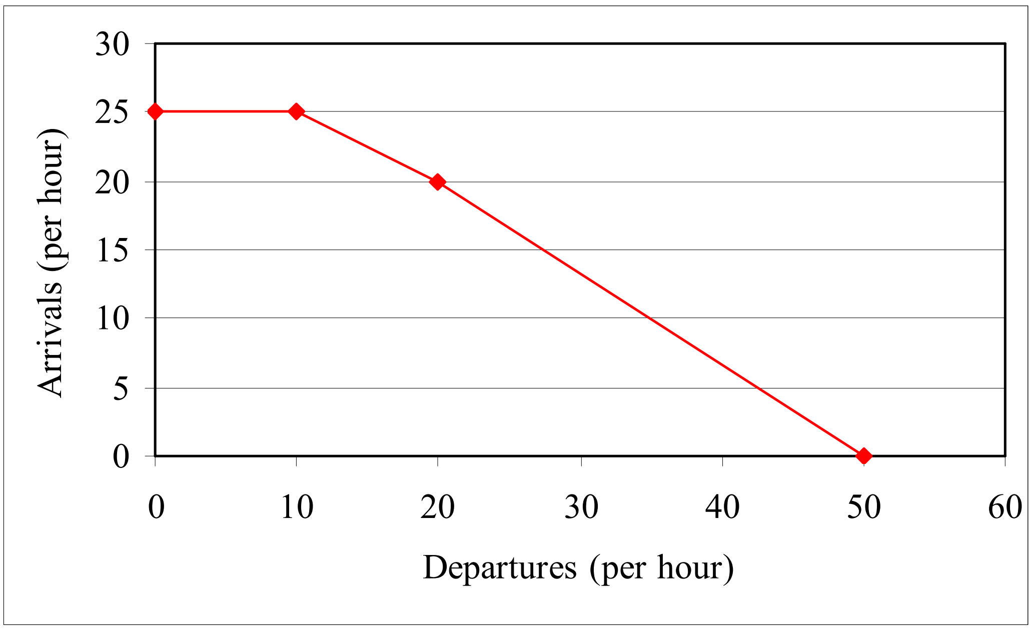

Figure 13: Sample Departure-Arrival Capacity Diagram (e.g., pareto frontier)

Section 3.4.4 Applying ACM to Other Airports in the Target Airport Set

For airports with layouts that do not fit into ACM preset configuration patterns, we select a runway configuration in ACM that has similar characteristics to the one sought and use such configuration for runway capacity analysis.

Section 3.5 Supplementary Runway Capacity Analysis Data

The FAA has completed a capacity analysis for the top 103 major airports in the US. This includes the 35 busiest commercial airports in the nation. Considering the complexity of operations and configurations of these 103 airports, we adopt the capacity numbers provided by the FAA in the TSAM airport capacity module.

Among these 103 airports, 4 airports are eliminated from the study because they are not in the continental U.S. These are Ted Stevens Anchorage International Airport (ANC), Juneau International Airport (JNU), and Honolulu International Airport (HNL). Fort Lauderdale Executive Airport (FXE) is also included in this short list. The numbers for the rest 99 airports in the FAA analysis will be integrated into this study to provide a full coverage of all target airports.

CHAPTER 4 RESULTS AND VERIFICATION

This chapter presents the analysis and results of the airport runway capacity calculations. The processes described in Chapter 3 are executed for 2320 airports making up the target airport set. The runway capacities for each airport are estimated and saved. The results for the aforementioned processes are presented in the following order: First the composition of the Target Airport Set is described in Sections 4.1 and 4.2. The chapter also discusses the results of estimated operations and aircraft mix at each airport. Section 4.3 presents the capacity numbers for different weather scenarios at all airports. Section 4.4 prescribes the application of the model developed using future operational parameters and the result. The verification of the model results is described in Section 4.5.

Section 4.1 Composition of the Target Airport Set

After examination of all airports in the FAA Airport Master Record, the research found 2,320 qualified airports for this study. These airports have runway(s) with a minimum effective length of 3,000 feet (corrected by altitude), and with a minimum width of 75 feet. The pavement attributes of qualified runways must be able to accommodate safe operations by Very Light Jets. Figure 14 shows the geographical distribution of the airports of interest. For a complete list of these airports, please refer to Appendix F.

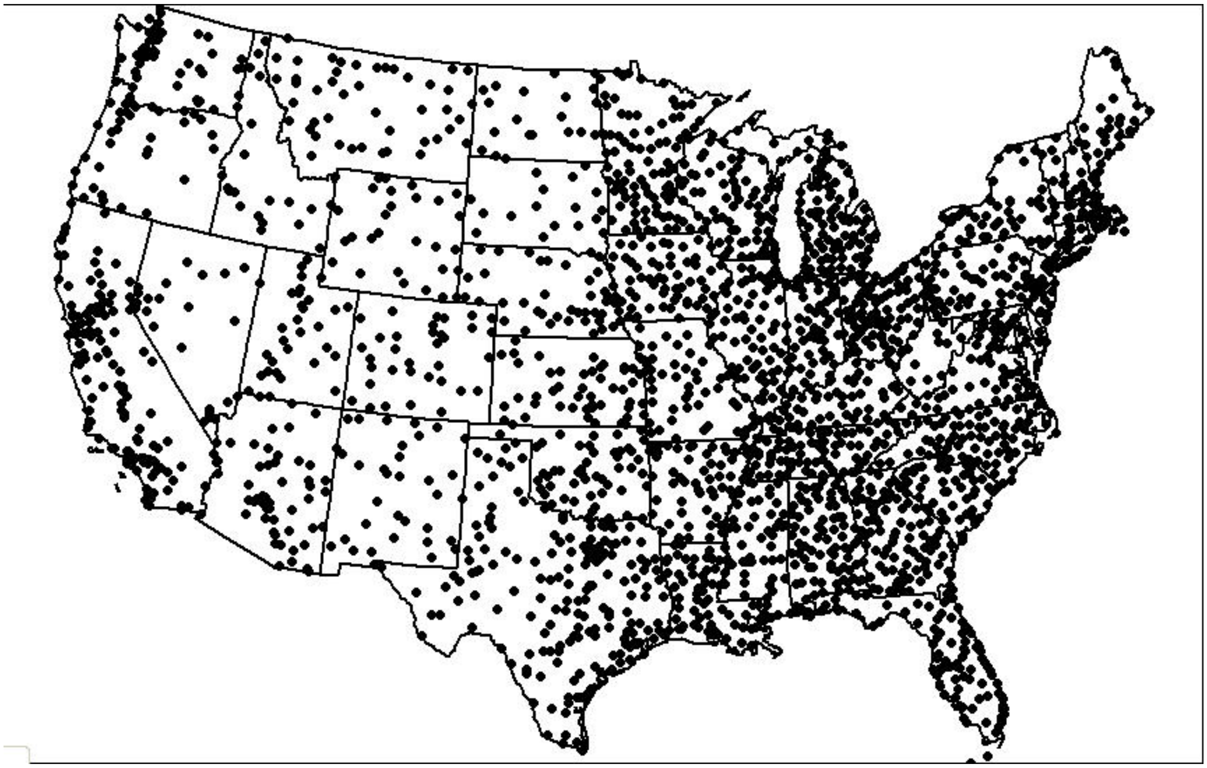

Figure 14: Map of the Target Airport Set.

To have a clearer picture about the composition of the Target Airport Set, the study analyzes the Airport Set by its characteristics. This series of analyses starts with the number of operations at each airport: commercial flights or non-commercial flights (general aviation operations). Figure 15 illustrates the number and percentage of airports with two operation types. The figure shows that 27% of 2,320 target airports have commercial operations, and the rest of them have only general aviation operations.

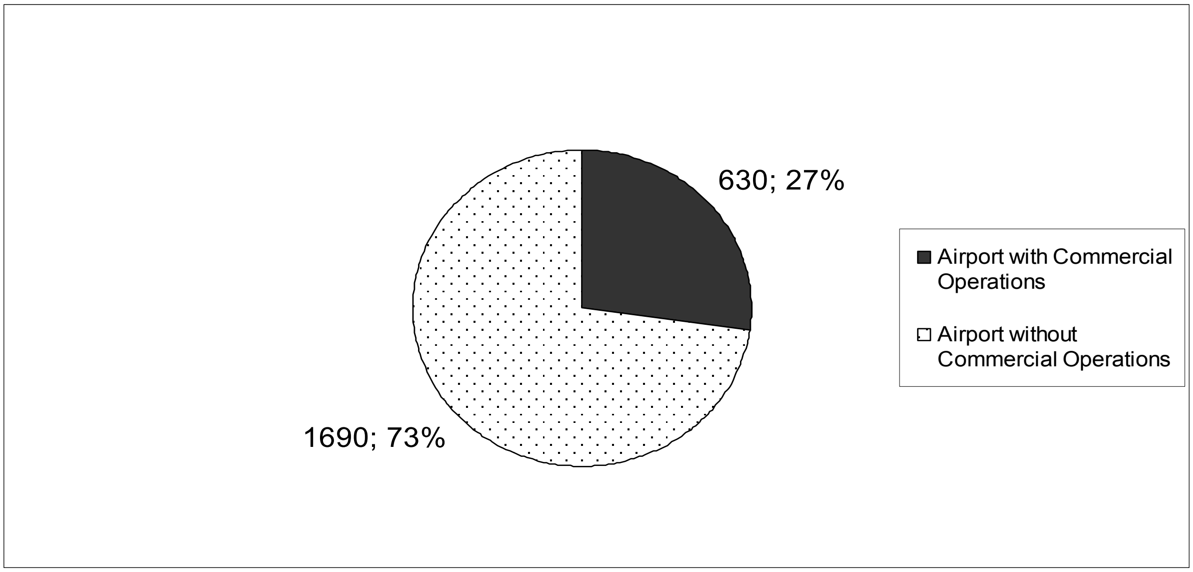

Figure 15: Airport Composition by Operation Type.

The Target Airport Set is also studied by the number of runways at each airport. Figure 16 demonstrates the distribution of airports by number of runways. The result shows that 92% of airports in the Target Airport Set have two or fewer runways.

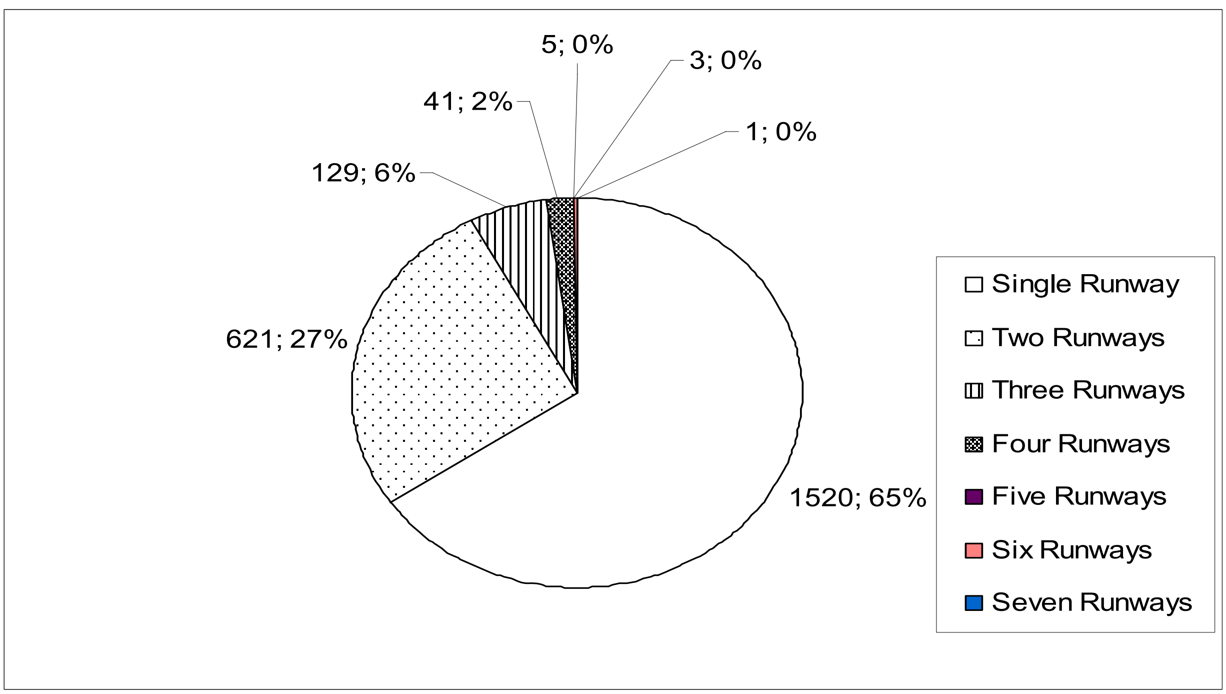

Figure 16: Airport Composition by Number of Runways.
(Number of Airport; Percentage)

The availability of an air traffic control tower or availability of air traffic services on site is also an important factor affecting the runway capacity of an airport. This is more important in bad weather. The study finds that 1,811 airports (i.e., 78% of them) do not have a control tower on site.

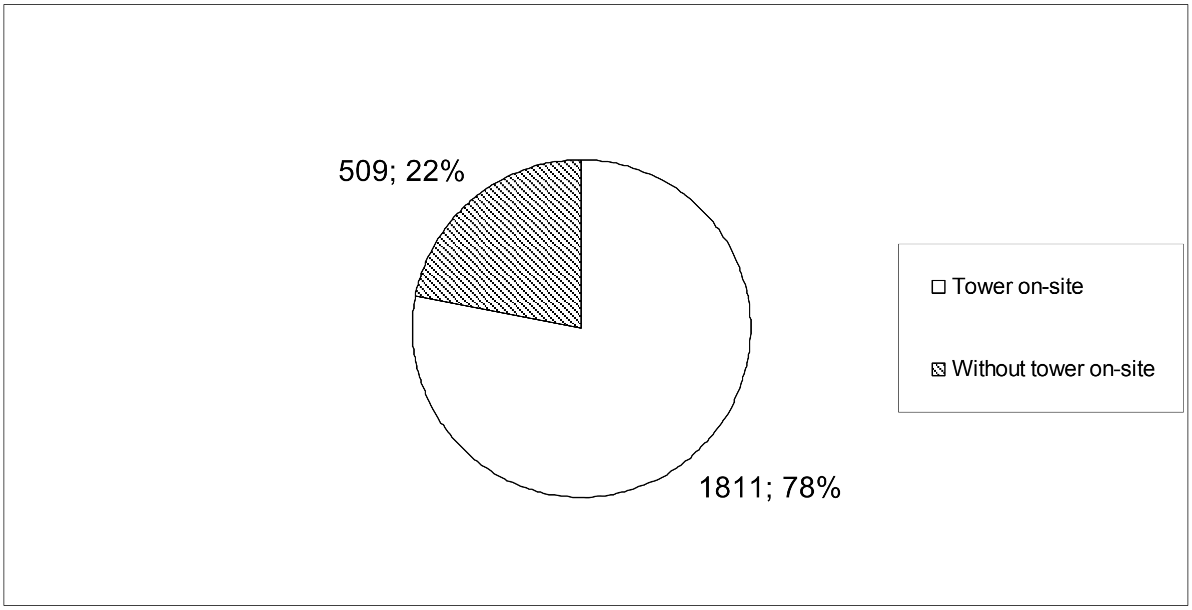

Figure 17: Airport Composition by Availability of Air Traffic Control Tower.

Section 4.2 Operations and Aircraft Mix at the Target Airport Set

As mentioned in Section 3.5, the study uses the FAA capacity data for 99 major airports. Therefore, the operation and aircraft mix estimations discussed here do not include those airports. The remaining number of airports to be referred to in this section is 2,221.

For commercial operations, the 2004 T-100 data indicates that a total of 219 aircraft types were used for operations at 531 airports. The study matches the specification data for these aircraft with the FAA weight class criteria and estimates a weight class table for these aircraft. This table is used to sort the T-100 operation data into appropriate aircraft classes, and to calculate aircraft mix at each airport. Please refer to Appendix G for a complete list of aircraft type codes used in the analysis.

In the year 2004, the T-100 data the study contains 53,094 commercial departure operation records at 531 airports. The same dataset contains 53,845 commercial arrival flight records at 531 airports. The numbers of commercial flights operated by each weight class at each airport are calculated. For non-commercial operations, the number of flights is estimated using the number of aircraft based at each airport. The combinations of the commercial and non-commercial flights are thus available for 2,221 airports.

Section 4.3 Airport Capacity

The capacity analysis and its output are presented as a set of standard pareto frontiers (arrival-departure diagrams) for each airport. To illustrate the analysis we use Chadron Municipal Airport (Airport ID: CDR) at Chadron, Nebraska, as an example. An aerial map picture of CDR is shown in Figure 18.

Figure 18: Aerial Photo of Chadron Municipal Airport (Photo Courtesy of Microsoft).

Figure 18 shows CDR with two intersecting runways: runway 2-20 and runway 11-29. The runway classification Matlab program examines the runway layout and finds that the configuration at CDR belongs to “Two Intersecting Runways”, which is model 6 in the ACM. For this model, the ACM has two preset operational strategies:

Strategy 2: One runway used for arrivals only, the other one used for departures only.

Strategy 3: One runway used for both arrivals and departures, the other one used for departures only.

The operational data for CDR indicates that the aircraft mix is 100% small aircraft. Therefore, 1:0:0:0 will be the aircraft mix parameter for the ACM. Combining the aircraft mix with other input data, the ACM model calculates runway capacities for both operational strategies at CDR. The results are shown in Tables 12 and 13.

Table 12: Runway Capacity Results for CDR under VMC Condition.

Strategy 2	Arrival Capacity (ops/hr)	Departure Capacity (ops/hr)	Total Capacity (ops/hr)
Departure Priority	0.0	66.7	66.7
10% Arrival	6.5	58.1	64.6
20% Arrival	12.5	50.1	62.6
30% Arrival	18.3	42.6	60.9
40% Arrival	23.6	35.5	59.1
50% Arrival	28.7	28.7	57.4
60% Arrival	33.6	22.4	56.0
70% Arrival	33.8	14.5	48.3
80% Arrival	33.8	8.5	42.3
90% Arrival	33.8	3.8	37.6
Arrival Priority	33.8	22.0	55.8

Strategy 3	Arrival Capacity (ops/hr)	Departure Capacity (ops/hr)	Total Capacity (ops/hr)
Departure Priority	0.0	133.3	133.3
10% Arrival	11.3	101.5	112.8
20% Arrival	19.6	78.2	97.8
30% Arrival	25.9	60.4	86.3
40% Arrival	30.9	46.3	77.2
50% Arrival	34.9	34.9	69.8
60% Arrival	35.3	23.5	58.8
70% Arrival	35.3	15.1	50.4
80% Arrival	35.3	8.8	44.1
90% Arrival	35.3	3.9	39.2
Arrival Priority	35.3	33.8	69.1

Table 13: Runway Capacity Results for CDR under MMC Condition.

Strategy 2	Arrival Capacity (ops/hr)	Departure Capacity (ops/hr)	Total Capacity (ops/hr)
Departure Priority	0.0	66.7	66.7
10% Arrival	6.5	58.1	64.6
20% Arrival	12.5	50.1	62.6
30% Arrival	18.3	42.6	60.9
40% Arrival	23.6	35.5	59.1
50% Arrival	28.7	28.7	57.4
60% Arrival	33.6	22.4	56.0
70% Arrival	33.8	14.5	48.3
80% Arrival	33.8	8.5	42.3
90% Arrival	33.8	3.8	37.6
Arrival Priority	33.8	22.0	55.8

Strategy 3	Arrival Capacity (ops/hr)	Departure Capacity (ops/hr)	Total Capacity (ops/hr)
Departure Priority	0.0	133.3	133.3
10% Arrival	11.3	101.5	112.8
20% Arrival	19.6	78.2	97.8
30% Arrival	25.9	60.4	86.3
40% Arrival	30.9	46.3	77.2
50% Arrival	34.9	34.9	69.8
60% Arrival	35.3	23.5	58.8
70% Arrival	35.3	15.1	50.4
80% Arrival	35.3	8.8	44.1
90% Arrival	35.3	3.9	39.2
Arrival Priority	35.3	33.8	69.1

Table 14: Runway Capacity Results for CDR under IMC Condition.

Strategy 2	Arrival Capacity (ops/hr)	Departure Capacity (ops/hr)	Total Capacity (ops/hr)
Departure Priority	0.0	14.5	14.5
10% Arrival	1.6	14.5	16.1
20% Arrival	3.6	14.5	18.1
30% Arrival	6.2	14.5	20.7
40% Arrival	7.5	11.3	18.8
50% Arrival	7.5	7.5	15.0
60% Arrival	7.5	5.0	12.5
70% Arrival	7.5	3.2	10.7
80% Arrival	7.5	1.9	9.4
90% Arrival	7.5	0.8	8.3
Arrival Priority	7.5	14.5	22.0

Strategy 3	Arrival Capacity (ops/hr)	Departure Capacity (ops/hr)	Total Capacity (ops/hr)
Departure Priority	0.0	14.5	14.5
10% Arrival	1.6	14.5	16.1
20% Arrival	3.6	14.5	18.1
30% Arrival	6.2	14.5	20.7
40% Arrival	7.5	11.3	18.8
50% Arrival	7.5	7.5	15.0
60% Arrival	7.5	5.0	12.5
70% Arrival	7.5	3.2	10.7
80% Arrival	7.5	1.9	9.4
90% Arrival	7.5	0.8	8.3
Arrival Priority	7.5	14.5	22.0

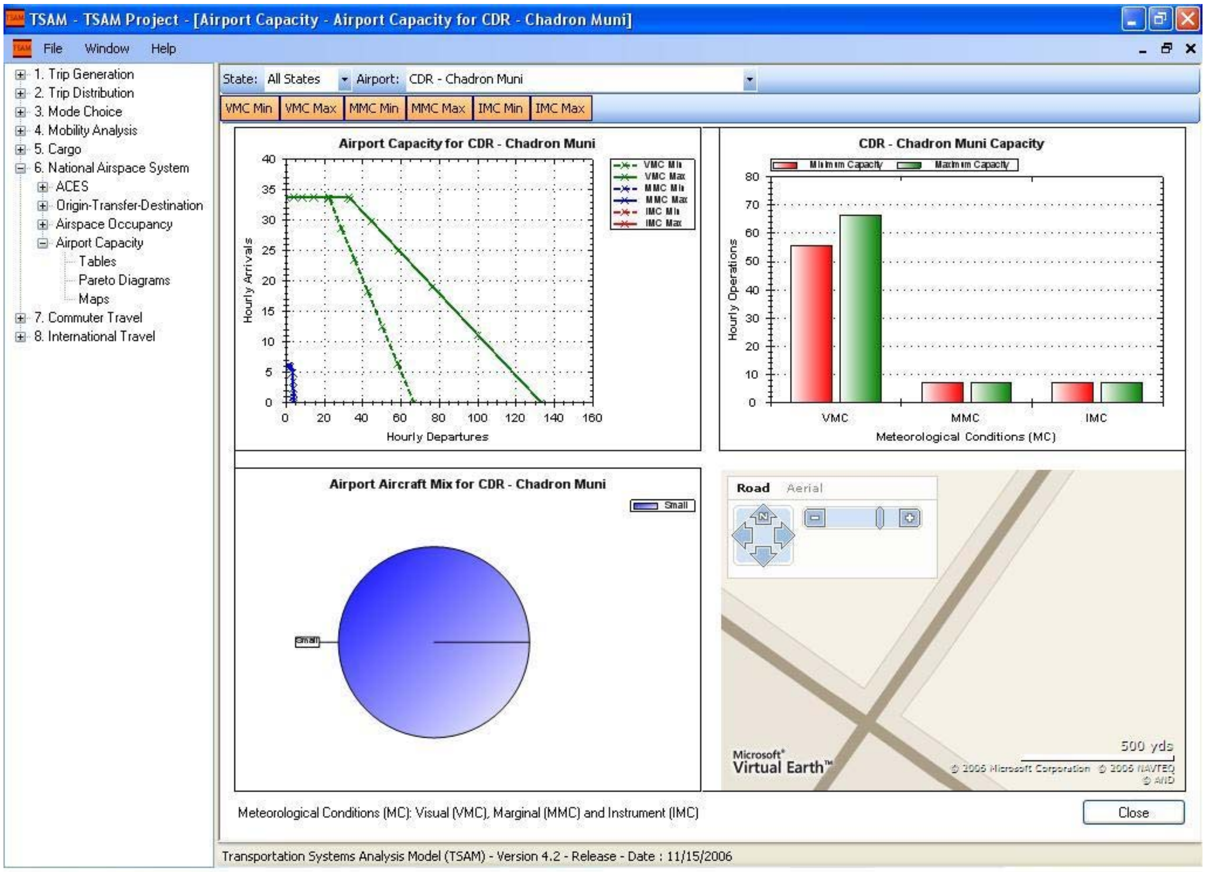

Figure 19: Screen Capture of TSAM Runway Capacity Display Information (CDR airport).

To present the runway capacity assessment in a systematic way, we integrate the capacity results into the TSAM model. Figure 19 illustrates the TSAM interface. The user can access any airport using the drop-down menu on the first row of the interface. The graph on the upper-left side reflects the departure-arrival capacity diagram (pareto frontier) for different weather scenarios. The user can select the desired weather scenarios from six preset scenarios: VMC, MMC, or IMC (lower and upper boundaries). The bars in the upper-right side viewpoint show a comparison of "arrival-priority capacity" between the six weather scenarios. The pie chart on the lower-left side illustrates the percentage aircraft mix assigned to each of four aircraft weight classes. The lower-right side viewpoint shows the map/aerial picture of the airfield using Microsoft virtual earth. With these four graphs, a user can easily browse the runway capacity information for any airport.

The process described above applies to all 2320 airports. Figures 20 and 21 present graphically the capacity for all 2,320 airports in the NAS for two weather conditions and for upper and lower-boundary capacities, respectively.

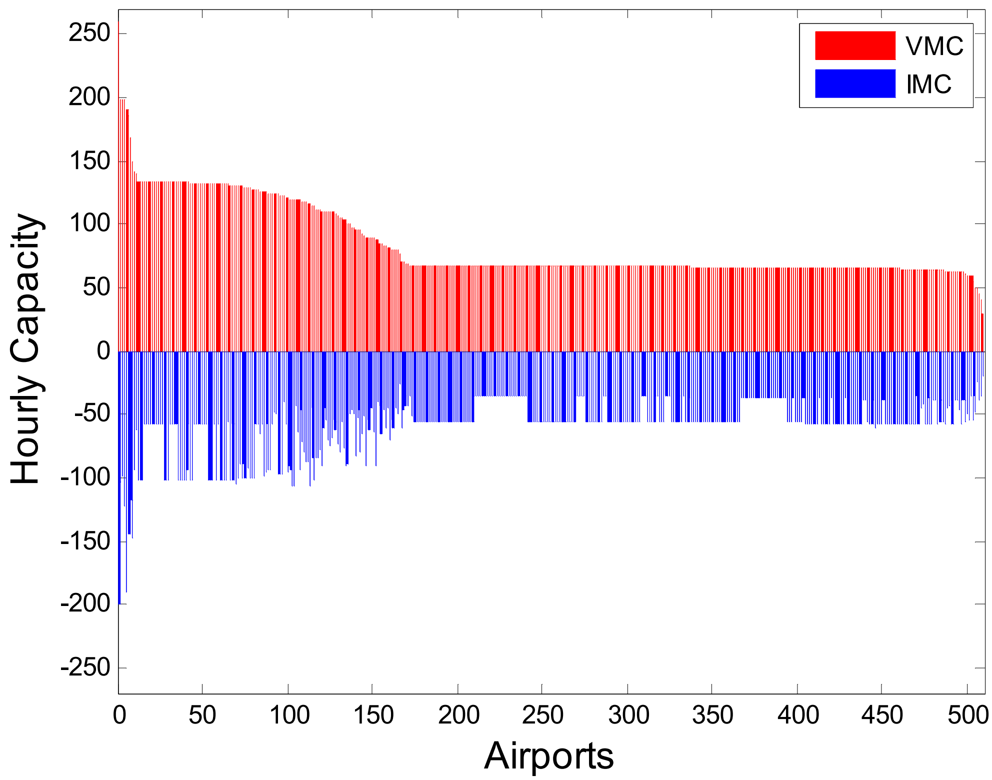

Figure 20: Maximum Runway Capacity under Upper-Boundary Operational Strategy at Towered Airports.

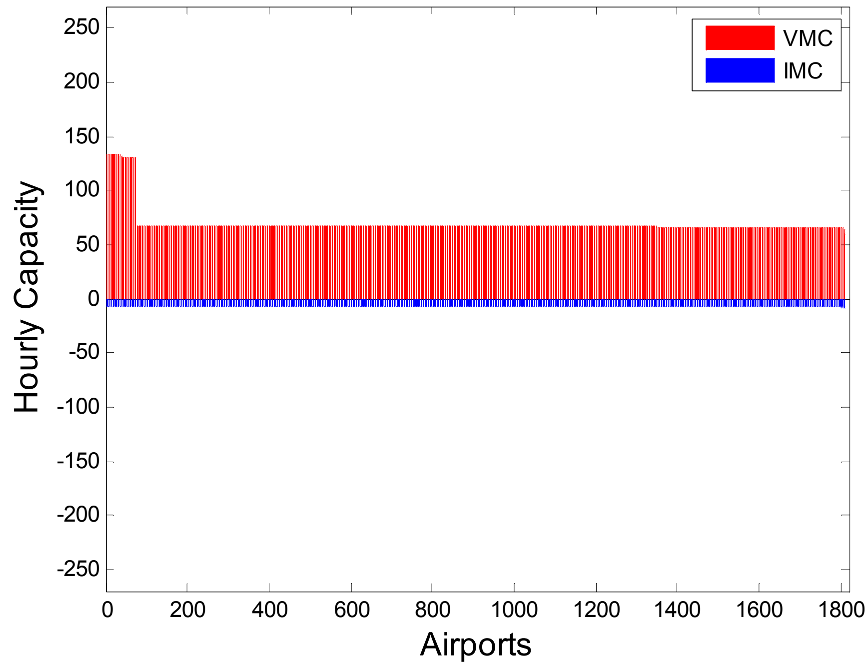

Figure 21: Maximum Runway Capacity under Upper-Boundary Operational Strategy at Non-Towered Airports.

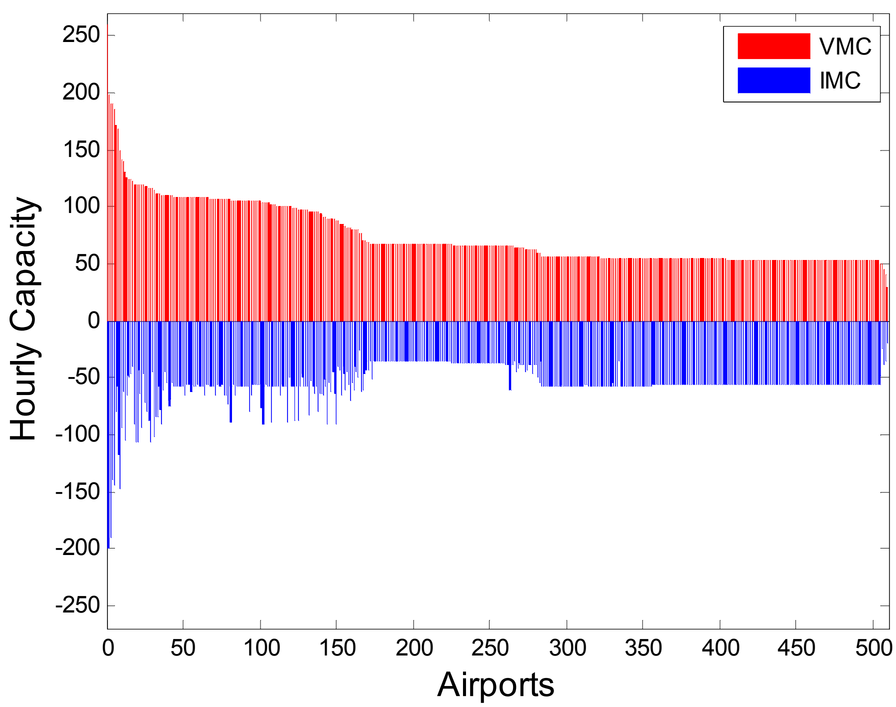

Figure 22: Maximum Runway Capacity under Lower-Boundary Operational Strategy at Towered Airports.

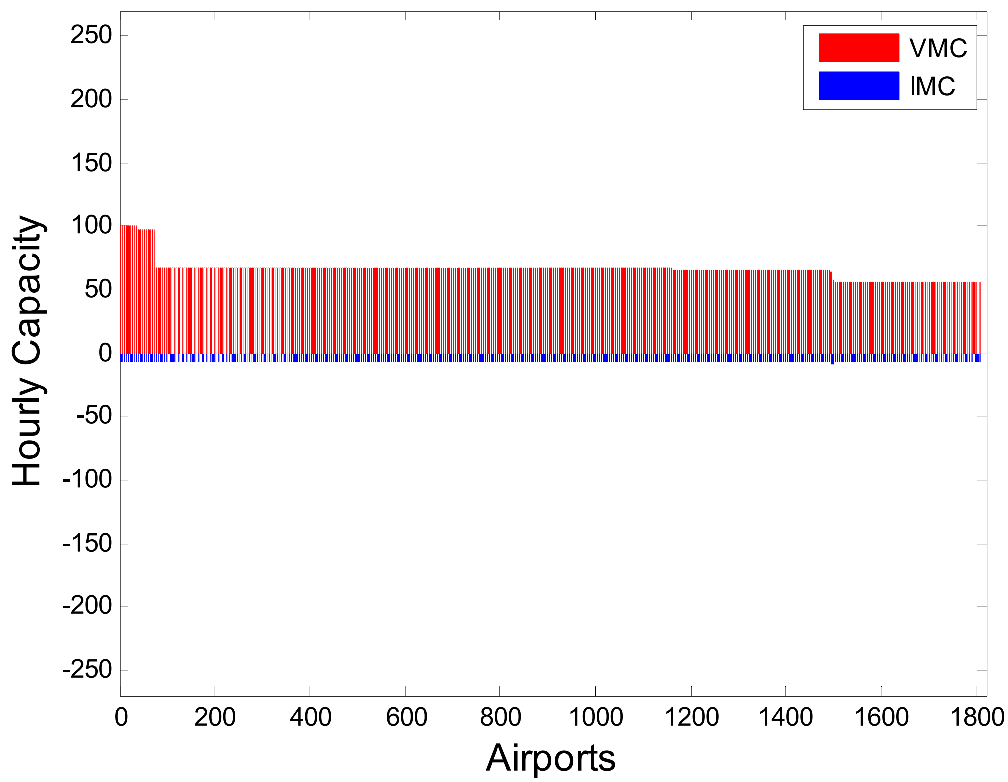

Figure 23: Maximum Runway Capacity under Lower-Boundary Operational Strategy at Non-Towered Airports.

Figures 20 through 23 show the capacity differential between VMC and IMC conditions. For airports with air traffic control tower on site, the capacity difference between these two weather scenarios comes from different separation requirements. The level of the loss of runway capacity is affected by the airport configuration and applicable operational strategies. For non-towered airports, the IMC runway capacity is constrained by the one-in one-out operation requirement. Only one flight can be performed in controlled airspace of the airport. That is the reason that all of these non-towered airports have the same runway capacity number under IMC condition. Hence, it is obvious that bad weather causes a serious degradation for system-wide airport runway capacity, especially for non-towered airports.

Figures 24 through 29 show the upper and lower-boundary airport capacities for three weather conditions: VMC, MMC, and IMC. From a system-wide point of view, it is clear that runway operational strategies do not contribute to the capacity of the system as much as weather conditions.

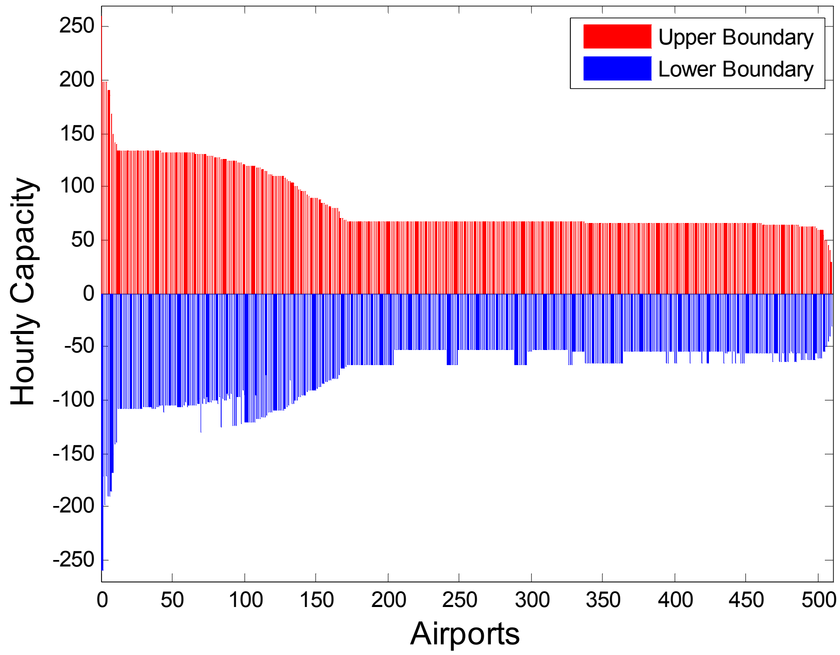

Figure 24: Maximum Runway Capacity Comparison between Upper and Lower Boundary Operational Strategies for VMC Condition at Towered Airports.

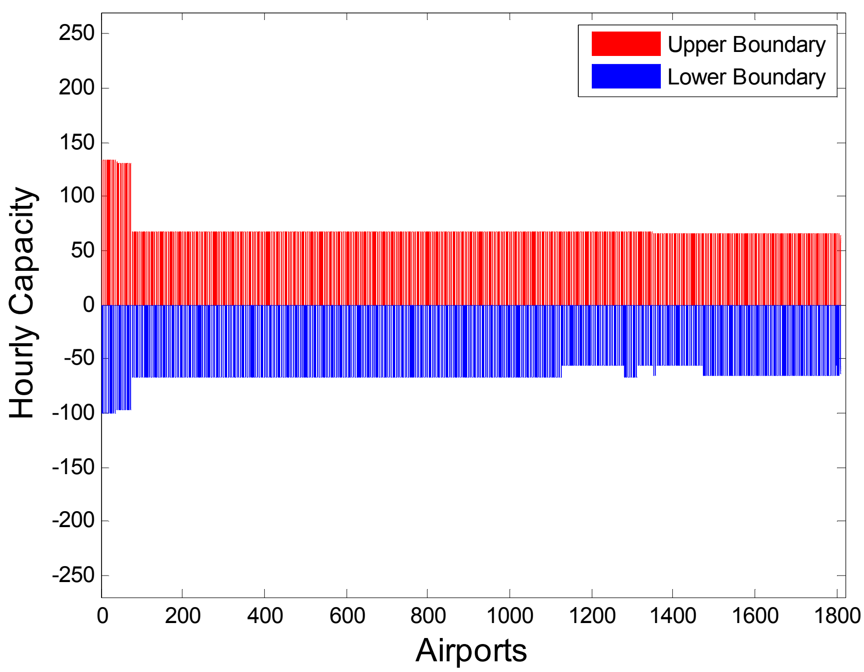

Figure 25: Maximum Runway Capacity Comparison between Upper and Lower Boundary Operational Strategies for VMC Condition at Non-Towered Airports.

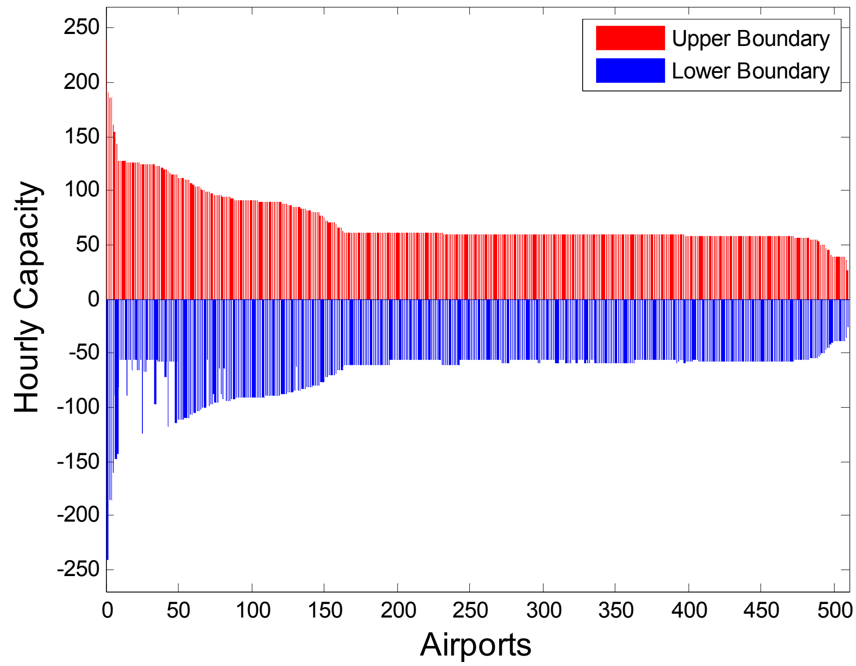

Figure 26: Maximum Runway Capacity Comparison between Upper and Lower Boundary Operational Strategies for MMC Condition at Towered Airports.

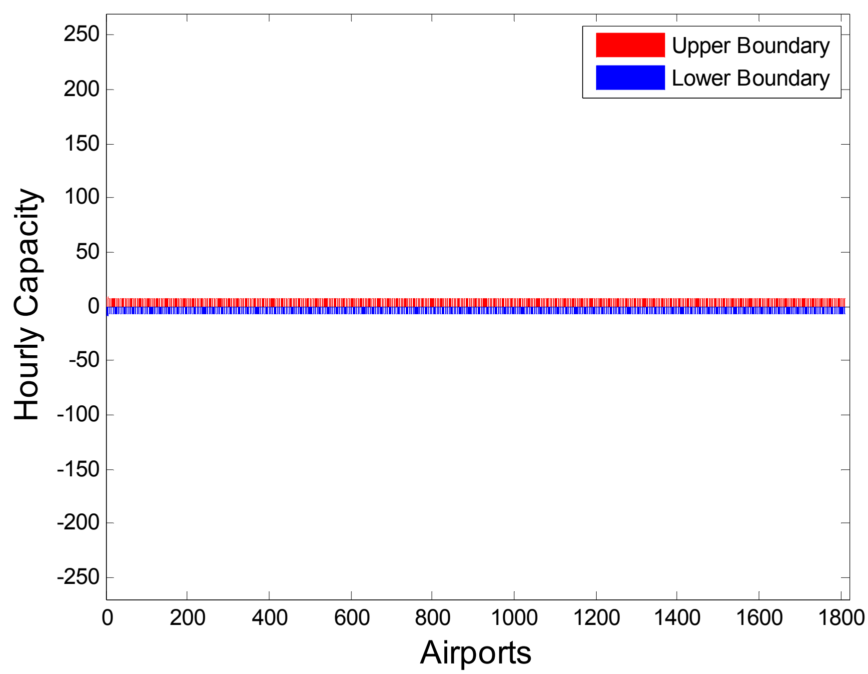

Figure 27: Maximum Runway Capacity Comparison between Upper and Lower Boundary Operational Strategies for MMC Condition at Non-Towered Airports.

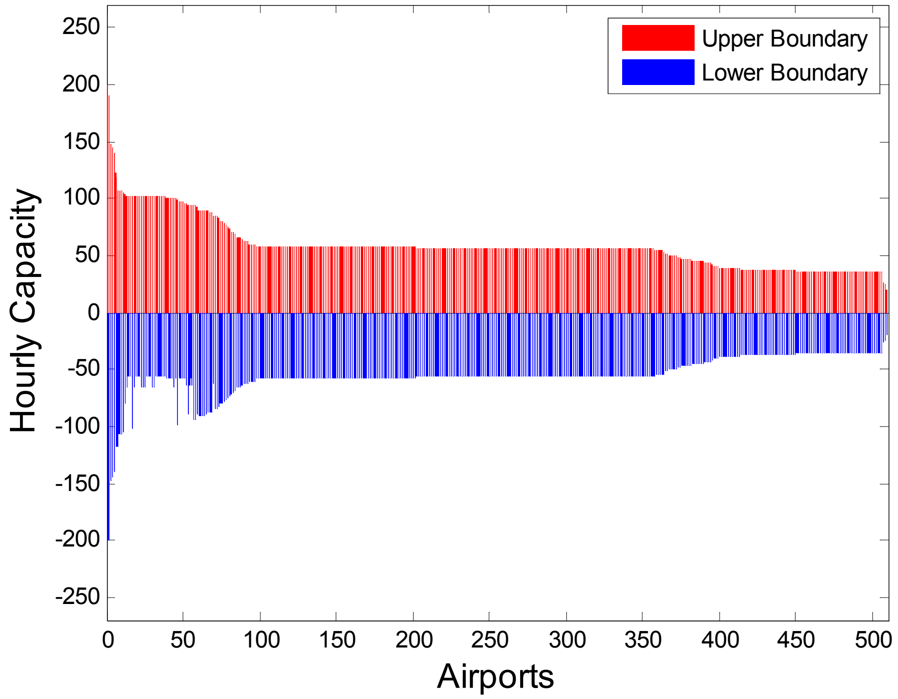

Figure 28: Maximum Runway Capacity Comparison between Upper and Lower Boundary Operational Strategies for IMC Condition at Towered Airports.

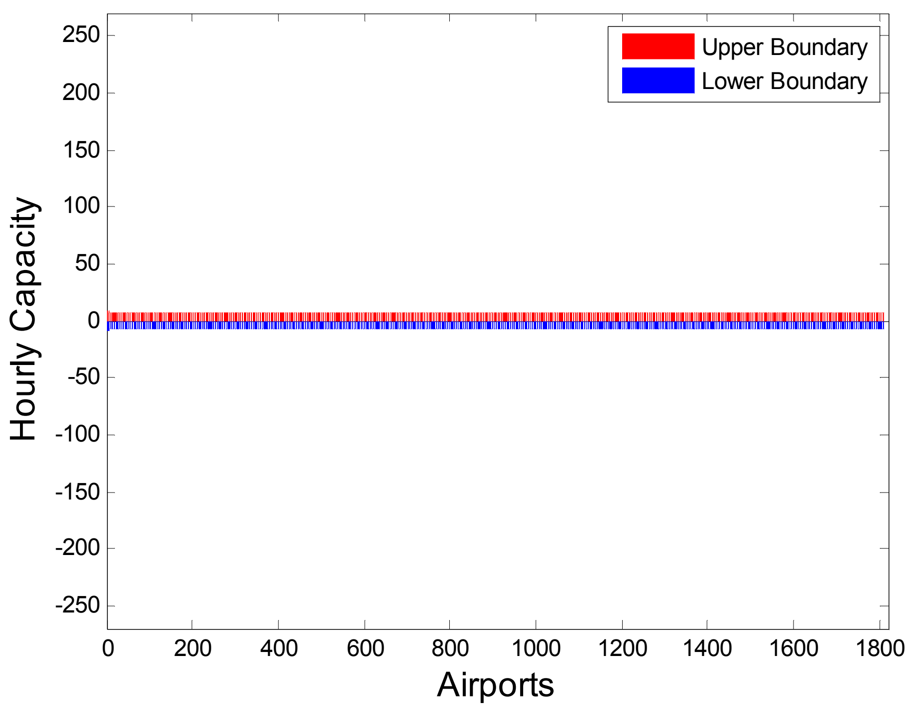

Figure 29: Maximum Runway Capacity Comparison between Upper and Lower Boundary Operational Strategies for IMC Condition at Non-Towered Airports.

Figures 30 through 35 show the hourly airport runway capacity maps for 2,320 continental U.S. airports. These maps provide a visual reference for the distribution and magnitude of airport capacity in the contiguous U.S. States. Inspection of these maps shows a significance difference in the capacity between VMC and MMC/IMC weather conditions. Different operational strategies also cause differences in runway capacities. However, the influence of operational strategies to runway capacity is not as influential as that of the weather.

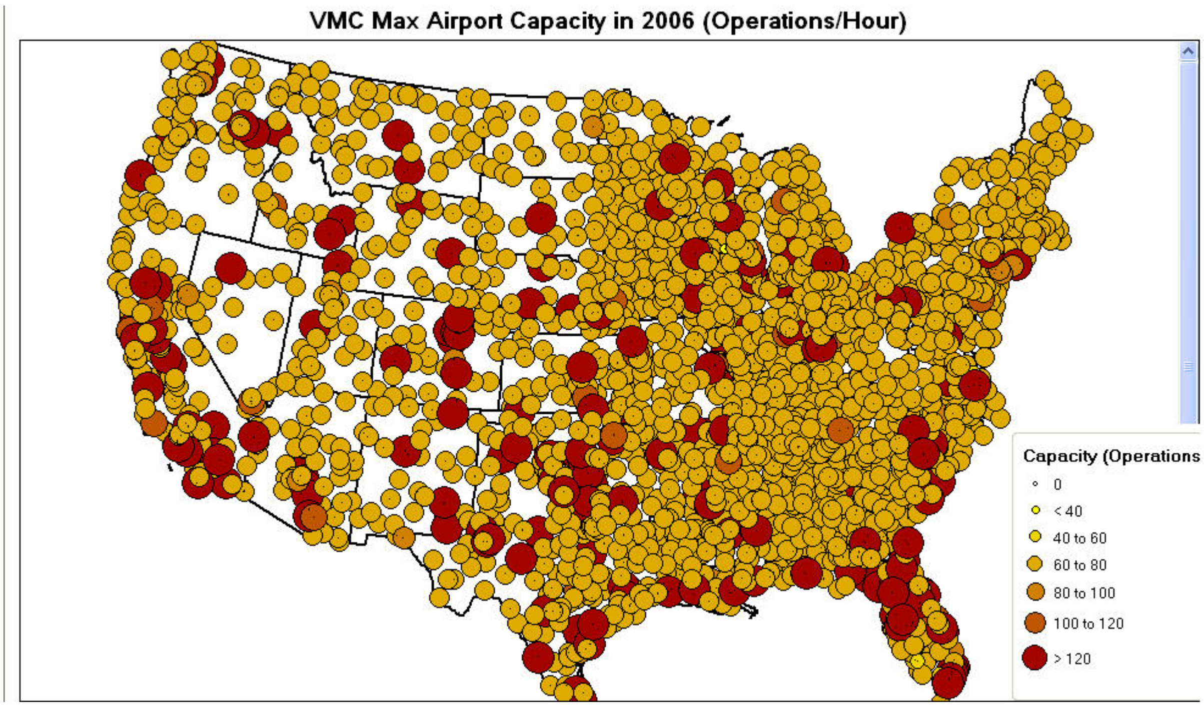

Figure 30: VMC Airport Capacity under Upper-boundary Strategy.

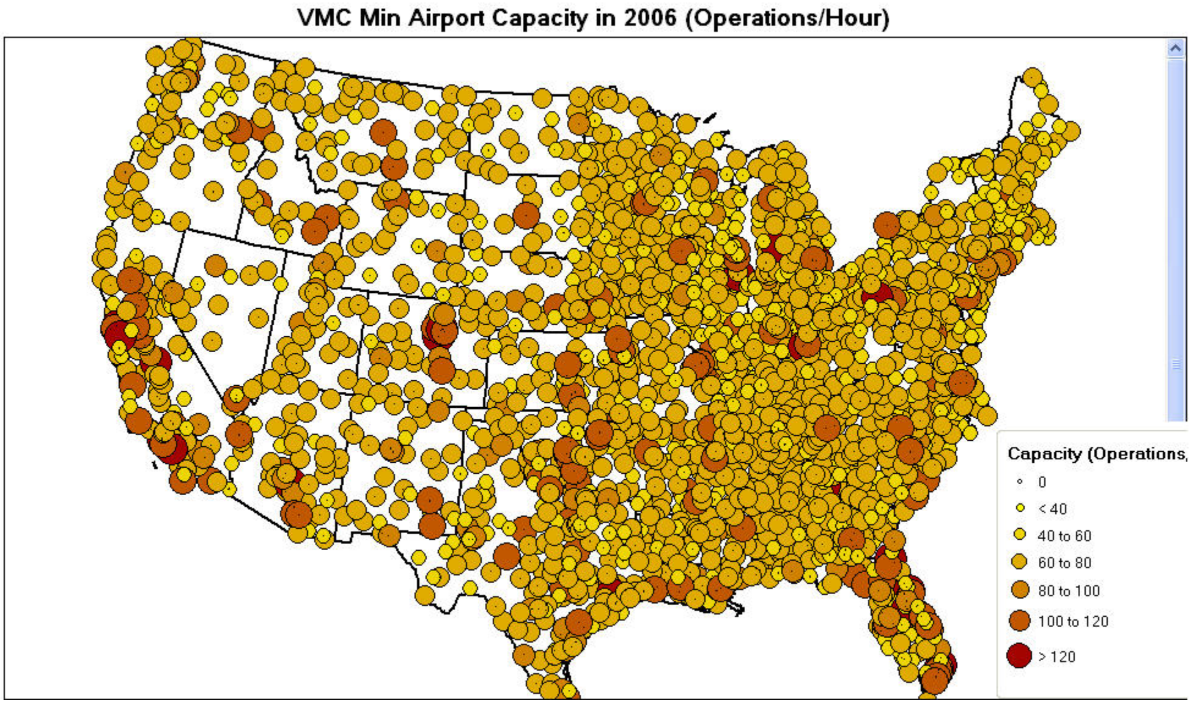

Figure 31: VMC Airport Capacity under Lower-boundary Strategy.

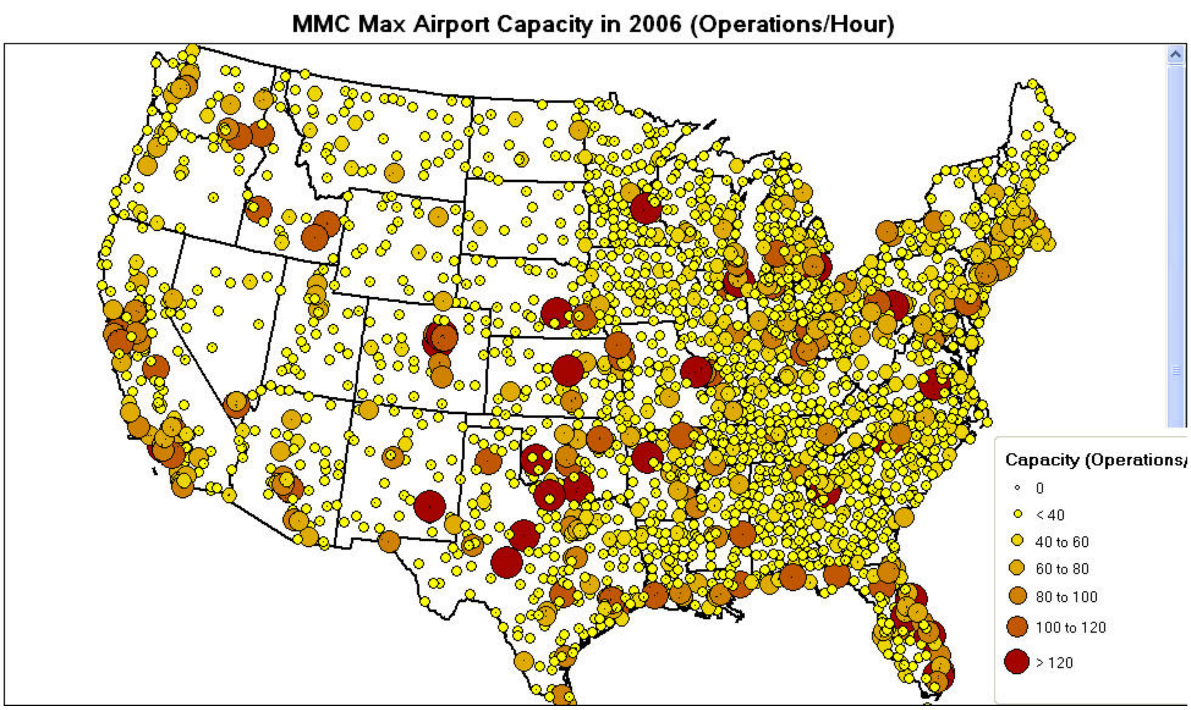

Figure 32: MMC Airport Capacity under Upper-boundary Strategy.

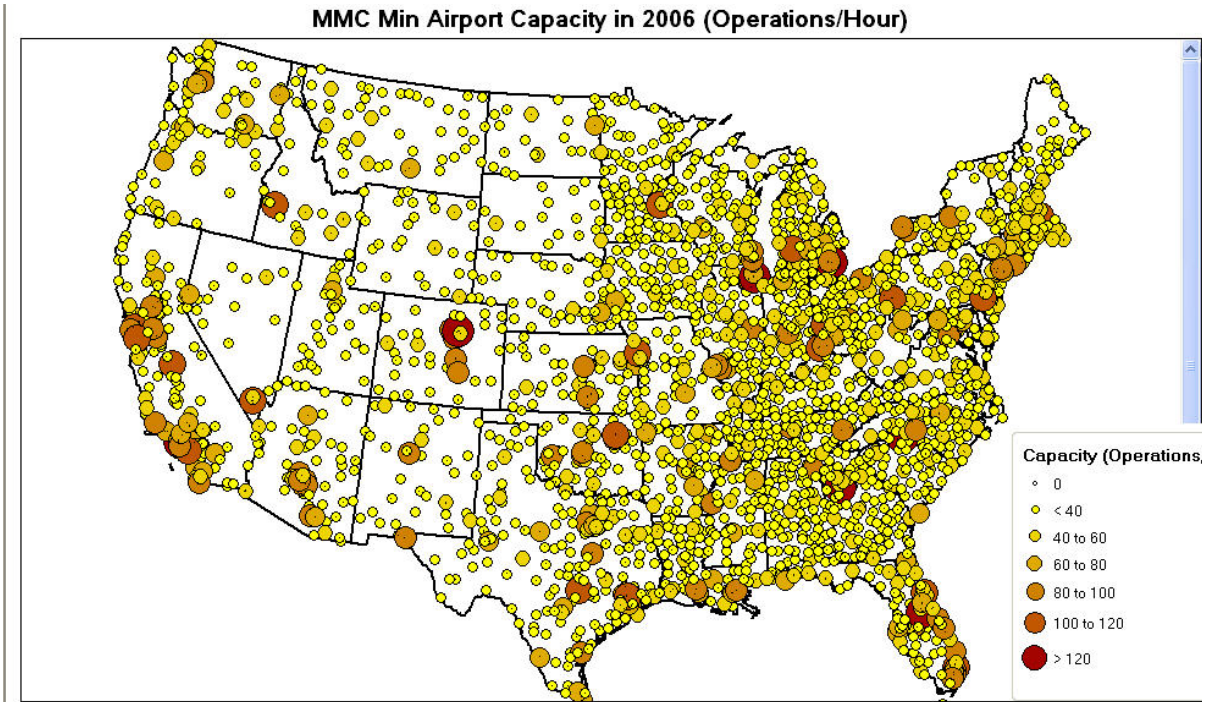

Figure 33: MMC Airport Capacity under Lower-boundary Strategy.

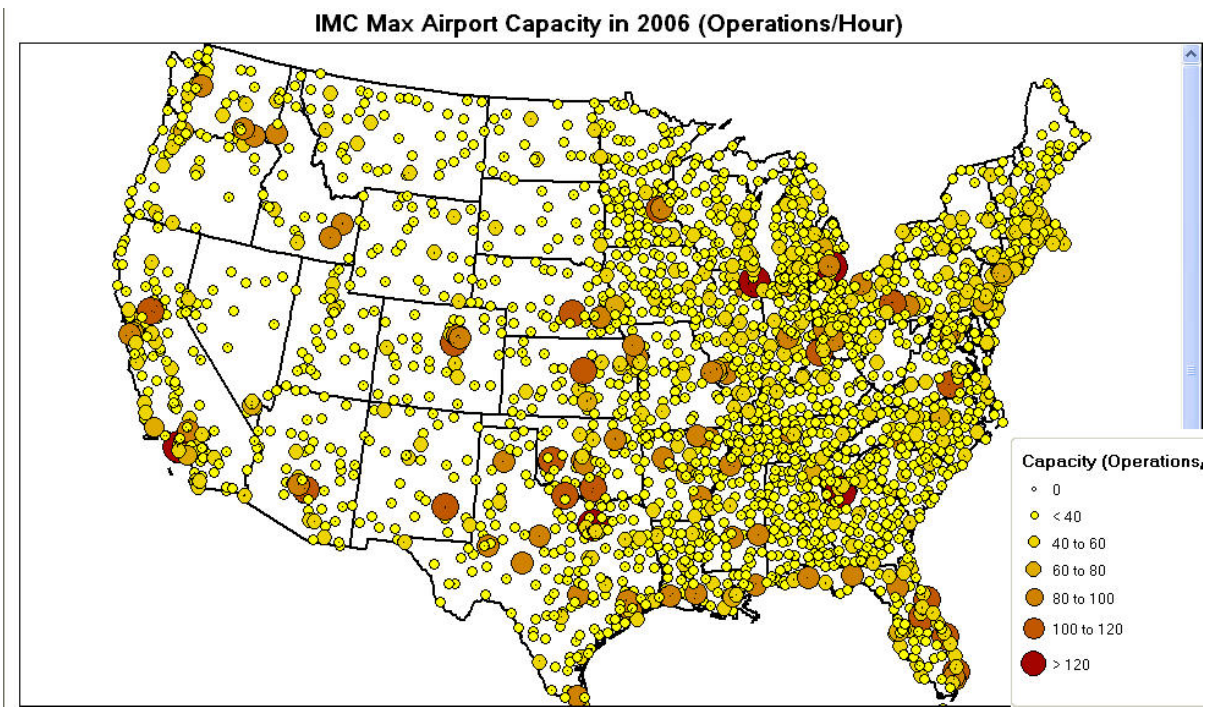

Figure 34: IMC Airport Capacity under Upper-boundary Strategy.

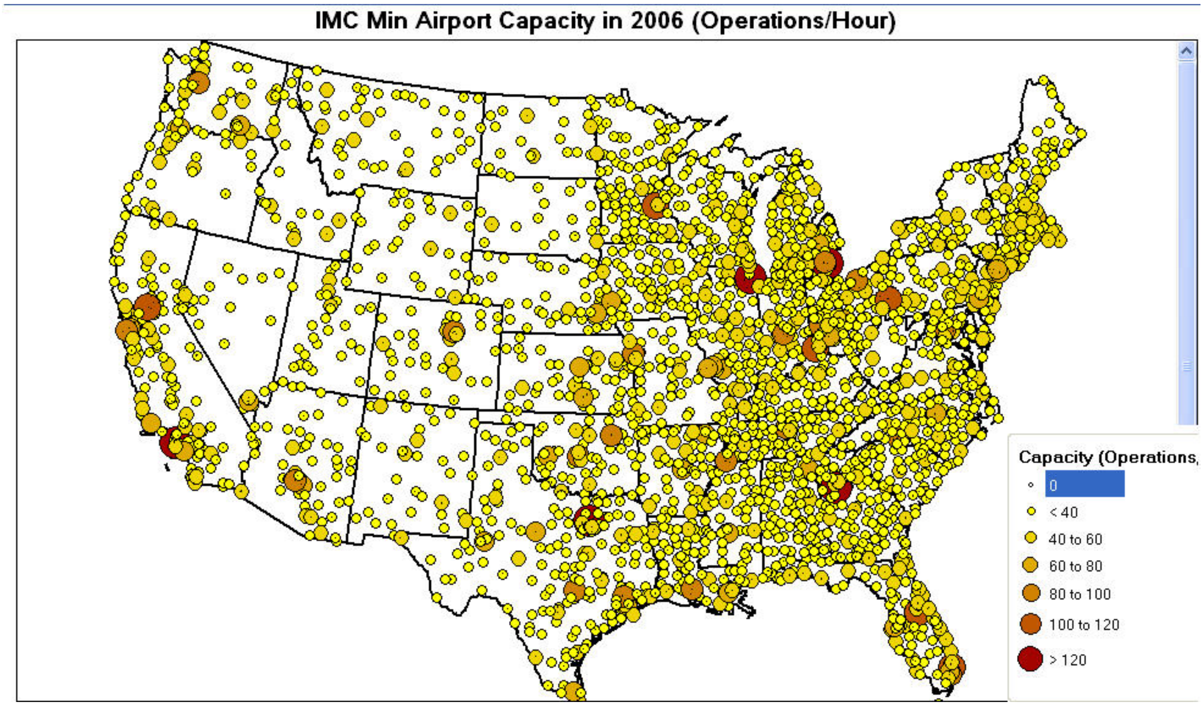

Figure 35: IMC Airport Capacity under Lower-boundary Strategy.

Section 4.4 Application of Future Scenarios

New technology might be able to mitigate some of the shortcomings in future airport capacity. For example, modern navigation aids can reduce the uncertainty of the location of aircraft; revised procedures and future vehicles might also be able to operate with reduced runway separation requirements. All these changes could create a totally different airport operating environment and bring substantial gains to airport runway capacity. To emulate possible scenarios in the future, users will be able to change several operational parameters through the user interface in TSAM. This will permit the assessment and comparison of improvements introduced by new technologies and regulations. Consider a non-towered airport as an example. Under current regulations allowing only one flight, a departure or an arrival, at a time in IFR conditions, yields very low runway capacities (e.g., 3-5 operations per hour). In IFR conditions, airports with more than one runway (say two closely spaced parallel runways) perform as single runway airports. This restriction limits the usage of infrastructure resources. In the future the use of new procedures and technology could allow such restrictions to be relaxed. The following case study will demonstrate the significance of technology and new procedures to runway capacity.

Edward F Knapp State Airport (MPV) at Barre/Montpelier, Vermont, can only operate as a single runway airport in IMC weather conditions because it lacks a control tower at the site. The airport has two runways. Under current operation procedures, MPV has an hourly capacity of 7.2 flights/hr. If a new technology enables the using of its both runways at the same time in IMC condition, the capacity could be boosted to 22 flights an hour. This demonstrates the capability of the model developed in this study to study technology-capacity tradeoffs.

Table 15: Comparison of Runway Capacity of Edward F Knapp State Airport under Current Operation Rules and a Future Scenario.

Future Scenario	Arrival Capacity (ops/hr)	Departure Capacity (ops/hr)	Total Capacity (ops/hr)
Departure Priority	0.0	14.5	14.5
10% Arrival	1.6	14.5	16.1
20% Arrival	3.6	14.5	18.1
30% Arrival	6.2	14.5	20.7
40% Arrival	7.5	11.3	18.8
50% Arrival	7.5	7.5	15
60% Arrival	7.5	5	12.5
70% Arrival	7.5	3.2	10.7
80% Arrival	7.5	1.9	9.4
90% Arrival	7.5	0.8	8.3
Arrival Priority	7.5	14.5	22

Current Rules	Arrival Capacity (ops/hr)	Departure Capacity (ops/hr)	Total Capacity (ops/hr)
Departure Priority	0.0	4.0	4.0
10% Arrival	0.4	3.9	4.3
20% Arrival	1.0	3.8	4.8
30% Arrival	1.6	3.7	5.4
40% Arrival	2.4	3.6	6.1
50% Arrival	3.5	3.5	7.0
60% Arrival	5.0	3.3	8.3
70% Arrival	6.0	2.6	8.5
80% Arrival	6.1	1.5	7.6
90% Arrival	6.1	0.7	6.8
Arrival Priority	6.1	1.1	7.2

Section 4.5 Verification

To verify the capacity estimates in this study, we select three airports to compare the airport capacity estimates with published values in the open literature. Although the general aviation airports comprise the majority of the targets of this study, the lack of capacity data for such airports makes it difficult to perform meaningful comparisons. Therefore, the validation comparisons will be made for commercial airports only.

The first airport chosen to verify capacity estimates is San Diego International-Lindbergh Field Airport (SAN). SAN has one runway, which fits the model 1 in ACM. The aircraft mix at SAN is 15:74:8:3 (Small: Large: B757: Heavy) [T-100 Carrier Data, 2004].

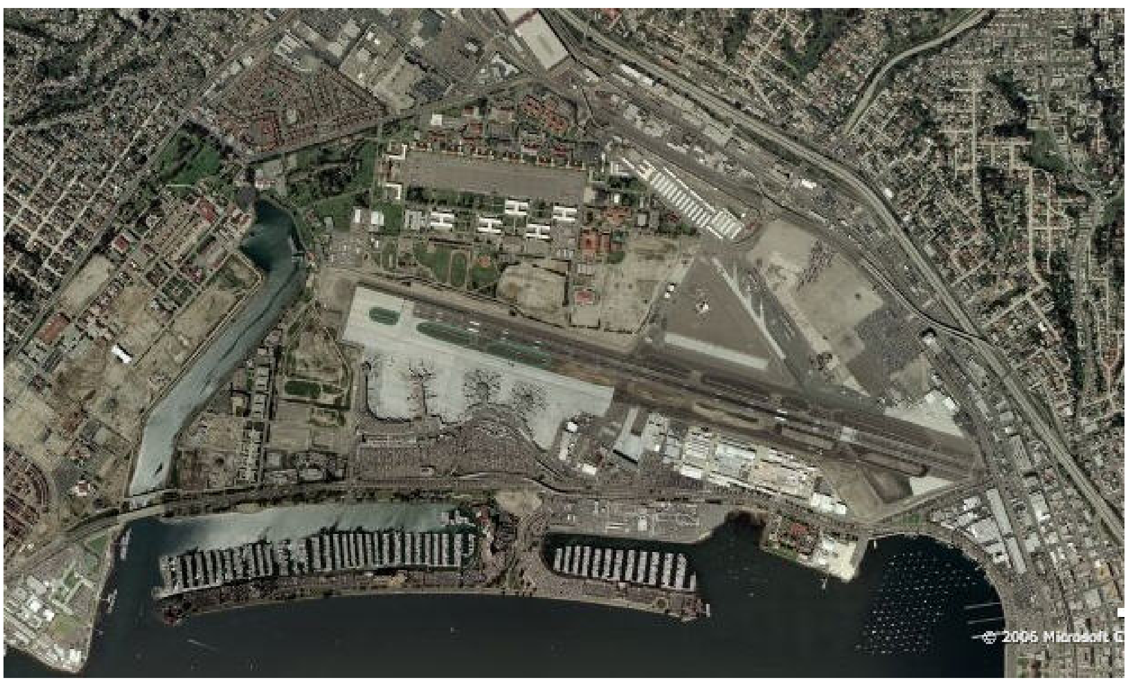

Figure 36: Aerial Photo of San Diego International-Lindbergh Field Airport (Photo Courtesy of Microsoft)

Table 16 compares the capacity obtained by the ACM program used in this study and the capacity analysis documented in the FAA Benchmark Report.

Table 16: Comparison of Runway Capacities at SAN between Different Data Sources.

	Capacity Number (flights/hour)		Percentage Difference (%)
Weather Condition	ACM Used in The Study	FAA Benchmark Report	
VMC	61	56-58	7.02
MMC	60.5	56-58	6.14
IMC	51.8	48-50	5.71

In the second example we estimate the capacity of New York La Guardia Airport (LGA). LGA has two runways and the aircraft mix is 7:83:9:1.

Figure 37: Aerial Photo of New York La Guardia Airport (Photo Courtesy of Microsoft).

Table 17: Comparison of Runway Capacities at LGA between Different Data Sources.

	Capacity Number (flights/hour)		Percentage Difference (%)
Weather Condition	ACM Used in The Study	FAA Benchmark Report	
VMC	86.9	78-85	6.63
MMC	66.2	74-85	16.73
IMC	66.2	69-74	7.41

We also examined Los Angeles International Airport (LAX) with the aircraft mix 16:54:11:19.

Figure 38: Aerial Photo of Los Angeles International Airport (Photo Courtesy of Microsoft).

Table 18: Comparison of Runway Capacities at LAX between Different Data Sources.

Weather Condition	Capacity Number (flights/hour)		Percentage Difference (%)
	ACM Used in The Study	FAA Benchmark Report	
VMC	143	137-148	0.35
MMC	131	126-132	1.55
IMC	131	117-124	8.71

The next example is Phoenix Sky Harbor International Airport (PHX) with aircraft mix of 3:89:6:2.

Figure 39: Aerial Photo of Phoenix Sky Harbor International Airport (Photo Courtesy of Microsoft).

Table 19: Comparison of Runway Capacities at PHX between Different Data Sources.

	Capacity Number (flights/hour)		Percentage of Differences (%)
Weather Condition	ACM Used in The Study	FAA Benchmark Report	
VMC	118	128-150	15.11
MMC	101	108-118	10.62
IMC	101	108-118	10.62

The last example considered is Atlanta Hartsfield-Jackson International Airport (ATL). The aircraft mix at ATL is 0:80:8:12. Although ATL opened their 5^{th} runway this May, the data in the Benchmark Report and aircraft mix information are all based on a four-runway scenario. This is the scenario used in this study.

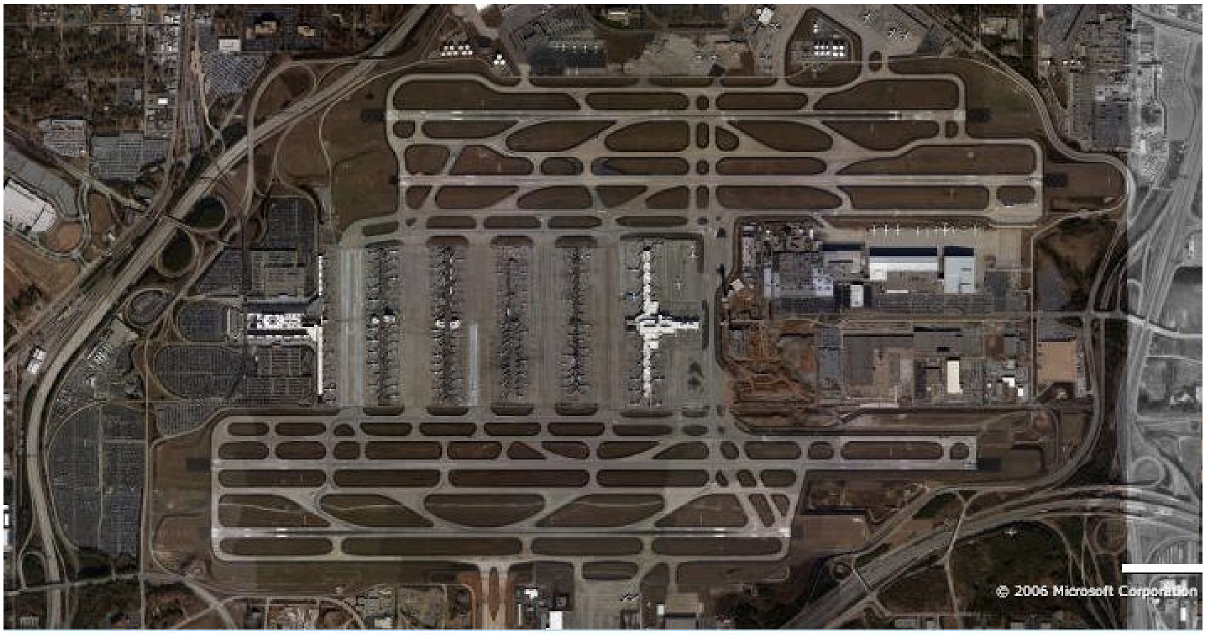

Figure 40: Aerial Photo of Atlanta Hartsfield-Jackson International Airport.
(Photo Courtesy of Microsoft)

Table 20: Comparison of Runway Capacities at ATL between Different Data Sources.

Weather Condition	Capacity Number (flights/hour) ACM Used in The Study	Capacity Number (flights/hour) FAA Benchmark Report	Percentage Difference (%)
VMC	171.1	180-188	7.01
MMC	156.7	172-174	9.42
IMC	137.2	158-162	14.25

All the five cases, with quite different runway configurations, demonstrate the reasonableness of the capacity estimates using ACM. As more airports experience recurring congestion, like Atlanta and New York La Guardia Airports, air traffic control personnel reduce the buffer of separation for handling more traffic and thus reducing delays at the airport. Therefore, it is possible to reason that observed discrepancies in capacity could well be due to variations in the local operational procedures at each airport. The verification shows that the program and process used in this study is able to deliver reliable airport runway capacity estimates. Nevertheless, there are many airports in our study that do not have reference capacity data to compare with. The hope is that this research can provide and serve as guidance for future studies.

CHAPTER 5 CONCLUSION

Section 5.1 Summary of Objectives

This study developed a nation-wide airport runway capacity model framework using the FAA ACM model. The study has been carried out to investigate potential capacity shortfalls to handle upcoming Very Light Jet (VLJ) services for qualified airports in the National Airspace System (NAS). The result of this research will help understand the current status of airport capacity in the NAS and attempt to find potential congestion in the future airport network. The major benefit of the study is to help FAA, NASA and the Joint Planning Development Office (JPDO) plan the airport infrastructure to handle future demands.

To achieve the goals, the study uses several aviation databases to quantify the aircraft mix, runway infrastructure and operations of 2320 airports. With these parameters, the FAA Airfield Capacity Model (ACM) can perform runway capacity estimation for different weather scenarios for each airport of interest. The result of our analysis is a capacity envelope graph that indicates operational limits for different aircraft arrival-departure combinations under several weather scenarios.

Section 5.2 Summary of Results

The result of airport runway capacity estimation for 2,320 airports in the NAS has been integrated into the TSAM model. With this model, users can easily assess the change of runway capacity under different arrival-departure composition trends, for different weather scenarios, and aircraft mixes. The model developed also integrates road and aerial maps of the airport.

Comparing the capacity between different weather conditions, the study found that many of the 2,320 airports examined have a VMC capacity at around 67 operations per hour, and at IMC capacity around 23 operations per hour.

The difference between upper-boundary and lower-boundary operational strategies is not that apparent for most airports, and this situation is common for all three weather conditions. However, the deficit of capacity is very obvious when comparing the VMC and IMC conditions. Therefore, although IMC weather conditions only account for 10% of all operations in the NAS, there is a significant payoff at improving the capacity under IMC conditions, with new technologies and procedures. Several studies suggest that up to 70% of the delays in the system occur under IMC conditions.

The verification of the study using data from the FAA Benchmark Report 2004 found that the capacity numbers estimated by the study compare favorably for airports with little congestion like San Diego. For congested airports like Atlanta and New York La Guardia Airport, ATC personnel might compress and reduce the buffer of separation to accommodate more operations for the purpose of reducing delays. This helps explain the difference between the FAA numbers and the capacity results estimated by our study using ACM.

Section 5.3 Recommendations and Future Research

The following recommendations for future research can be made from this study.

During the research process, one of the most time-consuming tasks has been the collection and integration of information from multipe databases. The inconsistency among data from different sources leads to ambiguity. Hence, it would be beneficial to all users if future aviation databases can be integrated or have some universal data identifier.

The runway capacity estimation model used in this study, the FAA Airfield Capacity Model (ACM), should probably be upgraded. ACM was developed more than twenty years ago; therefore, a reprogrammed version with a new user interface would be better to ease computations. The default parameters and algorithms need to be revised to comply with new and future runway standards. It would be useful if the ACM program can leave some user-defined functions for added flexibility. For example, the model does not consider the practical operational situation at non-towered airports. In the IFR non-towered airport scenario, the operations can only be performed in a one-in, one-out mode. We modeled such situation in ACM by adjusting the seperation matrices for IMC non-towered conditions. Automation of such procedures and algorithmic modifications will make the ACM more suitable for new and future studies.

Although the runway is the most important element in many airport systems to handle flights, there are also other factors that affect the ability of airports to accommodate operations. The lack of nation-wide database about airport infrastructure beyond the runway presents a problem for good assessments of total airport capacity. A comprehensive database that includes the number and location of taxiways at each airport, the configuration of apron areas, would provide invaluable information to be used in airport capacity analyses. We would also like to see terrain information of airport and adjacent area to be included, especially for some non-towered airports.

REFERENCES

1. Federal Aviation Administration (2006). "FAA Aerospace Forecasts Fiscal Years 2006-2017". http://www.faa.gov/data_statistics/aviation/aerospace_forecasts/2006-2017/media/FAA%20Aerospace%20Forecast.pdf. Visited October 24, 2006.
2. Air Transport Action Group (2005). "The Economic & Social Benefits of Air Transport". http://www.iata.org/NR/rdonlyres/5C57FE77-67FF-499C-A071-4E5E2216D728/0/ATAG_EconomicSocial_Benefits_Air_Transport.pdf. Visited October 24, 2006.
3. The Subcommittee on Aviation (2001). "S. 633, Hearing on Aviation Delay Prevention Act". The Committee on Commerce, Science, and Transportation, U.S. Senate.
4. Office of the Law Revision Counsel (2006). "United States Code, 49 U.S.C. 47102 (10)". U.S. House of Representatives.
5. National Business Aviation Association (2005). "Training Guidelines for Single Pilot Operations of Very Light Jets and Technically Advanced Aircraft". http://web.nbaa.org/public/ops/safety/vlj/VLJ_Training_Guidelines.pdf. Visited October 24, 2006.
6. Trani, A.A., Baik, H., Swingle, H., Ashiabor, S. (2003). "An Integrated Model to Study the Small Aircraft Transportation System (SATS)". Transportation Research Board 82nd Annual Meeting.
7. Trani, A.A., Baik, H., Hinze, N., Ashiabor, S., Viken, J.K., Dollyhigh, S. (2006). "Nationwide Impacts of Very Light Jet Traffic in the Future Next Generation Air Transportation System (NGATS)". 6th AIAA Aviation Technology, Integration and Operations Conference (ATIO).
8. McCartney, S. (2006). "Air-Taxi Services Threaten to Jam Airports". Wall Street Journal.
9. Wells, A.T. (2000). "Airport Planning & Management". McGraw-Hill.
10. Federal Aviation Administration (2004). "Airport Capacity Benchmark Report 2004". http://www.faa.gov/events/benchmarks/DOWNLOAD/pdf/2004_Benchmark_Report.pdf. Visited November 6, 2006.

11. Blumstein, A. (1959). “The Landing Capacity of a Runway”. Operations Research, 7, p.752-763.
12. Trani, A.A. (2005). Class notes of Analysis of Air Transportation System. http://128.173.204.63/courses/cee5614/cee5614_pub/Airport_capacity_intro.pdf. Virginia Tech.
13. Gilbo, E.P. (1993). “Airport Capacity: Representation, Estimation, Optimization”. IEEE Transactions on Control Systems Technology. Vol. 1, No. 3. P.144-154.
14. Donohue, G.L. (1999). “A Macroscopic Air Transportation Capacity Model: Metrics and Delay Correlation”. New Concepts and Methods in Air Traffic Management. SpringerVerlag.
15. Welch, J.D. and Lloyd, R.T. (2001). “Estimating Airport System Delay Performance”. 4th USA/ Europe Air Traffic Management R&D Seminar.
16. Inniss, T.R. and Ball, M.O. (2002). “Estimating One-Parameter Airport Arrival Capacity Distributions for Air Traffic Flow Management”. http://www.isr.umd.edu/NEXTOR/thesis/Innis-Ball-2002.pdf. Visited November 6, 2006.
17. Hansen, M., Tsao, J., Huang, S.A, and Wei, W. (1998). “Empirical Analysis of Airport Capacity Enhancement Impacts: A Case Study of DFW Airport”. http://www.nextor.org/pubs/RR-98-17.pdf. Visited November 6, 2006.
18. Pitfield, D.E., Jerrard, E.A. (1999). “Monte Carlo comes to Rome: a note on the estimation of unconstrained runway capacity at Rome Fiumucino International Airport”. *Journal of Air Transport Management.* Vol. 5, P.185-192.
19. Bazargan, M., Fleming, K., Subramanian, P. (2002). “A Simulation Study to Investigate Runway Capacity Using TAAM”. Proceedings of the 2002 Winter Simulation Conference. San Diego, CA.
20. Lang, S., Mundra, A.D., Cooper Jr., W.W., Levy, B.S., Lunsford, C.R., Smith III, A.P., Tittsworth, J.A. (2003). “A Phased Approach to - - Increase Airport Capacity through Safe Reduction of Existing Wake Turbulence Constraints”. The MITRE Corporation. http://www.mitrecaasd.org/library/documents/atm2003_paper85.pdf. Visited November 12, 2006.

21. The Subcommittee on Technology, Environment, and Aviation (1995). "Application of FAA Wake Vortex Research to Safety". The Committee on Science, Space, and Technology, U.S. House of Representatives.
22. Rutishauser, D.K., O'Connor, C.J. (2001). "Aircraft Wake Vortex Spacing System Performance Update and Validation Study". National Aeronautics and Space Administration. http://library-dspace.larc.nasa.gov/dspace/jsp/bitstream/2002/10564/1/NASA-2001-tm211240.pdf Visited November 13, 2006.
23. O'Connor, C.J., Rutishauser, D.K. (2001). "Enhanced Airport Capacity through Safe, Dynamic Reductions in Aircraft Separation: NASA's Aircraft Vortex Spacing System (AVOSS). National Aeronautics and Space Administration.
24. Reynolds-Feighan, A.J., Button, K.J. (1999). "An assessment of the capacity and congestion levels at European airports". *Journal of Air Transport Management.* Vol. 5, P.113-134.
25. Hamzawi, S.G. (1992). "Lack of Airport Capacity: Exploration of Alternative Solutions". *Transportation research. Part A: general*. Vol.26A, No.1. P. 47-58
26. Federal Aviation Administration Airport Master Record (5010) http://www.faa.gov/airports%5Fairtraffic/airports/airport%5Fsafety/airportdata%5F5010/. Visited November 13, 2006
27. Federal Aviation Administration (2006). "Very Light Jet (VLJ) Operations and Impact". http://www.fly.faa.gov/tfmlearning/files/Very_Light_Jets.ppt. Visited October 25, 2006.
28. Federal Aviation Administration. "Pilot/Controller Glossary". http://www.faa.gov/ATPUBS/PCG/A.HTM. Visited November 13, 2006.
29. Pfeiderer, E.M. (2005). "Relationship of the Aircraft Mix Index with Performance and Objective Workload Evaluation Research Measure and Controllers' Subjective Complexity Ratings". Federal Aviation Administration. http://www.faa.gov/library/reports/medical/oamtechreports/2000s/media/0516.pdf#search=%22aircraft%20mix%22. Visited November 13, 2006
30. Bureau of Transportation Statistics. "Air Carrier Statistics Data, T-100 Carrier Data". http://www.transtats.bts.gov/Tables.asp?DB_ID=111&DB_Name=Air%20Carrier%2

0Statistics%20%28Form%2041%20Traffic%29-%20All%20Carriers&DB_Short_Name=Air%20Carriers.

31. Federal Aviation Administration (2006). "Aircraft Characteristics Database". http://www.faa.gov/arp/airchardb/. Accessed July 26, 2006
32. Baik, H., Ashiabor, S., Trani, A.A. (2006). "Development of an Airport Choice Model for General Aviation Operations".
33. Bonnefoy, P.A., Hansman, R.J. (2006). "Preliminary Analysis of the Potential Impacts of Very Light Jets on the Air Transportation System". MIT.
34. BUCHair (2004). "BUCHair BizJets Database".
35. Swedish, W.J. (1981). "Upgraded FAA Airfield Capacity Model Volume 1: Supplemental Users' Guide". The MITRE Corporation.

APPENDIX A ACRONYMS

ACM	Airfield Capacity Model
ATSL	Air Transportation Systems Laboratory
BTS	Bureau of Transportation Statistics
DOT	Department of Transportation
ETMS	Enhanced Traffic Management System
FAA	Federal Aviation Administration
GDP	Ground Delay Program
IFR	Instrument Flight Rule
IMC	Instrument Meteorological Conditions
MMC	Marginal Meteorological Conditions
NAS	National Airspace System
NASA	National Aeronautics and Space Administration
NGATS	Next Generation Air Transportation System
OAG	Official Airline Guides
ROT	Runway Occupancy Time
TAAM	Total Airspace and Airport Modeler
TAF	Terminal Area Forecast
TSAM	Transportation Systems Analysis Model
TRACON	Terminal RAdar CONtro
VFR	Visual Flight Rule
VLJ	Very Light Jet
VMC	Visual Meteorological Conditions

APPENDIX B RUNWAY SURFACE TYPES AND CONDITIONS ACCEPTED

Type	Description
ASPH	Asphalt or bituminous concrete
ASPH-CONC	Partial asphalt and partial Portland cement concrete, with runway end markings
ASPH-CONC-F	Partial asphalt and partial Portland cement concrete, with runway end markings in fair condition
ASPH-CONC-G	Partial asphalt and partial Portland cement concrete, with runway end markings in good condition
ASPH-CONC-P	Partial asphalt and partial Portland cement concrete, with runway end markings in poor condition
ASPH-E	
ASPH-F	Asphalt or bituminous concrete, with runway end marking in fair condition
ASPH-G	Asphalt or bituminous concrete, with runway end marking in good condition
ASPH-P	Asphalt or bituminous concrete, with runway end marking in poor condition
CONC	Portland cement concrete
CONC-E	
CONC-F	Portland cement concrete, with runway end markings in fair condition
CONC-G	Portland cement concrete, with runway end markings in good condition
CONC-P	Portland cement concrete, with runway end markings in poor condition

APPENDIX C T-100 SEGMENT DATA FIELDS LIST AND DESCRIPTION

Field Name	Description
Year	Year
Quarter	Quarter (1-4)
Month	Month
Origin	Origin Airport Code
OriginCityName	Origin Airport, City Name
OriginCityNum	Origin Airport, City Code
OriginState	Origin Airport, State
OriginStateFips	Origin Airport, State FIPS Code
OriginStateName	Origin Airport, State Name
OriginCountry	Origin Airport, Country
OriginCountryName	Origin Airport, Country Name
OriginWac	Origin Airport, World Area Code
Dest	Destination Airport Code
DestCityName	Dest Airport, City Name
DestCityNum	Destination Airport, City Code
DestState	Destination Airport, State
DestStateFips	Destination Airport, State FIPS Code
DestStateName	Destination Airport, State Name
DestCountry	Destination Airport, Country
DestCountryName	Destination Airport, Country Name
DestWac	Destination Airport, World Area Code
AirlineID	An identification number assigned by US DOT to identify a unique airline (carrier).
UniqueCarrier	Unique Carrier Code. When the same code has been used by multiple carriers, a numeric suffix is used for earlier users, for example, PA, PA(1), PA(2). Use this field for analysis across a range of years.
UniqueCarrierName	Unique Carrier Name. When the same name has been used by multiple carriers, a numeric suffix is used for earlier users, for example, Air Caribbean, Air Caribbean (1).
UniqCarrierEntity	Unique Carrier Entity. This field distinguishes entities used by two or more carriers with a numeric suffix, for example, 06038 and 06038(1).
CarrierRegion	Operating Region Of Entity
Carrier	Code assigned by IATA and commonly used to identify a carrier. As the same code may have been assigned to different carriers over time, the code is not always unique. For analysis, use the Unique Carrier Code.
CarrierName	Carrier Name
CarrierGroup	Carrier Group Code
CarrierGroupNew	Carrier Group New
DestStateFips	Destination Airport, State FIPS Code

DestStateName	Destination Airport, State Name
DestCountry	Destination Airport, Country
DestCountryName	Destination Airport, Country Name
DestWac	Destination Airport, World Area Code
AirlineID	An identification number assigned by US DOT to identify a unique airline (carrier). A unique airline (carrier) is defined as one holding and reporting under the same DOT certificate regardless of its Code, Name, or holding company/corporation.
UniqueCarrier	Unique Carrier Code. When the same code has been used by multiple carriers, a numeric suffix is used for earlier users, for example, PA, PA(1), PA(2). Use this field for analysis across a range of years.
UniqueCarrierName	Unique Carrier Name. When the same name has been used by multiple carriers, a numeric suffix is used for earlier users, for example, Air Caribbean, Air Caribbean (1).
UniqCarrierEntity	Unique Carrier Entity. This field distinguishes entities used by two or more carriers with a numeric suffix, for example, 06038 and 06038(1).
CarrierRegion	Operating Region Of Entity
Carrier	Code assigned by IATA and commonly used to identify a carrier. As the same code may have been assigned to different carriers over time, the code is not always unique. For analysis, use the Unique Carrier Code.
CarrierName	Carrier Name
CarrierGroup	Carrier Group Code
CarrierGroupNew	Carrier Group New
Distance	Distance
DistanceGroup	Distance Intervals, every 500 Miles, for Flight Segment
Class	Service Class
AircraftGroup	Aircraft Group
AircraftType	Aircraft Type
AircraftConfig	Aircraft Configuration
DepScheduled	Departures Scheduled
DepPerformed	Departures Performed
Payload	Payload, in Pounds
Seats	Available Seats
Passengers	Non-Stop Segment Passengers Transported
Freight	Non-Stop Segment Freight Transported (pounds)
Mail	Non-Stop Segment Mail Transported (pounds)
RampToRamp	Ramp-to-Ramp Time, in Minutes
AirTime	Air Time, in Minutes
DataSource	Source of Data (D=Domestic ,I=International)

[BTS Air Carrier Statistics Form 41 Traffic]

APPENDIX D AIRFIELD CAPACITY MODEL OUTPUT FILE EXAMPLE

```
                       10/26/2006     23:56:43
           ** FAA Airfield Capacity Model - Revised April 1999 **


1M1    0 0 0VMC
  12   1   1
RWY1   1 1 0
1.00 .00 .00 .00
RWY2   2 1 0
1.00 .00 .00 .00
ARBAR1 1 2 0
 38. 40. 43. 48.
ARBAR2 2 2 0
 38. 40. 43. 48.
DLTAIJ 0 4 0
 1.9 1.9 1.9 1.9 2.7 1.9 1.9 1.9 3.5 3.0 3.0 2.7 4.5 3.6 3.6 2.7
APPSPD 0 5 0
 120 130 140 145
DRBAR  0 6 0
  34  34  39  39
DDSR   0 7 0
  45  45  45  45  60  60  60  60 120 120 120 120 120 120 120 120
GAMA   0 8 0
   7   7   7   7
TGRBAR 0 9 0
 15. 15. 15. 15.
OPENV  010 0
 125 73210.0 2.0
TWOIN  011 0
   110.010.0 10. 10. 10. 10. 10. 10. 10. 10.
ADSR   112 0
 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10.
ADSR   212 0
 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10.
DICBR  013 0
 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0
SIGMAS 019 0
  8. 30.  0.  6.  6.
OTHERS 020 0
 .05 .05 3.0 5.05000 3.0 .007777  10  20  30  40  50  60  70  80  90
BDD    022 0
  0.  0.  0. 25.  0.  0.  0. 20.  0.  0.  0. 10. 25. 20. 10.  0.
BAA    024 0
  0.  0.  0. 25.  0.  0.  0. 20.  0.  0.  0. 10. 25. 20. 10.  0.
ALTARR 025 1
 2.0 3000.    0.  .0


TWO INTER BEYOND, AWAY, ARR ON #1, DEP ON #2
```

==============
CAPACITY TABLE
==============

NO.	POINT	ARRIVALS	DEPARTS	TOTAL
1	Arrival Priority	33.8	66.7	100.5
2	20. sec Max Gap Stretch	33.8	66.7	100.5
	Departure Priority	.0	66.7	66.7

TO OBTAIN 10 PERCENT ARRIVALS, OPERATE
AT POINT 1 FOR 21 PERCENT OF THE HOUR, AND
AT DEPARTURE PRIORITY POINT FOR 79 PERCENT

*** AIRFIELD HOURLY RUNWAY CAPACITY ***

TOTAL = 74.1 ARRIVALS = 7.4 DEPARTURES = 66.7

***** ***** ***** ***** ***** *****

TO OBTAIN 20 PERCENT ARRIVALS, OPERATE
AT POINT 1 FOR 49 PERCENT OF THE HOUR, AND
AT DEPARTURE PRIORITY POINT FOR 51 PERCENT

*** AIRFIELD HOURLY RUNWAY CAPACITY ***

TOTAL = 83.3 ARRIVALS = 16.7 DEPARTURES = 66.7

***** ***** ***** ***** ***** *****

TO OBTAIN 30 PERCENT ARRIVALS, OPERATE
AT POINT 1 FOR 84 PERCENT OF THE HOUR, AND
AT DEPARTURE PRIORITY POINT FOR 16 PERCENT

*** AIRFIELD HOURLY RUNWAY CAPACITY ***

TOTAL = 95.2 ARRIVALS = 28.6 DEPARTURES = 66.7

***** ***** ***** ***** ***** *****

TO OBTAIN 40 PERCENT ARRIVALS,
AVAILABLE DEPARTURE CAPACITY IS REDUCED BY 15.9 OPERATIONS PER HOUR

```
               ***  AIRFIELD HOURLY RUNWAY CAPACITY  ***

               TOTAL =    84.6  ARRIVALS =   33.8  DEPARTURES =   50.8


         *****     *****     *****     *****     *****     *****


TO OBTAIN  50 PERCENT ARRIVALS,
AVAILABLE DEPARTURE CAPACITY IS REDUCED BY  32.8  OPERATIONS PER HOUR


               ***  AIRFIELD HOURLY RUNWAY CAPACITY  ***

               TOTAL =    67.7  ARRIVALS =   33.8  DEPARTURES =   33.8


         *****     *****     *****     *****     *****     *****


TO OBTAIN  60 PERCENT ARRIVALS,
AVAILABLE DEPARTURE CAPACITY IS REDUCED BY  44.1  OPERATIONS PER HOUR


               ***  AIRFIELD HOURLY RUNWAY CAPACITY  ***

               TOTAL =    56.4  ARRIVALS =   33.8  DEPARTURES =   22.6


         *****     *****     *****     *****     *****     *****


TO OBTAIN  70 PERCENT ARRIVALS,
AVAILABLE DEPARTURE CAPACITY IS REDUCED BY  52.2  OPERATIONS PER HOUR


               ***  AIRFIELD HOURLY RUNWAY CAPACITY  ***

               TOTAL =    48.4  ARRIVALS =   33.8  DEPARTURES =   14.5


         *****     *****     *****     *****     *****     *****


TO OBTAIN  80 PERCENT ARRIVALS,
AVAILABLE DEPARTURE CAPACITY IS REDUCED BY  58.2  OPERATIONS PER HOUR


               ***  AIRFIELD HOURLY RUNWAY CAPACITY  ***

               TOTAL =    42.3  ARRIVALS =   33.8  DEPARTURES =    8.5


         *****     *****     *****     *****     *****     *****
```

TO OBTAIN 90 PERCENT ARRIVALS,
AVAILABLE DEPARTURE CAPACITY IS REDUCED BY 62.9 OPERATIONS PER HOUR

*** AIRFIELD HOURLY RUNWAY CAPACITY ***

TOTAL =   37.6 ARRIVALS =   33.8 DEPARTURES =   3.8

*****   *****   *****   *****   *****   *****

APPENDIX E-1 MATLAB PROGRAM FILE FOR SINGLE RUNWAY AIRPORT

```
%
************************************************************************
% Program to get capacities for single runway airports with ACM

% In order to get correct categorization result here, it's VERY IMPORTANT
% to sort runway ends excel file with airportID, runwayID, and runway-end
ID

% Required input file: VLJ-Single Runways.xls, which contains runway
information

% Once the input file is ready, this Matlab program connects it with ACM
so that ACM
% can calculates capacity number for airports listed in the input file.
% The output of the program is the upper-boundary and lower-boundary
% The output is also send to an excel file which will have capacity
% informatio for all airports of interest

% Output file: output_Single, which contains three tables that have
capcity number
% for VMC, MMC, and IMC weather scenario respectively

% Coded by Yueh-Ting Chen, April 7, 2006
% Modified: Oct 23,2006
%
************************************************************************



%%  load RunwayInfo and generate input file for ACM
% First load runway data from excel file.
% Second distribute all airports into four categories with respect of
number of runways.
clc

[runwayData, runwayList]   =
xlsread('D:\YuehTing\Research_Project\Data\VLJ-Single Runway.xls');
[ArrArr_Data, ArrArr_Text] =
xlsread('D:\YuehTing\Research_Project\Data\ArrArr_Matrix.xls');  %
separation data for all scenarios
[DepDep_Data, DepDep_Text] =
xlsread('D:\YuehTing\Research_Project\Data\DepDep_Matrix.xls');  %
separation data for all scenarios

% runwayData: numeric data in excel file;
% runwayList: string data in excel file;
% each "runway end" occupies a row in excel file;

[RunwayEnd_No NoColumnRunwayData] = size(runwayData);     % [how many
runway ends, how many columns in file "runwayData"]
```

```
NoColumnRunwayList                = size(runwayList,2);   % how many
columns in file "runwayList"
output_Single_VMC = [];
output_Single_MMC = [];
output_Single_IMC = [];
weather = ['VMC';'MMC';'IMC'];

for wi = 1:3;       % wi, weather index; VMC, MMC, IMC
    output  = {};
    apIndex = 0;    % airport Index, for the use of ourput matrix

    for i = 2 : RunwayEnd_No
        if ~strcmp(runwayList{i,1},runwayList{i-1,1})
            apIndex = apIndex + 1;
            filename = strcat(runwayList{i,1},'_',weather(wi,:));
            fullname = strcat(filename,'.inp');
            fp1 = fopen(fullname,'w');

% start to build input file for ACM
            fprintf(fp1,'%-
6s%2d%2d%2d%3s\n',runwayList{i,1},0,0,0,weather(wi,:));
            fprintf(fp1,'%4d%4d%4d\n',1,3,0);

            fprintf(fp1,'%-6s%2d%2d%2d\n','RWY1',1,1,0);          %
aircraft mix on runway 1 (small to heavy)
            fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',runwayData(i-
1,20),runwayData(i-1,21),runwayData(i-1,22),runwayData(i-1,23)); %
******************need modification**************

            fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR1',1,2,0);        %
average arr. ROT for all aircraft type at runway1
            fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);

            fprintf(fp1,'%-6s%2d%2d%2d\n','DLTAIJ',0,4,0);        %
minimum arrival separations, in nm, for i followed by j, from smallest to
largest

            if  strcmp(runwayList{i,20},'Y');                    % see if
this aiport has tower on site

                switch wi                                      %
separation for aiport with tower
                    case 1

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(2,2),ArrArr_Data(2,3),ArrArr_Data(2,4),ArrArr_Data(2,5),...

ArrArr_Data(3,2),ArrArr_Data(3,3),ArrArr_Data(3,4),ArrArr_Data(3,5),...

ArrArr_Data(4,2),ArrArr_Data(4,3),ArrArr_Data(4,4),ArrArr_Data(4,5),...

ArrArr_Data(5,2),ArrArr_Data(5,3),ArrArr_Data(5,4),ArrArr_Data(5,5));
                    otherwise
```

```
fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(9,2),ArrArr_Data(9,3),ArrArr_Data(9,4),ArrArr_Data(9,5),...

ArrArr_Data(10,2),ArrArr_Data(10,3),ArrArr_Data(10,4),ArrArr_Data(10,5),..
.

ArrArr_Data(11,2),ArrArr_Data(11,3),ArrArr_Data(11,4),ArrArr_Data(11,5),..
.

ArrArr_Data(12,2),ArrArr_Data(12,3),ArrArr_Data(12,4),ArrArr_Data(12,5));
                end
            else                                                %
separation for aiport without tower
                switch wi
                    case 1

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(2,9),ArrArr_Data(2,10),ArrArr_Data(2,11),ArrArr_Data(2,12),...

ArrArr_Data(3,9),ArrArr_Data(3,10),ArrArr_Data(3,11),ArrArr_Data(3,12),...

ArrArr_Data(4,9),ArrArr_Data(4,10),ArrArr_Data(4,11),ArrArr_Data(4,12),...

ArrArr_Data(5,9),ArrArr_Data(5,10),ArrArr_Data(5,11),ArrArr_Data(5,12));
                    otherwise

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(9,9),ArrArr_Data(9,10),ArrArr_Data(9,11),ArrArr_Data(9,12),...

ArrArr_Data(10,9),ArrArr_Data(10,10),ArrArr_Data(10,11),ArrArr_Data(10,12)
,...

ArrArr_Data(11,9),ArrArr_Data(11,10),ArrArr_Data(11,11),ArrArr_Data(11,12)
,...

ArrArr_Data(12,9),ArrArr_Data(12,10),ArrArr_Data(12,11),ArrArr_Data(12,12)
);
                end
            end

            fprintf(fp1,'%-6s%2d%2d%2d\n','APPSPD',0,5,0);        %
approach speed, in knots
            fprintf(fp1,'%4d%4d%4d%4d\n',120,130,140,145);

            fprintf(fp1,'%-6s%2d%2d%2d\n','DRBAR',0,6,0);         %
departure runway occupancy time, in seconds
            fprintf(fp1,'%4d%4d%4d%4d\n',34,34,39,39);
```

```
            fprintf(fp1,'%-6s%2d%2d%2d\n','DDSR',0,7,0);          %
minimum departure separations, in seconds, for i followed by j

            if  strcmp(runwayList{i,20},'Y');                   % see if
this aiport has tower on site
                switch wi                                       %
separation for aiport with tower
                    case 1

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(2,2),DepDep_Data(2,3),DepDep_Data(2,4),DepDep_Data(2,5),...

DepDep_Data(3,2),DepDep_Data(3,3),DepDep_Data(3,4),DepDep_Data(3,5),...

DepDep_Data(4,2),DepDep_Data(4,3),DepDep_Data(4,4),DepDep_Data(4,5),...

DepDep_Data(5,2),DepDep_Data(5,3),DepDep_Data(5,4),DepDep_Data(5,5));
                    otherwise

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(9,2),DepDep_Data(9,3),DepDep_Data(9,4),DepDep_Data(9,5),...

DepDep_Data(10,2),DepDep_Data(10,3),DepDep_Data(10,4),DepDep_Data(10,5),..
.

DepDep_Data(11,2),DepDep_Data(11,3),DepDep_Data(11,4),DepDep_Data(11,5),..
.

DepDep_Data(12,2),DepDep_Data(12,3),DepDep_Data(12,4),DepDep_Data(12,5));
                end
            else                                               %
separation for aiport without tower
                switch wi
                    case 1

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(2,9),DepDep_Data(2,10),DepDep_Data(2,11),DepDep_Data(2,12),...

DepDep_Data(3,9),DepDep_Data(3,10),DepDep_Data(3,11),DepDep_Data(3,12),...

DepDep_Data(4,9),DepDep_Data(4,10),DepDep_Data(4,11),DepDep_Data(4,12),...

DepDep_Data(5,9),DepDep_Data(5,10),DepDep_Data(5,11),DepDep_Data(5,12));
                    otherwise

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(9,9),DepDep_Data(9,10),DepDep_Data(9,11),DepDep_Data(9,12),...

DepDep_Data(10,9),DepDep_Data(10,10),DepDep_Data(10,11),DepDep_Data(10,12)
,...

DepDep_Data(11,9),DepDep_Data(11,10),DepDep_Data(11,11),DepDep_Data(11,12)
,...
```

```
DepDep_Data(12,9),DepDep_Data(12,10),DepDep_Data(12,11),DepDep_Data(12,12)
);
                end
            end

            fprintf(fp1,'%-6s%2d%2d%2d\n','GAMA',0,8,0);       % length of
final approach path, in nm
            if  (~strcmp(runwayList{i,20},'Y')) && (wi == 1);
                fprintf(fp1,'%4d%4d%4d%4d\n',15,15,15,15);
            else fprintf(fp1,'%4d%4d%4d%4d\n',7,7,7,7);
            end

            fprintf(fp1,'%-6s%2d%2d%2d\n','TGRBAR',0,9,0);     % touch-
and-go runway occupancy time, in seconds
            fprintf(fp1,'%4d%4d%4d%4d\n',15,15,15,15);

            fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',0,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);


            fprintf(fp1,'%-6s%2d%2d%2d\n','DICBR',0,13,0);     % the
separation(nm) between departure I and arrival J on the crossing runway;
needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...
                3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3);


            fprintf(fp1,'%-6s%2d%2d%2d\n','SIGMAS',0,19,0);
            if  (~strcmp(runwayList{i,20},'Y')) && (wi == 1);
                fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f\n',8,30,0,6,6);
            else
                fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f\n',8,18,0,6,6);
            end
            % sigma of arrival ROT          (seconds),
            % sigma of interarrival time    (seconds),
            % sigma of touch and go ROT     (seconds),
            % sigma of cleared-to-roll time(seconds),
            % sigma of dep. ROT             (seconds)

            switch wi
                case 1      % VMC
                    visi = 5.0;
                    ceil = 5000;
                case 2      % MMC
                    visi = 2.5;          % needs check
                    ceil = 900;          % needs check
                case 3      % IMC
```

```
                visi = 0.0;
                ceil = 0;
            end %114% switch wi
            fprintf(fp1,'%-6s%2d%2d%2d\n','OTHERS',0,20,0);
            if (wi ~= 1) && (~strcmp(runwayList{i,20},'Y'))

fprintf(fp1,'%4.2f%4.2f%4.1f%4.1f%4.0f%4.1f%4.2f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f%4.0f\n',...

0.05,0.05,15,visi,ceil,3,0,7777,10,20,30,40,50,60,70,80,90);
            else

fprintf(fp1,'%4.2f%4.2f%4.1f%4.1f%4.0f%4.1f%4.2f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f%4.0f\n',...

0.05,0.05,3,visi,ceil,3,0,7777,10,20,30,40,50,60,70,80,90);
            end
            % %of violation for arrival ROT,
            % %of violation for interarrival time,
            % dep/arr separation(nm),
            % visibility(statute miles),
            % ceiling(feet),
            % glide slope angle (degrees),
            % proportion of touch and go operations,
            % arrival percentages: 8888 (arrival priority and departure
priority), 10%~90%

            fprintf(fp1,'%-6s%2d%2d%2d\n','BDD',0,22,0);        % buffer
time (seconds), between simultaneous dep. on close-spaced parallel runways

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                0,0,0,25,0,0,0,20,0,0,0,10,25,20,10,0);

            fprintf(fp1,'%-6s%2d%2d%2d\n','BAA',0,24,0);        % buffer
time (seconds), between simultaneous arr. on close-spaced parallel runways

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                0,0,0,25,0,0,0,20,0,0,0,10,25,20,10,0);

            fprintf(fp1,'%-6s%2d%2d%2d\n','ALTARR',0,25,1);
            fprintf(fp1,'%4.1f%6.0f%6.0f%4.1f\n',2,3000,0,0);
            % diagonal separation (nm); separation between runway
centerlines (feet);
            % thresholds displacement (feet); relative displacement of
gates for two runways (nm);

        %    fprintf(fp1,'%-6s%2D%2D%2D\n','INCIAT',0,26,1);      %
maximum number of iterations on the f.e.d. mix; convergence
        %    fprintf(fp1,'%4.0f%4.3f%4.0f%4.0f\n',0.05,0.05,0.20,0.1);
            % criterion for f.e.d mix iterations; maximum number of points
for
            % arrival capacity is calculated; increment by which the
arrival gaps are stretched (seconds);
            fclose(fp1);
```

```
%%  finish the building of ACM input file, then output the file and run
with ACM
            dos(['ACMrun ',filename]);


%%  capture the result of ACM from output files and output these number to
output matrix
            CurrentOutputFile = fopen(strcat(filename,'.res'),'r');
            totLines = 0;   % the serial number of line which is reading
in this specific output file
            percent  = 0;   % index for arrival percentage
            output{apIndex,1} = runwayList{i,1};    % the first column is
airport ID
            output{apIndex,2} = 1;                  % model index in ACM
            output{apIndex,3} = 3;                  % operation strategy
index in ACM

            while (~feof(CurrentOutputFile))
                totLines = totLines + 1;
                aLine    = fgetl(CurrentOutputFile);

                if ~isempty(findstr(aLine,'Arrival Priority'))      % look
for the line which has keyword "Arrival Priority"
                    output{apIndex,4} = str2double(aLine(37:45));   % save
Arrival Priority Arrival capacity number in the struct "output"
                    output{apIndex,5} = str2double(aLine(49:56));   % save
Arrival Priority Departure capacity number in the struct "output"
                    output{apIndex,6} = str2double(aLine(37:45)) +
str2double(aLine(49:56));
                elseif ~isempty(findstr(aLine,'Departure Priority'))    %
look for the line which has keyword "Departure Priority"
                    output{apIndex,7} = str2double(aLine(37:45));    %
save Departure Priority Arrival capacity number in the struct "output"
                    output{apIndex,8} = str2double(aLine(49:56));    %
save Departure Priority Departure capacity number in the struct "output"
                    output{apIndex,9} = str2double(aLine(37:45)) +
str2double(aLine(49:56));
                elseif ~isempty(findstr(aLine,'TOTAL ='))               %
look for the line which has keyword "TOTAL"
                    percent = percent + 1;

                    switch percent
                        case 1
                            output{apIndex,10} = str2double(aLine(44:50));
% save 10%-Arrival arr number
                            output{apIndex,11} = str2double(aLine(66:70));
% save 10%-Arrival dep number
                            output{apIndex,12} = str2double(aLine(44:50))
+ str2double(aLine(66:70));
                        case 2
                            output{apIndex,13} = str2double(aLine(44:50));
% save 20%-Arrival arr number
```

```
                                output{apIndex,14} = str2double(aLine(66:70));
% save 20%-Arrival dep number
                                output{apIndex,15} = str2double(aLine(44:50))
+ str2double(aLine(66:70));
                            case 3
                                output{apIndex,16} = str2double(aLine(44:50));
% save 30%-Arrival arr number
                                output{apIndex,17} = str2double(aLine(66:70));
% save 30%-Arrival dep number
                                output{apIndex,18} = str2double(aLine(44:50))
+ str2double(aLine(66:70));
                            case 4
                                output{apIndex,19} = str2double(aLine(44:50));
% save 40% Arrival number in the struct "output"
                                output{apIndex,20} = str2double(aLine(66:70));
% save 40% Arrival number in the struct "output"
                                output{apIndex,21} = str2double(aLine(44:50))
+ str2double(aLine(66:70));
                            case 5
                                output{apIndex,22} = str2double(aLine(44:50));
% save 50% Arrival number in the struct "output"
                                output{apIndex,23} = str2double(aLine(66:70));
% save 50% Arrival number in the struct "output"
                                output{apIndex,24} = str2double(aLine(44:50))
+ str2double(aLine(66:70));
                            case 6
                                output{apIndex,25} = str2double(aLine(44:50));
% save 60% Arrival number in the struct "output"
                                output{apIndex,26} = str2double(aLine(66:70));
% save 60% Arrival number in the struct "output"
                                output{apIndex,27} = str2double(aLine(44:50))
+ str2double(aLine(66:70));
                            case 7
                                output{apIndex,28} = str2double(aLine(44:50));
% save 70% Arrival number in the struct "output"
                                output{apIndex,29} = str2double(aLine(66:70));
% save 70% Arrival number in the struct "output"
                                output{apIndex,30} = str2double(aLine(44:50))
+ str2double(aLine(66:70));
                            case 8
                                output{apIndex,31} = str2double(aLine(44:50));
% save 80% Arrival number in the struct "output"
                                output{apIndex,32} = str2double(aLine(66:70));
% save 80% Arrival number in the struct "output"
                                output{apIndex,33} = str2double(aLine(44:50))
+ str2double(aLine(66:70));
                            case 9
                                output{apIndex,34} = str2double(aLine(44:50));
% save 90% Arrival number in the struct "output"
                                output{apIndex,35} = str2double(aLine(66:70));
% save 90% Arrival number in the struct "
                                output{apIndex,36} = str2double(aLine(44:50))
+ str2double(aLine(66:70));
                        end %170% switch percent
                    end %156% if ~isempty(findstr(aLine,'Arrival Priority'))
                end  %160% while (~feof(CurrentOutputFile))
```

```
            fclose(CurrentOutputFile); % finish reading the specific
output file


%%  output and send files to correspond folders

            movefile(fullname,'d:\YuehTing\Research_Project\Single Runway
ACM\ACMinput\');

movefile(strcat(filename,'.res'),'d:\YuehTing\Research_Project\Single
Runway ACM\ACMoutput\');
        end %40% if ~strcmp(runwayList{i,1},runwayList{i-1,1})
    end %39% for i = 2 : RunwayEnd_No

    switch wi
        case 1
            output_Single_VMC = output;
        case 2
            output_Single_MMC = output;
        case 3
            output_Single_IMC = output;
    end %219% switch wi
end %35% for wi = 1:3;

save output_Single output_Single_VMC output_Single_MMC output_Single_IMC

xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',output_Single_VMC,'VMC_Min','A805');
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',output_Single_MMC,'MMC_Min','A805');
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',output_Single_IMC,'IMC_Min','A805');
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',output_Single_VMC,'VMC_Max','A805');
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',output_Single_MMC,'MMC_Max','A805');
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',output_Single_IMC,'IMC_Max','A805');
```

APPENDIX E-2 MATLAB PROGRAM FILE FOR MULTI-RUNWAY AIRPORT-RUNWAY CLASSIFICATION

```
%
*************************************************************************
% Program to calculate capacities for multiple runway airports from ACM

% divide all multiple runway airports into four categories:
% two runways
% three runways
% four runways
% five or more runways

% In order to get correct categorization result here, it's VERY IMPORTANT
% to sort runway ends excel file with airportID, runwayID, and runway-end
ID

% then deal with the configurations within each category.

% distribute all two runway airports into four different configurations
% distribute all three runway airports into four different configurations
% distribute all four runway airports into four different configurations

% load characteristics of each airport and calculate capacity

% Coded by Yueh-Ting Chen, April 7, 2006
% Last Modifaction: Nov 17,2006
%
*************************************************************************


%% load RunwayInfo and Categorization
% DESCRIPTIVE TEXT
% First load runway data from excel file.
% Second distribute all airports into four categories with respect of
number of runways.

[runwayData, runwayList]          =
xlsread('D:\YuehTing\Research_Project\Data\VLJ-Multiple Runways.xls');
 % runwayData: numeric data in excel file;
 % runwayList: string data in excel file;
 % each "runway end" occupies a row in excel file;

[RunwayEnd_No NoColumnRunwayData] = size(runwayData);       % [how many
runway ends, how many columns in file "runwayData"]
NoColumnRunwayList                = size(runwayList,2);     % how many
columns in file "runwayList"

for s = 1 : RunwayEnd_No
```

```
    strs1 = (runwayList(s+1, NoColumnRunwayList - 1));         % strs1 =
Latitude
    strs2 = (runwayList(s+1, NoColumnRunwayList));             % strs2 =
Longitude
    runwayData(s, NoColumnRunwayData + 1) = str2angle(strs1);  % change
unit of Lat from (degree,minutes,seconds) to degree
    runwayData(s, NoColumnRunwayData + 2) = str2angle(strs2);  % change
unit of Long from (degree,minutes,seconds) to degree
end %38% for s = 1 : RunwayEnd_No

% Initialization
Two_RWY_Ends_No             = 0;     % number of ends for "two-runway" case
Three_RWY_Ends_No           = 0;     % number of ends for "three-runway"
case
Four_RWY_Ends_No            = 0;     % number of ends for "four-runway"
case
More_Than_Four_RWY_Ends_No = 0;     % number of ends for case of "five or
more runway"

% assort all runway end-based runway information with appropriate folder
% (e.g. 'runway with four-end', 'runway with five-end')

for n = 1:RunwayEnd_No
    totNumberOfRunwayEnd = runwayData(n,1);
    switch totNumberOfRunwayEnd
        case 4      % find four-end two runways airport
            Two_RWY_Ends_No = Two_RWY_Ends_No + 1;                  % the
number of runway ends
            Data_of_Two_RWY(Two_RWY_Ends_No,:) = runwayData(n,:);
            List_of_Two_RWY(Two_RWY_Ends_No,:) = runwayList(n+1,:);
            % the first row in runwayList file is index, so the correct
data we want should be row (n+1)
        case 6      % find six-end three runways airport
            Three_RWY_Ends_No = Three_RWY_Ends_No + 1;              % the
number of runway ends
            Data_of_Three_RWY(Three_RWY_Ends_No,:) = runwayData(n,:);
            List_of_Three_RWY(Three_RWY_Ends_No,:) = runwayList(n+1,:);
            % the first row in runwayList file is index, so the correct
data we want should be row (n+1)
        case 8     % find eight-end four runways airport
            Four_RWY_Ends_No = Four_RWY_Ends_No + 1;                  %
the number of runway ends
            Data_of_Four_RWY(Four_RWY_Ends_No,:) = runwayData(n,:);
            List_of_Four_RWY(Four_RWY_Ends_No,:) = runwayList(n+1,:);
            % the first row in runwayList file is index, so the correct
data we want should be row (n+1)
        otherwise % find more than four runways airport
            More_Than_Four_RWY_Ends_No = More_Than_Four_RWY_Ends_No + 1;
% the number of runway ends
            Data_of_More_Than_Four_RWY(More_Than_Four_RWY_Ends_No,:) =
runwayData(n,:);
            List_of_More_Than_Four_RWY(More_Than_Four_RWY_Ends_No,:) =
runwayList(n+1,:);
            % the first row in runwayList file is index, so the correct
data we want should be row (n+1)
    end %58% switch totNumberOfRunwayEnd
```

```
end %56% for n = 1:RunwayEnd_No

save Two_RWY_Characteristics Data_of_Two_RWY List_of_Two_RWY;
save Three_RWY_Characteristics Data_of_Three_RWY List_of_Three_RWY;
save Four_RWY_Characteristics Data_of_Four_RWY List_of_Four_RWY;
save More_Than_Four_RWY_Characteristics Data_of_More_Than_Four_RWY
List_of_More_Than_Four_RWY;



%%   Two runways
% Now we deal with all two-runway data.
% Distribute information in "Data_of_Two_RWY" and "List_of_Two_RWY" into
% subset files.
% There are four subset for all two-runway cases:
Two_Parallel_No        = 0;     % number of airports have two-parallel-
runway;                        model  2 in ACM
Two_Intersect_No       = 0;     % number of airports have two-intersect-
runway;                       model  6 in ACM
Two_OpenV_No           = 0;     % number of airports have two-openV-
runway;                          model  5 in ACM
Two_InterBeyonThres_No = 0;     % number of airports have two-
intersecting-beyond-threshold-runway; model 12 in ACM

% Initialization
NumApTwo               = 0;     % number of airports belong to these four
sets
NumEndTwo = size(Data_of_Two_RWY,1);   % how many rows in file
"Data_of_Two_RWY", in other word, how many
                                % ends for all two runway cases
for rN = 1:4:(NumEndTwo-3)
    NumApTwo = NumApTwo + 1;    % count number of two-runway airports
    % coordinates of end points on both line segments
    x1 = [Data_of_Two_RWY(rN,25)    Data_of_Two_RWY(rN+1,25)];       % x
coordinates(two ends) of line 1
    x2 = [Data_of_Two_RWY(rN+2,25)  Data_of_Two_RWY(rN+3,25)];       % x
coordinates(two ends) of line 2
    y1 = [Data_of_Two_RWY(rN,24)    Data_of_Two_RWY(rN+1,24)];       % y
coordinates(two ends) of line 1
    y2 = [Data_of_Two_RWY(rN+2,24)  Data_of_Two_RWY(rN+3,24)];       % y
coordinates(two ends) of line 2

    dist1 = distdim(distance(y1(1),x1(1),y2(1),x2(1)),'deg','ft');   %
calculate distance between two center line
    dist2 = distdim(distance(y1(1),x1(1),y2(2),x2(2)),'deg','ft');   %
calculate distance between two center line
    dist3 = distdim(distance(y1(2),x1(2),y2(1),x2(1)),'deg','ft');   %
calculate distance between two center line
    dist4 = distdim(distance(y1(2),x1(2),y2(2),x2(2)),'deg','ft');   %
calculate distance between two center line
    ShortestL1L2 = min([dist1,dist2,dist3,dist4]);                   % the
shortest distance between two line segments

    if  strncmp(List_of_Two_RWY{rN,17},List_of_Two_RWY{rN+2,17},2)  % see
if the Base-End ID is the same
        Two_Parallel_No = Two_Parallel_No + 1;                      % if
so, then these pair will be parallel RWY
```

```
        Config_Two(NumApTwo,:) = [List_of_Two_RWY(rN,:) 2 {ShortestL1L2}
'N/A' Data_of_Two_RWY(rN,20) Data_of_Two_RWY(rN,21) Data_of_Two_RWY(rN,22)
Data_of_Two_RWY(rN,23)];     % ********* new try ***************

    else
        delta  = det([(y1(1)-y1(2)) (x1(2)-x1(1)); (y2(1)-y2(2)) (x2(2)-
x2(1))]);
        deltaX = det([(y1(2)*x1(1)-y1(1)*x1(2)) -(x1(2)-x1(1));
(y2(2)*x2(1)-y2(1)*x2(2)) -(x2(2)-x2(1))]);
        deltaY = det([(y1(1)-y1(2)) -(y1(2)*x1(1)-y1(1)*x1(2)); (y2(1)-
y2(2)) -(y2(2)*x2(1)-y2(1)*x2(2))]);
        x = deltaX/ delta;      % X coordinate for intersecting point
        y = deltaY/ delta;      % Y coordinate for intersecting point

        %
*********************************************************************
        angle1 = azimuth('rh',y1(1),x1(1),y1(2),x1(2));    % calculate the
azimuth of runway1
        angle2 = azimuth('rh',y2(1),x2(1),y2(2),x2(2));    % calculate the
azimuth of runway2
        angle = abs(angle1-angle2);                        % calculate the
angle between two RWYs
        %
*********************************************************************

        if ((x1(1)>= x) && (x >= x1(2)))||((x1(1) <= x) && (x <= x1(2)))
            if ((x2(1)>= x) && (x >= x2(2)))||((x2(1) <= x) && (x <=
x2(2)))  % see if the intersecting point lies on both line segments
                Two_Intersect_No = Two_Intersect_No + 1;
% this runway pair is intersecting runways
                Config_Two(NumApTwo,:) = [List_of_Two_RWY(rN,:) 6 'N/A'
'N/A' Data_of_Two_RWY(rN,20) Data_of_Two_RWY(rN,21) Data_of_Two_RWY(rN,22)
Data_of_Two_RWY(rN,23)];
            else
                Two_InterBeyonThres_No = Two_InterBeyonThres_No + 1;
% this runway pair is intersect-beyond-threshold runways
                Config_Two(NumApTwo,:) = [List_of_Two_RWY(rN,:) 12
{ShortestL1L2} {angle} Data_of_Two_RWY(rN,20) Data_of_Two_RWY(rN,21)
Data_of_Two_RWY(rN,22) Data_of_Two_RWY(rN,23)];
            end  %151% if ((x2(1)>= x) && (x >= x2(2)))||((x2(1) <= x) &&
(x <= x2(2)))
        elseif ((x2(1)>= x) && (x >= x2(2)))||((x2(1) <= x) && (x <=
x2(2)))
            Two_InterBeyonThres_No = Two_InterBeyonThres_No + 1;
% this runway pair is intersect-beyond-threshold runways
            Config_Two(NumApTwo,:) = [List_of_Two_RWY(rN,:) 12
{ShortestL1L2} {angle} Data_of_Two_RWY(rN,20) Data_of_Two_RWY(rN,21)
Data_of_Two_RWY(rN,22) Data_of_Two_RWY(rN,23)];
        else
            Two_OpenV_No = Two_OpenV_No + 1;
% or they will be openV pair
            Config_Two(NumApTwo,:) = [List_of_Two_RWY(rN,:) 5
{ShortestL1L2} {angle} Data_of_Two_RWY(rN,20) Data_of_Two_RWY(rN,21)
Data_of_Two_RWY(rN,22) Data_of_Two_RWY(rN,23)];
        end  %136% ((x1(1)>= x) && (x >= x1(2)))||((x1(1) <= x) && (x <=
x1(2)))
```

```
    end  %117% List_Two_Parallel(rN,17) == List_Two_Parallel(rN+2,17)
end  %110% rN = 1:4:(a-3)

save TwoProfiles Two_Parallel_No Two_Intersect_No Two_OpenV_No
Two_InterBeyonThres_No

%%   Three runways
% % DESCRIPTIVE TEXT
% Now we deal with all three runways data.
% Further distributing all information in "Data_of_Three_RWY" and
% "List_of_Three_RWY" into sub-type files.
% There are four sub-types for all two-runway cases:
Three_Parallel_No        = 0;     % number of airports have three-
parallel-runway;                      model  3 in ACM
Three_Intersect_No       = 0;     % number of airports have three-
intersect-runway;                     model  7 in ACM
Three_OpenV_No           = 0;     % number of airports have three-openV-
runway;                         model 10 in ACM
Three_InterBeyonThres_No = 0;     % number of airports have three-
intersecting-beyond-threshold-runway; model 13 in ACM
Three_Non_ACM            = 0;     % number of three runway airports unable
to use ACM

% Initialization
NumApThree               = 0;     % number of airports belong to three-
runway case
NumEndThree = size(Data_of_Three_RWY,1);

for rN = 1:6:(NumEndThree-5)
    NumApThree = NumApThree + 1;    % count number of two-runway airports
    % coordinates of end points of these three runways
    x1 = [Data_of_Three_RWY(rN,25)   Data_of_Three_RWY(rN+1,25)];  % x
coordinates(two ends) of line 1
    x2 = [Data_of_Three_RWY(rN+2,25) Data_of_Three_RWY(rN+3,25)];  % x
coordinates(two ends) of line 2
    x3 = [Data_of_Three_RWY(rN+4,25) Data_of_Three_RWY(rN+5,25)];  % x
coordinates(two ends) of line 3
    y1 = [Data_of_Three_RWY(rN,24)   Data_of_Three_RWY(rN+1,24)];  % y
coordinates(two ends) of line 1
    y2 = [Data_of_Three_RWY(rN+2,24) Data_of_Three_RWY(rN+3,24)];  % y
coordinates(two ends) of line 2
    y3 = [Data_of_Three_RWY(rN+4,24) Data_of_Three_RWY(rN+5,24)];  % y
coordinates(two ends) of line 3

    dist121 = distdim(distance(y1(1),x1(1),y2(1),x2(1)),'deg','ft');   %
calculate distance between line 1 and line 2
    dist122 = distdim(distance(y1(1),x1(1),y2(2),x2(2)),'deg','ft');   %
calculate distance between line 1 and line 2
    dist123 = distdim(distance(y1(2),x1(2),y2(1),x2(1)),'deg','ft');   %
calculate distance between line 1 and line 2
    dist124 = distdim(distance(y1(2),x1(2),y2(2),x2(2)),'deg','ft');   %
calculate distance between line 1 and line 2
    dist231 = distdim(distance(y2(1),x2(1),y3(1),x3(1)),'deg','ft');   %
calculate distance between line 2 and line 3
    dist232 = distdim(distance(y2(1),x2(1),y3(2),x3(2)),'deg','ft');   %
calculate distance between line 2 and line 3
```

```
    dist233 = distdim(distance(y2(2),x2(2),y3(1),x3(1)),'deg','ft');   %
calculate distance between line 2 and line 3
    dist234 = distdim(distance(y2(2),x2(2),y3(2),x3(2)),'deg','ft');   %
calculate distance between line 2 and line 3
    dist131 = distdim(distance(y1(1),x1(1),y3(1),x3(1)),'deg','ft');   %
calculate distance between line 1 and line 3
    dist132 = distdim(distance(y1(1),x1(1),y3(2),x3(2)),'deg','ft');   %
calculate distance between line 1 and line 3
    dist133 = distdim(distance(y1(2),x1(2),y3(1),x3(1)),'deg','ft');   %
calculate distance between line 1 and line 3
    dist134 = distdim(distance(y1(2),x1(2),y3(2),x3(2)),'deg','ft');   %
calculate distance between line 1 and line 3

    shortestL1L2 = min([dist121,dist122,dist123,dist124]);             %
the shortest distance between two line segments
    shortestL2L3 = min([dist231,dist232,dist233,dist234]);             %
the shortest distance between two line segments
    shortestL1L3 = min([dist131,dist132,dist133,dist134]);             %
the shortest distance between two line segments

    shortestL1L2L3 = min([shortestL1L2,shortestL2L3,shortestL1L3]);
    longestL1L2L3  = max([shortestL1L2,shortestL2L3,shortestL1L3]);

    ID1 = sscanf(List_of_Three_RWY{rN,17},'%f',2);
    ID2 = sscanf(List_of_Three_RWY{rN+2,17},'%f',2);
    ID3 = sscanf(List_of_Three_RWY{rN+4,17},'%f',2);

% set 1
    if (shortestL1L2 < 2500) && (abs(ID1-ID2) <= 0)
        if (abs(ID3-ID2) <= 0) && (shortestL2L3 > 700)
            Three_Parallel_No = Three_Parallel_No + 1;    % if so, then
these runways will be parallel RWY
            Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:) 3
{shortestL2L3} 'N/A' Data_of_Three_RWY(rN,20) Data_of_Three_RWY(rN,21)
Data_of_Three_RWY(rN,22) Data_of_Three_RWY(rN,23)];
        else
            delta23  = det([(y2(1)-y2(2)) (x2(2)-x2(1)); (y3(1)-y3(2))
(x3(2)-x3(1))]);
            delta23X = det([(y2(2)*x2(1)-y2(1)*x2(2)) -(x2(2)-x2(1));
(y3(2)*x3(1)-y3(1)*x3(2)) -(x3(2)-x3(1))]);
            delta23Y = det([(y2(1)-y2(2)) -(y2(2)*x2(1)-y2(1)*x2(2));
(y3(1)-y3(2)) -(y3(2)*x3(1)-y3(1)*x3(2))]);
            x23 = delta23X/ delta23;      % X coordinate for intersecting
point
            y23 = delta23Y/ delta23;      % Y coordinate for intersecting
point

            delta13  = det([(y1(1)-y1(2)) (x1(2)-x1(1)); (y3(1)-y3(2))
(x3(2)-x3(1))]);
            delta13X = det([(y1(2)*x1(1)-y1(1)*x1(2)) -(x1(2)-x1(1));
(y3(2)*x3(1)-y3(1)*x3(2)) -(x3(2)-x3(1))]);
            delta13Y = det([(y1(1)-y1(2)) -(y1(2)*x1(1)-y1(1)*x1(2));
(y3(1)-y3(2)) -(y3(2)*x3(1)-y3(1)*x3(2))]);
            x13 = delta23X/ delta23;      % X coordinate for intersecting
point
```

```
            y13 = delta23Y/ delta23;      % Y coordinate for intersecting
point

            angle1 = azimuth('rh',y1(1),x1(1),y1(2),x1(2));    % calculate
the azimuth of runway2
            angle2 = azimuth('rh',y2(1),x2(1),y2(2),x2(2));    % calculate
the azimuth of runway2
            angle3 = azimuth('rh',y3(1),x3(1),y3(2),x3(2));    % calculate
the azimuth of runway3
            angle23 = abs(angle2-angle3);                      % calculate
the angle between two RWYs
            angle13 = abs(angle1-angle3);                      % calculate
the angle between two RWYs

            if (((x1(1)>= x13) && (x13 >= x1(2))) || ((x1(1) <= x13) &&
(x13 <= x1(2)))) || ...
                    (((x2(1)>= x23) && (x23 >= x2(2))) || ((x2(1) <= x23)
&& (x23 <= x2(2))))
                if (((x3(1)>= x13) && (x13 >= x3(2))) || ((x3(1) <= x13)
&& (x13 <= x3(2)))) && ...
                        (((x3(1)>= x23) && (x23 >= x3(2))) || ((x3(1) <=
x23) && (x23 <= x3(2))))% see if the intersecting point lies on both line
segments
                    Three_Intersect_No = Three_Intersect_No + 1;
% three intersecting runways
                    Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
7 {shortestL2L3} 'N/A' Data_of_Three_RWY(rN,20) Data_of_Three_RWY(rN,21)
Data_of_Three_RWY(rN,22) Data_of_Three_RWY(rN,23)];
                elseif (((x3(1) > x3(2)) && (x3(2) > x13)) || ((x3(2) >
x3(1)) && (x3(1) > x13))) && ...
                        (((x3(1) > x3(2)) && (x3(2) > x23)) || ((x3(2) >
x3(1)) && (x3(1) > x23))) &&...
                        shortestL1L3 < shortestL2L3
                    Three_InterBeyonThres_No = Three_InterBeyonThres_No +
1;    % this runway pair is intersect-beyond-threshold runways
                    Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
13 {shortestL1L3} {angle13} Data_of_Three_RWY(rN,20)
Data_of_Three_RWY(rN,21) Data_of_Three_RWY(rN,22)
Data_of_Three_RWY(rN,23)];
                elseif (((x3(1) > x3(2)) && (x3(2) > x13)) || ((x3(2) >
x3(1)) && (x3(1) > x13))) && ...
                        (((x3(1) > x3(2)) && (x3(2) > x23)) || ((x3(2) >
x3(1)) && (x3(1) > x23))) &&...
                        shortestL1L3 > shortestL2L3
                    Three_InterBeyonThres_No = Three_InterBeyonThres_No +
1;    % this runway pair is intersect-beyond-threshold runways
                    Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
13 {shortestL2L3} {angle23} Data_of_Three_RWY(rN,20)
Data_of_Three_RWY(rN,21) Data_of_Three_RWY(rN,22)
Data_of_Three_RWY(rN,23)];
                else
                    Three_Non_ACM = Three_Non_ACM + 1;
                    Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
                end %if shortestL1L3 < shortestL2L3
```

```
            elseif (((x1(1) > x1(2)) && (x1(2) > x13)) || ((x1(2) > x1(1))
&& (x1(1) > x13))) &&...
                    (((x2(1) > x2(2)) && (x2(2) > x23)) || ((x2(2) >
x2(1)) && (x2(1) > x23))) &&...
                    (((x3(1) > x3(2)) && (x3(2) > x13)) || ((x3(2) >
x3(1)) && (x3(1) > x13))) &&...
                    (((x3(1) > x3(2)) && (x3(2) > x23)) || ((x3(2) >
x3(1)) && (x3(1) > x23)))
                 Three_OpenV_No = Three_OpenV_No + 1;
                 if shortestL1L3 < shortestL2L3
                     Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
10 {shortestL1L3} {angle13} Data_of_Three_RWY(rN,20)
Data_of_Three_RWY(rN,21) Data_of_Three_RWY(rN,22)
Data_of_Three_RWY(rN,23)];
                 else
                     Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
10 {shortestL2L3} {angle23} Data_of_Three_RWY(rN,20)
Data_of_Three_RWY(rN,21) Data_of_Three_RWY(rN,22)
Data_of_Three_RWY(rN,23)];
                 end %if shortestL1L3 < shortestL2L3
            else
                Three_Non_ACM = Three_Non_ACM + 1;
                Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
            end
        end
% end of set 1

% set 2
    elseif (shortestL2L3 < 2500) && (abs(ID2-ID3) <= 0)
        if (abs(ID3-ID1) <= 0) && (shortestL1L2 >= 2500)
            Three_Parallel_No = Three_Parallel_No + 1;    % if so, then
these runways will be parallel RWY
            Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:) 3
{shortestL1L3} 'N/A' Data_of_Three_RWY(rN,20) Data_of_Three_RWY(rN,21)
Data_of_Three_RWY(rN,22) Data_of_Three_RWY(rN,23)];
        else
            delta12  = det([(y1(1)-y1(2)) (x1(2)-x1(1)); (y2(1)-y2(2))
(x2(2)-x2(1))]);
            delta12X = det([(y1(2)*x1(1)-y1(1)*x1(2)) -(x1(2)-x1(1));
(y2(2)*x2(1)-y2(1)*x2(2)) -(x2(2)-x2(1))]);
            delta12Y = det([(y1(1)-y1(2)) -(y1(2)*x1(1)-y1(1)*x1(2));
(y2(1)-y2(2)) -(y2(2)*x2(1)-y2(1)*x2(2))]);
            x12 = delta12X/ delta12;      % X coordinate for intersecting
point
            y12 = delta12Y/ delta12;      % Y coordinate for intersecting
point

            delta13  = det([(y1(1)-y1(2)) (x1(2)-x1(1)); (y3(1)-y3(2))
(x3(2)-x3(1))]);
            delta13X = det([(y1(2)*x1(1)-y1(1)*x1(2)) -(x1(2)-x1(1));
(y3(2)*x3(1)-y3(1)*x3(2)) -(x3(2)-x3(1))]);
            delta13Y = det([(y1(1)-y1(2)) -(y1(2)*x1(1)-y1(1)*x1(2));
(y3(1)-y3(2)) -(y3(2)*x3(1)-y3(1)*x3(2))]);
            x13 = delta13X/ delta13;      % X coordinate for intersecting
point
```

```
            y13 = delta13Y/ delta13;      % Y coordinate for intersecting
point

            angle1 = azimuth('rh',y1(1),x1(1),y1(2),x1(2));    % calculate
the azimuth of runway2
            angle2 = azimuth('rh',y2(1),x2(1),y2(2),x2(2));    % calculate
the azimuth of runway2
            angle3 = azimuth('rh',y3(1),x3(1),y3(2),x3(2));    % calculate
the azimuth of runway3
            angle12 = abs(angle1-angle2);                      % calculate
the angle between two RWYs
            angle13 = abs(angle1-angle3);                      % calculate
the angle between two RWYs

            if (((x2(1)>= x12) && (x12 >= x2(2))) || ((x2(1) <= x12) &&
(x12 <= x2(2)))) || ...
                     (((x3(1)>= x13) && (x13 >= x3(2))) || ((x3(1) <= x13)
&& (x13 <= x3(2))))
                if (((x1(1)>= x12) && (x12 >= x1(2))) || ((x1(1) <= x12)
&& (x12 <= x1(2)))) && ...
                        (((x1(1)>= x13) && (x13 >= x1(2))) || ((x1(1) <=
x13) && (x13 <= x1(2)))) % see if the intersecting point lies on both line
segments
                    Three_Intersect_No = Three_Intersect_No + 1;
% three intersecting runways
                    Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
7 {shortestL1L3} 'N/A' Data_of_Three_RWY(rN,20) Data_of_Three_RWY(rN,21)
Data_of_Three_RWY(rN,22) Data_of_Three_RWY(rN,23)];
                elseif (((x1(1) > x1(2)) && (x1(2) > x12)) || ((x1(2) >
x1(1)) && (x1(1) > x12))) && ...
                        (((x1(1) > x1(2)) && (x1(2) > x13)) || ((x1(2) >
x1(1)) && (x1(1) > x13))) &&...
                        shortestL1L2 < shortestL1L3
                    Three_InterBeyonThres_No = Three_InterBeyonThres_No +
1;    % this runway pair is intersect-beyond-threshold runways
                    Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
13 {shortestL1L2} {angle12} Data_of_Three_RWY(rN,20)
Data_of_Three_RWY(rN,21) Data_of_Three_RWY(rN,22)
Data_of_Three_RWY(rN,23)];
                elseif (((x1(1) > x1(2)) && (x1(2) > x12)) || ((x1(2) >
x1(1)) && (x1(1) > x12))) && ...
                        (((x1(1) > x1(2)) && (x1(2) > x13)) || ((x1(2) >
x1(1)) && (x1(1) > x13))) &&...
                        shortestL1L3 < shortestL1L2
                    Three_InterBeyonThres_No = Three_InterBeyonThres_No +
1;    % this runway pair is intersect-beyond-threshold runways
                    Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
13 {shortestL1L3} {angle13} Data_of_Three_RWY(rN,20)
Data_of_Three_RWY(rN,21) Data_of_Three_RWY(rN,22)
Data_of_Three_RWY(rN,23)];
                else
                    Three_Non_ACM = Three_Non_ACM + 1;
                    Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
                end %if shortestL1L3 < shortestL2L3
```

```
            elseif (((x2(1) > x2(2)) && (x2(2) > x12)) || ((x2(2) > x2(1))
&& (x2(1) > x12))) &&...
                     (((x3(1) > x3(2)) && (x3(2) > x13)) || ((x3(2) >
x3(1)) && (x3(1) > x13))) &&...
                     (((x1(1) > x1(2)) && (x1(2) > x12)) || ((x1(2) >
x1(1)) && (x1(1) > x12))) &&...
                     (((x1(1) > x1(2)) && (x1(2) > x13)) || ((x1(2) >
x1(1)) && (x1(1) > x13)))
                 Three_OpenV_No = Three_OpenV_No + 1;
                 if shortestL1L2 < shortestL1L3
                     Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
10 {shortestL1L2} {angle12} Data_of_Three_RWY(rN,20)
Data_of_Three_RWY(rN,21) Data_of_Three_RWY(rN,22)
Data_of_Three_RWY(rN,23)];
                 else
                     Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
10 {shortestL1L3} {angle13} Data_of_Three_RWY(rN,20)
Data_of_Three_RWY(rN,21) Data_of_Three_RWY(rN,22)
Data_of_Three_RWY(rN,23)];
                 end %if shortestL1L3 < shortestL2L3
            else
                Three_Non_ACM = Three_Non_ACM + 1;
                Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:)
'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
            end
        end
% end of set 2

% set 3
    elseif (shortestL1L2 >= 2500) && (shortestL2L3 >= 2500) && (abs(ID1-
ID2) <= 1) && (abs(ID2-ID3) <= 1)
        Three_Parallel_No = Three_Parallel_No + 1;    % if so, then these
runways will be parallel RWY
        Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:) 3
{shortestL1L3} 'N/A' Data_of_Three_RWY(rN,20) Data_of_Three_RWY(rN,21)
Data_of_Three_RWY(rN,22) Data_of_Three_RWY(rN,23)];

    elseif ((abs(ID1-ID2) <= 0) && (abs(ID2-ID3) > 1)) || ((abs(ID2-ID3)
<= 0) && (abs(ID1-ID2) > 1))
        delta23  = det([(y2(1)-y2(2)) (x2(2)-x2(1)); (y3(1)-y3(2)) (x3(2)-
x3(1))]);
        delta23X = det([(y2(2)*x2(1)-y2(1)*x2(2)) -(x2(2)-x2(1));
(y3(2)*x3(1)-y3(1)*x3(2)) -(x3(2)-x3(1))]);
        delta23Y = det([(y2(1)-y2(2)) -(y2(2)*x2(1)-y2(1)*x2(2)); (y3(1)-
y3(2)) -(y3(2)*x3(1)-y3(1)*x3(2))]);
        x23 = delta23X/ delta23;      % X coordinate for intersecting
point
        y23 = delta23Y/ delta23;      % Y coordinate for intersecting
point

        delta13  = det([(y1(1)-y1(2)) (x1(2)-x1(1)); (y3(1)-y3(2)) (x3(2)-
x3(1))]);
        delta13X = det([(y1(2)*x1(1)-y1(1)*x1(2)) -(x1(2)-x1(1));
(y3(2)*x3(1)-y3(1)*x3(2)) -(x3(2)-x3(1))]);
        delta13Y = det([(y1(1)-y1(2)) -(y1(2)*x1(1)-y1(1)*x1(2)); (y3(1)-
y3(2)) -(y3(2)*x3(1)-y3(1)*x3(2))]);
```

```
        x13 = delta23X/ delta23;      % X coordinate for intersecting
point
        y13 = delta23Y/ delta23;      % Y coordinate for intersecting
point

        if (((x3(1)>= x13) && (x13 >= x3(2))) || ((x3(1) <= x13) && (x13
<= x3(2)))) &&...
                (((x3(1)>= x23) && (x23 >= x3(2))) || ((x3(1) <= x23) &&
(x23 <= x3(2)))) &&...
                (((x1(1)>= x13) && (x13 >= x1(2))) || ((x1(1) <= x13) &&
(x13 <= x1(2)))) &&...
                (((x2(1)>= x23) && (x23 >= x2(2))) || ((x2(1) <= x23) &&
(x23 <= x2(2)))) % see if the intersecting point lies on both line
segments
            Three_Intersect_No = Three_Intersect_No + 1;               %
three intersecting runways
            Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:) 7
{shortestL2L3} 'N/A' Data_of_Three_RWY(rN,20) Data_of_Three_RWY(rN,21)
Data_of_Three_RWY(rN,22) Data_of_Three_RWY(rN,23)];
        elseif (((x1(1)>= x13) && (x13 >= x1(2))) || ((x1(1) <= x13) &&
(x13 <= x1(2)))) &&...
                (((x1(1)>= x23) && (x23 >= x1(2))) || ((x1(1) <= x23) &&
(x23 <= x1(2)))) &&...
                (((x3(1)>= x13) && (x13 >= x3(2))) || ((x3(1) <= x13) &&
(x13 <= x3(2)))) &&...
                (((x2(1)>= x12) && (x12 >= x2(2))) || ((x2(1) <= x12) &&
(x12 <= x2(2)))) % see if the intersecting point lies on both line
segments
            Three_Intersect_No = Three_Intersect_No + 1;               %
three intersecting runways
            Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:) 7
{shortestL2L3} 'N/A' Data_of_Three_RWY(rN,20) Data_of_Three_RWY(rN,21)
Data_of_Three_RWY(rN,22) Data_of_Three_RWY(rN,23)];
        else
            Three_Non_ACM = Three_Non_ACM + 1;
            Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:) 'N/A'
'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
        end



    else
        Three_Non_ACM = Three_Non_ACM + 1;
        Config_Three(NumApThree,:) = [List_of_Three_RWY(rN,:) 'N/A' 'N/A'
'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
    end
end

save ThreeProfiles Three_Parallel_No Three_Intersect_No Three_OpenV_No
Three_InterBeyonThres_No Three_Non_ACM

%%   Four runways
% % Now we deal with all four runways data.
% % Further distributing all information in "Data_of_four_RWY" and
% % "List_of_four_RWY" into sub-type files
% % There are four sub-types for all four-runway cases:
```

```
Four_Parallel_No        = 0;     % number of airports have four-parallel-
runway;                         model 4 in ACM
Four_Intersect_No       = 0;     % number of airports have four-intersect-
runway;                        model 15 in ACM
Four_OpenV_No           = 0;     % number of airports have four-openV-
runway;                           model 11 in ACM
Four_InterBeyonThres_No = 0;     % number of airports have four-
intersecting-beyond-threshold-runway;   model 14 in ACM
Four_Non_ACM            = 0;

% Initialization
NumApFour               = 0;     % number of airports belong to four-
runway case
numEndFour = size(Data_of_Four_RWY,1);   % how many rows in file
"Data_of_Four_RWY", in other word, how many
                                 % ends for all four runway cases

for rN = 1:8:(numEndFour-7)      % row Number
    NumApFour = NumApFour + 1;   % count number of four-runway airports
    ID1 = sscanf(List_of_Four_RWY{rN,17}  ,'%f',2);
    ID2 = sscanf(List_of_Four_RWY{rN+2,17},'%f',2);
    ID3 = sscanf(List_of_Four_RWY{rN+4,17},'%f',2);
    ID4 = sscanf(List_of_Four_RWY{rN+6,17},'%f',2);

    if (abs(ID1-ID2) <= 1) && (abs(ID3-ID4) <= 1)
        % coordinates of end points on both line segments
        x1 = [Data_of_Four_RWY(rN,25)   Data_of_Four_RWY(rN+1,25)];
% x coordinates(two ends) of line 1
        x2 = [Data_of_Four_RWY(rN+2,25) Data_of_Four_RWY(rN+3,25)];
% x coordinates(two ends) of line 2
        x3 = [Data_of_Four_RWY(rN+4,25) Data_of_Four_RWY(rN+5,25)];
% x coordinates(two ends) of line 3
        x4 = [Data_of_Four_RWY(rN+6,25) Data_of_Four_RWY(rN+7,25)];
% x coordinates(two ends) of line 4
        y1 = [Data_of_Four_RWY(rN,24)   Data_of_Four_RWY(rN+1,24)];
% y coordinates(two ends) of line 1
        y2 = [Data_of_Four_RWY(rN+2,24) Data_of_Four_RWY(rN+3,24)];
% y coordinates(two ends) of line 2
        y3 = [Data_of_Four_RWY(rN+4,24) Data_of_Four_RWY(rN+5,24)];
% y coordinates(two ends) of line 3
        y4 = [Data_of_Four_RWY(rN+6,24) Data_of_Four_RWY(rN+7,24)];
% y coordinates(two ends) of line 4

        dist121 = distdim(distance(y1(1),x1(1),y2(1),x2(1)),'deg','ft');
% calculate distance between line 1 and line 2
        dist122 = distdim(distance(y1(1),x1(1),y2(2),x2(2)),'deg','ft');
% calculate distance between line 1 and line 2
        dist123 = distdim(distance(y1(2),x1(2),y2(1),x2(1)),'deg','ft');
% calculate distance between line 1 and line 2
        dist124 = distdim(distance(y1(2),x1(2),y2(2),x2(2)),'deg','ft');
% calculate distance between line 1 and line 2
        dist341 = distdim(distance(y3(1),x3(1),y4(1),x4(1)),'deg','ft');
% calculate distance between line 1 and line 3
        dist342 = distdim(distance(y3(1),x3(1),y4(2),x4(2)),'deg','ft');
% calculate distance between line 1 and line 3
```

```
        dist343 = distdim(distance(y3(2),x3(2),y4(1),x4(1)),'deg','ft');
% calculate distance between line 1 and line 3
        dist344 = distdim(distance(y3(2),x3(2),y4(2),x4(2)),'deg','ft');
% calculate distance between line 1 and line 3

        dist231 = distdim(distance(y2(1),x2(1),y3(1),x3(1)),'deg','ft');
% calculate distance between line 2 and line 3
        dist232 = distdim(distance(y2(1),x2(1),y3(2),x3(2)),'deg','ft');
% calculate distance between line 2 and line 3
        dist233 = distdim(distance(y2(2),x2(2),y3(1),x3(1)),'deg','ft');
% calculate distance between line 2 and line 3
        dist234 = distdim(distance(y2(2),x2(2),y3(2),x3(2)),'deg','ft');
% calculate distance between line 2 and line 3
        dist131 = distdim(distance(y1(1),x1(1),y3(1),x3(1)),'deg','ft');
% calculate distance between line 1 and line 3
        dist132 = distdim(distance(y1(1),x1(1),y3(2),x3(2)),'deg','ft');
% calculate distance between line 1 and line 3
        dist133 = distdim(distance(y1(2),x1(2),y3(1),x3(1)),'deg','ft');
% calculate distance between line 1 and line 3
        dist134 = distdim(distance(y1(2),x1(2),y3(2),x3(2)),'deg','ft');
% calculate distance between line 1 and line 3

        dist241 = distdim(distance(y2(1),x2(1),y4(1),x4(1)),'deg','ft');
% calculate distance between line 2 and line 4
        dist242 = distdim(distance(y2(1),x2(1),y4(2),x4(2)),'deg','ft');
% calculate distance between line 2 and line 4
        dist243 = distdim(distance(y2(2),x2(2),y4(1),x4(1)),'deg','ft');
% calculate distance between line 2 and line 4
        dist244 = distdim(distance(y2(2),x2(2),y4(2),x4(2)),'deg','ft');
% calculate distance between line 2 and line 4
        dist141 = distdim(distance(y1(1),x1(1),y4(1),x4(1)),'deg','ft');
% calculate distance between line 1 and line 4
        dist142 = distdim(distance(y1(1),x1(1),y4(2),x4(2)),'deg','ft');
% calculate distance between line 1 and line 4
        dist143 = distdim(distance(y1(2),x1(2),y4(1),x4(1)),'deg','ft');
% calculate distance between line 1 and line 4
        dist144 = distdim(distance(y1(2),x1(2),y4(2),x4(2)),'deg','ft');
% calculate distance between line 1 and line 4

        shortestL1L2 = min([dist121,dist122,dist123,dist124]);
% the shortest distance between two line segments
        shortestL3L4 = min([dist341,dist342,dist343,dist344]);
% the shortest distance between two line segments

        shortestL2L3 = min([dist231,dist232,dist233,dist234]);
% the shortest distance between two line segments
        shortestL1L3 = min([dist131,dist132,dist133,dist134]);
% the shortest distance between two line segments
        shortestL2L4 = min([dist241,dist242,dist243,dist244]);
% the shortest distance between two line segments
        shortestL1L4 = min([dist141,dist142,dist143,dist144]);
% the shortest distance between two line segments
        shortestL1L2L3L4 = min([shortestL2L3 , shortestL1L3 , shortestL2L4
, shortestL1L4]);

% set 1
```

```
        if (shortestL1L2 < 2500) && (shortestL3L4 < 2500)
            if abs(ID3-ID2) <= 1
                Four_No = Four_No + 1;    % if so, then these runways will
be parallel RWY
                Config_Four(NumApFour,:) = [List_of_Four_RWY(rN,:) 4
{shortestL2L3} 'N/A' Data_of_Four_RWY(rN,20) Data_of_Four_RWY(rN,21)
Data_of_Four_RWY(rN,22) Data_of_Four_RWY(rN,23)];
            else
                delta23  = det([(y2(1)-y2(2)) (x2(2)-x2(1)); (y3(1)-y3(2))
(x3(2)-x3(1))]);
                delta23X = det([(y2(2)*x2(1)-y2(1)*x2(2)) -(x2(2)-x2(1));
(y3(2)*x3(1)-y3(1)*x3(2)) -(x3(2)-x3(1))]);
                delta23Y = det([(y2(1)-y2(2)) -(y2(2)*x2(1)-y2(1)*x2(2));
(y3(1)-y3(2)) -(y3(2)*x3(1)-y3(1)*x3(2))]);
                x23 = delta23X/ delta23;      % X coordinate for
intersecting point
                y23 = delta23Y/ delta23;      % Y coordinate for
intersecting point

                delta13  = det([(y1(1)-y1(2)) (x1(2)-x1(1)); (y3(1)-y3(2))
(x3(2)-x3(1))]);
                delta13X = det([(y1(2)*x1(1)-y1(1)*x1(2)) -(x1(2)-x1(1));
(y3(2)*x3(1)-y3(1)*x3(2)) -(x3(2)-x3(1))]);
                delta13Y = det([(y1(1)-y1(2)) -(y1(2)*x1(1)-y1(1)*x1(2));
(y3(1)-y3(2)) -(y3(2)*x3(1)-y3(1)*x3(2))]);
                x13 = delta13X/ delta13;      % X coordinate for
intersecting point
                y13 = delta13Y/ delta13;      % Y coordinate for
intersecting point

                delta24  = det([(y2(1)-y2(2)) (x2(2)-x2(1)); (y4(1)-y4(2))
(x4(2)-x4(1))]);
                delta24X = det([(y2(2)*x2(1)-y2(1)*x2(2)) -(x2(2)-x2(1));
(y4(2)*x4(1)-y4(1)*x4(2)) -(x4(2)-x4(1))]);
                delta24Y = det([(y2(1)-y2(2)) -(y2(2)*x2(1)-y2(1)*x2(2));
(y4(1)-y4(2)) -(y4(2)*x4(1)-y4(1)*x4(2))]);
                x24 = delta24X/ delta24;      % X coordinate for
intersecting point
                y24 = delta24Y/ delta24;      % Y coordinate for
intersecting point

                delta14  = det([(y1(1)-y1(2)) (x1(2)-x1(1)); (y4(1)-y4(2))
(x4(2)-x4(1))]);
                delta14X = det([(y1(2)*x1(1)-y1(1)*x1(2)) -(x1(2)-x1(1));
(y4(2)*x4(1)-y4(1)*x4(2)) -(x4(2)-x4(1))]);
                delta14Y = det([(y1(1)-y1(2)) -(y1(2)*x1(1)-y1(1)*x1(2));
(y4(1)-y4(2)) -(y4(2)*x4(1)-y4(1)*x4(2))]);
                x14 = delta14X/ delta14;      % X coordinate for
intersecting point
                y14 = delta14Y/ delta14;      % Y coordinate for
intersecting point

                angle2 = azimuth('rh',y2(1),x2(1),y2(2),x2(2));    %
calculate the azimuth of runway2
                angle3 = azimuth('rh',y3(1),x3(1),y3(2),x3(2));    %
calculate the azimuth of runway3
```

```
                angle23 = abs(angle2-angle3);                     %
calculate the angle between two RWYs
                if ((((x1(1)>= x13) && (x13 >= x1(2))) || ((x1(1) <= x13)
&& (x13 <= x1(2)))) && ...
                        (((x1(1)>= x14) && (x14 >= x1(2))) || ((x1(1) <=
x14) && (x14 <= x1(2))))) && ...
                        ((((x2(1)>= x23) && (x23 >= x2(2))) || ((x2(1) <=
x23) && (x23 <= x2(2)))) &&...
                        (((x2(1)>= x24) && (x24 >= x2(2))) || ((x2(1) <=
x24) && (x24 <= x2(2)))))
                    if (((x3(1)>= x13) && (x13 >= x3(2))) || ((x3(1) <=
x13) && (x13 <= x3(2)))) && ...
                            (((x3(1)>= x23) && (x23 >= x3(2))) || ((x3(1)
<= x23) && (x23 <= x3(2)))) &&...
                            (((x4(1)>= x14) && (x14 >= x4(2))) || ((x4(1)
<= x14) && (x14 <= x4(2)))) && ...
                            (((x4(1)>= x24) && (x24 >= x4(2))) || ((x4(1)
<= x24) && (x24 <= x4(2)))) % see if the intersecting point lies on both
line segments
                        Four_Intersect_No = Four_Intersect_No + 1;
% four intersecting runways
                        Config_Four(NumApFour,:) = [List_of_Four_RWY(rN,:)
15 'N/A' 'N/A' Data_of_Four_RWY(rN,20) Data_of_Four_RWY(rN,21)
Data_of_Four_RWY(rN,22) Data_of_Four_RWY(rN,23)];
                    elseif (((x3(1) > x3(2)) && (x3(2) > x13)) || ((x3(2)
> x3(1)) && (x3(1) > x13))) && ...
                            (((x3(1) > x3(2)) && (x3(2) > x23)) || ((x3(2)
> x3(1)) && (x3(1) > x23))) &&...
                            (((x4(1) > x4(2)) && (x4(2) > x14)) || ((x4(2)
> x4(1)) && (x4(1) > x14))) && ...
                            (((x4(1) > x4(2)) && (x4(2) > x24)) || ((x4(2)
> x4(1)) && (x4(1) > x24))) % see if the intersecting point lies on both
line segments
                            Four_InterBeyonThres_No =
Four_InterBeyonThres_No + 1;    % this runway pair is intersect-beyond-
threshold runways
                            Config_Four(NumApThree,:) =
[List_of_Four_RWY(rN,:) 14 {shortestL1L2L3L4} {angle23}
Data_of_Four_RWY(rN,20) Data_of_Four_RWY(rN,21) Data_of_Four_RWY(rN,22)
Data_of_Four_RWY(rN,23)];
                    else
                        Four_Non_ACM = Four_Non_ACM + 1;
                        Config_Four(NumApFour,:) = [List_of_Four_RWY(rN,:)
'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
                    end %if shortestL1L3 < shortestL2L3
                elseif (((x1(1) > x1(2)) && (x1(2) > x13)) || ((x1(2) >
x1(1)) && (x1(1) > x13))) &&...
                        (((x3(1) > x3(2)) && (x3(2) > x13)) || ((x3(2) >
x3(1)) && (x3(1) > x13))) &&...
                        (((x2(1) > x2(2)) && (x2(2) > x23)) || ((x2(2) >
x2(1)) && (x2(1) > x23))) &&...
                        (((x3(1) > x3(2)) && (x3(2) > x23)) || ((x3(2) >
x3(1)) && (x3(1) > x23))) &&...
                        (((x1(1) > x1(2)) && (x1(2) > x14)) || ((x1(2) >
x1(1)) && (x1(1) > x14))) &&...
```

```
                        (((x4(1) > x4(2)) && (x4(2) > x14)) || ((x4(2) >
x4(1)) && (x4(1) > x14))) &&...
                        (((x2(1) > x2(2)) && (x2(2) > x24)) || ((x2(2) >
x2(1)) && (x2(1) > x24))) &&...
                        (((x4(1) > x4(2)) && (x4(2) > x24)) || ((x4(2) >
x4(1)) && (x4(1) > x24)))
                    Four_OpenV_No = Four_OpenV_No + 1;
                    Config_Four(NumApFour,:) = [List_of_Four_RWY(rN,:) 11
{shortestL1L2L3L4} {angle23} Data_of_Four_RWY(rN,20)
Data_of_Four_RWY(rN,21) Data_of_Four_RWY(rN,22) Data_of_Four_RWY(rN,23)];
                 else
                    Four_Non_ACM = Four_Non_ACM + 1;
                    Config_Four(NumApFour,:) = [List_of_Four_RWY(rN,:)
'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
                 end %if ((((x1(1)>= x13) && (x13 >= x1(2))) || ((x1(1) <=
x13) && (x13 <= x1(2)))) && ...
            end %if abs(ID3-ID2) <= 1
    % end of set 1

    % set 2
        else
            Four_Non_ACM = Four_Non_ACM + 1;
            Config_Four(NumApFour,:) = [List_of_Four_RWY(rN,:) 'N/A' 'N/A'
'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
        end
    else
        Four_Non_ACM = Four_Non_ACM + 1;
        Config_Four(NumApFour,:) = [List_of_Four_RWY(rN,:) 'N/A' 'N/A'
'N/A' 'N/A' 'N/A' 'N/A' 'N/A'];
    end
end

save FourProfiles Four_Parallel_No Four_Intersect_No Four_OpenV_No
Four_InterBeyonThres_No Four_Non_ACM

save Config Config_Two Config_Three Config_Four
save NumAp NumApTwo NumApThree NumApFour
```

APPENDIX E-3 MATLAB PROGRAM FILE FOR MULTI-RUNWAY AIRPORT-PARSER

```
% ****************************************************************
% Load characteristics of two-runway, three-runway, and four-runway
airports and create txt file
% for each airport to serve as input file of ACM
%
% Coded by Yueh-Ting Chen, May 1,2006
% Last Modifaction: Nov 17,2006
% ****************************************************************

%% Load airport characteristic data

clc
% load NumAp
% load runway configuration data for all airports
% load Config
% load number of airports data
input.alternatingArrivlas   = 0;
% alternating arrivals are to be conducted if RWY are 2500 feet apart.
weather                     = ['VMC';'MMC';'IMC'];
% three weather scenarios: VMC, MMC, IMC

[runwayData,runwayList]     =
xlsread('D:\YuehTing\Research_Project\Data\VLJ-Multiple Runways.xls');
% runway data for all airports
[ArrArr_Data, ArrArr_Text]  =
xlsread('D:\YuehTing\Research_Project\Data\ArrArr_Matrix.xls');
% separation data for all scenarios
[DepDep_Data, DepDep_Text]  =
xlsread('D:\YuehTing\Research_Project\Data\DepDep_Matrix.xls');
% separation data for all scenarios

fid = fopen('outputFileNames.dat', 'a+');
% open a file named "outputFileNames.dat" to save all files generated for
ACM

%% For Two Runways Airport

for wi = 1:3;               % Run the program for three weather
conditions; VMC, MMC, IMC.
    for i = 1 : NumApTwo    % Run the program for all two-runway airports.

        % Assign ACM model number and strategy number to each airport by
following procedure.
        if  (~strcmp(Config_Two{i,20},'Y')) && (wi ~=1);    % For non-
towered airport under MMC/IMC condition,
            input.modelNumber = 1;                          % Treat this
multi-runway airport as single runway airport, capacity calculation-wised.
            input.operationalStrategy = 3;
        else                                                % Otherwise,
treat this airport by its real configuration.
```

```
            input.modelNumber = Config_Two{i,27};           % Model number
in ACM: parallel(2) / intersecting(6) / open-v(5) /
inter.beyond.threshold(12);
            CenterlineSeparation = Config_Two{i,28};        % (feet)
            switch input.modelNumber
                case 2                                      % parallel
runways
                    if CenterlineSeparation >=4300          % judging the
appropriate operational strategies
                        input.operationalStrategy = [2 4 5 6];          %
Far
                    elseif (4300 > CenterlineSeparation) &&
(CenterlineSeparation >=3500)
                        input.operationalStrategy = [8 10 11 12];       %
Medium
                    elseif (3500 > CenterlineSeparation) &&
(CenterlineSeparation >=2500)
                        input.operationalStrategy = [14 16 17 18];      %
Near
                    elseif (2500 > CenterlineSeparation) &&
(CenterlineSeparation >= 700)
                        input.operationalStrategy = [20 22 23 24];      %
Close
                    else                                                %
for runway separation < 700
                        input.modelNumber = 1;
                        input.operationalStrategy = 3;              %
Should be treated as single runway
                        disp(['Runway centerline separation at '
Config_Two{i,1} ' is less than 700 feet']);
                    end %28% CenterlineSeparation >=4300
                case 5                                                  %
open-v
                    input.operationalStrategy = [2 3 4 5];
                case 6                                                  %
intersecting
                    input.operationalStrategy = [2 3];
                otherwise                                               %
model 12, intersecting beyond threshold
                    input.operationalStrategy = [1 2 3 4];
            end  %26% input.modelNumber == 2;
        end

        % Assign filename to each ACM input text file.
        % The file name is in the following format:
        % AirportID_WeatherCondition_ModelNumber_StrategyNumber.inp
        for OS = 1:size(input.operationalStrategy,2);    % OS, Operational
Strategy.
            filename =
strcat(Config_Two{i,1},'_',weather(wi,:),'_',num2str(input.modelNumber),..
.
                '_',num2str(input.operationalStrategy(OS)));
            fullname = strcat(filename,'.inp');
            fp1 = fopen(fullname,'w');

            % start to build ACM input file
```

```
            fprintf(fp1,'%-
6s%2d%2d%2d%3s\n',Config_Two{i,1},0,0,0,weather(wi,:));

fprintf(fp1,'%4d%4d%4d\n',input.modelNumber,input.operationalStrategy(OS),
input.alternatingArrivlas);

            fprintf(fp1,'%-6s%2d%2d%2d\n','RWY1',1,1,0);        % aircraft
mix on runway 1 (small to heavy)

fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',Config_Two{i,30},Config_Two{i,31},Con
fig_Two{i,32},Config_Two{i,33});
            if input.modelNumber ~= 1;
                fprintf(fp1,'%-6s%2d%2d%2d\n','RWY2',2,1,0);    % aircraft
mix on runway 2 (small to heavy)

fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',Config_Two{i,30},Config_Two{i,31},Con
fig_Two{i,32},Config_Two{i,33});
            end %64% if input.modelNumber ~= 1;

            fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR1',1,2,0);      % average
arr. ROT for all aircraft type at runway1
            fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);
            if input.modelNumber ~= 1;
                fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR2',2,2,0);  % average
arr. ROT for all aircraft type at runway2
                fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);
            end %70% if input.modelNumber ~= 1;

            fprintf(fp1,'%-6s%2d%2d%2d\n','DLTAIJ',0,4,0);      % minimum
arrival separations, in nm, for i followed by j, from smallest to largest

            if  strcmp(Config_Two{i,20},'Y');                   % see if
this aiport has tower on site
                switch wi                                       %
separation for aiport with tower
                    case 1

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(2,2),ArrArr_Data(2,3),ArrArr_Data(2,4),ArrArr_Data(2,5),...

ArrArr_Data(3,2),ArrArr_Data(3,3),ArrArr_Data(3,4),ArrArr_Data(3,5),...

ArrArr_Data(4,2),ArrArr_Data(4,3),ArrArr_Data(4,4),ArrArr_Data(4,5),...

ArrArr_Data(5,2),ArrArr_Data(5,3),ArrArr_Data(5,4),ArrArr_Data(5,5));
                    otherwise

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(9,2),ArrArr_Data(9,3),ArrArr_Data(9,4),ArrArr_Data(9,5),...

ArrArr_Data(10,2),ArrArr_Data(10,3),ArrArr_Data(10,4),ArrArr_Data(10,5),..
.
```

```
ArrArr_Data(11,2),ArrArr_Data(11,3),ArrArr_Data(11,4),ArrArr_Data(11,5),..
.

ArrArr_Data(12,2),ArrArr_Data(12,3),ArrArr_Data(12,4),ArrArr_Data(12,5));
                end
            else                                                    %
separation for aiport without tower
                switch wi
                    case 1

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(2,9),ArrArr_Data(2,10),ArrArr_Data(2,11),ArrArr_Data(2,12),...

ArrArr_Data(3,9),ArrArr_Data(3,10),ArrArr_Data(3,11),ArrArr_Data(3,12),...

ArrArr_Data(4,9),ArrArr_Data(4,10),ArrArr_Data(4,11),ArrArr_Data(4,12),...

ArrArr_Data(5,9),ArrArr_Data(5,10),ArrArr_Data(5,11),ArrArr_Data(5,12));
                    otherwise

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(9,9),ArrArr_Data(9,10),ArrArr_Data(9,11),ArrArr_Data(9,12),...

ArrArr_Data(10,9),ArrArr_Data(10,10),ArrArr_Data(10,11),ArrArr_Data(10,12)
,...

ArrArr_Data(11,9),ArrArr_Data(11,10),ArrArr_Data(11,11),ArrArr_Data(11,12)
,...

ArrArr_Data(12,9),ArrArr_Data(12,10),ArrArr_Data(12,11),ArrArr_Data(12,12)
);
                end
            end

            fprintf(fp1,'%-6s%2d%2d%2d\n','APPSPD',0,5,0);        %
approach speed, in knots
            fprintf(fp1,'%4d%4d%4d%4d\n',120,130,140,145);

            fprintf(fp1,'%-6s%2d%2d%2d\n','DRBAR',0,6,0);         %
departure runway occupancy time, in seconds
            fprintf(fp1,'%4d%4d%4d%4d\n',34,34,39,39);

            fprintf(fp1,'%-6s%2d%2d%2d\n','DDSR',0,7,0);          %
minimum departure separations, in seconds, for i followed by j

            if  strcmp(Config_Two{i,20},'Y');                      % see if
this aiport has tower on site
                switch wi                                           %
separation for aiport with tower
                    case 1
```

```
fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(2,2),DepDep_Data(2,3),DepDep_Data(2,4),DepDep_Data(2,5),...

DepDep_Data(3,2),DepDep_Data(3,3),DepDep_Data(3,4),DepDep_Data(3,5),...

DepDep_Data(4,2),DepDep_Data(4,3),DepDep_Data(4,4),DepDep_Data(4,5),...

DepDep_Data(5,2),DepDep_Data(5,3),DepDep_Data(5,4),DepDep_Data(5,5));
                    otherwise

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(9,2),DepDep_Data(9,3),DepDep_Data(9,4),DepDep_Data(9,5),...

DepDep_Data(10,2),DepDep_Data(10,3),DepDep_Data(10,4),DepDep_Data(10,5),..
.

DepDep_Data(11,2),DepDep_Data(11,3),DepDep_Data(11,4),DepDep_Data(11,5),..
.

DepDep_Data(12,2),DepDep_Data(12,3),DepDep_Data(12,4),DepDep_Data(12,5));
                end
            else                                            %
separation for aiport without tower
                switch wi
                    case 1

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(2,9),DepDep_Data(2,10),DepDep_Data(2,11),DepDep_Data(2,12),...

DepDep_Data(3,9),DepDep_Data(3,10),DepDep_Data(3,11),DepDep_Data(3,12),...

DepDep_Data(4,9),DepDep_Data(4,10),DepDep_Data(4,11),DepDep_Data(4,12),...

DepDep_Data(5,9),DepDep_Data(5,10),DepDep_Data(5,11),DepDep_Data(5,12));
                    otherwise

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(9,9),DepDep_Data(9,10),DepDep_Data(9,11),DepDep_Data(9,12),...

DepDep_Data(10,9),DepDep_Data(10,10),DepDep_Data(10,11),DepDep_Data(10,12)
,...

DepDep_Data(11,9),DepDep_Data(11,10),DepDep_Data(11,11),DepDep_Data(11,12)
,...

DepDep_Data(12,9),DepDep_Data(12,10),DepDep_Data(12,11),DepDep_Data(12,12)
);
                end
            end

            fprintf(fp1,'%-6s%2d%2d%2d\n','GAMA',0,8,0);       % length of
final approach path, in nm
```

```
            if (wi ~= 1) && (~strcmp(Config_Two{i,20},'Y'))
                fprintf(fp1,'%4d%4d%4d%4d\n',15,15,15,15);
            else fprintf(fp1,'%4d%4d%4d%4d\n',7,7,7,7);
            end

            fprintf(fp1,'%-6s%2d%2d%2d\n','TGRBAR',0,9,0);     % touch-
and-go runway occupancy time, in seconds
            fprintf(fp1,'%4d%4d%4d%4d\n',15,15,15,15);

            % if input.modelNumber == 5||12
            if (input.modelNumber == 5)||(input.modelNumber == 12)
                fprintf(fp1,'%-6s%2d%2d\n','OPENV',0,10);          % for
open-V/ intersecting beyond threshold

fprintf(fp1,'%4.0f%4.0f%4.1f%4.1f\n',Config_Two{i,29},CenterlineSeparation
,10,2.0);    % angle(degress); distance between threshold of runways
            end %89% if Config_Two{i,24} ~= 'N/A'

            %  elseif input.modelNumber == 6, intersecting
            fprintf(fp1,'%-6s%2d%2d%2d\n','TWOIN',0,11,0);     % airborne
intersection indicator; needed for 6-1,6-3

fprintf(fp1,'%4.0f%4.1f%4.1f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f\n',1,
10,10,10,10,10,10,10,10,10,10);

            fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',1,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);

            fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',2,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);

            fprintf(fp1,'%-6s%2d%2d%2d\n','DICBR',0,13,0);     % the
separation(nm) between departure I and arrival J on the crossing runway;
needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...
                3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3);

            fprintf(fp1,'%-6s%2d%2d%2d\n','SIGMAS',0,19,0);
            if  (~strcmp(Config_Two{i,20},'Y')) && (wi == 1);
                fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f\n',8,30,0,6,6);
            else
                fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f\n',8,18,0,6,6);
            end
```

```
            % sigma of arrival ROT         (seconds),
            % sigma of interarrival time   (seconds),
            % sigma of touch and go ROT    (seconds),
            % sigma of cleared-to-roll time(seconds),
            % sigma of dep. ROT            (seconds)

            switch wi
                case 1      % VMC
                    visi = 5.0;
                    ceil = 5000;
                case 2      % MMC
                    visi = 2.5;
                    ceil = 900;
                case 3      % IMC
                    visi = 0.0;
                    ceil = 0;
            end %114% switch wi

            fprintf(fp1,'%-6s%2d%2d%2d\n','OTHERS',0,20,0);
            if (~strcmp(Config_Two{i,20},'Y')) && (wi ~= 1);

fprintf(fp1,'%4.2f%4.2f%4.1f%4.1f%4.0f%4.1f%4.2f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f%4.0f\n',...

0.05,0.05,15,visi,ceil,3,0,7777,10,20,30,40,50,60,70,80,90);
            else

fprintf(fp1,'%4.2f%4.2f%4.1f%4.1f%4.0f%4.1f%4.2f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f%4.0f\n',...
                    0.05,0.05,
3,visi,ceil,3,0,7777,10,20,30,40,50,60,70,80,90);
            end
            % percentage of violation for arrival ROT,
            % percentage of violation for interarrival time,
            % dep/arr separation(nm),
            % visibility(statute miles),
            % ceiling(feet),
            % glide slope angle (degrees),
            % proportion of touch and go operations,
            % arrival percentages: 7777 (both arrival priority and
departure priority), 10%~90%

            fprintf(fp1,'%-6s%2d%2d%2d\n','BDD',0,22,0);        % buffer
time (seconds), between simultaneous dep. on close-spaced parallel runways

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                0,0,0,25,0,0,0,20,0,0,0,10,25,20,10,0);

            fprintf(fp1,'%-6s%2d%2d%2d\n','BAA',0,24,0);        % buffer
time (seconds), between simultaneous arr. on close-spaced parallel runways

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                0,0,0,25,0,0,0,20,0,0,0,10,25,20,10,0);
```

```
            fprintf(fp1,'%-6s%2d%2d%2d\n','ALTARR',0,25,1);
            fprintf(fp1,'%4.1f%6.0f%6.0f%4.1f\n',2,3000,0,0);
            % diagonal separation (nm); separation between runway
centerlines (feet);
            % thresholds displacement (feet); relative displacement of
gates for two runways (nm);

            fclose(fp1);

            dos(['ACMrun ',filename]);
            movefile(fullname,'d:\YuehTing\Research_Project\ACMinput\');

movefile(strcat(filename,'.res'),'d:\YuehTing\Research_Project\ACMoutput\'
);
            fprintf(fid,'%-
72s\n',strcat('d:\YuehTing\Research_Project\ACMoutput\',filename,'.res'));

        end %53% for OS = 1:size(input.operationalStrategy,2);
    end %27% for i = 1 : NumApTwo
end %26% for wi = 1:3;



%% For Three

for wi = 1:3;       % wi, weather index; VMC, MMC, IMC
    for i = 1 : NumApThree

        % Assign ACM model number and strategy number to each airport by
following procedure.
        input.modelNumber = Config_Three{i,27};         % parallel(3) /
intersecting(7) / open-v(10) / inter.beyond.threshold(13)
        CenterlineSeparation = Config_Three{i,28};      % (feet)
        if ~strcmp(Config_Three{i,27},'N/A')
            if  (~strcmp(Config_Three{i,20},'Y')) && (wi ~=1);  % For non-
towered airport under MMC/IMC condition,
                input.modelNumber = 1;                          % Treat
this multi-runway airport as single runway airport, capacity calculation-
wised.
                input.operationalStrategy = 3;
            else                                                %
Otherwise, treat this airport by its real configuration.
                switch input.modelNumber
                    case 3                                      % parallel
runways
                        if CenterlineSeparation >=4300          % judging
the appropriate operational strategies
                            input.operationalStrategy = [3 6 8 10 12 14 16
24 27];      % Far
                        elseif (4300 > CenterlineSeparation) &&
(CenterlineSeparation >=3500)
                            input.operationalStrategy = [11 13 15 23 26];
% Medium
                        elseif (3500 > CenterlineSeparation) &&
(CenterlineSeparation >=2500)
```

```
                            input.operationalStrategy = [2 5 7 9];
% Near
                        else
                            input.operationalStrategy = [1 4 17 18 19 20
21 22 25];     % Close
                        end
                    case 10
% open-v
                        input.operationalStrategy = [1 2 3 4 5];
                    case 7
% intersecting
                        if CenterlineSeparation >=3500
% judging the appropriate operational strategies
                            input.operationalStrategy = [2 4];
% Far
                        elseif (3500 > CenterlineSeparation) &&
(CenterlineSeparation >=700)
                            input.operationalStrategy = [1 3];
% Medium
                        else
                            input.modelNumber = 6;
                            input.operationalStrategy = [2 3];
% Should be treated as single runway
                        end
                    case 13
% model 13, intersecting beyond threshold
                        input.operationalStrategy = [1 2 3 4];
                    otherwise
                        break
                end  %26% switch input.modelNumber
            end

            % Assign filename to each ACM input text file.
            % The file name is in the following format:
            % AirportID_WeatherCondition_ModelNumber_StrategyNumber.inp
            for OS = 1:size(input.operationalStrategy,2);    % OS,
Operational Strategy.
                filename =
strcat(Config_Three{i,1},'_',weather(wi,:),'_',num2str(input.modelNumber),
...
                    '_',num2str(input.operationalStrategy(OS)));
                fullname = strcat(filename,'.inp');
                fp1 = fopen(fullname,'w');

                % start to build input file for ACM
                fprintf(fp1,'%-
6s%2d%2d%2d%3s\n',Config_Three{i,1},0,0,0,weather(wi,:));

fprintf(fp1,'%4d%4d%4d\n',input.modelNumber,input.operationalStrategy(OS),
input.alternatingArrivlas);

                fprintf(fp1,'%-6s%2d%2d%2d\n','RWY1',1,1,0);            %
aircraft mix on runway 1 (small to heavy)

fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',Config_Three{i,30},Config_Three{i,31}
,Config_Three{i,32},Config_Three{i,33});
```

```
                if input.modelNumber ~= 1;
                    fprintf(fp1,'%-6s%2d%2d%2d\n','RWY2',2,1,0);
% aircraft mix on runway 2 (small to heavy)

fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',Config_Three{i,30},Config_Three{i,31}
,Config_Three{i,32},Config_Three{i,33});
                end
                if (input.modelNumber ~= 6) &&  (input.modelNumber ~= 1);
                    fprintf(fp1,'%-6s%2d%2d%2d\n','RWY3',3,1,0);        %
aircraft mix on runway 3 (small to heavy)

fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',Config_Three{i,30},Config_Three{i,31}
,Config_Three{i,32},Config_Three{i,33});
                end

                fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR1',1,2,0);          %
average arr. ROT for all aircraft type at runway1
                fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);
                if input.modelNumber ~= 1;
                    fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR2',2,2,0);
% average arr. ROT for all aircraft type at runway1
                    fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);
                end
                if (input.modelNumber ~= 6) &&  (input.modelNumber ~= 1);
                    fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR3',3,2,0);      %
average arr. ROT for all aircraft type at runway1
                    fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);
                end

                fprintf(fp1,'%-6s%2d%2d%2d\n','DLTAIJ',0,4,0);          %
minimum arrival separations, in nm, for i followed by j

                if  strcmp(Config_Three{i,20},'Y');                     %
see if this aiport has tower on site
                    switch wi                                           %
separation for aiport with tower
                        case 1

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(2,2),ArrArr_Data(2,3),ArrArr_Data(2,4),ArrArr_Data(2,5),...

ArrArr_Data(3,2),ArrArr_Data(3,3),ArrArr_Data(3,4),ArrArr_Data(3,5),...

ArrArr_Data(4,2),ArrArr_Data(4,3),ArrArr_Data(4,4),ArrArr_Data(4,5),...

ArrArr_Data(5,2),ArrArr_Data(5,3),ArrArr_Data(5,4),ArrArr_Data(5,5));
                        otherwise

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(9,2),ArrArr_Data(9,3),ArrArr_Data(9,4),ArrArr_Data(9,5),...
```

```
ArrArr_Data(10,2),ArrArr_Data(10,3),ArrArr_Data(10,4),ArrArr_Data(10,5),..
.
ArrArr_Data(11,2),ArrArr_Data(11,3),ArrArr_Data(11,4),ArrArr_Data(11,5),..
.
ArrArr_Data(12,2),ArrArr_Data(12,3),ArrArr_Data(12,4),ArrArr_Data(12,5));
                    end
                else                                                 %
separation for aiport without tower
                    switch wi
                        case 1

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(2,9),ArrArr_Data(2,10),ArrArr_Data(2,11),ArrArr_Data(2,12),...

ArrArr_Data(3,9),ArrArr_Data(3,10),ArrArr_Data(3,11),ArrArr_Data(3,12),...

ArrArr_Data(4,9),ArrArr_Data(4,10),ArrArr_Data(4,11),ArrArr_Data(4,12),...

ArrArr_Data(5,9),ArrArr_Data(5,10),ArrArr_Data(5,11),ArrArr_Data(5,12));
                        otherwise

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(9,9),ArrArr_Data(9,10),ArrArr_Data(9,11),ArrArr_Data(9,12),...

ArrArr_Data(10,9),ArrArr_Data(10,10),ArrArr_Data(10,11),ArrArr_Data(10,12)
,...

ArrArr_Data(11,9),ArrArr_Data(11,10),ArrArr_Data(11,11),ArrArr_Data(11,12)
,...

ArrArr_Data(12,9),ArrArr_Data(12,10),ArrArr_Data(12,11),ArrArr_Data(12,12)
);
                    end
                end

                fprintf(fp1,'%-6s%2d%2d%2d\n','APPSPD',0,5,0);        %
approach speed, in knots
                fprintf(fp1,'%4d%4d%4d%4d\n',120,130,140,145);

                fprintf(fp1,'%-6s%2d%2d%2d\n','DRBAR',0,6,0);         %
departure runway occupancy time, in seconds
                fprintf(fp1,'%4d%4d%4d%4d\n',34,34,39,39);

                fprintf(fp1,'%-6s%2d%2d%2d\n','DDSR',0,7,0);          %
minimum departure separations, in seconds, for i followed by j

                if  strcmp(Config_Three{i,20},'Y');                    %
see if this aiport has tower on site
```

```
                    switch wi                                       %
separation for aiport with tower
                        case 1

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(2,2),DepDep_Data(2,3),DepDep_Data(2,4),DepDep_Data(2,5),...

DepDep_Data(3,2),DepDep_Data(3,3),DepDep_Data(3,4),DepDep_Data(3,5),...

DepDep_Data(4,2),DepDep_Data(4,3),DepDep_Data(4,4),DepDep_Data(4,5),...

DepDep_Data(5,2),DepDep_Data(5,3),DepDep_Data(5,4),DepDep_Data(5,5));
                        otherwise

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(9,2),DepDep_Data(9,3),DepDep_Data(9,4),DepDep_Data(9,5),...

DepDep_Data(10,2),DepDep_Data(10,3),DepDep_Data(10,4),DepDep_Data(10,5),..
.

DepDep_Data(11,2),DepDep_Data(11,3),DepDep_Data(11,4),DepDep_Data(11,5),..
.

DepDep_Data(12,2),DepDep_Data(12,3),DepDep_Data(12,4),DepDep_Data(12,5));
                    end
                else                                                %
separation for aiport without tower
                    switch wi
                        case 1

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(2,9),DepDep_Data(2,10),DepDep_Data(2,11),DepDep_Data(2,12),...

DepDep_Data(3,9),DepDep_Data(3,10),DepDep_Data(3,11),DepDep_Data(3,12),...

DepDep_Data(4,9),DepDep_Data(4,10),DepDep_Data(4,11),DepDep_Data(4,12),...

DepDep_Data(5,9),DepDep_Data(5,10),DepDep_Data(5,11),DepDep_Data(5,12));
                        otherwise

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(9,9),DepDep_Data(9,10),DepDep_Data(9,11),DepDep_Data(9,12),...

DepDep_Data(10,9),DepDep_Data(10,10),DepDep_Data(10,11),DepDep_Data(10,12)
,...

DepDep_Data(11,9),DepDep_Data(11,10),DepDep_Data(11,11),DepDep_Data(11,12)
,...

DepDep_Data(12,9),DepDep_Data(12,10),DepDep_Data(12,11),DepDep_Data(12,12)
);
                    end
                end
```

```
                fprintf(fp1,'%-6s%2d%2d%2d\n','GAMA',0,8,0);         %
length of final approach path, in nm
                if (wi ~= 1) && (~strcmp(Config_Three{i,20},'Y'))
                    fprintf(fp1,'%4d%4d%4d%4d\n',15,15,15,15);
                else fprintf(fp1,'%4d%4d%4d%4d\n',7,7,7,7);
                end

                fprintf(fp1,'%-6s%2d%2d%2d\n','TGRBAR',0,9,0);       %
touch-and-go runway occupancy time, in seconds
                fprintf(fp1,'%4d%4d%4d%4d\n',15,15,15,15);


                % if input.modelNumber == 10||13
                if (input.modelNumber == 10)||(input.modelNumber == 13)
                    fprintf(fp1,'%-6s%2d%2d\n','OPENV',0,10);         %
for open-V/ intersecting beyond threshold

fprintf(fp1,'%4.0f%4.0f%4.1f%4.1f\n',Config_Three{i,29},CenterlineSeparati
on,10,2.0);    % angle(degress); distance between threshold of runways
                end %89% if Config_Two{i,24} ~= 'N/A'

                %  elseif input.modelNumber == 7, intersecting
                fprintf(fp1,'%-6s%2d%2d%2d\n','TWOIN',0,11,0);     %
airborne intersection indicatior; needed for 6-1,6-3

fprintf(fp1,'%4.0f%4.1f%4.1f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f\n',1,
10,10,10,10,10,10,10,10,10,10);

                fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',1,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                    10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);

                fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',2,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                    10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);

                fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',3,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                    10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);
```

```
                fprintf(fp1,'%-6s%2d%2d%2d\n','DICBR',0,13,0);     % the
separation(nm) between departure I and arrival J on the crossing runway;
needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...
                    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3);

                fprintf(fp1,'%-6s%2d%2d%2d\n','SIGMAS',0,19,0);
                if  (~strcmp(Config_Three{i,20},'Y')) && (wi == 1);
                    fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f\n',8,30,0,6,6);
                else
                    fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f\n',8,18,0,6,6);
                end
                % sigma of arrival ROT         (seconds),
                % sigma of interarrival time   (seconds),
                % sigma of touch and go ROT    (seconds),
                % sigma of cleared-to-roll time(seconds),
                % sigma of dep. ROT            (seconds)

                switch wi
                    case 1      % VMC
                        visi = 5.0;
                        ceil = 5000;
                    case 2      % MMC
                        visi = 2.5;          % needs check
                        ceil = 900;          % needs check
                    case 3      % IMC
                        visi = 0.0;
                        ceil = 0;
                end %114% switch wi
                fprintf(fp1,'%-6s%2d%2d%2d\n','OTHERS',0,20,0);
                if (~strcmp(Config_Three{i,20},'Y')) && (wi ~= 1);

fprintf(fp1,'%4.2f%4.2f%4.1f%4.1f%4.0f%4.1f%4.2f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f%4.0f\n',...

0.05,0.05,15,visi,ceil,3,0,7777,10,20,30,40,50,60,70,80,90);
                else

fprintf(fp1,'%4.2f%4.2f%4.1f%4.1f%4.0f%4.1f%4.2f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f%4.0f\n',...

0.05,0.05,3,visi,ceil,3,0,7777,10,20,30,40,50,60,70,80,90);
                end
                % %of violation for arrival ROT, %of violation for
interarrival time,
                % dep/arr separation(nm), visibility(statute miles),
ceiling(feet),
                % glide slope angle (degrees), proportion of touch and go
operations,
                % arrival percentages: 8888 (arrival priority and
departure priority), 50%
```

```
                fprintf(fp1,'%-6s%2d%2d%2d\n','BDD',0,22,0);        %
buffer time (seconds), between simultaneous dep. on close-spaced parallel
runways

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                    0,0,0,25,0,0,0,20,0,0,0,10,25,20,10,0);

                fprintf(fp1,'%-6s%2d%2d%2d\n','BAA',0,24,0);        %
buffer time (seconds), between simultaneous arr. on close-spaced parallel
runways

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                    0,0,0,25,0,0,0,20,0,0,0,10,25,20,10,0);

                fprintf(fp1,'%-6s%2d%2d%2d\n','ALTARR',0,25,1);
                fprintf(fp1,'%4.1f%6.0f%6.0f%4.1f\n',2,3000,0,0);
                % diagonal separation (nm); separation between runway
centerlines (feet);
                % thresholds displacement (feet); relative displacement of
gates for two runways (nm);

                %    fprintf(fp1,'%-6s%2D%2D%2D\n','INCIAT',0,26,1);
% maximum number of iterations on the f.e.d. mix; convergence
                %
fprintf(fp1,'%4.0f%4.3f%4.0f%4.0f\n',0.05,0.05,0.20,0.1);
                % criterion for f.e.d mix iterations; maximum number of
points for
                % arrival capacity is calculated; increment by which the
arrival gaps are stretched (seconds);
                fclose(fp1);

                dos(['ACMrun ',filename]);

movefile(fullname,'d:\YuehTing\Research_Project\ACMinput\');

movefile(strcat(filename,'.res'),'d:\YuehTing\Research_Project\ACMoutput\'
);
                fprintf(fid,'%-
72s\n',strcat('d:\YuehTing\Research_Project\ACMoutput\',filename,'.res'));

            end %53% for OS = 1:size(input.operationalStrategy,2);
        end %27% for i = 1 : NumApTwo
    end
end %26% for wi = 1:3;


%% For Four Runway Airport

for wi = 1:3;       % wi, weather index; VMC, MMC, IMC
    for i = 1 : NumApFour
        input.modelNumber = Config_Four{i,27};                  %
parallel(4) / intersecting(15) / open-v(11) / inter.beyond.threshold(14)
        CenterlineSeparation = Config_Four{i,28};               % (feet)
```

```
        if ~strcmp(Config_Four{i,27},'N/A')
            if  (~strcmp(Config_Four{i,20},'Y')) && (wi ~=1);   % For non-
towered airport under MMC/IMC condition,
                input.modelNumber = 1;                          % Treat
this multi-runway airport as single runway airport, capacity calculation-
wised.
                input.operationalStrategy = 3;
            else
                switch input.modelNumber
                    case 4          % parallel runways
                        if CenterlineSeparation >=4300         % judging
the appropriate operational strategies
                            input.operationalStrategy = [2 4 6 8 10 12 14
16 18 20 22 23];   % Far
                        elseif 4300 > CenterlineSeparation
                            input.operationalStrategy = [1 3 5 7 9 11 13
15 17 19 21 25];    % Medium
                        end  %28% if CenterlineSeparation >=4300
                    case 11          % open-v
                        input.operationalStrategy = [1 2 3 4];           %
                    case 15          % intersecting
                        input.operationalStrategy = [1 2];
                    case 14
                        input.operationalStrategy = [1 2 3 4];
                    otherwise       % model 14, intersecting beyond
threshold
                        break
                end  %26% switch input.modelNumber
            end

            % Assign filename to each ACM input text file.
            % The file name is in the following format:
            % AirportID_WeatherCondition_ModelNumber_StrategyNumber.inp
            for OS = 1:size(input.operationalStrategy,2);    % OS,
Operational Strategy.
                filename =
strcat(Config_Four{i,1},'_',weather(wi,:),'_',num2str(input.modelNumber),.
..
                    '_',num2str(input.operationalStrategy(OS)));
                fullname = strcat(filename,'.inp');
                fp1 = fopen(fullname,'w');

                % start to build input file for ACM
                fprintf(fp1,'%-
6s%2d%2d%2d%3s\n',Config_Four{i,1},0,0,0,weather(wi,:));

fprintf(fp1,'%4d%4d%4d\n',input.modelNumber,input.operationalStrategy(OS),
input.alternatingArrivlas);

                fprintf(fp1,'%-6s%2d%2d%2d\n','RWY1',1,1,0);           %
aircraft mix on runway 1 (small to heavy)

fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',Config_Four{i,30},Config_Four{i,31},C
onfig_Four{i,32},Config_Four{i,33});
                if input.modelNumber ~= 1;
```

```
                    fprintf(fp1,'%-6s%2d%2d%2d\n','RWY2',2,1,0);        %
aircraft mix on runway 2 (small to heavy)

fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',Config_Four{i,30},Config_Four{i,31},C
onfig_Four{i,32},Config_Four{i,33});
                    fprintf(fp1,'%-6s%2d%2d%2d\n','RWY3',3,1,0);        %
aircraft mix on runway 3 (small to heavy)

fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',Config_Four{i,30},Config_Four{i,31},C
onfig_Four{i,32},Config_Four{i,33});
                    fprintf(fp1,'%-6s%2d%2d%2d\n','RWY4',4,1,0);        %
aircraft mix on runway 4 (small to heavy)

fprintf(fp1,'%4.2f%4.2f%4.2f%4.2f\n',Config_Four{i,30},Config_Four{i,31},C
onfig_Four{i,32},Config_Four{i,33});
                end

                fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR1',1,2,0);
% average arr. ROT for all aircraft type at runway1
                fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);
                if input.modelNumber ~= 1;
                    fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR2',2,2,0);
% average arr. ROT for all aircraft type at runway1
                    fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);
                    fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR3',3,2,0);
% average arr. ROT for all aircraft type at runway1
                    fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);
                    fprintf(fp1,'%-6s%2d%2d%2d\n','ARBAR4',4,2,0);
% average arr. ROT for all aircraft type at runway1
                    fprintf(fp1,'%4d%4d%4d%4d\n',38,40,43,48);
                end

                fprintf(fp1,'%-6s%2d%2d%2d\n','DLTAIJ',0,4,0);        %
minimum arrival separations, in nm, for i followed by j

                if  strcmp(Config_Four{i,20},'Y');                    %
see if this aiport has tower on site
                    switch wi                                          %
separation for aiport with tower
                        case 1

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(2,2),ArrArr_Data(2,3),ArrArr_Data(2,4),ArrArr_Data(2,5),...

ArrArr_Data(3,2),ArrArr_Data(3,3),ArrArr_Data(3,4),ArrArr_Data(3,5),...

ArrArr_Data(4,2),ArrArr_Data(4,3),ArrArr_Data(4,4),ArrArr_Data(4,5),...

ArrArr_Data(5,2),ArrArr_Data(5,3),ArrArr_Data(5,4),ArrArr_Data(5,5));
                        otherwise

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...
```

```
ArrArr_Data(9,2),ArrArr_Data(9,3),ArrArr_Data(9,4),ArrArr_Data(9,5),...

ArrArr_Data(10,2),ArrArr_Data(10,3),ArrArr_Data(10,4),ArrArr_Data(10,5),..
.

ArrArr_Data(11,2),ArrArr_Data(11,3),ArrArr_Data(11,4),ArrArr_Data(11,5),..
.

ArrArr_Data(12,2),ArrArr_Data(12,3),ArrArr_Data(12,4),ArrArr_Data(12,5));
                    end
                else                                                %
separation for aiport without tower
                    switch wi
                        case 1

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(2,9),ArrArr_Data(2,10),ArrArr_Data(2,11),ArrArr_Data(2,12),...

ArrArr_Data(3,9),ArrArr_Data(3,10),ArrArr_Data(3,11),ArrArr_Data(3,12),...

ArrArr_Data(4,9),ArrArr_Data(4,10),ArrArr_Data(4,11),ArrArr_Data(4,12),...

ArrArr_Data(5,9),ArrArr_Data(5,10),ArrArr_Data(5,11),ArrArr_Data(5,12));
                        otherwise

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...

ArrArr_Data(9,9),ArrArr_Data(9,10),ArrArr_Data(9,11),ArrArr_Data(9,12),...

ArrArr_Data(10,9),ArrArr_Data(10,10),ArrArr_Data(10,11),ArrArr_Data(10,12)
,...

ArrArr_Data(11,9),ArrArr_Data(11,10),ArrArr_Data(11,11),ArrArr_Data(11,12)
,...

ArrArr_Data(12,9),ArrArr_Data(12,10),ArrArr_Data(12,11),ArrArr_Data(12,12)
);
                    end
                end

                fprintf(fp1,'%-6s%2d%2d%2d\n','APPSPD',0,5,0);        %
approach speed, in knots
                fprintf(fp1,'%4d%4d%4d%4d\n',120,130,140,145);

                fprintf(fp1,'%-6s%2d%2d%2d\n','DRBAR',0,6,0);         %
departure runway occupancy time, in seconds
                fprintf(fp1,'%4d%4d%4d%4d\n',34,34,39,39);

                fprintf(fp1,'%-6s%2d%2d%2d\n','DDSR',0,7,0);          %
minimum departure separations, in seconds, for i followed by j
```

```
                if  strcmp(Config_Four{i,20},'Y');                    %
see if this aiport has tower on site
                    switch wi                                         %
separation for aiport with tower
                        case 1

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(2,2),DepDep_Data(2,3),DepDep_Data(2,4),DepDep_Data(2,5),...

DepDep_Data(3,2),DepDep_Data(3,3),DepDep_Data(3,4),DepDep_Data(3,5),...

DepDep_Data(4,2),DepDep_Data(4,3),DepDep_Data(4,4),DepDep_Data(4,5),...

DepDep_Data(5,2),DepDep_Data(5,3),DepDep_Data(5,4),DepDep_Data(5,5));
                        otherwise

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(9,2),DepDep_Data(9,3),DepDep_Data(9,4),DepDep_Data(9,5),...

DepDep_Data(10,2),DepDep_Data(10,3),DepDep_Data(10,4),DepDep_Data(10,5),..
.

DepDep_Data(11,2),DepDep_Data(11,3),DepDep_Data(11,4),DepDep_Data(11,5),..
.

DepDep_Data(12,2),DepDep_Data(12,3),DepDep_Data(12,4),DepDep_Data(12,5));
                    end
                else                                                  %
separation for aiport without tower
                    switch wi
                        case 1

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(2,9),DepDep_Data(2,10),DepDep_Data(2,11),DepDep_Data(2,12),...

DepDep_Data(3,9),DepDep_Data(3,10),DepDep_Data(3,11),DepDep_Data(3,12),...

DepDep_Data(4,9),DepDep_Data(4,10),DepDep_Data(4,11),DepDep_Data(4,12),...

DepDep_Data(5,9),DepDep_Data(5,10),DepDep_Data(5,11),DepDep_Data(5,12));
                        otherwise

fprintf(fp1,'%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d\n',...

DepDep_Data(9,9),DepDep_Data(9,10),DepDep_Data(9,11),DepDep_Data(9,12),...

DepDep_Data(10,9),DepDep_Data(10,10),DepDep_Data(10,11),DepDep_Data(10,12)
,...

DepDep_Data(11,9),DepDep_Data(11,10),DepDep_Data(11,11),DepDep_Data(11,12)
,...

DepDep_Data(12,9),DepDep_Data(12,10),DepDep_Data(12,11),DepDep_Data(12,12)
);
```

```
                end
            end

            fprintf(fp1,'%-6s%2d%2d%2d\n','GAMA',0,8,0);          %
length of final approach path, in nm
            if (wi ~= 1) && (~strcmp(Config_Four{i,20},'Y'))
                fprintf(fp1,'%4d%4d%4d%4d\n',15,15,15,15);
            else fprintf(fp1,'%4d%4d%4d%4d\n',7,7,7,7);
            end

            fprintf(fp1,'%-6s%2d%2d%2d\n','TGRBAR',0,9,0);        %
touch-and-go runway occupancy time, in seconds
            fprintf(fp1,'%4d%4d%4d%4d\n',15,15,15,15);

            % if input.modelNumber == 11||14
            if (input.modelNumber == 11)||(input.modelNumber == 14)
                fprintf(fp1,'%-6s%2d%2d\n','OPENV',0,10);          %
for open-V/ intersecting beyond threshold

fprintf(fp1,'%4.0f%4.0f%4.1f%4.1f\n',Config_Four{i,29},CenterlineSeparatio
n,10,2.0);    % angle(degress); distance between threshold of runways
            end %89% if Config_Two{i,24} ~= 'N/A'

            fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',1,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);

            fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',2,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);

            fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',3,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);

            fprintf(fp1,'%-6s%2d%2d%2d\n','ADSR',4,12,0);      % the
separation(seconds) between arrival I(runway IRUM) and departure J on the
crossing runway; needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10);
```

```
                fprintf(fp1,'%-6s%2d%2d%2d\n','DICBR',0,13,0);     % the
separation(nm) between departure I and arrival J on the crossing runway;
needed for 6-2, 6-3, 12-3, 12-4

fprintf(fp1,'%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%4.1f%
4.1f%4.1f%4.1f%4.1f\n',...
                    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3);

                fprintf(fp1,'%-6s%2d%2d%2d\n','SIGMAS',0,19,0);
                if  (~strcmp(Config_Four{i,20},'Y')) && (wi == 1);
                    fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f\n',8,30,0,6,6);
                else
                    fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f\n',8,18,0,6,6);
                end
                % sigma of arrival ROT         (seconds),
                % sigma of interarrival time   (seconds),
                % sigma of touch and go ROT    (seconds),
                % sigma of cleared-to-roll time(seconds),
                % sigma of dep. ROT            (seconds)

                switch wi
                    case 1      % VMC
                        visi = 5.0;
                        ceil = 5000;
                    case 2      % MMC
                        visi = 2.5;          % needs check
                        ceil = 900;          % needs check
                    case 3      % IMC
                        visi = 0.0;
                        ceil = 0;
                end %114% switch wi
                fprintf(fp1,'%-6s%2d%2d%2d\n','OTHERS',0,20,0);
                if (~strcmp(Config_Four{i,20},'Y')) && (wi ~= 1);

fprintf(fp1,'%4.2f%4.2f%4.1f%4.1f%4.0f%4.1f%4.2f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f%4.0f\n',...

0.05,0.05,15,visi,ceil,3,0,7777,10,20,30,40,50,60,70,80,90);
                else

fprintf(fp1,'%4.2f%4.2f%4.1f%4.1f%4.0f%4.1f%4.2f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f%4.0f\n',...

0.05,0.05,3,visi,ceil,3,0,7777,10,20,30,40,50,60,70,80,90);
                end
                % %of violation for arrival ROT, %of violation for
interarrival time,
                % dep/arr separation(nm), visibility(statute miles),
ceiling(feet),
                % glide slope angle (degrees), proportion of touch and go
operations,
                % arrival percentages: 8888 (arrival priority and
departure priority), 50%
```

```
                fprintf(fp1,'%-6s%2d%2d%2d\n','BDD',0,22,0);        %
buffer time (seconds), between simultaneous dep. on close-spaced parallel
runways

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                    0,0,0,25,0,0,0,20,0,0,0,10,25,20,10,0);

                fprintf(fp1,'%-6s%2d%2d%2d\n','BAA',0,24,0);        %
buffer time (seconds), between simultaneous arr. on close-spaced parallel
runways

fprintf(fp1,'%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%4.0f%
4.0f%4.0f%4.0f%4.0f\n',...
                    0,0,0,25,0,0,0,20,0,0,0,10,25,20,10,0);

                fprintf(fp1,'%-6s%2d%2d%2d\n','ALTARR',0,25,1);
                fprintf(fp1,'%4.1f%6.0f%6.0f%4.1f\n',2,3000,0,0);
                % diagonal separation (nm); separation between runway
centerlines (feet);
                % thresholds displacement (feet); relative displacement of
gates for two runways (nm);

                fclose(fp1);

                dos(['ACMrun ',filename]);

movefile(fullname,'d:\YuehTing\Research_Project\ACMinput\');

movefile(strcat(filename,'.res'),'d:\YuehTing\Research_Project\ACMoutput\'
);
                fprintf(fid,'%-
72s\n',strcat('d:\YuehTing\Research_Project\ACMoutput\',filename,'.res'));

            end %53% for OS = 1:size(input.operationalStrategy,2);
        end
    end %27% for i = 1 : NumApTwo
end %26% for wi = 1:3;
fclose(fid);
```

APPENDIX E-4 MATLAB PROGRAM FILE FOR MULTI-RUNWAY AIRPORT-RESULT COLLECTION

```
% ***********************************************************************
% "ACM_extractor.m"
% Extract information from ACM output file and save them to another
abstract file
%
% Coded by Yueh-Ting Chen, May 1, 2006
% Helped by Dr. Baik, May 22, 2006
% Last Modifaction: Nov 17,2006
%
************************************************************************

%% extract the capacity data from each airport file
%
clc

totLinesOutput_InVMC = 0;
totLinesOutput_InMMC = 0;
totLinesOutput_InIMC = 0;

output_VMC = [];
output_MMC = [];
output_IMC = [];

FP_outputFileNames = fopen('D:\YuehTing\Research_Project\Multiple
Runway\outputFileNames.dat','r');    % "outputFileNames.dat" lists all the
output file names.
totFilesOpened = 0;

while (~feof(FP_outputFileNames))
    totFilesOpened      = totFilesOpened + 1;
    aFileName           = fgetl(FP_outputFileNames);     % read line in
"outputFileNames.dat"
    FP_aOutputFile      = fopen(aFileName,'r');          % open the file
whose name previously read
    totLines            = 0;                             % the serial
number of line which is reading in this specific file
    percent             = 0;
    thisAirportId       = aFileName(40:42);
    weather             = aFileName(44:46);

    while (~feof(FP_aOutputFile))
        totLines            = totLines + 1;
        aLine               = fgetl(FP_aOutputFile);
        if totLines == 7
            modelNo         = str2double(aLine(1:5));
            strategy        = str2double(aLine(8:10));
        elseif ~isempty(findstr(aLine,'Arrival Priority'))      % look for
the line which has keyword "Arrival Priority"
            capArrPriArr    = str2double(aLine(37:45));    % save Arrival
Priority Arrival capacity number in the struct "output"
```

```
            capArrPriDep    = str2double(aLine(49:56));    % save Arrival
Priority Departure capacity number in the struct "output"
        elseif ~isempty(findstr(aLine,'Departure Priority'))    % look for
the line which has keyword "Departure Priority"
            capDepPriArr    = str2double(aLine(37:45));    % save
Departure Priority Arrival capacity number in the struct "output"
            capDepPriDep    = str2double(aLine(49:56));    % save
Departure Priority Departure capacity number in the struct "output"
            if capDepPriDep == 0;
                break
            end
        elseif ~isempty(findstr(aLine,'TOTAL ='))              % look for
the line which has keyword "TOTAL"
            percent         = percent + 1;
            switch percent
                case 1
                    capTenArr      = str2double(aLine(44:50));   % save 10%
Arrival number in the struct "output"
                    capTenDep      = str2double(aLine(66:70));   % save 10%
Arrival number in the struct "output"
                case 2
                    capTwentyArr   = str2double(aLine(44:50));   % save 20%
Arrival number in the struct "output"
                    capTwentyDep   = str2double(aLine(66:70));   % save 20%
Arrival number in the struct "output"
                case 3
                    capThirtyArr   = str2double(aLine(44:50));   % save 30%
Arrival number in the struct "output"
                    capThirtyDep   = str2double(aLine(66:70));   % save 30%
Arrival number in the struct "output"
                case 4
                    capFourtyArr   = str2double(aLine(44:50));   % save 40%
Arrival number in the struct "output"
                    capFourtyDep   = str2double(aLine(66:70));   % save 40%
Arrival number in the struct "output"
                case 5
                    capFiftyArr    = str2double(aLine(44:50));   % save 50%
Arrival number in the struct "output"
                    capFiftyDep    = str2double(aLine(66:70));   % save 50%
Arrival number in the struct "output"
                case 6
                    capSixtyArr    = str2double(aLine(44:50));   % save 60%
Arrival number in the struct "output"
                    capSixtyDep    = str2double(aLine(66:70));   % save 60%
Arrival number in the struct "output"
                case 7
                    capSeventyArr = str2double(aLine(44:50));   % save 70%
Arrival number in the struct "output"
                    capSeventyDep = str2double(aLine(66:70));   % save 70%
Arrival number in the struct "output"
                case 8
                    capEightyArr   = str2double(aLine(44:50));   % save 80%
Arrival number in the struct "output"
                    capEightyDep   = str2double(aLine(66:70));   % save 80%
Arrival number in the struct "output"
                case 9
```

```
                    capNinetyArr  = str2double(aLine(44:50));   % save 90%
Arrival number in the struct "output"
                    capNinetyDep  = str2double(aLine(66:70));   % save 90%
Arrival number in the struct "output"
            end
        end %28% if (totLines == 6)||(totLines == 7)
    end %25% while (~feof(FP_aOutputFile))    end %25% while
(~feof(FP_aOutputFile))

    if capDepPriDep == 0;
        continue
    end

    switch weather

        case 'VMC'
            totLinesOutput_InVMC = totLinesOutput_InVMC + 1;
            output_VMC{totLinesOutput_InVMC, 1}  = thisAirportId;
            output_VMC{totLinesOutput_InVMC, 2}  = modelNo;
            output_VMC{totLinesOutput_InVMC, 3}  = strategy;
            output_VMC{totLinesOutput_InVMC, 4}  = capArrPriArr;
            output_VMC{totLinesOutput_InVMC, 5}  = capArrPriDep;
            output_VMC{totLinesOutput_InVMC, 6}  = capArrPriArr +
capArrPriDep;
            output_VMC{totLinesOutput_InVMC, 7}  = 0;
            output_VMC{totLinesOutput_InVMC, 8}  = capDepPriDep;
            output_VMC{totLinesOutput_InVMC, 9}  = 0 + capDepPriDep;
            output_VMC{totLinesOutput_InVMC, 10} = capTenArr;
            output_VMC{totLinesOutput_InVMC, 11} = capTenDep;
            output_VMC{totLinesOutput_InVMC, 12} = capTenArr + capTenDep;
            output_VMC{totLinesOutput_InVMC, 13} = capTwentyArr;
            output_VMC{totLinesOutput_InVMC, 14} = capTwentyDep;
            output_VMC{totLinesOutput_InVMC, 15} = capTwentyArr +
capTwentyDep;
            output_VMC{totLinesOutput_InVMC, 16} = capThirtyArr;
            output_VMC{totLinesOutput_InVMC, 17} = capThirtyDep;
            output_VMC{totLinesOutput_InVMC, 18} = capThirtyArr +
capThirtyDep;
            output_VMC{totLinesOutput_InVMC, 19} = capFourtyArr;
            output_VMC{totLinesOutput_InVMC, 20} = capFourtyDep;
            output_VMC{totLinesOutput_InVMC, 21} = capFourtyArr +
capFourtyDep;
            output_VMC{totLinesOutput_InVMC, 22} = capFiftyArr;
            output_VMC{totLinesOutput_InVMC, 23} = capFiftyDep;
            output_VMC{totLinesOutput_InVMC, 24} = capFiftyArr +
capFiftyDep;
            output_VMC{totLinesOutput_InVMC, 25} = capSixtyArr;
            output_VMC{totLinesOutput_InVMC, 26} = capSixtyDep;
            output_VMC{totLinesOutput_InVMC, 27} = capSixtyArr +
capSixtyDep;
            output_VMC{totLinesOutput_InVMC, 28} = capSeventyArr;
            output_VMC{totLinesOutput_InVMC, 29} = capSeventyDep;
            output_VMC{totLinesOutput_InVMC, 30} = capSeventyArr +
capSeventyDep;
            output_VMC{totLinesOutput_InVMC, 31} = capEightyArr;
            output_VMC{totLinesOutput_InVMC, 32} = capEightyDep;
```

```
            output_VMC{totLinesOutput_InVMC, 33} = capEightyArr +
capEightyDep;
            output_VMC{totLinesOutput_InVMC, 34} = capNinetyArr;
            output_VMC{totLinesOutput_InVMC, 35} = capNinetyDep;
            output_VMC{totLinesOutput_InVMC, 36} = capNinetyArr +
capNinetyDep;

        case 'MMC'
            totLinesOutput_InMMC = totLinesOutput_InMMC + 1;
            output_MMC{totLinesOutput_InMMC, 1}  = thisAirportId;
            output_MMC{totLinesOutput_InMMC, 2}  = modelNo;
            output_MMC{totLinesOutput_InMMC, 3}  = strategy;
            output_MMC{totLinesOutput_InMMC, 4}  = capArrPriArr;
            output_MMC{totLinesOutput_InMMC, 5}  = capArrPriDep;
            output_MMC{totLinesOutput_InMMC, 6}  = capArrPriArr +
capArrPriDep;
            output_MMC{totLinesOutput_InMMC, 7}  = 0;
            output_MMC{totLinesOutput_InMMC, 8}  = capDepPriDep;
            output_MMC{totLinesOutput_InMMC, 9}  = 0 + capDepPriDep;
            output_MMC{totLinesOutput_InMMC, 10} = capTenArr;
            output_MMC{totLinesOutput_InMMC, 11} = capTenDep;
            output_MMC{totLinesOutput_InMMC, 12} = capTenArr + capTenDep;
            output_MMC{totLinesOutput_InMMC, 13} = capTwentyArr;
            output_MMC{totLinesOutput_InMMC, 14} = capTwentyDep;
            output_MMC{totLinesOutput_InMMC, 15} = capTwentyArr +
capTwentyDep;
            output_MMC{totLinesOutput_InMMC, 16} = capThirtyArr;
            output_MMC{totLinesOutput_InMMC, 17} = capThirtyDep;
            output_MMC{totLinesOutput_InMMC, 18} = capThirtyArr +
capThirtyDep;
            output_MMC{totLinesOutput_InMMC, 19} = capFourtyArr;
            output_MMC{totLinesOutput_InMMC, 20} = capFourtyDep;
            output_MMC{totLinesOutput_InMMC, 21} = capFourtyArr +
capFourtyDep;
            output_MMC{totLinesOutput_InMMC, 22} = capFiftyArr;
            output_MMC{totLinesOutput_InMMC, 23} = capFiftyDep;
            output_MMC{totLinesOutput_InMMC, 24} = capFiftyArr +
capFiftyDep;
            output_MMC{totLinesOutput_InMMC, 25} = capSixtyArr;
            output_MMC{totLinesOutput_InMMC, 26} = capSixtyDep;
            output_MMC{totLinesOutput_InMMC, 27} = capSixtyArr +
capSixtyDep;
            output_MMC{totLinesOutput_InMMC, 28} = capSeventyArr;
            output_MMC{totLinesOutput_InMMC, 29} = capSeventyDep;
            output_MMC{totLinesOutput_InMMC, 30} = capSeventyArr +
capSeventyDep;
            output_MMC{totLinesOutput_InMMC, 31} = capEightyArr;
            output_MMC{totLinesOutput_InMMC, 32} = capEightyDep;
            output_MMC{totLinesOutput_InMMC, 33} = capEightyArr +
capEightyDep;
            output_MMC{totLinesOutput_InMMC, 34} = capNinetyArr;
            output_MMC{totLinesOutput_InMMC, 35} = capNinetyDep;
            output_MMC{totLinesOutput_InMMC, 36} = capNinetyArr +
capNinetyDep;

        case 'IMC'
```

```
            totLinesOutput_InIMC = totLinesOutput_InIMC + 1;
            output_IMC{totLinesOutput_InIMC, 1}  = thisAirportId;
            output_IMC{totLinesOutput_InIMC, 2}  = modelNo;
            output_IMC{totLinesOutput_InIMC, 3}  = strategy;
            output_IMC{totLinesOutput_InIMC, 4}  = capArrPriArr;
            output_IMC{totLinesOutput_InIMC, 5}  = capArrPriDep;
            output_IMC{totLinesOutput_InIMC, 6}  = capArrPriArr +
capArrPriDep;
            output_IMC{totLinesOutput_InIMC, 7}  = 0;
            output_IMC{totLinesOutput_InIMC, 8}  = capDepPriDep;
            output_IMC{totLinesOutput_InIMC, 9}  = 0 + capDepPriDep;
            output_IMC{totLinesOutput_InIMC, 10} = capTenArr;
            output_IMC{totLinesOutput_InIMC, 11} = capTenDep;
            output_IMC{totLinesOutput_InIMC, 12} = capTenArr + capTenDep;
            output_IMC{totLinesOutput_InIMC, 13} = capTwentyArr;
            output_IMC{totLinesOutput_InIMC, 14} = capTwentyDep;
            output_IMC{totLinesOutput_InIMC, 15} = capTwentyArr +
capTwentyDep;
            output_IMC{totLinesOutput_InIMC, 16} = capThirtyArr;
            output_IMC{totLinesOutput_InIMC, 17} = capThirtyDep;
            output_IMC{totLinesOutput_InIMC, 18} = capThirtyArr +
capThirtyDep;
            output_IMC{totLinesOutput_InIMC, 19} = capFourtyArr;
            output_IMC{totLinesOutput_InIMC, 20} = capFourtyDep;
            output_IMC{totLinesOutput_InIMC, 21} = capFourtyArr +
capFourtyDep;
            output_IMC{totLinesOutput_InIMC, 22} = capFiftyArr;
            output_IMC{totLinesOutput_InIMC, 23} = capFiftyDep;
            output_IMC{totLinesOutput_InIMC, 24} = capFiftyArr +
capFiftyDep;
            output_IMC{totLinesOutput_InIMC, 25} = capSixtyArr;
            output_IMC{totLinesOutput_InIMC, 26} = capSixtyDep;
            output_IMC{totLinesOutput_InIMC, 27} = capSixtyArr +
capSixtyDep;
            output_IMC{totLinesOutput_InIMC, 28} = capSeventyArr;
            output_IMC{totLinesOutput_InIMC, 29} = capSeventyDep;
            output_IMC{totLinesOutput_InIMC, 30} = capSeventyArr +
capSeventyDep;
            output_IMC{totLinesOutput_InIMC, 31} = capEightyArr;
            output_IMC{totLinesOutput_InIMC, 32} = capEightyDep;
            output_IMC{totLinesOutput_InIMC, 33} = capEightyArr +
capEightyDep;
            output_IMC{totLinesOutput_InIMC, 34} = capNinetyArr;
            output_IMC{totLinesOutput_InIMC, 35} = capNinetyDep;
            output_IMC{totLinesOutput_InIMC, 36} = capNinetyArr +
capNinetyDep;
    end
    fclose(FP_aOutputFile);         % finish reading the specific output
file
end %18% while (~feof(FP_outputFileNames))
fclose(FP_outputFileNames);         % finish reading "outputFileNames.dat"

save output_Multi output_VMC output_MMC output_IMC
```

APPENDIX E-5 MATLAB PROGRAM FILE FOR MULTI-RUNWAY AIRPORT-RESULT COMPARISON

```
% ************************************************************************
% "CapacityAnalyzer.m"
% Compare the capacity data got from ACM_extractor.m, and find the minimum
and maximum capacity
% number from different strategies for each weather scenario at each
airport
%
% Coded by Yueh-Ting Chen, May 11, 2006
% Last Modifaction: Nov 17,2006
%
************************************************************************

clc
load output_Multi

weather = ['VMC';'MMC';'IMC'];
FinalCapacityMin = [];
FinalCapacityMax = [];

for wi = 1:3
    SI = 0;             % Serial Number Index for after-comparison-
capacity-result set of minimum capacity
    SX = 0;             % Serial Number Index for after-comparison-
capacity-result set of maximum capacity

    switch wi           % weather index, for three weather scenarios
        case 1
            output = output_VMC;
        case 2
            output = output_MMC;
        case 3
            output = output_IMC;
    end

    tempmin = output(1,:);      % temporary row for minimum capacity
    tempmax = output(1,:);      % temporary row for maximum capacity

% **************************** For Finding Mimimum Capacity
***************************************
    for RI = 2:size(output,1);       % how many sets of capacity number to
check/compare, total: 4158 sets
        if  strcmp(output(RI,1),output(RI-1,1))
%             output{RI,37} == output{RI-1,37}    % compare the airportID
between row RI and RI+1

% since there are three weather cases(for each operational

% strategy), and they are generated in order, so the next

% case(row) we should compare with is the row (RI+3). We
```

```
% check whether they are still the same airport ID, if so,

% then start to compare capacity number.

            if output{RI,6}(1,1) <= tempmin{6}(1,1)
                tempmin = output(RI,:);

% if ID and weather of (RI+3) are both the same as

% RI's, then we compare their capacity. If current

% one (RI) is larger, then we keep it, and go to next

% case (another +3) to compare again.
            end %29% if (output(RI+3).ArrPriArr + output(RI+3).ArrPriDep)
<= (Critical.ArrPriArr + Critical.ArrPriDep)
        else
            SI = SI + 1;
            switch wi
                case 1
                    FinalCapacity_Min_VMC(SI,:) = tempmin(1,:);
                case 2
                    FinalCapacity_Min_MMC(SI,:) = tempmin(1,:);
                case 3
                    FinalCapacity_Min_IMC(SI,:) = tempmin(1,:);
            end
            tempmin = output(RI,:);
        end %
        if RI == size(output,1)
            switch wi
                case 1
                    FinalCapacity_Min_VMC(SI+1,:) = tempmin(1,:);
                case 2
                    FinalCapacity_Min_MMC(SI+1,:) = tempmin(1,:);
                case 3
                    FinalCapacity_Min_IMC(SI+1,:) = tempmin(1,:);
            end
        end
    end

%
**************************************************************************
***********************

% **************************** For Finding Mimimum Capacity
***************************************
    for RX = 2:size(output,1);       % how many sets of capacity number to
check/compare, total: 4158 sets
        if  strcmp(output(RX,1),output(RX-1,1))
%         if output{RX,37} == output{RX-1,37}    % compare the airportID
between row RX and RX+1

% since there are three weather cases(for each operational

% strategy), and they are generated in order, so the next
```

```
% case(row) we should compare with is the row (RX+3). We

% check whether they are still the same airport ID, if so,

% then start to compare capacity number.

            if output{RX,6}(1,1) >= tempmax{6}(1,1)
                tempmax = output(RX,:);

% if ID and weather of (RX+3) are both the same as

% RX's, then we compare their capacity. If current

% one (RX) is larger, then we keep it, and go to next

% case (another +3) to compare again.
            end %29% if (output(RX+3).ArrPriArr + output(RX+3).ArrPriDep)
<= (Critical.ArrPriArr + Critical.ArrPriDep)
        else
            SX = SX + 1;
            switch wi
                case 1
                    FinalCapacity_Max_VMC(SX,:) = tempmax(1,:);
                case 2
                    FinalCapacity_Max_MMC(SX,:) = tempmax(1,:);
                case 3
                    FinalCapacity_Max_IMC(SX,:) = tempmax(1,:);
            end
            tempmax = output(RX,:);
        end %
        if RX == size(output,1)
            switch wi
                case 1
                    FinalCapacity_Max_VMC(SX+1,:) = tempmax(1,:);
                case 2
                    FinalCapacity_Max_MMC(SX+1,:) = tempmax(1,:);
                case 3
                    FinalCapacity_Max_IMC(SX+1,:) = tempmax(1,:);
            end
        end
    end
%
**************************************************************************
***********************


end

save FinalCapacityMax FinalCapacity_Max_VMC FinalCapacity_Max_MMC
FinalCapacity_Max_IMC
save FinalCapacityMin FinalCapacity_Min_VMC FinalCapacity_Min_MMC
FinalCapacity_Min_IMC

xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',FinalCapacity_Min_VMC,'VMC_Min','A193');
```

```
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',FinalCapacity_Min_MMC,'MMC_Min','A193');
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',FinalCapacity_Min_IMC,'IMC_Min','A193');
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',FinalCapacity_Max_VMC,'VMC_Max','A193');
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',FinalCapacity_Max_MMC,'MMC_Max','A193');
xlswrite('D:\YuehTing\Research_Project\Data\Capacity Result
V6',FinalCapacity_Max_IMC,'IMC_Max','A193');
```

APPENDIX F LIST OF THE TARGET AIRPORTS

Airport_Index	State_Name	Airport_ID	Airport_Name	City_Name
1	AL	02A	GRAGG-WADE FIELD	CLANTON
2	AL	04A	FRANK SIKES	LUVERNE
3	AL	06A	MOTON FIELD MUNI	TUSKEGEE
4	AL	07A	FRANKLIN FIELD	UNION SPRINGS
5	AL	08A	WETUMPKA MUNI	WETUMPKA
6	AL	09A	BUTLER-CHOCTAW COUNTY	BUTLER
7	AL	0A8	BIBB COUNTY	CENTREVILLE
8	AL	0J4	FLORALA MUNI	FLORALA
9	AL	0J6	HEADLAND MUNI	HEADLAND
10	AL	0R1	ATMORE MUNI	ATMORE
11	AL	11A	CLAYTON MUNI	CLAYTON
12	AL	12J	BREWTON MUNI	BREWTON
13	AL	14J	CARL FOLSOM	ELBA
14	AL	1A9	PRATTVILLE - GROUBY FIELD	PRATTVILLE
15	AL	1M4	POSEY FIELD	HALEYVILLE
16	AL	1R8	BAY MINETTE MUNI	BAY MINETTE
17	AL	20A	ROBBINS FIELD	ONEONTA
18	AL	23A	MALLARD	YORK
19	AL	26A	ASHLAND/LINEVILLE	ASHLAND/LINEVILLE
20	AL	2R5	ST ELMO	ST ELMO
21	AL	33J	GENEVA MUNI	GENEVA
22	AL	3A1	FOLSOM FIELD	CULLMAN
23	AL	3A7	EUTAW MUNI	EUTAW
24	AL	3M2	DOUBLE SPRINGS-WINSTON COUNTY	DOUBLE SPRINGS
25	AL	3M8	NORTH PICKENS	REFORM
26	AL	41A	REEVES	TALLASSEE
27	AL	4A6	SCOTTSBORO MUNI-WORD FIELD	SCOTTSBORO
28	AL	4A9	ISBELL FIELD	FORT PAYNE
29	AL	4R3	JACKSON MUNI	JACKSON
30	AL	4R4	H L SONNY CALLAHAN	FAIRHOPE

31	AL	5M0	ROUNTREE FIELD	HARTSELLE
32	AL	5R1	ROY WILCOX	CHATOM
33	AL	5R4	FOLEY MUNI	FOLEY
34	AL	60A	BRUNDIDGE MUNI	BRUNDIDGE
35	AL	61A	CAMDEN MUNI	CAMDEN
36	AL	67A	FORT DEPOSIT-LOWNDES COUNTY	FORT DEPOSIT
37	AL	70A	FREDDIE JONES FIELD	LINDEN
38	AL	71A	PINE HILL MUNI	PINE HILL
39	AL	71J	BLACKWELL FIELD	OZARK
40	AL	79J	ANDALUSIA-OPP	ANDALUSIA/OPP
41	AL	7A0	GREENSBORO MUNI	GREENSBORO
42	AL	7A2	DEMOPOLIS MUNI	DEMOPOLIS
43	AL	7A3	LANETT MUNI	LANETT
44	AL	7A5	ROANOKE MUNI	ROANOKE
45	AL	7A6	STEVENSON	STEVENSON
46	AL	8A0	THE ALBERTVILLE MUNI-THOMAS J BRUMLIK FLD	ALBERTVILLE
47	AL	8A1	GUNTERSVILLE MUNI - JOE STARNES FIELD	GUNTERSVILLE
48	AL	9A4	LAWRENCE COUNTY	COURTLAND
49	AL	A08	VAIDEN FIELD	MARION
50	AL	AIV	GEORGE DOWNER	ALICEVILLE
51	AL	ALX	THOMAS C RUSSELL FLD	ALEXANDER CITY
52	AL	ANB	ANNISTON METROPOLITAN	ANNISTON
53	AL	ASN	TALLADEGA MUNI	TALLADEGA
54	AL	AUO	AUBURN-OPELIKA ROBERT G. PITTS	AUBURN
55	AL	BFM	MOBILE DOWNTOWN	MOBILE
56	AL	BHM	BIRMINGHAM INTL	BIRMINGHAM
57	AL	C22	CENTRE MUNI	CENTRE
58	AL	DCU	PRYOR FIELD RGNL	DECATUR
59	AL	DHN	DOTHAN RGNL	DOTHAN
60	AL	EDN	ENTERPRISE MUNI	ENTERPRISE
61	AL	EET	SHELBY COUNTY	ALABASTER
62	AL	EKY	BESSEMER	BESSEMER
63	AL	EUF	WEEDON FIELD	EUFAULA

64	AL	GAD	GADSDEN MUNI	GADSDEN
65	AL	GZH	MIDDLETON FIELD	EVERGREEN
66	AL	HAB	MARION COUNTY-RANKIN FITE	HAMILTON
67	AL	HSV	HUNTSVILLE INTL-CARL T JONES FIELD	HUNTSVILLE
68	AL	JFX	WALKER COUNTY-BEVILL FIELD	JASPER
69	AL	JKA	JACK EDWARDS	GULF SHORES
70	AL	M22	RUSSELLVILLE MUNI	RUSSELLVILLE
71	AL	M55	LAMAR COUNTY	VERNON
72	AL	M95	RICHARD ARTHUR FIELD	FAYETTE
73	AL	MDQ	MADISON COUNTY EXECUTIVE	HUNTSVILLE
74	AL	MGM	MONTGOMERY RGNL (DANNELLY FIELD)	MONTGOMERY
75	AL	MOB	MOBILE REGIONAL	MOBILE
76	AL	MSL	NORTHWEST ALABAMA RGNL	MUSCLE SHOALS
77	AL	MVC	MONROE COUNTY	MONROEVILLE
78	AL	PLR	ST CLAIR COUNTY	PELL CITY
79	AL	PRN	MAC CRENSHAW MEMORIAL	GREENVILLE
80	AL	SCD	MERKEL FIELD SYLACAUGA MUNI	SYLACAUGA
81	AL	SEM	CRAIG FIELD	SELMA
82	AL	TCL	TUSCALOOSA REGIONAL	TUSCALOOSA
83	AL	TOI	TROY MUNI	TROY
84	AR	0M0	BILLY FREE MUNICIPAL	DUMAS
85	AR	1M1	NORTH LITTLE ROCK MUNI	NORTH LITTLE ROCK
86	AR	32A	DANVILLE MUNI	DANVILLE
87	AR	42A	MELBOURNE MUNI - JOHN E MILLER FIELD	MELBOURNE
88	AR	4A5	SEARCY COUNTY	MARSHALL
89	AR	4M1	CARROLL COUNTY	BERRYVILLE
90	AR	4M3	CARLISLE MUNI	CARLISLE
91	AR	5M5	CRYSTAL LAKE	DECATUR
92	AR	6M0	HAZEN MUNI	HAZEN
93	AR	6M7	MARIANNA/LEE COUNTY-STEVE EDWARDS FIELD	MARIANNA
94	AR	7M1	MC GEHEE MUNI	MC GEHEE
95	AR	AGO	MAGNOLIA MUNI	MAGNOLIA
96	AR	ARG	WALNUT RIDGE REGIONAL	WALNUT RIDGE

97	AR	ASG	SPRINGDALE MUNI	SPRINGDALE
98	AR	AWM	WEST MEMPHIS MUNI	WEST MEMPHIS
99	AR	BPK	OZARK REGIONAL	MOUNTAIN HOME
100	AR	BVX	BATESVILLE REGIONAL	BATESVILLE
101	AR	BYH	ARKANSAS INTERNATIONAL	BLYTHEVILLE
102	AR	CDH	HARRELL FIELD	CAMDEN
103	AR	CRT	Z M JACK STELL FIELD	CROSSETT
104	AR	CVK	SHARP COUNTY REGIONAL	ASH FLAT
105	AR	CWS	DENNIS F CANTRELL FIELD	CONWAY
106	AR	DEQ	J LYNN HELMS SEVIER COUNTY	DE QUEEN
107	AR	ELD	SOUTH ARKANSAS REGIONAL AT GOODWIN FIELD	EL DORADO
108	AR	FLP	MARION COUNTY REGIONAL	FLIPPIN
109	AR	FSM	FORT SMITH REGIONAL	FORT SMITH
110	AR	FYV	DRAKE FIELD	FAYETTEVILLE
111	AR	H35	CLARKSVILLE MUNI	CLARKSVILLE
112	AR	HBZ	HEBER SPRINGS MUNI	HEBER SPRINGS
113	AR	HEE	THOMPSON-ROBBINS	HELENA/WEST HELENA
114	AR	HKA	BLYTHEVILLE MUNI	BLYTHEVILLE
115	AR	HOT	MEMORIAL FIELD	HOT SPRINGS
116	AR	HRO	BOONE COUNTY	HARRISON
117	AR	JBR	JONESBORO MUNI	JONESBORO
118	AR	LIT	ADAMS FIELD	LITTLE ROCK
119	AR	LLQ	MONTICELLO MUNI/ELLIS FIELD	MONTICELLO
120	AR	M18	HOPE MUNI	HOPE
121	AR	M19	NEWPORT MUNI	NEWPORT
122	AR	M32	LAKE VILLAGE MUNI	LAKE VILLAGE
123	AR	M36	FRANK FEDERER MEMORIAL	BRINKLEY
124	AR	M60	WOODRUFF COUNTY	AUGUSTA
125	AR	M65	WYNNE MUNI	WYNNE
126	AR	M70	POCAHONTAS MUNI	POCAHONTAS
127	AR	M89	DEXTER B FLORENCE MEMORIAL FIELD	ARKADELPHIA
128	AR	MEZ	MENA INTERMOUNTAIN MUNICIPAL	MENA
129	AR	MPJ	PETIT JEAN PARK	MORRILTON

130	AR	PBF	GRIDER FIELD	PINE BLUFF
131	AR	PGR	KIRK FIELD	PARAGOULD
132	AR	ROG	ROGERS MUNICIPAL-CARTER FIELD	ROGERS
133	AR	RUE	RUSSELLVILLE RGNL	RUSSELLVILLE
134	AR	SGT	STUTTGART MUNI	STUTTGART
135	AR	SLG	SMITH FIELD	SILOAM SPRINGS
136	AR	SRC	SEARCY MUNI	SEARCY
137	AR	TXK	TEXARKANA REGIONAL-WEBB FIELD	TEXARKANA
138	AR	XNA	NORTHWEST ARKANSAS RGNL	FAYETTEVILLE/SPRINGDALE/
139	AZ	0V7	KAYENTA	KAYENTA
140	AZ	AVQ	MARANA REGIONAL	TUCSON
141	AZ	AZC	COLORADO CITY MUNI	COLORADO CITY
142	AZ	BXK	BUCKEYE MUNI	BUCKEYE
143	AZ	CFT	GREENLEE COUNTY	CLIFTON/MORENCI
144	AZ	CGZ	CASA GRANDE MUNI	CASA GRANDE
145	AZ	CHD	CHANDLER MUNI	CHANDLER
146	AZ	CMR	H.A. CLARK MEMORIAL FIELD	WILLIAMS
147	AZ	D68	TOWN OF SPRINGERVILLE MUNI	SPRINGERVILLE
148	AZ	DGL	DOUGLAS MUNI	DOUGLAS
149	AZ	DUG	BISBEE DOUGLAS INTL	DOUGLAS BISBEE
150	AZ	DVT	PHOENIX DEER VALLEY	PHOENIX
151	AZ	E24	WHITERIVER	WHITERIVER
152	AZ	E25	WICKENBURG MUNI	WICKENBURG
153	AZ	E63	GILA BEND MUNI	GILA BEND
154	AZ	E77	SAN MANUEL	SAN MANUEL
155	AZ	E95	BENSON MUNICIPAL	BENSON
156	AZ	FFZ	FALCON FLD	MESA
157	AZ	FHU	SIERRA VISTA MUNI-LIBBY AAF	FORT HUACHUCA SIERRA VISTA
158	AZ	FLG	FLAGSTAFF PULLIAM	FLAGSTAFF
159	AZ	GCN	GRAND CANYON NATIONAL PARK	GRAND CANYON
160	AZ	GEU	GLENDALE MUNICIPAL	GLENDALE
161	AZ	GYR	PHOENIX GOODYEAR	GOODYEAR
162	AZ	HII	LAKE HAVASU CITY	LAKE HAVASU CITY

163	AZ	IFP	LAUGHLIN/BULLHEAD INTERNATIONAL	BULLHEAD CITY
164	AZ	IGM	KINGMAN	KINGMAN
165	AZ	INW	WINSLOW-LINDBERGH REGIONAL	WINSLOW
166	AZ	IWA	WILLIAMS GATEWAY	PHOENIX
167	AZ	MZJ	PINAL AIRPARK	MARANA
168	AZ	OLS	NOGALES INTL	NOGALES
169	AZ	P08	COOLIDGE MUNI	COOLIDGE
170	AZ	P13	SAN CARLOS APACHE	GLOBE
171	AZ	P14	HOLBROOK MUNI	HOLBROOK
172	AZ	P20	AVI SUQUILLA	PARKER
173	AZ	P23	SELIGMAN	SELIGMAN
174	AZ	P33	COCHISE COUNTY	WILLCOX
175	AZ	P52	COTTONWOOD	COTTONWOOD
176	AZ	PAN	PAYSON	PAYSON
177	AZ	PGA	PAGE MUNI	PAGE
178	AZ	PHX	PHOENIX SKY HARBOR INTL	PHOENIX
179	AZ	PRC	ERNEST A. LOVE FIELD	PRESCOTT
180	AZ	RQE	WINDOW ROCK	WINDOW ROCK
181	AZ	RYN	RYAN FIELD	TUCSON
182	AZ	SAD	SAFFORD REGIONAL	SAFFORD
183	AZ	SDL	SCOTTSDALE	SCOTTSDALE
184	AZ	SEZ	SEDONA	SEDONA
185	AZ	SJN	ST JOHNS INDUSTRIAL AIR PARK	ST JOHNS
186	AZ	SOW	SHOW LOW REGIONAL	SHOW LOW
187	AZ	T03	TUBA CITY	TUBA CITY
188	AZ	TUS	TUCSON INTL	TUCSON
189	AZ	TYL	TAYLOR	TAYLOR
190	AZ	YUM	YUMA MCAS/YUMA INTL	YUMA
191	CA	0Q5	SHELTER COVE	SHELTER COVE
192	CA	2Q3	YOLO COUNTY-DAVIS/WOODLAND/WINTERS	DAVIS/WOODLAND/WINTERS
193	CA	3O7	HOLLISTER MUNI	HOLLISTER
194	CA	49X	CHEMEHUEVI VALLEY	CHEMEHUEVI VALLEY
195	CA	ACV	ARCATA	ARCATA/EUREKA

196	CA	APC	NAPA COUNTY	NAPA
197	CA	APV	APPLE VALLEY	APPLE VALLEY
198	CA	AUN	AUBURN MUNI	AUBURN
199	CA	BFL	MEADOWS FIELD	BAKERSFIELD
200	CA	BIH	EASTERN SIERRA REGIONAL	BISHOP
201	CA	BLH	BLYTHE	BLYTHE
202	CA	BNG	BANNING MUNI	BANNING
203	CA	BUR	BOB HOPE	BURBANK
204	CA	C80	NEW COALINGA MUNI	COALINGA
205	CA	C83	BYRON	BYRON
206	CA	CCB	CABLE	UPLAND
207	CA	CCR	BUCHANAN FIELD	CONCORD
208	CA	CEC	JACK MC NAMARA FIELD	CRESCENT CITY
209	CA	CIC	CHICO MUNI	CHICO
210	CA	CMA	CAMARILLO	CAMARILLO
211	CA	CNO	CHINO	CHINO
212	CA	CRQ	MC CLELLAN-PALOMAR	CARLSBAD
213	CA	CXL	CALEXICO INTL	CALEXICO
214	CA	DAG	BARSTOW-DAGGETT	DAGGETT
215	CA	DLO	DELANO MUNI	DELANO
216	CA	DVO	GNOSS FIELD	NOVATO
217	CA	E16	SOUTH COUNTY ARPT OF SANTA CLARA COUNTY	SAN MARTIN
218	CA	EED	NEEDLES	NEEDLES
219	CA	EKA	MURRAY FIELD	EUREKA
220	CA	EMT	EL MONTE	EL MONTE
221	CA	F70	FRENCH VALLEY	MURRIETA/TEMECULA
222	CA	FAT	FRESNO YOSEMITE INTERNATIONAL	FRESNO
223	CA	FOT	ROHNERVILLE	FORTUNA
224	CA	HAF	HALF MOON BAY	HALF MOON BAY
225	CA	HHR	JACK NORTHROP FIELD/HAWTHORNE MUNICIPAL	HAWTHORNE
226	CA	HJO	HANFORD MUNI	HANFORD
227	CA	HMT	HEMET-RYAN	HEMET
228	CA	HWD	HAYWARD EXECUTIVE	HAYWARD

229	CA	IPL	IMPERIAL COUNTY	IMPERIAL
230	CA	IYK	INYOKERN	INYOKERN
231	CA	KIC	MESA DEL REY	KING CITY
232	CA	L04	HOLTVILLE	HOLTVILLE
233	CA	L08	BORREGO VALLEY	BORREGO SPRINGS
234	CA	L12	REDLANDS MUNI	REDLANDS
235	CA	L45	BAKERSFIELD MUNICIPAL	BAKERSFIELD
236	CA	L67	RIALTO MUNI /MIRO FLD/	RIALTO
237	CA	LAX	LOS ANGELES INTL	LOS ANGELES
238	CA	LGB	LONG BEACH /DAUGHERTY FIELD/	LONG BEACH
239	CA	LHM	LINCOLN REGIONAL/KARL HARDER FIELD	LINCOLN
240	CA	LPC	LOMPOC	LOMPOC
241	CA	LSN	LOS BANOS MUNI	LOS BANOS
242	CA	LVK	LIVERMORE MUNI	LIVERMORE
243	CA	MAE	MADERA MUNI	MADERA
244	CA	MCC	MC CLELLAN AIRFIELD	SACRAMENTO
245	CA	MCE	MERCED MUNICIPAL/MACREADY FIELD	MERCED
246	CA	MER	CASTLE	ATWATER
247	CA	MHR	SACRAMENTO MATHER	SACRAMENTO
248	CA	MHV	MOJAVE	MOJAVE
249	CA	MIT	SHAFTER-MINTER FIELD	SHAFTER
250	CA	MMH	MAMMOTH YOSEMITE	MAMMOTH LAKES
251	CA	MOD	MODESTO CITY-CO-HARRY SHAM FLD	MODESTO
252	CA	MYF	MONTGOMERY FIELD	SAN DIEGO
253	CA	MYV	YUBA COUNTY	MARYSVILLE
254	CA	O02	NERVINO	BECKWOURTH
255	CA	O05	ROGERS FIELD	CHESTER
256	CA	O16	GARBERVILLE	GARBERVILLE
257	CA	O17	NEVADA COUNTY AIR PARK	GRASS VALLEY
258	CA	O22	COLUMBIA	COLUMBIA
259	CA	O48	LITTLE RIVER	LITTLE RIVER
260	CA	O52	SUTTER COUNTY	YUBA CITY
261	CA	O69	PETALUMA MUNI	PETALUMA

262	CA	O88	RIO VISTA MUNI	RIO VISTA
263	CA	OAK	METROPOLITAN OAKLAND INTL	OAKLAND
264	CA	OAR	MARINA MUNICIPAL	MARINA
265	CA	ONT	ONTARIO INTL	ONTARIO
266	CA	OVE	OROVILLE MUNI	OROVILLE
267	CA	OXR	OXNARD	OXNARD
268	CA	PMD	PALMDALE REGIONAL/USAF PLANT 42	PALMDALE
269	CA	POC	BRACKETT FIELD	LA VERNE
270	CA	PRB	PASO ROBLES MUNI	PASO ROBLES
271	CA	PSP	PALM SPRINGS INTERNATIONAL	PALM SPRINGS
272	CA	PTV	PORTERVILLE MUNI	PORTERVILLE
273	CA	PVF	PLACERVILLE	PLACERVILLE
274	CA	RAL	RIVERSIDE MUNI	RIVERSIDE
275	CA	RBL	RED BLUFF MUNI	RED BLUFF
276	CA	RDD	REDDING MUNI	REDDING
277	CA	RIU	RANCHO MURIETA	RANCHO MURIETA
278	CA	RNM	RAMONA	RAMONA
279	CA	SAC	SACRAMENTO EXECUTIVE	SACRAMENTO
280	CA	SAN	SAN DIEGO INTL	SAN DIEGO
281	CA	SBA	SANTA BARBARA MUNI	SANTA BARBARA
282	CA	SBD	SAN BERNARDINO INTERNATIONAL	SAN BERNARDINO
283	CA	SBP	SAN LUIS COUNTY REGIONAL	SAN LUIS OBISPO
284	CA	SCK	STOCKTON METROPOLITAN	STOCKTON
285	CA	SDM	BROWN FIELD MUNI	SAN DIEGO
286	CA	SEE	GILLESPIE FIELD	SAN DIEGO/EL CAJON
287	CA	SFO	SAN FRANCISCO INTERNATIONAL	SAN FRANCISCO
288	CA	SIY	SISKIYOU COUNTY	MONTAGUE
289	CA	SJC	NORMAN Y. MINETA SAN JOSE INTERNATIONAL	SAN JOSE
290	CA	SMF	SACRAMENTO INTERNATIONAL	SACRAMENTO
291	CA	SMO	SANTA MONICA MUNI	SANTA MONICA
292	CA	SMX	SANTA MARIA PUB/CAPT G ALLAN HANCOCK FLD	SANTA MARIA
293	CA	SNA	JOHN WAYNE AIRPORT-ORANGE COUNTY	SANTA ANA
294	CA	SNS	SALINAS MUNI	SALINAS

295	CA	STS	CHARLES M. SCHULZ - SONOMA COUNTY	SANTA ROSA
296	CA	TCY	TRACY MUNI	TRACY
297	CA	TLR	MEFFORD FIELD	TULARE
298	CA	TNP	TWENTYNINE PALMS	TWENTYNINE PALMS
299	CA	TOA	ZAMPERINI FIELD	TORRANCE
300	CA	TRK	TRUCKEE-TAHOE	TRUCKEE
301	CA	TRM	JACQUELINE COCHRAN REGIONAL	PALM SPRINGS
302	CA	TVL	LAKE TAHOE	SOUTH LAKE TAHOE
303	CA	UKI	UKIAH MUNI	UKIAH
304	CA	VCB	NUT TREE	VACAVILLE
305	CA	VCV	SOUTHERN CALIFORNIA LOGISTICS	VICTORVILLE
306	CA	VIS	VISALIA MUNI	VISALIA
307	CA	VNY	VAN NUYS	VAN NUYS
308	CA	WHP	WHITEMAN	LOS ANGELES
309	CA	WJF	GENERAL WM J FOX AIRFIELD	LANCASTER
310	CA	WLW	WILLOWS-GLENN COUNTY	WILLOWS
311	CA	WVI	WATSONVILLE MUNI	WATSONVILLE
312	CO	0V2	HARRIET ALEXANDER FIELD	SALIDA
313	CO	1V6	FREMONT COUNTY	CANON CITY
314	CO	1V9	BLAKE FIELD	DELTA
315	CO	2V1	STEVENS FIELD	PAGOSA SPRINGS
316	CO	2V2	VANCE BRAND	LONGMONT
317	CO	2V5	WRAY MUNI	WRAY
318	CO	4V0	RANGELY	RANGELY
319	CO	7V1	CENTRAL COLORADO REGIONAL	BUENA VISTA
320	CO	AKO	COLORADO PLAINS REGIONAL	AKRON
321	CO	ALS	SAN LUIS VALLEY REGIONAL/BERGMAN FIELD	ALAMOSA
322	CO	APA	CENTENNIAL	DENVER
323	CO	ASE	ASPEN-PITKIN CO/SARDY FIELD	ASPEN
324	CO	BJC	JEFFCO	DENVER
325	CO	CEZ	CORTEZ MUNI	CORTEZ
326	CO	COS	CITY OF COLORADO SPRINGS MUNI	COLORADO SPRINGS
327	CO	DEN	DENVER INTL	DENVER

328	CO	DRO	DURANGO-LA PLATA COUNTY	DURANGO
329	CO	EGE	EAGLE COUNTY REGIONAL	EAGLE
330	CO	FNL	FORT COLLINS-LOVELAND MUNI	FORT COLLINS/LOVELAND
331	CO	FTG	FRONT RANGE	DENVER
332	CO	GJT	WALKER FIELD	GRAND JUNCTION
333	CO	GUC	GUNNISON-CRESTED BUTTE REGIONAL	GUNNISON
334	CO	GXY	GREELEY-WELD COUNTY	GREELEY
335	CO	HDN	YAMPA VALLEY	HAYDEN
336	CO	HEQ	HOLYOKE	HOLYOKE
337	CO	ITR	KIT CARSON COUNTY	BURLINGTON
338	CO	LAA	LAMAR MUNI	LAMAR
339	CO	LHX	LA JUNTA MUNI	LA JUNTA
340	CO	MTJ	MONTROSE REGIONAL	MONTROSE
341	CO	PUB	PUEBLO MEMORIAL	PUEBLO
342	CO	RIL	GARFIELD COUNTY REGIONAL	RIFLE
343	CO	STK	STERLING MUNI	STERLING
344	CO	TAD	PERRY STOKES	TRINIDAD
345	CO	TEX	TELLURIDE REGIONAL	TELLURIDE
346	CT	4B8	ROBERTSON FIELD	PLAINVILLE
347	CT	BDL	BRADLEY INTL	WINDSOR LOCKS
348	CT	BDR	IGOR I SIKORSKY MEMORIAL	BRIDGEPORT
349	CT	DXR	DANBURY MUNI	DANBURY
350	CT	GON	GROTON-NEW LONDON	GROTON (NEW LONDON)
351	CT	HFD	HARTFORD-BRAINARD	HARTFORD
352	CT	HVN	TWEED-NEW HAVEN	NEW HAVEN
353	CT	IJD	WINDHAM	WILLIMANTIC
354	CT	MMK	MERIDEN MARKHAM MUNI	MERIDEN
355	CT	OXC	WATERBURY-OXFORD	OXFORD
356	DC	DCA	RONALD REAGAN WASHINGTON NATIONAL	WASHINGTON
357	DC	IAD	WASHINGTON DULLES INTERNATIONAL	WASHINGTON
358	DE	GED	SUSSEX COUNTY	GEORGETOWN
359	DE	ILG	NEW CASTLE	WILMINGTON
360	FL	1J0	TRI-COUNTY	BONIFAY

361	FL	24J	SUWANNEE COUNTY	LIVE OAK
362	FL	28J	PALATKA MUNI - LT. KAY LARKIN FIELD	PALATKA
363	FL	2IS	AIRGLADES	CLEWISTON
364	FL	2R4	PETER PRINCE FIELD	MILTON
365	FL	2RR	RIVER RANCH RESORT	RIVER RANCH
366	FL	40J	PERRY-FOLEY	PERRY
367	FL	42J	KEYSTONE AIRPARK	KEYSTONE HEIGHTS
368	FL	55J	FERNANDINA BEACH MUNI	FERNANDINA BEACH
369	FL	AAF	APALACHICOLA MUNI	APALACHICOLA
370	FL	APF	NAPLES MUNI	NAPLES
371	FL	AVO	AVON PARK EXECUTIVE	AVON PARK
372	FL	BCT	BOCA RATON	BOCA RATON
373	FL	BKV	HERNANDO COUNTY	BROOKSVILLE
374	FL	BOW	BARTOW MUNI	BARTOW
375	FL	CEW	BOB SIKES	CRESTVIEW
376	FL	CGC	CRYSTAL RIVER	CRYSTAL RIVER
377	FL	CHN	WAUCHULA MUNI	WAUCHULA
378	FL	CLW	CLEARWATER AIR PARK	CLEARWATER
379	FL	COI	MERRITT ISLAND	MERRITT ISLAND
380	FL	CRG	CRAIG MUNI	JACKSONVILLE
381	FL	CTY	CROSS CITY	CROSS CITY
382	FL	DAB	DAYTONA BEACH INTL	DAYTONA BEACH
383	FL	DED	DELAND MUNI-SIDNEY H TAYLOR FIELD	DELAND
384	FL	DTS	DESTIN-FORT WALTON BEACH	DESTIN
385	FL	EVB	NEW SMYRNA BEACH MUNI	NEW SMYRNA BEACH
386	FL	EYW	KEY WEST INTL	KEY WEST
387	FL	F45	NORTH PALM BEACH COUNTY GENERAL AVIATION	WEST PALM BEACH
388	FL	FLL	FORT LAUDERDALE/HOLLYWOOD INTL	FORT LAUDERDALE
389	FL	FMY	PAGE FIELD	FORT MYERS
390	FL	FPR	ST LUCIE COUNTY INTL	FORT PIERCE
391	FL	FXE	FORT LAUDERDALE EXECUTIVE	FORT LAUDERDALE
392	FL	GIF	WINTER HAVEN'S GILBERT	WINTER HAVEN
393	FL	GNV	GAINESVILLE RGNL	GAINESVILLE

394	FL	HEG	HERLONG	JACKSONVILLE
395	FL	HWO	NORTH PERRY	HOLLYWOOD
396	FL	IMM	IMMOKALEE	IMMOKALEE
397	FL	ISM	KISSIMMEE GATEWAY	ORLANDO
398	FL	JAX	JACKSONVILLE INTL	JACKSONVILLE
399	FL	LAL	LAKELAND LINDER RGNL	LAKELAND
400	FL	LCQ	LAKE CITY MUNI	LAKE CITY
401	FL	LEE	LEESBURG REGIONAL	LEESBURG
402	FL	LNA	PALM BEACH COUNTY PARK	WEST PALM BEACH
403	FL	MAI	MARIANNA MUNI	MARIANNA
404	FL	MCO	ORLANDO INTL	ORLANDO
405	FL	MIA	MIAMI INTL	MIAMI
406	FL	MKY	MARCO ISLAND	MARCO ISLAND
407	FL	MLB	MELBOURNE INTL	MELBOURNE
408	FL	MTH	THE FLORIDA KEYS MARATHON	MARATHON
409	FL	OBE	OKEECHOBEE COUNTY	OKEECHOBEE
410	FL	OCF	OCALA INTL-JIM TAYLOR FIELD	OCALA
411	FL	OMN	ORMOND BEACH MUNI	ORMOND BEACH
412	FL	OPF	OPA LOCKA	MIAMI
413	FL	ORL	EXECUTIVE	ORLANDO
414	FL	PBI	PALM BEACH INTL	WEST PALM BEACH
415	FL	PCM	PLANT CITY MUNI	PLANT CITY
416	FL	PFN	PANAMA CITY-BAY CO INTL	PANAMA CITY
417	FL	PGD	CHARLOTTE COUNTY	PUNTA GORDA
418	FL	PHK	PALM BEACH CO GLADES	PAHOKEE
419	FL	PIE	ST PETERSBURG-CLEARWATER INTL	ST PETERSBURG-CLEARWATER
420	FL	PMP	POMPANO BEACH AIRPARK	POMPANO BEACH
421	FL	PNS	PENSACOLA REGIONAL	PENSACOLA
422	FL	RSW	SOUTHWEST FLORIDA INTL	FORT MYERS
423	FL	SEF	SEBRING REGIONAL	SEBRING
424	FL	SFB	ORLANDO SANFORD INTL	ORLANDO
425	FL	SGJ	ST AUGUSTINE	ST AUGUSTINE
426	FL	SPG	ALBERT WHITTED	ST PETERSBURG

427	FL	SRQ	SARASOTA/BRADENTON INTL	SARASOTA/BRADENTON
428	FL	SUA	WITHAM FIELD	STUART
429	FL	TIX	SPACE COAST REGIONAL	TITUSVILLE
430	FL	TLH	TALLAHASSEE REGIONAL	TALLAHASSEE
431	FL	TMB	KENDALL-TAMIAMI EXECUTIVE	MIAMI
432	FL	TNT	DADE-COLLIER TRAINING AND TRANSITION	MIAMI
433	FL	TPA	TAMPA INTL	TAMPA
434	FL	TPF	PETER O KNIGHT	TAMPA
435	FL	VDF	VANDENBERG	TAMPA
436	FL	VNC	VENICE MUNI	VENICE
437	FL	VPS	EGLIN AFB	VALPARAISO
438	FL	VQQ	CECIL FIELD	JACKSONVILLE
439	FL	VRB	VERO BEACH MUNI	VERO BEACH
440	FL	X05	PILOT COUNTRY	BROOKSVILLE
441	FL	X06	ARCADIA MUNI	ARCADIA
442	FL	X07	LAKE WALES MUNI	LAKE WALES
443	FL	X13	CARRABELLE-THOMPSON	CARRABELLE
444	FL	X26	SEBASTIAN MUNI	SEBASTIAN
445	FL	X35	DUNNELLON/MARION CO & PARK OF COMMERCE	DUNNELLON
446	FL	X47	FLAGLER COUNTY	BUNNELL
447	FL	X51	HOMESTEAD GENERAL AVIATION	HOMESTEAD
448	FL	X59	VALKARIA	VALKARIA
449	FL	X60	WILLISTON MUNI	WILLISTON
450	FL	ZPH	ZEPHYRHILLS MUNI	ZEPHYRHILLS
451	GA	09J	JEKYLL ISLAND	JEKYLL ISLAND
452	GA	15J	COOK COUNTY	ADEL
453	GA	16J	DAWSON MUNI	DAWSON
454	GA	17J	DONALSONVILLE MUNI	DONALSONVILLE
455	GA	18A	FRANKLIN COUNTY	CANON
456	GA	19A	JACKSON COUNTY	JEFFERSON
457	GA	27A	ELBERT COUNTY-PATZ FIELD	ELBERTON
458	GA	2J2	LIBERTY COUNTY	HINESVILLE
459	GA	2J3	LOUISVILLE MUNI	LOUISVILLE

460	GA	3J7	GREENE COUNTY REGIONAL	GREENSBORO
461	GA	46A	BLAIRSVILLE	BLAIRSVILLE
462	GA	47A	CHEROKEE COUNTY	CANTON
463	GA	4A4	POLK COUNTY AIRPORT-CORNELIUS MOORE FIELD	CEDARTOWN
464	GA	4A7	CLAYTON COUNTY - TARA FIELD	HAMPTON
465	GA	4J2	BERRIEN CO	NASHVILLE
466	GA	4J5	QUITMAN BROOKS COUNTY	QUITMAN
467	GA	4J6	ST MARYS	ST MARYS
468	GA	53A	DR. C P SAVAGE SR.	MONTEZUMA
469	GA	5A9	ROOSEVELT MEMORIAL	WARM SPRINGS
470	GA	6A2	GRIFFIN-SPALDING COUNTY	GRIFFIN
471	GA	70J	CAIRO-GRADY COUNTY	CAIRO
472	GA	82A	MARION COUNTY	BUENA VISTA
473	GA	9A1	COVINGTON MUNI	COVINGTON
474	GA	ABY	SOUTHWEST GEORGIA REGIONAL	ALBANY
475	GA	ACJ	SOUTHER FIELD	AMERICUS
476	GA	AGS	AUGUSTA RGNL AT BUSH FIELD	AUGUSTA
477	GA	AHN	ATHENS/BEN EPPS	ATHENS
478	GA	AJR	HABERSHAM COUNTY	CORNELIA
479	GA	AMG	BACON COUNTY	ALMA
480	GA	ATL	HARTSFIELD - JACKSON ATLANTA INTL	ATLANTA
481	GA	AYS	WAYCROSS-WARE COUNTY	WAYCROSS
482	GA	AZE	HAZLEHURST	HAZLEHURST
483	GA	BGE	DECATUR COUNTY INDUSTRIAL AIR PARK	BAINBRIDGE
484	GA	BHC	BAXLEY MUNI	BAXLEY
485	GA	BIJ	EARLY COUNTY	BLAKELY
486	GA	BQK	BRUNSWICK GOLDEN ISLES	BRUNSWICK
487	GA	BXG	BURKE COUNTY	WAYNESBORO
488	GA	CCO	NEWNAN COWETA COUNTY	NEWNAN
489	GA	CKF	CRISP COUNTY-CORDELE	CORDELE
490	GA	CSG	COLUMBUS METROPOLITAN	COLUMBUS
491	GA	CTJ	WEST GEORGIA RGNL - O V GRAY FIELD	CARROLLTON
492	GA	CWV	CLAXTON-EVANS COUNTY	CLAXTON

493	GA	CZL	TOM B. DAVID FLD	CALHOUN
494	GA	DBN	W H 'BUD' BARRON	DUBLIN
495	GA	DNL	DANIEL FIELD	AUGUSTA
496	GA	DNN	DALTON MUNI	DALTON
497	GA	DQH	DOUGLAS MUNI	DOUGLAS
498	GA	EZM	HEART OF GEORGIA RGNL	EASTMAN
499	GA	FFC	PEACHTREE CITY-FALCON FIELD	ATLANTA
500	GA	FTY	FULTON COUNTY AIRPORT-BROWN FIELD	ATLANTA
501	GA	FZG	FITZGERALD MUNI	FITZGERALD
502	GA	GVL	LEE GILMER MEMORIAL	GAINESVILLE
503	GA	HOE	HOMERVILLE	HOMERVILLE
504	GA	HQU	THOMSON-MCDUFFIE COUNTY	THOMSON
505	GA	IIY	WASHINGTON-WILKES COUNTY	WASHINGTON
506	GA	JES	JESUP-WAYNE COUNTY	JESUP
507	GA	JYL	PLANTATION ARPK	SYLVANIA
508	GA	JZP	PICKENS COUNTY	JASPER
509	GA	LGC	LAGRANGE-CALLAWAY	LAGRANGE
510	GA	LZU	GWINNETT COUNTY - BRISCOE FIELD	LAWRENCEVILLE
511	GA	MAC	MACON DOWNTOWN	MACON
512	GA	MCN	MIDDLE GEORGIA REGIONAL	MACON
513	GA	MGR	MOULTRIE MUNI	MOULTRIE
514	GA	MHP	METTER MUNI	METTER
515	GA	MLJ	BALDWIN COUNTY	MILLEDGEVILLE
516	GA	MQW	TELFAIR-WHEELER	MC RAE
517	GA	MUL	SPENCE	MOULTRIE
518	GA	OKZ	KAOLIN FIELD	SANDERSVILLE
519	GA	OPN	THOMASTON-UPSON COUNTY	THOMASTON
520	GA	PDK	DEKALB-PEACHTREE	ATLANTA
521	GA	PIM	CALLAWAY GARDENS-HARRIS COUNTY	PINE MOUNTAIN
522	GA	PXE	PERRY-HOUSTON COUNTY	PERRY
523	GA	RMG	RICHARD B RUSSELL	ROME
524	GA	RVJ	SWINTON SMITH FLD AT REIDSVILLE MUNICIPAL	REIDSVILLE
525	GA	RYY	COBB COUNTY-MC COLLUM FIELD	ATLANTA

526	GA	SAV	SAVANNAH/HILTON HEAD INTERNATIONAL	SAVANNAH
527	GA	SBO	EMANUEL COUNTY	SWAINSBORO
528	GA	SSI	MALCOLM MC KINNON	BRUNSWICK
529	GA	SYV	SYLVESTER	SYLVESTER
530	GA	TBR	STATESBORO-BULLOCH COUNTY	STATESBORO
531	GA	TMA	HENRY TIFT MYERS	TIFTON
532	GA	TOC	TOCCOA RG LETOURNEAU FIELD	TOCCOA
533	GA	TVI	THOMASVILLE REGIONAL	THOMASVILLE
534	GA	VDI	VIDALIA REGIONAL	VIDALIA
535	GA	VLD	VALDOSTA RGNL	VALDOSTA
536	GA	VPC	CARTERSVILLE	CARTERSVILLE
537	GA	WDR	WINDER-BARROW	WINDER
538	IA	AIO	ATLANTIC MUNI	ATLANTIC
539	IA	ALO	WATERLOO REGIONAL	WATERLOO
540	IA	AMW	AMES MUNI	AMES
541	IA	AWG	WASHINGTON MUNI	WASHINGTON
542	IA	AXA	ALGONA MUNI	ALGONA
543	IA	BNW	BOONE MUNI	BOONE
544	IA	BRL	SOUTHEAST IOWA REGIONAL	BURLINGTON
545	IA	CBF	COUNCIL BLUFFS MUNI	COUNCIL BLUFFS
546	IA	CCY	CHARLES CITY MUNI	CHARLES CITY
547	IA	CID	THE EASTERN IOWA	CEDAR RAPIDS
548	IA	CIN	ARTHUR N NEU	CARROLL
549	IA	CKP	CHEROKEE MUNI	CHEROKEE
550	IA	CNC	CHARITON MUNI	CHARITON
551	IA	CSQ	CRESTON MUNI	CRESTON
552	IA	CWI	CLINTON MUNI	CLINTON
553	IA	DBQ	DUBUQUE REGIONAL	DUBUQUE
554	IA	DEH	DECORAH MUNI	DECORAH
555	IA	DNS	DENISON MUNI	DENISON
556	IA	DSM	DES MOINES INTL	DES MOINES
557	IA	DVN	DAVENPORT MUNI	DAVENPORT
558	IA	EBS	WEBSTER CITY MUNI	WEBSTER CITY

559	IA	EFW	JEFFERSON MUNI	JEFFERSON
560	IA	EOK	KEOKUK MUNI	KEOKUK
561	IA	EST	ESTHERVILLE MUNI	ESTHERVILLE
562	IA	FFL	FAIRFIELD MUNI	FAIRFIELD
563	IA	FOD	FORT DODGE REGIONAL	FORT DODGE
564	IA	FSW	FORT MADISON MUNI	FORT MADISON
565	IA	FXY	FOREST CITY MUNI	FOREST CITY
566	IA	GGI	GRINNELL REGIONAL	GRINNELL
567	IA	HNR	HARLAN MUNI	HARLAN
568	IA	HPT	HAMPTON MUNI	HAMPTON
569	IA	I75	OSCEOLA MUNICIPAL	OSCEOLA
570	IA	ICL	SCHENCK FIELD	CLARINDA
571	IA	IFA	IOWA FALLS MUNI	IOWA FALLS
572	IA	IIB	INDEPENDENCE MUNI	INDEPENDENCE
573	IA	IKV	ANKENY REGIONAL	ANKENY
574	IA	IOW	IOWA CITY MUNI	IOWA CITY
575	IA	LRJ	LE MARS MUNI	LE MARS
576	IA	MCW	MASON CITY MUNI	MASON CITY
577	IA	MIW	MARSHALLTOWN MUNI	MARSHALLTOWN
578	IA	MPZ	MOUNT PLEASANT MUNI	MOUNT PLEASANT
579	IA	MUT	MUSCATINE MUNI	MUSCATINE
580	IA	MXO	MONTICELLO REGIONAL	MONTICELLO
581	IA	OLZ	OELWEIN MUNI	OELWEIN
582	IA	OOA	OSKALOOSA MUNI	OSKALOOSA
583	IA	OTM	OTTUMWA INDUSTRIAL	OTTUMWA
584	IA	OXV	KNOXVILLE MUNI	KNOXVILLE
585	IA	PEA	PELLA MUNI	PELLA
586	IA	PRO	PERRY MUNI	PERRY
587	IA	RDK	RED OAK MUNI	RED OAK
588	IA	SDA	SHENANDOAH MUNI	SHENANDOAH
589	IA	SHL	SHELDON MUNI	SHELDON
590	IA	SKI	SAC CITY MUNI	SAC CITY
591	IA	SLB	STORM LAKE MUNI	STORM LAKE

592	IA	SPW	SPENCER MUNI	SPENCER
593	IA	SUX	SIOUX GATEWAY/COL. BUD DAY FIELD	SIOUX CITY
594	IA	TNU	NEWTON MUNI	NEWTON
595	IA	TVK	CENTERVILLE MUNI	CENTERVILLE
596	IA	TZT	BELLE PLAINE MUNI	BELLE PLAINE
597	ID	1U7	BEAR LAKE COUNTY	PARIS
598	ID	65S	BOUNDARY COUNTY	BONNERS FERRY
599	ID	AOC	ARCO-BUTTE COUNTY	ARCO
600	ID	BOI	BOISE AIR TERMINAL/GOWEN FLD	BOISE
601	ID	COE	COEUR D'ALENE AIR TERM	COEUR D'ALENE
602	ID	DIJ	DRIGGS-REED MEML	DRIGGS
603	ID	EUL	CALDWELL INDUSTRIAL	CALDWELL
604	ID	GNG	GOODING MUNI	GOODING
605	ID	IDA	IDAHO FALLS RGNL	IDAHO FALLS
606	ID	JER	JEROME COUNTY	JEROME
607	ID	LWS	LEWISTON-NEZ PERCE COUNTY	LEWISTON
608	ID	MYL	MC CALL MUNICIPAL	MC CALL
609	ID	PIH	POCATELLO REGIONAL	POCATELLO
610	ID	S67	NAMPA MUNI	NAMPA
611	ID	S80	IDAHO COUNTY	GRANGEVILLE
612	ID	S83	SHOSHONE COUNTY	KELLOGG
613	ID	SUN	FRIEDMAN MEMORIAL	HAILEY
614	ID	SZT	SANDPOINT	SANDPOINT
615	ID	TWF	JOSLIN FIELD - MAGIC VALLEY RGNL	TWIN FALLS
616	ID	U76	MOUNTAIN HOME MUNI	MOUNTAIN HOME
617	IL	06C	SCHAUMBURG REGIONAL	CHICAGO/SCHAUMBURG
618	IL	1H2	EFFINGHAM COUNTY MEMORIAL	EFFINGHAM
619	IL	1H8	CASEY MUNI	CASEY
620	IL	2H0	SHELBY COUNTY	SHELBYVILLE
621	IL	3LF	LITCHFIELD MUNI	LITCHFIELD
622	IL	AAA	LOGAN COUNTY	LINCOLN
623	IL	AJG	MOUNT CARMEL MUNI	MOUNT CARMEL
624	IL	ALN	ST LOUIS REGIONAL	ALTON/ST LOUIS

625	IL	ARR	AURORA MUNI	CHICAGO/AURORA
626	IL	BLV	SCOTT AFB/MIDAMERICA	BELLEVILLE
627	IL	BMI	CENTRAL IL REGL ARPT AT BLOOMINGTON-NORMAL	BLOOMINGTON/NORMAL
628	IL	C09	MORRIS MUNI - JAMES R. WASHBURN FIELD	MORRIS
629	IL	C15	PEKIN MUNI	PEKIN
630	IL	C73	DIXON MUNI-CHARLES R. WALGREEN FIELD	DIXON
631	IL	C75	MARSHALL COUNTY	LACON
632	IL	CIR	CAIRO RGNL	CAIRO
633	IL	CMI	UNIVERSITY OF ILLINOIS-WILLARD	CHAMPAIGN/URBANA
634	IL	CPS	ST LOUIS DOWNTOWN	CAHOKIA/ST LOUIS
635	IL	CTK	INGERSOLL	CANTON
636	IL	CUL	CARMI MUNI	CARMI
637	IL	DEC	DECATUR	DECATUR
638	IL	DKB	DE KALB TAYLOR MUNI	DE KALB
639	IL	DNV	VERMILION COUNTY	DANVILLE
640	IL	DPA	DUPAGE	CHICAGO/WEST CHICAGO
641	IL	ENL	CENTRALIA MUNI	CENTRALIA
642	IL	EZI	KEWANEE MUNI	KEWANEE
643	IL	FEP	ALBERTUS	FREEPORT
644	IL	FOA	FLORA MUNI	FLORA
645	IL	FWC	FAIRFIELD MUNI	FAIRFIELD
646	IL	GBG	GALESBURG MUNI	GALESBURG
647	IL	GRE	GREENVILLE	GREENVILLE
648	IL	H96	BENTON MUNI	BENTON
649	IL	HSB	HARRISBURG-RALEIGH	HARRISBURG
650	IL	I63	MOUNT STERLING MUNICIPAL	MOUNT STERLING
651	IL	IJX	JACKSONVILLE MUNI	JACKSONVILLE
652	IL	IKK	GREATER KANKAKEE	KANKAKEE
653	IL	LOT	LEWIS UNIVERSITY	CHICAGO/ROMEOVILLE
654	IL	LWV	LAWRENCEVILLE-VINCENNES INTL	LAWRENCEVILLE
655	IL	M30	METROPOLIS MUNI	METROPOLIS
656	IL	MDH	SOUTHERN ILLINOIS	CARBONDALE/MURPHYSBORO
657	IL	MDW	CHICAGO MIDWAY INTL	CHICAGO

658	IL	MLI	QUAD CITY INTL	MOLINE
659	IL	MQB	MACOMB MUNI	MACOMB
660	IL	MTO	COLES COUNTY MEMORIAL	MATTOON/CHARLESTON
661	IL	MVN	MOUNT VERNON	MOUNT VERNON
662	IL	MWA	WILLIAMSON COUNTY REGIONAL	MARION
663	IL	OLY	OLNEY-NOBLE	OLNEY-NOBLE
664	IL	ORD	CHICAGO O'HARE INTL	CHICAGO
665	IL	PIA	GREATER PEORIA REGIONAL	PEORIA
666	IL	PNT	PONTIAC MUNICIPAL	PONTIAC
667	IL	PRG	EDGAR COUNTY	PARIS
668	IL	PWK	PALWAUKEE MUNI	CHICAGO/PROSPECT HEIGHTS/WHEELING
669	IL	RFD	CHICAGO/ROCKFORD INTERNATIONAL	ROCKFORD
670	IL	RPJ	ROCHELLE MUNI AIRPORT-KORITZ FIELD	ROCHELLE
671	IL	RSV	ROBINSON MUNI	ROBINSON
672	IL	SLO	SALEM-LECKRONE	SALEM
673	IL	SPI	ABRAHAM LINCOLN CAPITAL	SPRINGFIELD
674	IL	SQI	WHITESIDE CO ARPT-JOS H BITTORF FLD	STERLING/ROCKFALLS
675	IL	TAZ	TAYLORVILLE MUNI	TAYLORVILLE
676	IL	TIP	RANTOUL NATL AVN CNTR-FRANK ELLIOTT FLD	RANTOUL
677	IL	UGN	WAUKEGAN REGIONAL	CHICAGO/WAUKEGAN
678	IL	UIN	QUINCY REGIONAL-BALDWIN FIELD	QUINCY
679	IL	VLA	VANDALIA MUNI	VANDALIA
680	IL	VYS	ILLINOIS VALLEY RGNL-WALTER A DUNCAN FIELD	PERU
681	IN	05C	GRIFFITH-MERRILLVILLE	GRIFFITH
682	IN	2R2	HENDRICKS COUNTY-GORDON GRAHAM FLD	INDIANAPOLIS
683	IN	4I7	PUTNAM COUNTY	GREENCASTLE
684	IN	AID	ANDERSON MUNICIPAL-DARLINGTON FIELD	ANDERSON
685	IN	ANQ	TRI-STATE STEUBEN COUNTY	ANGOLA
686	IN	ASW	WARSAW MUNI	WARSAW
687	IN	BAK	COLUMBUS MUNI	COLUMBUS
688	IN	BFR	VIRGIL I GRISSOM MUNI	BEDFORD
689	IN	BMG	MONROE COUNTY	BLOOMINGTON
690	IN	C62	KENDALLVILLE MUNI	KENDALLVILLE

691	IN	CEV	METTEL FIELD	CONNERSVILLE
692	IN	CFJ	CRAWFORDSVILLE MUNI	CRAWFORDSVILLE
693	IN	DCY	DAVIESS COUNTY	WASHINGTON
694	IN	EKM	ELKHART MUNI	ELKHART
695	IN	EVV	EVANSVILLE REGIONAL	EVANSVILLE
696	IN	EYE	EAGLE CREEK AIRPARK	INDIANAPOLIS
697	IN	FKR	FRANKFORT MUNI	FRANKFORT
698	IN	FRH	FRENCH LICK MUNI	FRENCH LICK
699	IN	FWA	FORT WAYNE INTERNATIONAL	FORT WAYNE
700	IN	GEZ	SHELBYVILLE MUNI	SHELBYVILLE
701	IN	GGP	LOGANSPORT/CASS COUNTY	LOGANSPORT
702	IN	GSH	GOSHEN MUNI	GOSHEN
703	IN	GWB	DE KALB COUNTY	AUBURN
704	IN	GYY	GARY/CHICAGO INTERNATIONAL	GARY
705	IN	HFY	GREENWOOD MUNI	INDIANAPOLIS
706	IN	HHG	HUNTINGTON MUNI	HUNTINGTON
707	IN	HNB	HUNTINGBURG	HUNTINGBURG
708	IN	HUF	TERRE HAUTE INTERNATIONAL-HULMAN FIELD	TERRE HAUTE
709	IN	IMS	MADISON MUNI	MADISON
710	IN	IND	INDIANAPOLIS INTL	INDIANAPOLIS
711	IN	IWH	WABASH MUNI	WABASH
712	IN	JVY	CLARK REGIONAL	JEFFERSONVILLE
713	IN	LAF	PURDUE UNIVERSITY	LAFAYETTE
714	IN	MGC	MICHIGAN CITY MUNI	MICHIGAN CITY
715	IN	MIE	DELAWARE COUNTY - JOHNSON FIELD	MUNCIE
716	IN	MQJ	MOUNT COMFORT	INDIANAPOLIS
717	IN	MZZ	MARION MUNI	MARION
718	IN	OKK	KOKOMO MUNI	KOKOMO
719	IN	OVO	NORTH VERNON	NORTH VERNON
720	IN	OXI	STARKE COUNTY	KNOX
721	IN	PLD	PORTLAND MUNI	PORTLAND
722	IN	PPO	LA PORTE MUNI	LA PORTE
723	IN	RID	RICHMOND MUNI	RICHMOND

724	IN	RWN	ARENS FIELD	WINAMAC
725	IN	SBN	SOUTH BEND REGIONAL	SOUTH BEND
726	IN	SER	FREEMAN MUNI	SEYMOUR
727	IN	SIV	SULLIVAN COUNTY	SULLIVAN
728	IN	SMD	SMITH FIELD	FORT WAYNE
729	IN	TEL	PERRY COUNTY MUNI	TELL CITY
730	IN	TYQ	INDIANAPOLIS EXECUTIVE	INDIANAPOLIS
731	IN	UMP	INDIANAPOLIS METROPOLITAN	INDIANAPOLIS
732	IN	VPZ	PORTER COUNTY MUNI	VALPARAISO
733	KS	3JC	FREEMAN FIELD	JUNCTION CITY
734	KS	9K8	KINGMAN AIRPORT - CLYDE CESSNA FIELD	KINGMAN
735	KS	AAO	COLONEL JAMES JABARA	WICHITA
736	KS	ADT	ATWOOD-RAWLINS COUNTY CITY-COUNTY	ATWOOD
737	KS	BEC	BEECH FACTORY	WICHITA
738	KS	CBK	SHALZ FIELD	COLBY
739	KS	CFV	COFFEYVILLE MUNI	COFFEYVILLE
740	KS	CNU	CHANUTE MARTIN JOHNSON	CHANUTE
741	KS	CYW	CLAY CENTER MUNI	CLAY CENTER
742	KS	DDC	DODGE CITY REGIONAL	DODGE CITY
743	KS	EGT	WELLINGTON MUNI	WELLINGTON
744	KS	EMP	EMPORIA MUNI	EMPORIA
745	KS	EQA	CAPTAIN JACK THOMAS/EL DORADO	EL DORADO
746	KS	EWK	NEWTON-CITY-COUNTY	NEWTON
747	KS	FLV	SHERMAN AAF	FORT LEAVENWORTH
748	KS	FOE	FORBES FIELD	TOPEKA
749	KS	FSK	FORT SCOTT MUNI	FORT SCOTT
750	KS	GBD	GREAT BEND MUNI	GREAT BEND
751	KS	GCK	GARDEN CITY REGIONAL	GARDEN CITY
752	KS	GLD	RENNER FLD /GOODLAND MUNI/	GOODLAND
753	KS	HLC	HILL CITY MUNI	HILL CITY
754	KS	HQG	HUGOTON MUNI	HUGOTON
755	KS	HRU	HERINGTON REGIONAL	HERINGTON
756	KS	HUT	HUTCHINSON MUNI	HUTCHINSON

757	KS	HYS	HAYS RGNL	HAYS
758	KS	ICT	WICHITA MID-CONTINENT	WICHITA
759	KS	IDP	INDEPENDENCE MUNI	INDEPENDENCE
760	KS	IXD	NEW CENTURY AIRCENTER	OLATHE
761	KS	K78	ABILENE MUNI	ABILENE
762	KS	K79	JETMORE MUNI	JETMORE
763	KS	K88	ALLEN COUNTY	IOLA
764	KS	LBL	LIBERAL MUNI	LIBERAL
765	KS	LQR	LARNED-PAWNEE COUNTY	LARNED
766	KS	LWC	LAWRENCE MUNI	LAWRENCE
767	KS	MEJ	MEADE MUNI	MEADE
768	KS	MHK	MANHATTAN RGNL	MANHATTAN
769	KS	MPR	MC PHERSON	MC PHERSON
770	KS	OEL	OAKLEY MUNI	OAKLEY
771	KS	OJC	JOHNSON COUNTY EXECUTIVE	OLATHE
772	KS	OWI	OTTAWA MUNI	OTTAWA
773	KS	PPF	TRI-CITY	PARSONS
774	KS	PTS	ATKINSON MUNI	PITTSBURG
775	KS	PTT	PRATT INDUSTRIAL	PRATT
776	KS	RSL	RUSSELL MUNI	RUSSELL
777	KS	SLN	SALINA MUNI	SALINA
778	KS	TOP	PHILIP BILLARD MUNI	TOPEKA
779	KS	UKL	COFFEY COUNTY	BURLINGTON
780	KS	ULS	ULYSSES	ULYSSES
781	KS	WLD	STROTHER FIELD	WINFIELD/ARKANSAS CITY
782	KY	0I8	CYNTHIANA-HARRISON COUNTY	CYNTHIANA
783	KY	1A6	MIDDLESBORO-BELL COUNTY	MIDDLESBORO
784	KY	1M9	LAKE BARKLEY STATE PARK	CADIZ
785	KY	27K	GEORGETOWN SCOTT COUNTY - MARSHALL FLD	GEORGETOWN
786	KY	2I0	MADISONVILLE MUNI	MADISONVILLE
787	KY	2I3	ROUGH RIVER STATE PARK	FALLS-OF-ROUGH
788	KY	2M0	PRINCETON-CALDWELL COUNTY	PRINCETON
789	KY	3I6	PAINTSVILLE-PRESTONSBURG-COMBS FIELD	PAINTSVILLE

790	KY	4M7	RUSSELLVILLE-LOGAN COUNTY	RUSSELLVILLE
791	KY	5M9	MARION-CRITTENDEN COUNTY	MARION
792	KY	6I2	LEBANON-SPRINGFIELD	SPRINGFIELD
793	KY	AAS	TAYLOR COUNTY	CAMPBELLSVILLE
794	KY	BRY	SAMUELS FIELD	BARDSTOWN
795	KY	BWG	BOWLING GREEN-WARREN COUNTY RGNL	BOWLING GREEN
796	KY	CEY	KYLE-OAKLEY FIELD	MURRAY
797	KY	CVG	CINCINNATI/NORTHERN KENTUCKY INTERNATIONAL	COVINGTON
798	KY	DVK	STUART POWELL FIELD	DANVILLE
799	KY	DWU	ASHLAND-BOYD COUNTY	ASHLAND
800	KY	EHR	HENDERSON CITY-COUNTY	HENDERSON
801	KY	EKQ	WAYNE COUNTY	MONTICELLO
802	KY	EKX	ADDINGTON FIELD	ELIZABETHTOWN
803	KY	FFT	CAPITAL CITY	FRANKFORT
804	KY	FGX	FLEMING-MASON	FLEMINGSBURG
805	KY	GLW	GLASGOW MUNI	GLASGOW
806	KY	HVC	HOPKINSVILLE-CHRISTIAN COUNTY	HOPKINSVILLE
807	KY	I35	TUCKER-GUTHRIE MEMORIAL	HARLAN
808	KY	I39	MADISON	RICHMOND
809	KY	I93	BRECKINRIDGE COUNTY	HARDINSBURG
810	KY	IOB	MOUNT STERLING-MONTGOMERY COUNTY	MOUNT STERLING
811	KY	JKL	JULIAN CARROLL	JACKSON
812	KY	K20	WENDELL H FORD	HAZARD
813	KY	K22	BIG SANDY REGIONAL	PRESTONSBURG
814	KY	K24	RUSSELL COUNTY	JAMESTOWN
815	KY	K62	GENE SNYDER	FALMOUTH
816	KY	LEX	BLUE GRASS	LEXINGTON
817	KY	LOU	BOWMAN FIELD	LOUISVILLE
818	KY	LOZ	LONDON-CORBIN ARPT-MAGEE FLD	LONDON
819	KY	M21	MUHLENBERG COUNTY	GREENVILLE
820	KY	M25	MAYFIELD GRAVES COUNTY	MAYFIELD
821	KY	M34	KENTUCKY DAM STATE PARK	GILBERTSVILLE
822	KY	OWB	OWENSBORO-DAVIESS COUNTY	OWENSBORO

823	KY	PAH	BARKLEY REGIONAL	PADUCAH
824	KY	PBX	PIKE COUNTY-HATCHER FIELD	PIKEVILLE
825	KY	SDF	LOUISVILLE INTL-STANDIFORD FIELD	LOUISVILLE
826	KY	SME	SOMERSET-PULASKI COUNTY-J.T. WILSON FIELD	SOMERSET
827	KY	TWT	STURGIS MUNI	STURGIS
828	KY	TZV	TOMPKINSVILLE-MONROE COUNTY	TOMPKINSVILLE
829	KY	W38	WILLIAMSBURG-WHITLEY COUNTY	WILLIAMSBURG
830	LA	0M8	BYERLEY	LAKE PROVIDENCE
831	LA	0M9	DELHI MUNI	DELHI
832	LA	0R3	ABBEVILLE CHRIS CRUSTA MEMORIAL	ABBEVILLE
833	LA	0R4	CONCORDIA PARISH	VIDALIA
834	LA	0R5	DAVID G JOYCE	WINNFIELD
835	LA	0R7	THE RED RIVER	COUSHATTA
836	LA	1L0	ST JOHN THE BAPTIST PARISH	RESERVE
837	LA	1R1	JENA	JENA
838	LA	1R4	WOODWORTH	WOODWORTH
839	LA	2F8	MOREHOUSE MEMORIAL	BASTROP
840	LA	2L0	PINEVILLE MUNI	PINEVILLE
841	LA	2R1	LE MAIRE MEML	JEANERETTE
842	LA	2R6	BUNKIE MUNICIPAL	BUNKIE
843	LA	2R7	FRANKLINTON	FRANKLINTON
844	LA	3F3	C E 'RUSTY' WILLIAMS	MANSFIELD
845	LA	3R2	LE GROS MEMORIAL	CROWLEY
846	LA	3R4	HART	MANY
847	LA	3R7	JENNINGS	JENNINGS
848	LA	4R7	EUNICE	EUNICE
849	LA	5F0	ARCADIA-BIENVILLE PARISH	ARCADIA
850	LA	5F3	HAYNESVILLE	HAYNESVILLE
851	LA	5R8	DE QUINCY INDUSTRIAL AIRPARK	DE QUINCY
852	LA	ACP	ALLEN PARISH	OAKDALE
853	LA	AEX	ALEXANDRIA INTL	ALEXANDRIA
854	LA	ARA	ACADIANA REGIONAL	NEW IBERIA
855	LA	ASD	SLIDELL	SLIDELL

856	LA	BTR	BATON ROUGE METROPOLITAN - RYAN FIELD	BATON ROUGE
857	LA	BXA	GEORGE R CARR MEMORIAL AIR FLD	BOGALUSA
858	LA	CWF	CHENNAULT INTL	LAKE CHARLES
859	LA	DRI	BEAUREGARD REGIONAL	DE RIDDER
860	LA	DTN	SHREVEPORT DOWNTOWN	SHREVEPORT
861	LA	ESF	ESLER REGIONAL	ALEXANDRIA
862	LA	F24	MINDEN-WEBSTER	MINDEN
863	LA	F86	COLUMBIA	COLUMBIA
864	LA	F88	JONESBORO	JONESBORO
865	LA	F89	WINNSBORO MUNICIPAL	WINNSBORO
866	LA	GAO	SOUTH LAFOURCHE	GALLIANO
867	LA	HDC	HAMMOND NORTHSHORE REGIONAL	HAMMOND
868	LA	HUM	HOUMA-TERREBONNE	HOUMA
869	LA	HZR	FALSE RIVER RGNL	NEW ROADS
870	LA	IER	NATCHITOCHES REGIONAL	NATCHITOCHES
871	LA	L32	JONESVILLE	JONESVILLE
872	LA	L33	TENSAS PARISH	ST JOSEPH
873	LA	L38	LOUISIANA REGIONAL	GONZALES
874	LA	L39	LEESVILLE	LEESVILLE
875	LA	L47	OLLA	OLLA
876	LA	L66	POLLOCK MUNICIPAL	POLLOCK
877	LA	L75	SOUTHLAND FIELD	SULPHUR
878	LA	LCH	LAKE CHARLES REGIONAL	LAKE CHARLES
879	LA	LFT	LAFAYETTE REGIONAL	LAFAYETTE
880	LA	M79	JOHN H HOOKS JR MEMORIAL	RAYVILLE
881	LA	M80	SCOTT	TALLULAH
882	LA	MKV	MARKSVILLE MUNICIPAL	MARKSVILLE
883	LA	MLU	MONROE REGIONAL	MONROE
884	LA	MSY	LOUIS ARMSTRONG NEW ORLEANS INTL	NEW ORLEANS
885	LA	NEW	LAKEFRONT	NEW ORLEANS
886	LA	OPL	ST LANDRY PARISH-AHART FIELD	OPELOUSAS
887	LA	PTN	HARRY P WILLIAMS MEMORIAL	PATTERSON
888	LA	RSN	RUSTON RGNL	RUSTON

889	LA	SHV	SHREVEPORT REGIONAL	SHREVEPORT
890	LA	SPH	SPRINGHILL	SPRINGHILL
891	LA	TVR	VICKSBURG TALLULAH RGNL	TALLULAH/VICKSBURG - MS
892	MA	0B5	TURNERS FALLS	MONTAGUE
893	MA	1B6	HOPEDALE INDUSTRIAL PARK	HOPEDALE
894	MA	1B9	MANSFIELD MUNI	MANSFIELD
895	MA	3B0	SOUTHBRIDGE MUNI	SOUTHBRIDGE
896	MA	3B2	MARSHFIELD MUNICIPAL - GEORGE HARLOW FIELD	MARSHFIELD
897	MA	ACK	NANTUCKET MEMORIAL	NANTUCKET
898	MA	AQW	HARRIMAN-AND-WEST	NORTH ADAMS
899	MA	BAF	BARNES MUNI	WESTFIELD/SPRINGFIELD
900	MA	BED	LAURENCE G HANSCOM FLD	BEDFORD
901	MA	BOS	GENERAL EDWARD LAWRENCE LOGAN INTL	BOSTON
902	MA	BVY	BEVERLY MUNI	BEVERLY
903	MA	CEF	WESTOVER ARB/METROPOLITAN	SPRINGFIELD/CHICOPEE
904	MA	CQX	CHATHAM MUNI	CHATHAM
905	MA	EWB	NEW BEDFORD REGIONAL	NEW BEDFORD
906	MA	FIT	FITCHBURG MUNI	FITCHBURG
907	MA	HYA	BARNSTABLE MUNI-BOARDMAN/POLANDO FIELD	HYANNIS
908	MA	LWM	LAWRENCE MUNI	LAWRENCE
909	MA	MVY	MARTHAS VINEYARD	VINEYARD HAVEN
910	MA	ORE	ORANGE MUNI	ORANGE
911	MA	ORH	WORCESTER REGIONAL	WORCESTER
912	MA	OWD	NORWOOD MEMORIAL	NORWOOD
913	MA	PSF	PITTSFIELD MUNI	PITTSFIELD
914	MA	PVC	PROVINCETOWN MUNI	PROVINCETOWN
915	MA	PYM	PLYMOUTH MUNI	PLYMOUTH
916	MA	TAN	TAUNTON MUNI	TAUNTON
917	MD	2W6	CAPT WALTER FRANCIS DUKE RGNL AT ST MARY'S	LEONARDTOWN
918	MD	BWI	BALTIMORE/WASHINGTON INTL THURGOOD MARSHAL	BALTIMORE
919	MD	CBE	GREATER CUMBERLAND REGIONAL	CUMBERLAND
920	MD	CGE	CAMBRIDGE-DORCHESTER	CAMBRIDGE
921	MD	DMW	CARROLL COUNTY REGIONAL/JACK B POAGE FIELD	WESTMINSTER

922	MD	ESN	EASTON/NEWNAM FIELD	EASTON
923	MD	FDK	FREDERICK MUNI	FREDERICK
924	MD	FME	TIPTON	FORT MEADE(ODENTON)
925	MD	GAI	MONTGOMERY COUNTY AIRPARK	GAITHERSBURG
926	MD	HGR	HAGERSTOWN REGIONAL-RICHARD A HENSON FLD	HAGERSTOWN
927	MD	MTN	MARTIN STATE	BALTIMORE
928	MD	OXB	OCEAN CITY MUNI	OCEAN CITY
929	MD	SBY	SALISBURY-OCEAN CITY WICOMICO REGIONAL	SALISBURY
930	ME	0B1	BETHEL REGIONAL	BETHEL
931	ME	1B0	DEXTER REGIONAL	DEXTER
932	ME	2B7	PITTSFIELD MUNI	PITTSFIELD
933	ME	3B1	GREENVILLE MUNI	GREENVILLE
934	ME	43B	DEBLOIS FLIGHT STRIP	DEBLOIS
935	ME	8B0	STEVEN A. BEAN MUNI	RANGELEY
936	ME	AUG	AUGUSTA STATE	AUGUSTA
937	ME	B19	BIDDEFORD MUNI	BIDDEFORD
938	ME	BGR	BANGOR INTL	BANGOR
939	ME	BHB	HANCOCK COUNTY-BAR HARBOR	BAR HARBOR
940	ME	BST	BELFAST MUNI	BELFAST
941	ME	CAR	CARIBOU MUNI	CARIBOU
942	ME	EPM	EASTPORT MUNI	EASTPORT
943	ME	FVE	NORTHERN AROOSTOOK REGIONAL	FRENCHVILLE
944	ME	HUL	HOULTON INTL	HOULTON
945	ME	IWI	WISCASSET	WISCASSET
946	ME	IZG	EASTERN SLOPES REGIONAL	FRYEBURG
947	ME	LEW	AUBURN/LEWISTON MUNI	AUBURN/LEWISTON
948	ME	MLT	MILLINOCKET MUNI	MILLINOCKET
949	ME	OLD	DEWITT FLD -OLD TOWN MUNI	OLD TOWN
950	ME	OWK	CENTRAL MAINE ARPT OF NORRIDGEWOCK	NORRIDGEWOCK
951	ME	PNN	PRINCETON MUNI	PRINCETON
952	ME	PQI	NORTHERN MAINE REGIONAL ARPT AT PRESQUE IS	PRESQUE ISLE
953	ME	PWM	PORTLAND INTL JETPORT	PORTLAND
954	ME	RKD	KNOX COUNTY REGIONAL	ROCKLAND

955	ME	SFM	SANFORD REGIONAL	SANFORD
956	ME	WVL	WATERVILLE ROBERT LAFLEUR	WATERVILLE
957	MI	13C	LAKEVIEW AIRPORT-GRIFFITH FIELD	LAKEVIEW
958	MI	35D	PADGHAM FIELD	ALLEGAN
959	MI	3BS	JACK BARSTOW	MIDLAND
960	MI	3CM	JAMES CLEMENTS MUNI	BAY CITY
961	MI	3FM	FREMONT MUNI	FREMONT
962	MI	3GM	GRAND HAVEN MEML AIRPARK	GRAND HAVEN
963	MI	3RC	ROSCOMMON CONSERVATION	ROSCOMMON
964	MI	3TR	JERRY TYLER MEML	NILES
965	MI	48D	CLARE MUNI	CLARE
966	MI	4D0	ABRAMS MUNI	GRAND LEDGE
967	MI	6D6	GREENVILLE MUNI	GREENVILLE
968	MI	6D9	IOSCO COUNTY	EAST TAWAS
969	MI	77G	MARLETTE	MARLETTE
970	MI	7D3	BALDWIN MUNI	BALDWIN
971	MI	83D	MACKINAC COUNTY	ST IGNACE
972	MI	8M8	GARLAND	LEWISTON
973	MI	9C8	EVART MUNI	EVART
974	MI	9D9	HASTINGS	HASTINGS
975	MI	9G2	PRICES	LINDEN
976	MI	ACB	ANTRIM COUNTY	BELLAIRE
977	MI	ADG	LENAWEE COUNTY	ADRIAN
978	MI	AMN	GRATIOT COMMUNITY	ALMA
979	MI	ANJ	SAULT STE MARIE MUNI/SANDERSON FIELD	SAULT STE MARIE
980	MI	APN	ALPENA COUNTY REGIONAL	ALPENA
981	MI	ARB	ANN ARBOR MUNI	ANN ARBOR
982	MI	AZO	KALAMAZOO/BATTLE CREEK INTERNATIONAL	KALAMAZOO
983	MI	BAX	HURON COUNTY MEMORIAL	BAD AXE
984	MI	BEH	SOUTHWEST MICHIGAN REGIONAL	BENTON HARBOR
985	MI	BIV	TULIP CITY	HOLLAND
986	MI	BTL	W K KELLOGG	BATTLE CREEK
987	MI	C04	OCEANA COUNTY	HART/SHELBY

988	MI	C20	ANDREWS UNIVERSITY AIRPARK	BERRIEN SPRINGS
989	MI	C91	DOWAGIAC MUNICIPAL	DOWAGIAC
990	MI	CAD	WEXFORD COUNTY	CADILLAC
991	MI	CFS	TUSCOLA AREA	CARO
992	MI	CIU	CHIPPEWA COUNTY INTL	SAULT STE MARIE
993	MI	CMX	HOUGHTON COUNTY MEMORIAL	HANCOCK
994	MI	CVX	CHARLEVOIX MUNI	CHARLEVOIX
995	MI	D95	DUPONT-LAPEER	LAPEER
996	MI	D98	ROMEO STATE	ROMEO
997	MI	DET	COLEMAN A. YOUNG MUNICIPAL	DETROIT
998	MI	DRM	DRUMMOND ISLAND	DRUMMOND ISLAND
999	MI	DTW	DETROIT METROPOLITAN WAYNE COUNTY	DETROIT
1000	MI	ERY	LUCE COUNTY	NEWBERRY
1001	MI	ESC	DELTA COUNTY	ESCANABA
1002	MI	FKS	FRANKFORT DOW MEMORIAL FIELD	FRANKFORT
1003	MI	FNT	BISHOP INTERNATIONAL	FLINT
1004	MI	FPK	FITCH H BEACH	CHARLOTTE
1005	MI	GDW	GLADWIN ZETTEL MEMORIAL	GLADWIN
1006	MI	GLR	GAYLORD REGIONAL	GAYLORD
1007	MI	GOV	GRAYLING AAF	GRAYLING
1008	MI	GRR	GERALD R. FORD INTERNATIONAL	GRAND RAPIDS
1009	MI	HAI	THREE RIVERS MUNI DR HAINES	THREE RIVERS
1010	MI	HTL	ROSCOMMON COUNTY	HOUGHTON LAKE
1011	MI	HYX	SAGINAW COUNTY H.W. BROWNE	SAGINAW
1012	MI	IMT	FORD	IRON MOUNTAIN KINGSFORD
1013	MI	IRS	KIRSCH MUNI	STURGIS
1014	MI	ISQ	SCHOOLCRAFT COUNTY	MANISTIQUE
1015	MI	IWD	GOGEBIC-IRON COUNTY	IRONWOOD
1016	MI	JXN	JACKSON COUNTY-REYNOLDS FIELD	JACKSON
1017	MI	JYM	HILLSDALE MUNI	HILLSDALE
1018	MI	LAN	CAPITAL CITY	LANSING
1019	MI	LDM	MASON COUNTY	LUDINGTON
1020	MI	LWA	SOUTH HAVEN AREA REGIONAL	SOUTH HAVEN

1021	MI	MBL	MANISTEE CO.-BLACKER	MANISTEE
1022	MI	MBS	MBS INTL	SAGINAW
1023	MI	MCD	MACKINAC ISLAND	MACKINAC ISLAND
1024	MI	MGN	HARBOR SPRINGS	HARBOR SPRINGS
1025	MI	MKG	MUSKEGON COUNTY	MUSKEGON
1026	MI	MNM	MENOMINEE-MARINETTE TWIN COUNTY	MENOMINEE
1027	MI	MOP	MOUNT PLEASANT MUNI	MOUNT PLEASANT
1028	MI	N98	BOYNE CITY MUNI	BOYNE CITY
1029	MI	OEB	BRANCH COUNTY MEMORIAL	COLDWATER
1030	MI	OGM	ONTONAGON COUNTY	ONTONAGON
1031	MI	ONZ	GROSSE ILE MUNI	DETROIT/GROSSE ILE
1032	MI	OSC	OSCODA-WURTSMITH	OSCODA
1033	MI	OZW	LIVINGSTON COUNTY	HOWELL
1034	MI	PHN	ST CLAIR COUNTY INTL	PORT HURON
1035	MI	PLN	PELLSTON REGIONAL AIRPORT OF EMMET COUNTY	PELLSTON
1036	MI	PTK	OAKLAND COUNTY INTERNATIONAL	PONTIAC
1037	MI	PZQ	PRESQUE ISLE COUNTY	ROGERS CITY
1038	MI	RCT	NARTRON FIELD	REED CITY
1039	MI	RMY	BROOKS FIELD	MARSHALL
1040	MI	RNP	OWOSSO COMMUNITY	OWOSSO
1041	MI	RQB	ROBEN-HOOD	BIG RAPIDS
1042	MI	SAW	SAWYER INTERNATIONAL	MARQUETTE
1043	MI	SJX	BEAVER ISLAND	BEAVER ISLAND
1044	MI	SLH	CHEBOYGAN COUNTY	CHEBOYGAN
1045	MI	TEW	MASON JEWETT FIELD	MASON
1046	MI	TTF	CUSTER	MONROE
1047	MI	TVC	CHERRY CAPITAL	TRAVERSE CITY
1048	MI	Y31	WEST BRANCH COMMUNITY	WEST BRANCH
1049	MI	Y70	IONIA COUNTY	IONIA
1050	MI	Y83	SANDUSKY CITY	SANDUSKY
1051	MI	Y89	KALKASKA CITY	KALKASKA
1052	MI	YIP	WILLOW RUN	DETROIT
1053	MN	04Y	HAWLEY MUNI	HAWLEY

1054	MN	12D	TOWER MUNI	TOWER
1055	MN	12Y	LE SUEUR MUNI	LE SUEUR
1056	MN	16D	PERHAM MUNI	PERHAM
1057	MN	48Y	PINEY PINECREEK BORDER	PINECREEK
1058	MN	7Y4	BAGLEY MUNI	BAGLEY
1059	MN	ACQ	WASECA MUNI	WASECA
1060	MN	ADC	WADENA MUNI	WADENA
1061	MN	AEL	ALBERT LEA MUNI	ALBERT LEA
1062	MN	AIT	AITKIN MUNI-STEVE KURTZ FIELD	AITKIN
1063	MN	ANE	ANOKA COUNTY-BLAINE ARPT(JANES FIELD)	MINNEAPOLIS
1064	MN	AQP	APPLETON MUNI	APPLETON
1065	MN	AUM	AUSTIN MUNI	AUSTIN
1066	MN	AXN	CHANDLER FIELD	ALEXANDRIA
1067	MN	BBB	BENSON MUNI	BENSON
1068	MN	BDE	BAUDETTE INTL	BAUDETTE
1069	MN	BFW	SILVER BAY MUNI	SILVER BAY
1070	MN	BJI	BEMIDJI REGIONAL	BEMIDJI
1071	MN	BRD	BRAINERD LAKES RGNL	BRAINERD
1072	MN	CBG	CAMBRIDGE MUNI	CAMBRIDGE
1073	MN	CHU	HOUSTON COUNTY	CALEDONIA
1074	MN	CKC	GRAND MARAIS/COOK COUNTY	GRAND MARAIS
1075	MN	CKN	CROOKSTON MUNI KIRKWOOD FLD	CROOKSTON
1076	MN	COQ	CLOQUET CARLTON COUNTY	CLOQUET
1077	MN	CQM	COOK MUNI	COOK
1078	MN	D37	WARREN MUNI	WARREN
1079	MN	D42	SPRINGFIELD MUNI	SPRINGFIELD
1080	MN	DLH	DULUTH INTL	DULUTH
1081	MN	DTL	DETROIT LAKES-WETHING FIELD	DETROIT LAKES
1082	MN	DXX	LAC QUI PARLE COUNTY	MADISON
1083	MN	DYT	SKY HARBOR	DULUTH
1084	MN	ELO	ELY MUNI	ELY
1085	MN	ETH	WHEATON MUNI	WHEATON
1086	MN	EVM	EVELETH-VIRGINIA MUNI	EVELETH

1087	MN	FCM	FLYING CLOUD	MINNEAPOLIS
1088	MN	FFM	FERGUS FALLS MUNI-EINAR MICKELSON FLD	FERGUS FALLS
1089	MN	FKA	FILLMORE COUNTY	PRESTON
1090	MN	FRM	FAIRMONT MUNI	FAIRMONT
1091	MN	FSE	FOSSTON MUNI	FOSSTON
1092	MN	GDB	GRANITE FALLS MUNI/LENZEN-ROE MEMORIAL FLD	GRANITE FALLS
1093	MN	GHW	GLENWOOD MUNI	GLENWOOD
1094	MN	GPZ	GRAND RAPIDS/ITASCA CO-GORDON NEWSTROM FLD	GRAND RAPIDS
1095	MN	GYL	GLENCOE MUNI	GLENCOE
1096	MN	HCD	HUTCHINSON MUNI-BUTLER FIELD	HUTCHINSON
1097	MN	HCO	HALLOCK MUNI	HALLOCK
1098	MN	HIB	CHISHOLM-HIBBING	HIBBING
1099	MN	HZX	ISEDOR IVERSON	MC GREGOR
1100	MN	ILL	WILLMAR MUNI-JOHN L RICE FIELD	WILLMAR
1101	MN	INL	FALLS INTL	INTERNATIONAL FALLS
1102	MN	JKJ	MOORHEAD MUNI	MOORHEAD
1103	MN	JMR	MORA MUNI	MORA
1104	MN	JYG	ST JAMES MUNI	ST JAMES
1105	MN	LJF	LITCHFIELD MUNI	LITCHFIELD
1106	MN	LVN	AIRLAKE	MINNEAPOLIS
1107	MN	LXL	LITTLE FALLS/MORRISON COUNTY-LINDBERGH FLD	LITTLE FALLS
1108	MN	MIC	CRYSTAL	MINNEAPOLIS
1109	MN	MJQ	JACKSON MUNI	JACKSON
1110	MN	MKT	MANKATO REGIONAL	MANKATO
1111	MN	MML	SOUTHWEST MINNESOTA RGNL MARSHALL/RYAN FLD	MARSHALL
1112	MN	MOX	MORRIS MUNI	MORRIS
1113	MN	MSP	MINNEAPOLIS-ST PAUL INTL/WOLD-CHAMBERLAIN/	MINNEAPOLIS
1114	MN	MVE	MONTEVIDEO-CHIPPEWA COUNTY	MONTEVIDEO
1115	MN	MWM	WINDOM MUNI	WINDOM
1116	MN	MZH	MOOSE LAKE CARLTON COUNTY	MOOSE LAKE
1117	MN	ONA	WINONA MUNI-MAX CONRAD FLD	WINONA
1118	MN	ORB	ORR REGIONAL	ORR
1119	MN	OTG	WORTHINGTON MUNI	WORTHINGTON

1120	MN	OVL	OLIVIA REGIONAL	OLIVIA
1121	MN	OWA	OWATONNA DEGNER RGNL	OWATONNA
1122	MN	PEX	PAYNESVILLE MUNI	PAYNESVILLE
1123	MN	PKD	PARK RAPIDS MUNI-KONSHOK FIELD	PARK RAPIDS
1124	MN	PNM	PRINCETON MUNI	PRINCETON
1125	MN	PQN	PIPESTONE MUNI	PIPESTONE
1126	MN	RGK	RED WING RGNL	RED WING
1127	MN	ROS	RUSH CITY RGNL	RUSH CITY
1128	MN	ROX	ROSEAU MUNI/RUDY BILLBERG FIELD	ROSEAU
1129	MN	RRT	WARROAD INTL-SWEDE CARLSON FIELD	WARROAD
1130	MN	RST	ROCHESTER INTERNATIONAL	ROCHESTER
1131	MN	RWF	REDWOOD FALLS MUNI	REDWOOD FALLS
1132	MN	SAZ	STAPLES MUNI	STAPLES
1133	MN	SBU	BLUE EARTH MUNI	BLUE EARTH
1134	MN	SGS	SOUTH ST PAUL MUNI-RICHARD E FLEMING FLD	SOUTH ST PAUL
1135	MN	STC	ST CLOUD REGIONAL	ST CLOUD
1136	MN	STP	ST PAUL DOWNTOWN HOLMAN FLD	ST PAUL
1137	MN	TOB	DODGE CENTER	DODGE CENTER
1138	MN	TVF	THIEF RIVER FALLS REGIONAL	THIEF RIVER FALLS
1139	MN	TWM	RICHARD B HELGESON	TWO HARBORS
1140	MN	ULM	NEW ULM MUNI	NEW ULM
1141	MN	VVV	ORTONVILLE MUNI-MARTINSON FIELD	ORTONVILLE
1142	MN	XVG	LONGVILLE MUNI	LONGVILLE
1143	MO	1H0	CREVE COEUR	ST LOUIS
1144	MO	9K4	SKYHAVEN	WARRENSBURG
1145	MO	AIZ	LEE C FINE MEMORIAL	KAISER LAKE OZARK
1146	MO	BUM	BUTLER MEMORIAL	BUTLER
1147	MO	CGI	CAPE GIRARDEAU REGIONAL	CAPE GIRARDEAU
1148	MO	COU	COLUMBIA REGIONAL	COLUMBIA
1149	MO	DMO	SEDALIA MEMORIAL	SEDALIA
1150	MO	DXE	DEXTER MUNI	DEXTER
1151	MO	EOS	NEOSHO HUGH ROBINSON	NEOSHO
1152	MO	EVU	NORTHWEST MISSOURI REGIONAL	MARYVILLE

1153	MO	EZZ	CAMERON MEMORIAL	CAMERON
1154	MO	FAM	FARMINGTON REGIONAL	FARMINGTON
1155	MO	GPH	CLAY COUNTY REGIONAL	MOSBY
1156	MO	H21	CAMDENTON MEMORIAL	CAMDENTON
1157	MO	H79	ELDON MODEL AIRPARK	ELDON
1158	MO	H88	FREDERICKTOWN REGIONAL	FREDERICKTOWN
1159	MO	HAE	HANNIBAL REGIONAL	HANNIBAL
1160	MO	IRK	KIRKSVILLE REGIONAL	KIRKSVILLE
1161	MO	JEF	JEFFERSON CITY MEML	JEFFERSON CITY
1162	MO	JLN	JOPLIN REGIONAL	JOPLIN
1163	MO	K02	PERRYVILLE MUNI	PERRYVILLE
1164	MO	LBO	FLOYD W. JONES LEBANON	LEBANON
1165	MO	LRY	LAWRENCE SMITH MEMORIAL	HARRISONVILLE
1166	MO	LXT	LEE'S SUMMIT MUNICIPAL	LEE'S SUMMIT
1167	MO	M05	CARUTHERSVILLE MEMORIAL	CARUTHERSVILLE
1168	MO	M17	BOLIVAR MUNICIPAL	BOLIVAR
1169	MO	M58	MONETT MUNI	MONETT
1170	MO	MAW	MALDEN MUNI	MALDEN
1171	MO	MBY	OMAR N BRADLEY	MOBERLY
1172	MO	MCI	KANSAS CITY INTL	KANSAS CITY
1173	MO	MHL	MARSHALL MEML MUNI	MARSHALL
1174	MO	MKC	CHARLES B. WHEELER DOWNTOWN	KANSAS CITY
1175	MO	MNF	MOUNTAIN VIEW	MOUNTAIN VIEW
1176	MO	MO6	WASHINGTON MEMORIAL	WASHINGTON
1177	MO	MO8	NORTH CENTRAL MISSOURI REGIONAL	BROOKFIELD
1178	MO	MYJ	MEXICO MEMORIAL	MEXICO
1179	MO	NVD	NEVADA MUNI	NEVADA
1180	MO	PLK	M. GRAHAM CLARK - TANEY COUNTY	POINT LOOKOUT
1181	MO	POF	POPLAR BLUFF MUNICIPAL	POPLAR BLUFF
1182	MO	SET	ST CHARLES COUNTY SMARTT	ST CHARLES
1183	MO	SGF	SPRINGFIELD-BRANSON REGIONAL	SPRINGFIELD
1184	MO	SIK	SIKESTON MEML MUNI	SIKESTON
1185	MO	STJ	ROSECRANS MEMORIAL	ST JOSEPH

1186	MO	STL	LAMBERT-ST LOUIS INTL	ST LOUIS
1187	MO	SUS	SPIRIT OF ST LOUIS	ST LOUIS
1188	MO	TBN	WAYNESVILLE RGNL ARPT AT FORNEY FIELD	FORT LEONARD WOOD
1189	MO	TKX	KENNETT MEMORIAL	KENNETT
1190	MO	TRX	TRENTON MUNI	TRENTON
1191	MO	UNO	WEST PLAINS MUNI	WEST PLAINS
1192	MO	UUV	SULLIVAN REGIONAL	SULLIVAN
1193	MO	VER	JESSE VIERTEL MEMORIAL	BOONVILLE
1194	MO	VIH	ROLLA NATIONAL	ROLLA/VICHY
1195	MS	1R7	BROOKHAVEN-LINCOLN COUNTY	BROOKHAVEN
1196	MS	25M	RIPLEY	RIPLEY
1197	MS	2M4	G. V. MONTGOMERY	FOREST
1198	MS	2R0	WAYNESBORO MUNI	WAYNESBORO
1199	MS	4R1	I H BASS JR MEMORIAL	LUMBERTON
1200	MS	66Y	DIAMONDHEAD	DIAMONDHEAD
1201	MS	87I	YAZOO COUNTY	YAZOO CITY
1202	MS	9M4	ACKERMAN CHOCTAW COUNTY	ACKERMAN
1203	MS	CKM	FLETCHER FIELD	CLARKSDALE
1204	MS	CRX	ROSCOE TURNER	CORINTH
1205	MS	GLH	MID DELTA RGNL	GREENVILLE
1206	MS	GNF	GRENADA MUNI	GRENADA
1207	MS	GPT	GULFPORT-BILOXI INTL	GULFPORT
1208	MS	GTR	GOLDEN TRIANGLE REGIONAL	COLUMBUS/W POINT/STARKVILLE
1209	MS	GWO	GREENWOOD-LEFLORE	GREENWOOD
1210	MS	HBG	BOBBY L CHAIN MUNI	HATTIESBURG
1211	MS	HEZ	HARDY-ANDERS FIELD NATCHEZ-ADAMS COUNTY	NATCHEZ
1212	MS	HKS	HAWKINS FIELD	JACKSON
1213	MS	HSA	STENNIS INTL	BAY ST LOUIS
1214	MS	IDL	INDIANOLA MUNI	INDIANOLA
1215	MS	JAN	JACKSON-EVERS INTL	JACKSON
1216	MS	LMS	LOUISVILLE WINSTON COUNTY	LOUISVILLE
1217	MS	LUL	HESLER-NOBLE FIELD	LAUREL
1218	MS	M11	COPIAH COUNTY	CRYSTAL SPRINGS

1219	MS	M13	POPLARVILLE-PEARL RIVER COUNTY	POPLARVILLE
1220	MS	M16	JOHN BELL WILLIAMS	RAYMOND
1221	MS	M23	JAMES H EASOM FIELD	NEWTON
1222	MS	M40	MONROE COUNTY	ABERDEEN/AMORY
1223	MS	M44	HOUSTON MUNI	HOUSTON
1224	MS	M72	NEW ALBANY-UNION CO	NEW ALBANY
1225	MS	M83	MCCHAREN FIELD	WEST POINT
1226	MS	MBO	BRUCE CAMPBELL FIELD	MADISON
1227	MS	MCB	MC COMB/PIKE COUNTY/JOHN E LEWIS FIELD	MC COMB
1228	MS	MEI	KEY FIELD	MERIDIAN
1229	MS	MJD	PICAYUNE MUNI	PICAYUNE
1230	MS	MPE	PHILADELPHIA MUNI	PHILADELPHIA
1231	MS	OLV	OLIVE BRANCH	OLIVE BRANCH
1232	MS	OSX	KOSCIUSKO-ATTALA COUNTY	KOSCIUSKO
1233	MS	PIB	HATTIESBURG-LAUREL REGIONAL	HATTIESBURG/LAUREL
1234	MS	PMU	PANOLA COUNTY	BATESVILLE
1235	MS	PQL	TRENT LOTT INTERNATIONAL	PASCAGOULA
1236	MS	RNV	CLEVELAND MUNI	CLEVELAND
1237	MS	STF	GEORGE M BRYAN	STARKVILLE
1238	MS	TUP	TUPELO RGNL	TUPELO
1239	MS	UBS	COLUMBUS-LOWNDES COUNTY	COLUMBUS
1240	MS	UOX	UNIVERSITY-OXFORD	OXFORD
1241	MS	UTA	TUNICA MUNICIPAL	TUNICA
1242	MS	VKS	VICKSBURG MUNI	VICKSBURG
1243	MT	1S3	TILLITT FIELD	FORSYTH
1244	MT	38S	DEER LODGE-CITY-COUNTY	DEER LODGE
1245	MT	3U3	BOWMAN FIELD	ANACONDA
1246	MT	3U7	BENCHMARK	BENCHMARK
1247	MT	48S	HARLEM	HARLEM
1248	MT	4U3	LIBERTY COUNTY	CHESTER
1249	MT	4U6	CIRCLE TOWN COUNTY	CIRCLE
1250	MT	6S0	BIG TIMBER	BIG TIMBER
1251	MT	6S5	RAVALLI COUNTY	HAMILTON

1252	MT	6S8	LAUREL MUNI	LAUREL
1253	MT	79S	FORT BENTON	FORT BENTON
1254	MT	7S0	RONAN	RONAN
1255	MT	88M	EUREKA	EUREKA
1256	MT	8S1	POLSON	POLSON
1257	MT	8U6	TERRY	TERRY
1258	MT	9S2	SCOBEY	SCOBEY
1259	MT	BHK	BAKER MUNI	BAKER
1260	MT	BIL	BILLINGS LOGAN INTL	BILLINGS
1261	MT	BTM	BERT MOONEY	BUTTE
1262	MT	BZN	GALLATIN FIELD	BOZEMAN
1263	MT	CII	CHOTEAU	CHOTEAU
1264	MT	CTB	CUT BANK MUNI	CUT BANK
1265	MT	DLN	DILLON	DILLON
1266	MT	GDV	DAWSON COMMUNITY	GLENDIVE
1267	MT	GGW	WOKAL FIELD/GLASGOW INTL	GLASGOW
1268	MT	GPI	GLACIER PARK INTL	KALISPELL
1269	MT	GTF	GREAT FALLS INTL	GREAT FALLS
1270	MT	HLN	HELENA REGIONAL	HELENA
1271	MT	HVR	HAVRE CITY-COUNTY	HAVRE
1272	MT	JDN	JORDAN	JORDAN
1273	MT	LVM	MISSION FIELD	LIVINGSTON
1274	MT	LWT	LEWISTOWN MUNI	LEWISTOWN
1275	MT	M46	COLSTRIP	COLSTRIP
1276	MT	M75	MALTA	MALTA
1277	MT	MLS	FRANK WILEY FIELD	MILES CITY
1278	MT	MSO	MISSOULA INTERNATIONAL	MISSOULA
1279	MT	OLF	L M CLAYTON	WOLF POINT
1280	MT	PWD	SHER-WOOD	PLENTYWOOD
1281	MT	RPX	ROUNDUP	ROUNDUP
1282	MT	S01	CONRAD	CONRAD
1283	MT	S59	LIBBY	LIBBY
1284	MT	S71	EDGAR G OBIE	CHINOOK

1285	MT	SBX	SHELBY	SHELBY
1286	MT	SDY	SIDNEY-RICHLAND MUNI	SIDNEY
1287	MT	THM	THOMPSON FALLS	THOMPSON FALLS
1288	MT	WYS	YELLOWSTONE	WEST YELLOWSTONE
1289	NC	1A5	MACON COUNTY	FRANKLIN
1290	NC	45J	ROCKINGHAM-HAMLET	ROCKINGHAM
1291	NC	5W8	SILER CITY MUNICIPAL	SILER CITY
1292	NC	60J	OCEAN ISLE	OCEAN ISLE BEACH
1293	NC	7W6	HYDE COUNTY	ENGELHARD
1294	NC	ACZ	HENDERSON FIELD	WALLACE
1295	NC	AFP	ANSON COUNTY	WADESBORO
1296	NC	AKH	GASTONIA MUNI	GASTONIA
1297	NC	ASJ	TRI-COUNTY	AHOSKIE
1298	NC	AVL	ASHEVILLE REGIONAL	ASHEVILLE
1299	NC	BUY	BURLINGTON-ALAMANCE RGNL	BURLINGTON
1300	NC	CLT	CHARLOTTE/DOUGLAS INTL	CHARLOTTE
1301	NC	CPC	COLUMBUS COUNTY MUNI	WHITEVILLE
1302	NC	CTZ	SAMPSON COUNTY	CLINTON
1303	NC	DPL	DUPLIN CO	KENANSVILLE
1304	NC	ECG	ELIZABETH CITY CG AIR STATION/RGNL	ELIZABETH CITY
1305	NC	EDE	NORTHEASTERN RGNL	EDENTON
1306	NC	EHO	SHELBY MUNI	SHELBY
1307	NC	EQY	MONROE REGIONAL	MONROE
1308	NC	EWN	CRAVEN COUNTY REGIONAL	NEW BERN
1309	NC	EXX	DAVIDSON COUNTY	LEXINGTON
1310	NC	EYF	CURTIS L BROWN JR FIELD	ELIZABETHTOWN
1311	NC	FAY	FAYETTEVILLE REGIONAL/GRANNIS FIELD	FAYETTEVILLE
1312	NC	FQD	RUTHERFORD CO - MARCHMAN FIELD	RUTHERFORDTON
1313	NC	GEV	ASHE COUNTY	JEFFERSON
1314	NC	GSO	PIEDMONT TRIAD INTERNATIONAL	GREENSBORO
1315	NC	GWW	GOLDSBORO-WAYNE MUNI	GOLDSBORO
1316	NC	HBI	ASHEBORO REGIONAL	ASHEBORO
1317	NC	HKY	HICKORY REGIONAL	HICKORY

1318	NC	HNZ	HENDERSON-OXFORD	OXFORD
1319	NC	HRJ	HARNETT COUNTY	ERWIN
1320	NC	HSE	BILLY MITCHELL	HATTERAS
1321	NC	IGX	HORACE WILLIAMS	CHAPEL HILL
1322	NC	ILM	WILMINGTON INTL	WILMINGTON
1323	NC	INT	SMITH REYNOLDS	WINSTON SALEM
1324	NC	IPJ	LINCOLNTON-LINCOLN COUNTY RGNL	LINCOLNTON
1325	NC	ISO	KINSTON RGNL JETPORT AT STALLINGS FLD	KINSTON
1326	NC	JNX	JOHNSTON COUNTY	SMITHFIELD
1327	NC	JQF	CONCORD REGIONAL	CONCORD
1328	NC	LBT	LUMBERTON MUNI	LUMBERTON
1329	NC	LHZ	FRANKLIN COUNTY	LOUISBURG
1330	NC	MCZ	MARTIN COUNTY	WILLIAMSTON
1331	NC	MEB	LAURINBURG-MAXTON	MAXTON
1332	NC	MQI	DARE COUNTY REGIONAL	MANTEO
1333	NC	MRH	MICHAEL J. SMITH FIELD	BEAUFORT
1334	NC	MRN	FOOTHILLS REGIONAL	MORGANTON
1335	NC	MWK	MOUNT AIRY/SURRY COUNTY	MOUNT AIRY
1336	NC	OAJ	ALBERT J ELLIS	JACKSONVILLE
1337	NC	OCW	WARREN FIELD	WASHINGTON
1338	NC	ONX	CURRITUCK COUNTY	CURRITUCK
1339	NC	PGV	PITT-GREENVILLE	GREENVILLE
1340	NC	PMZ	PLYMOUTH MUNI	PLYMOUTH
1341	NC	RDU	RALEIGH-DURHAM INTL	RALEIGH/DURHAM
1342	NC	RHP	ANDREWS-MURPHY	ANDREWS
1343	NC	RUQ	ROWAN COUNTY	SALISBURY
1344	NC	RWI	ROCKY MOUNT-WILSON RGNL	ROCKY MOUNT
1345	NC	RZZ	HALIFAX COUNTY	ROANOKE RAPIDS
1346	NC	SIF	ROCKINGHAM COUNTY NC SHILOH	REIDSVILLE
1347	NC	SOP	MOORE COUNTY	PINEHURST/SOUTHERN PINES
1348	NC	SUT	BRUNSWICK COUNTY	OAK ISLAND
1349	NC	SVH	STATESVILLE REGIONAL	STATESVILLE
1350	NC	TDF	PERSON COUNTY	ROXBORO

1351	NC	TTA	SANFORD-LEE COUNTY RGNL	SANFORD
1352	NC	UKF	WILKES COUNTY	NORTH WILKESBORO
1353	NC	VUJ	STANLY COUNTY	ALBEMARLE
1354	NC	W03	WILSON INDUSTRIAL AIR CENTER	WILSON
1355	NC	W40	MOUNT OLIVE MUNI	MOUNT OLIVE
1356	NC	ZEF	ELKIN MUNI	ELKIN
1357	ND	06D	ROLLA MUNI	ROLLA
1358	ND	46D	CARRINGTON MUNI	CARRINGTON
1359	ND	5N8	CASSELTON ROBERT MILLER RGNL	CASSELTON
1360	ND	6D8	BARNES COUNTY MUNI	VALLEY CITY
1361	ND	BIS	BISMARCK MUNI	BISMARCK
1362	ND	BPP	BOWMAN MUNI	BOWMAN
1363	ND	BWP	HARRY STERN	WAHPETON
1364	ND	D60	TIOGA MUNI	TIOGA
1365	ND	DIK	DICKINSON - THEODORE ROOSEVELT REGIONAL	DICKINSON
1366	ND	DVL	DEVILS LAKE MUNI	DEVILS LAKE
1367	ND	FAR	HECTOR INTL	FARGO
1368	ND	GAF	HUTSON FIELD	GRAFTON
1369	ND	GFK	GRAND FORKS INTL	GRAND FORKS
1370	ND	HEI	HETTINGER MUNI	HETTINGER
1371	ND	HZE	MERCER COUNTY REGIONAL	HAZEN
1372	ND	ISN	SLOULIN FLD INTL	WILLISTON
1373	ND	JMS	JAMESTOWN REGIONAL	JAMESTOWN
1374	ND	MOT	MINOT INTL	MINOT
1375	ND	PMB	PEMBINA MUNI	PEMBINA
1376	ND	S25	WATFORD CITY MUNI	WATFORD CITY
1377	ND	Y19	MANDAN MUNI	MANDAN
1378	NE	0G3	TECUMSEH MUNICIPAL	TECUMSEH
1379	NE	AFK	NEBRASKA CITY MUNICIPAL	NEBRASKA CITY
1380	NE	AHQ	WAHOO MUNI	WAHOO
1381	NE	AIA	ALLIANCE MUNI	ALLIANCE
1382	NE	ANW	AINSWORTH MUNI	AINSWORTH
1383	NE	AUH	AURORA MUNICIPAL - AL POTTER FIELD	AURORA

1384	NE	BBW	BROKEN BOW MUNI	BROKEN BOW
1385	NE	BFF	WESTERN NEB. RGNL/WILLIAM B. HEILIG FIELD	SCOTTSBLUFF
1386	NE	BIE	BEATRICE MUNICIPAL	BEATRICE
1387	NE	CDR	CHADRON MUNI	CHADRON
1388	NE	CEK	CRETE MUNICIPAL	CRETE
1389	NE	CZD	COZAD MUNI	COZAD
1390	NE	EAR	KEARNEY MUNI	KEARNEY
1391	NE	FBY	FAIRBURY MUNICIPAL	FAIRBURY
1392	NE	FET	FREMONT MUNI	FREMONT
1393	NE	FMZ	FAIRMONT STATE AIRFIELD	FAIRMONT
1394	NE	GRI	CENTRAL NEBRASKA REGIONAL	GRAND ISLAND
1395	NE	GRN	GORDON MUNI	GORDON
1396	NE	HDE	BREWSTER FIELD	HOLDREGE
1397	NE	HSI	HASTINGS MUNI	HASTINGS
1398	NE	IBM	KIMBALL MUNI/ROBERT E ARRAJ FIELD	KIMBALL
1399	NE	IML	IMPERIAL MUNI	IMPERIAL
1400	NE	JYR	YORK MUNICIPAL	YORK
1401	NE	K46	BLAIR MUNI	BLAIR
1402	NE	LBF	NORTH PLATTE RGNL AIRPORT LEE BIRD FIELD	NORTH PLATTE
1403	NE	LCG	WAYNE MUNI	WAYNE
1404	NE	LNK	LINCOLN	LINCOLN
1405	NE	LXN	JIM KELLY FIELD	LEXINGTON
1406	NE	MCK	MC COOK REGIONAL	MC COOK
1407	NE	MLE	MILLARD	OMAHA
1408	NE	OFK	KARL STEFAN MEMORIAL	NORFOLK
1409	NE	OGA	SEARLE FIELD	OGALLALA
1410	NE	OLU	COLUMBUS MUNI	COLUMBUS
1411	NE	OMA	EPPLEY AIRFIELD	OMAHA
1412	NE	ONL	THE O'NEILL MUNI-JOHN L BAKER FIELD	O'NEILL
1413	NE	PMV	PLATTSMOUTH MUNI	PLATTSMOUTH
1414	NE	RBE	ROCK COUNTY	BASSETT
1415	NE	SCB	SCRIBNER STATE	SCRIBNER
1416	NE	SNY	SIDNEY MUNICIPAL/LLOYD W. CARR FIELD	SIDNEY

1417	NE	SWT	SEWARD MUNICIPAL	SEWARD
1418	NE	TQE	TEKAMAH MUNI	TEKAMAH
1419	NE	VTN	MILLER FIELD	VALENTINE
1420	NH	8B1	HAWTHORNE-FEATHER AIRPARK	HILLSBORO
1421	NH	ASH	BOIRE FIELD	NASHUA
1422	NH	BML	BERLIN MUNI	BERLIN
1423	NH	CNH	CLAREMONT MUNI	CLAREMONT
1424	NH	CON	CONCORD MUNI	CONCORD
1425	NH	DAW	SKYHAVEN	ROCHESTER
1426	NH	EEN	DILLANT-HOPKINS	KEENE
1427	NH	HIE	MOUNT WASHINGTON REGIONAL	WHITEFIELD
1428	NH	LCI	LACONIA MUNI	LACONIA
1429	NH	LEB	LEBANON MUNI	LEBANON
1430	NH	MHT	MANCHESTER	MANCHESTER
1431	NH	PSM	PEASE INTERNATIONAL TRADEPORT	PORTSMOUTH
1432	NJ	1N4	WOODBINE MUNI	WOODBINE
1433	NJ	39N	PRINCETON	PRINCETON/ROCKY HILL
1434	NJ	ACY	ATLANTIC CITY INTERNATIONAL	ATLANTIC CITY
1435	NJ	BLM	MONMOUTH EXECUTIVE	BELMAR/FARMINGDALE
1436	NJ	CDW	ESSEX COUNTY	CALDWELL
1437	NJ	EWR	NEWARK LIBERTY INTL	NEWARK
1438	NJ	LDJ	LINDEN	LINDEN
1439	NJ	MIV	MILLVILLE MUNI	MILLVILLE
1440	NJ	MJX	ROBERT J. MILLER AIR PARK	TOMS RIVER
1441	NJ	MMU	MORRISTOWN MUNI	MORRISTOWN
1442	NJ	N14	FLYING W	LUMBERTON
1443	NJ	N81	HAMMONTON MUNI	HAMMONTON
1444	NJ	N87	TRENTON-ROBBINSVILLE	ROBBINSVILLE
1445	NJ	TEB	TETERBORO	TETERBORO
1446	NJ	TTN	TRENTON MERCER	TRENTON
1447	NJ	WWD	CAPE MAY COUNTY	WILDWOOD
1448	NM	0E0	MORIARTY	MORIARTY
1449	NM	24N	JICARILLA APACHE NATION	DULCE

1450	NM	5T6	DONA ANA COUNTY AT SANTA TERESA	SANTA TERESA
1451	NM	5V5	SHIPROCK AIRSTRIP	SHIPROCK
1452	NM	ABQ	ALBUQUERQUE INTL SUNPORT	ALBUQUERQUE
1453	NM	AEG	DOUBLE EAGLE II	ALBUQUERQUE
1454	NM	ALM	ALAMOGORDO-WHITE SANDS REGIONAL	ALAMOGORDO
1455	NM	ATS	ARTESIA MUNI	ARTESIA
1456	NM	AXX	ANGEL FIRE	ANGEL FIRE
1457	NM	CAO	CLAYTON MUNI ARPK	CLAYTON
1458	NM	CNM	CAVERN CITY AIR TRML	CARLSBAD
1459	NM	CVN	CLOVIS MUNI	CLOVIS
1460	NM	DMN	DEMING MUNI	DEMING
1461	NM	E06	LEA COUNTY-ZIP FRANKLIN MEMORIAL	LOVINGTON
1462	NM	E14	SAN JUAN PUEBLO	ESPANOLA
1463	NM	F37	CARRIZOZO MUNI	CARRIZOZO
1464	NM	FMN	FOUR CORNERS REGIONAL	FARMINGTON
1465	NM	FSU	FORT SUMNER MUNI	FORT SUMNER
1466	NM	GNT	GRANTS-MILAN MUNI	GRANTS
1467	NM	GUP	GALLUP MUNICIPAL	GALLUP
1468	NM	HOB	LEA COUNTY RGNL	HOBBS
1469	NM	I58	SANTA ROSA ROUTE 66	SANTA ROSA
1470	NM	LRU	LAS CRUCES INTERNATIONAL	LAS CRUCES
1471	NM	LSB	LORDSBURG MUNI	LORDSBURG
1472	NM	N24	QUESTA MUNI NR 2	QUESTA
1473	NM	ONM	SOCORRO MUNI	SOCORRO
1474	NM	ROW	ROSWELL INTERNATIONAL AIR CENTER	ROSWELL
1475	NM	RTN	RATON MUNICIPAL/CREWS FIELD	RATON
1476	NM	SAF	SANTA FE MUNI	SANTA FE
1477	NM	SRR	SIERRA BLANCA REGIONAL	RUIDOSO
1478	NM	SVC	GRANT COUNTY	SILVER CITY
1479	NM	TCC	TUCUMCARI MUNI	TUCUMCARI
1480	NM	TCS	TRUTH OR CONSEQUENCES MUNI	TRUTH OR CONSEQUENCES
1481	NV	0L7	JEAN	JEAN
1482	NV	4SD	RENO/STEAD	RENO

1483	NV	61B	BOULDER CITY MUNI	BOULDER CITY
1484	NV	67L	MESQUITE	MESQUITE
1485	NV	9U3	AUSTIN	AUSTIN
1486	NV	B08	SILVER SPRINGS	SILVER SPRINGS
1487	NV	BAM	BATTLE MOUNTAIN	BATTLE MOUNTAIN
1488	NV	CXP	CARSON	CARSON CITY
1489	NV	EKO	ELKO REGIONAL	ELKO
1490	NV	ELY	ELY ARPT /YELLAND FLD/	ELY
1491	NV	FLX	FALLON MUNI	FALLON
1492	NV	HND	HENDERSON EXECUTIVE	LAS VEGAS
1493	NV	HTH	HAWTHORNE INDUSTRIAL	HAWTHORNE
1494	NV	LAS	MC CARRAN INTL	LAS VEGAS
1495	NV	LOL	DERBY FIELD	LOVELOCK
1496	NV	LWL	WELLS MUNICIPAL/HARRIET FIELD	WELLS
1497	NV	MEV	MINDEN-TAHOE	MINDEN
1498	NV	O43	YERINGTON MUNI	YERINGTON
1499	NV	RNO	RENO/TAHOE INTERNATIONAL	RENO
1500	NV	TPH	TONOPAH	TONOPAH
1501	NV	U08	PERKINS FIELD	OVERTON
1502	NV	VGT	NORTH LAS VEGAS	LAS VEGAS
1503	NV	WMC	WINNEMUCCA MUNI	WINNEMUCCA
1504	NY	0G7	FINGER LAKES REGIONAL	SENECA FALLS
1505	NY	1B1	COLUMBIA COUNTY	HUDSON
1506	NY	4G6	HORNELL MUNI	HORNELL
1507	NY	5B2	SARATOGA COUNTY	SARATOGA SPRINGS
1508	NY	7G0	LEDGEDALE AIRPARK	BROCKPORT
1509	NY	7N1	CORNING-PAINTED POST	CORNING
1510	NY	9G3	AKRON	AKRON
1511	NY	ALB	ALBANY INTL	ALBANY
1512	NY	ART	WATERTOWN INTERNATIONAL	WATERTOWN
1513	NY	BGM	GREATER BINGHAMTON/EDWIN A LINK FIELD	BINGHAMTON
1514	NY	BQR	BUFFALO-LANCASTER	LANCASTER
1515	NY	BUF	BUFFALO NIAGARA INTL	BUFFALO

1516	NY	CZG	TRI-CITIES	ENDICOTT
1517	NY	D38	CANANDAIGUA	CANANDAIGUA
1518	NY	DKK	CHAUTAUQUA COUNTY/DUNKIRK	DUNKIRK
1519	NY	DSV	DANSVILLE MUNI	DANSVILLE
1520	NY	ELM	ELMIRA/CORNING REGIONAL	ELMIRA/CORNING
1521	NY	ELZ	WELLSVILLE MUNI ARPT - TARANTINE FLD	WELLSVILLE
1522	NY	FOK	FRANCIS S GABRESKI	WESTHAMPTON BEACH
1523	NY	FRG	REPUBLIC	FARMINGDALE
1524	NY	FZY	OSWEGO COUNTY	FULTON
1525	NY	GFL	FLOYD BENNETT MEMORIAL	GLENS FALLS
1526	NY	GVQ	GENESEE COUNTY	BATAVIA
1527	NY	H30	HAMILTON MUNICIPAL	HAMILTON
1528	NY	HPN	WESTCHESTER COUNTY	WHITE PLAINS
1529	NY	HTO	EAST HAMPTON	EAST HAMPTON
1530	NY	HWV	BROOKHAVEN	SHIRLEY
1531	NY	IAG	NIAGARA FALLS INTL	NIAGARA FALLS
1532	NY	ISP	LONG ISLAND MAC ARTHUR	ISLIP
1533	NY	ITH	ITHACA TOMPKINS REGIONAL	ITHACA
1534	NY	JFK	JOHN F KENNEDY INTL	NEW YORK
1535	NY	JHW	CHAUTAUQUA COUNTY/JAMESTOWN	JAMESTOWN
1536	NY	LGA	LA GUARDIA	NEW YORK
1537	NY	MAL	MALONE-DUFORT	MALONE
1538	NY	MGJ	ORANGE COUNTY	MONTGOMERY
1539	NY	MSS	MASSENA INTL-RICHARDS FIELD	MASSENA
1540	NY	MSV	SULLIVAN COUNTY INTL	MONTICELLO
1541	NY	N03	CORTLAND COUNTY-CHASE FIELD	CORTLAND
1542	NY	N23	SIDNEY MUNI	SIDNEY
1543	NY	N66	ONEONTA MUNI	ONEONTA
1544	NY	N89	JOSEPH Y RESNICK	ELLENVILLE
1545	NY	NY0	FULTON COUNTY	JOHNSTOWN
1546	NY	OGS	OGDENSBURG INTL	OGDENSBURG
1547	NY	OIC	LT WARREN EATON	NORWICH
1548	NY	OLE	CATTARAUGUS COUNTY-OLEAN	OLEAN

1549	NY	PBG	PLATTSBURGH INTL	PLATTSBURGH
1550	NY	PEO	PENN YAN	PENN YAN
1551	NY	PLB	CLINTON CO	PLATTSBURGH
1552	NY	POU	DUTCHESS COUNTY	POUGHKEEPSIE
1553	NY	RME	GRIFFISS AIRPARK	ROME
1554	NY	ROC	GREATER ROCHESTER INTERNATIONAL	ROCHESTER
1555	NY	SCH	SCHENECTADY COUNTY	SCHENECTADY
1556	NY	SLK	ADIRONDACK REGIONAL	SARANAC LAKE
1557	NY	SWF	STEWART INT'L	NEWBURGH
1558	NY	SYR	SYRACUSE HANCOCK INTL	SYRACUSE
1559	NY	UCA	ONEIDA COUNTY	UTICA
1560	OH	02G	COLUMBIANA COUNTY	EAST LIVERPOOL
1561	OH	0G6	WILLIAMS COUNTY	BRYAN
1562	OH	16G	SENECA COUNTY	TIFFIN
1563	OH	17G	PORT BUCYRUS-CRAWFORD COUNTY	BUCYRUS
1564	OH	1G0	WOOD COUNTY	BOWLING GREEN
1565	OH	1G5	MEDINA MUNICIPAL	MEDINA
1566	OH	22I	VINTON COUNTY	MC ARTHUR
1567	OH	29G	PORTAGE COUNTY	RAVENNA
1568	OH	3G3	WADSWORTH MUNI	WADSWORTH
1569	OH	3G4	ASHLAND COUNTY	ASHLAND
1570	OH	4G5	MONROE COUNTY	WOODSFIELD
1571	OH	4I3	KNOX COUNTY	MOUNT VERNON
1572	OH	56D	WYANDOT COUNTY	UPPER SANDUSKY
1573	OH	5A1	NORWALK-HURON COUNTY	NORWALK
1574	OH	5G7	BLUFFTON	BLUFFTON
1575	OH	6I6	DARBY DAN	COLUMBUS
1576	OH	8G6	HARRISON COUNTY	CADIZ
1577	OH	AKR	AKRON FULTON INTL	AKRON
1578	OH	AOH	LIMA ALLEN COUNTY	LIMA
1579	OH	AXV	NEIL ARMSTRONG	WAPAKONETA
1580	OH	BJJ	WAYNE COUNTY	WOOSTER
1581	OH	BKL	BURKE LAKEFRONT	CLEVELAND

1582	OH	CAK	AKRON-CANTON REGIONAL	AKRON
1583	OH	CDI	CAMBRIDGE MUNI	CAMBRIDGE
1584	OH	CGF	CUYAHOGA COUNTY	CLEVELAND
1585	OH	CLE	CLEVELAND-HOPKINS INTL	CLEVELAND
1586	OH	CMH	PORT COLUMBUS INTL	COLUMBUS
1587	OH	CQA	LAKEFIELD	CELINA
1588	OH	CYO	PICKAWAY COUNTY MEMORIAL	CIRCLEVILLE
1589	OH	DAY	JAMES M COX DAYTON INTL	DAYTON
1590	OH	DLZ	DELAWARE MUNI	DELAWARE
1591	OH	EDJ	BELLEFONTAINE RGNL	BELLEFONTAINE
1592	OH	EOP	PIKE COUNTY	WAVERLY
1593	OH	FDY	FINDLAY	FINDLAY
1594	OH	FZI	FOSTORIA METROPOLITAN	FOSTORIA
1595	OH	GAS	GALLIA-MEIGS REGIONAL	GALLIPOLIS
1596	OH	GQQ	GALION MUNI	GALION
1597	OH	HAO	BUTLER CO RGNL	HAMILTON
1598	OH	HOC	HIGHLAND COUNTY	HILLSBORO
1599	OH	HZY	ASHTABULA COUNTY	ASHTABULA
1600	OH	I12	SIDNEY MUNI	SIDNEY
1601	OH	I17	PIQUA AIRPORT- HARTZELL FIELD	PIQUA
1602	OH	I19	GREENE COUNTY-LEWIS A. JACKSON REGIONAL	DAYTON
1603	OH	I23	FAYETTE COUNTY	WASHINGTON COURT HOUSE
1604	OH	I40	RICHARD DOWNING	COSHOCTON
1605	OH	I43	JAMES A RHODES	JACKSON
1606	OH	I69	CLERMONT COUNTY	BATAVIA
1607	OH	I74	GRIMES FIELD	URBANA
1608	OH	I86	PERRY COUNTY	NEW LEXINGTON
1609	OH	I95	HARDIN COUNTY	KENTON
1610	OH	ILN	AIRBORNE AIRPARK	WILMINGTON
1611	OH	ISZ	CINCINNATI-BLUE ASH	CINCINNATI
1612	OH	LCK	RICKENBACKER INTERNATIONAL	COLUMBUS
1613	OH	LHQ	FAIRFIELD COUNTY	LANCASTER
1614	OH	LNN	WILLOUGHBY LOST NATION MUNI	WILLOUGHBY

1615	OH	LPR	LORAIN COUNTY REGIONAL	LORAIN/ELYRIA
1616	OH	LUK	CINCINNATI MUNI AIRPORT LUNKEN FIELD	CINCINNATI
1617	OH	MFD	MANSFIELD LAHM REGIONAL	MANSFIELD
1618	OH	MGY	DAYTON-WRIGHT BROTHERS	DAYTON
1619	OH	MNN	MARION MUNI	MARION
1620	OH	MRT	UNION COUNTY	MARYSVILLE
1621	OH	MWO	HOOK FIELD MUNI	MIDDLETOWN
1622	OH	OSU	OHIO STATE UNIVERSITY	COLUMBUS
1623	OH	PCW	CARL R KELLER FIELD	PORT CLINTON
1624	OH	PHD	HARRY CLEVER FIELD	NEW PHILADELPHIA
1625	OH	PMH	GREATER PORTSMOUTH REGIONAL	PORTSMOUTH
1626	OH	RZT	ROSS COUNTY	CHILLICOTHE
1627	OH	S24	SANDUSKY COUNTY REGIONAL	FREMONT
1628	OH	SGH	SPRINGFIELD-BECKLEY MUNI	SPRINGFIELD
1629	OH	TDZ	METCALF FIELD	TOLEDO
1630	OH	TOL	TOLEDO EXPRESS	TOLEDO
1631	OH	TSO	CARROLL COUNTY-TOLSON	CARROLLTON
1632	OH	TZR	BOLTON FIELD	COLUMBUS
1633	OH	UNI	OHIO UNIVERSITY SNYDER FIELD	ATHENS/ALBANY
1634	OH	USE	FULTON COUNTY	WAUSEON
1635	OH	UYF	MADISON COUNTY	LONDON
1636	OH	VES	DARKE COUNTY	VERSAILLES
1637	OH	VNW	VAN WERT COUNTY	VAN WERT
1638	OH	VTA	NEWARK-HEATH	NEWARK
1639	OH	YNG	YOUNGSTOWN-WARREN REGIONAL	YOUNGSTOWN/WARREN
1640	OH	ZZV	ZANESVILLE MUNI	ZANESVILLE
1641	OK	0F8	WILLIAM R. POGUE MUNI	SAND SPRINGS
1642	OK	1F0	ARDMORE DOWNTOWN EXECUTIVE	ARDMORE
1643	OK	2K4	SCOTT FIELD	MANGUM
1644	OK	37K	FALCONHEAD	BURNEYVILLE
1645	OK	3O4	SAYRE MUNI	SAYRE
1646	OK	4O4	MC CURTAIN COUNTY RGNL	IDABEL
1647	OK	ADH	ADA MUNI	ADA

1648	OK	ADM	ARDMORE MUNI	ARDMORE
1649	OK	AVK	ALVA RGNL	ALVA
1650	OK	AXS	ALTUS/QUARTZ MOUNTAIN RGNL	ALTUS
1651	OK	BVO	BARTLESVILLE MUNI	BARTLESVILLE
1652	OK	CHK	CHICKASHA MUNI	CHICKASHA
1653	OK	CLK	CLINTON REGIONAL	CLINTON
1654	OK	CSM	CLINTON-SHERMAN	CLINTON
1655	OK	DUA	EAKER FIELD	DURANT
1656	OK	DUC	HALLIBURTON FIELD	DUNCAN
1657	OK	ELK	ELK CITY MUNI	ELK CITY
1658	OK	F22	PERRY MUNI	PERRY
1659	OK	F29	CLARENCE E PAGE MUNI	OKLAHOMA CITY
1660	OK	F99	HOLDENVILLE MUNI	HOLDENVILLE
1661	OK	FDR	FREDERICK MUNI	FREDERICK
1662	OK	GAG	GAGE	GAGE
1663	OK	GCM	CLAREMORE RGNL	CLAREMORE
1664	OK	GMJ	GROVE MUNI	GROVE
1665	OK	GOK	GUTHRIE MUNI	GUTHRIE
1666	OK	GUY	GUYMON MUNI	GUYMON
1667	OK	H71	MID-AMERICA INDUSTRIAL	PRYOR
1668	OK	HBR	HOBART MUNI	HOBART
1669	OK	HHW	STAN STAMPER MUNI	HUGO
1670	OK	HSD	SUNDANCE AIRPARK	OKLAHOMA CITY
1671	OK	JSV	SALLISAW MUNI	SALLISAW
1672	OK	LAW	LAWTON-FORT SILL REGIONAL	LAWTON
1673	OK	MIO	MIAMI MUNI	MIAMI
1674	OK	MKO	DAVIS FIELD	MUSKOGEE
1675	OK	MLC	MC ALESTER REGIONAL	MC ALESTER
1676	OK	OJA	THOMAS P STAFFORD	WEATHERFORD
1677	OK	OKC	WILL ROGERS WORLD	OKLAHOMA CITY
1678	OK	OKM	OKMULGEE RGNL	OKMULGEE
1679	OK	OUN	UNIVERSITY OF OKLAHOMA WESTHEIMER	NORMAN
1680	OK	PNC	PONCA CITY RGNL	PONCA CITY

1681	OK	PVJ	PAULS VALLEY MUNI	PAULS VALLEY
1682	OK	PWA	WILEY POST	OKLAHOMA CITY
1683	OK	RKR	ROBERT S KERR	POTEAU
1684	OK	RVS	RICHARD LLOYD JONES JR	TULSA
1685	OK	SNL	SHAWNEE RGNL	SHAWNEE
1686	OK	SRE	SEMINOLE MUNI	SEMINOLE
1687	OK	SWO	STILLWATER RGNL	STILLWATER
1688	OK	TQH	TAHLEQUAH MUNI	TAHLEQUAH
1689	OK	TUL	TULSA INTL	TULSA
1690	OK	WDG	ENID WOODRING RGNL	ENID
1691	OK	WWR	WEST WOODWARD	WOODWARD
1692	OR	3S4	ILLINOIS VALLEY	CAVE JUNCTION
1693	OR	3S8	GRANTS PASS	GRANTS PASS
1694	OR	4S1	GOLD BEACH MUNI	GOLD BEACH
1695	OR	4S2	KEN JERNSTEDT AIRFIELD	HOOD RIVER
1696	OR	4S9	PORTLAND-MULINO	PORTLAND-MULINO
1697	OR	5S6	CAPE BLANCO STATE	DENMARK
1698	OR	9S9	LEXINGTON	LEXINGTON
1699	OR	AST	ASTORIA REGIONAL	ASTORIA
1700	OR	BKE	BAKER CITY MUNICIPAL	BAKER CITY
1701	OR	BNO	BURNS MUNI	BURNS
1702	OR	CVO	CORVALLIS MUNI	CORVALLIS
1703	OR	DLS	COLUMBIA GORGE REGIONAL/THE DALLES MUNI	THE DALLES
1704	OR	EUG	MAHLON SWEET FIELD	EUGENE
1705	OR	HIO	PORTLAND-HILLSBORO	PORTLAND
1706	OR	HRI	HERMISTON MUNI	HERMISTON
1707	OR	LGD	LA GRANDE/UNION COUNTY	LA GRANDE
1708	OR	LKV	LAKE COUNTY	LAKEVIEW
1709	OR	LMT	KLAMATH FALLS	KLAMATH FALLS
1710	OR	M50	BOARDMAN	BOARDMAN
1711	OR	MFR	ROGUE VALLEY INTERNATIONAL - MEDFORD	MEDFORD
1712	OR	MMV	MC MINNVILLE MUNI	MC MINNVILLE
1713	OR	ONO	ONTARIO MUNI	ONTARIO

1714	OR	ONP	NEWPORT MUNI	NEWPORT
1715	OR	OTH	NORTH BEND MUNI	NORTH BEND
1716	OR	PDT	EASTERN OREGON REGIONAL AT PENDLETON	PENDLETON
1717	OR	PDX	PORTLAND INTL	PORTLAND
1718	OR	RBG	ROSEBURG REGIONAL	ROSEBURG
1719	OR	RDM	ROBERTS FIELD	REDMOND
1720	OR	S03	ASHLAND MUNI-SUMNER PARKER FIELD	ASHLAND
1721	OR	S07	BEND MUNI	BEND
1722	OR	S12	ALBANY MUNI	ALBANY
1723	OR	S33	CITY-COUNTY	MADRAS
1724	OR	S47	TILLAMOOK	TILLAMOOK
1725	OR	SLE	MCNARY FLD	SALEM
1726	OR	SPB	SCAPPOOSE INDUSTRIAL AIRPARK	SCAPPOOSE
1727	OR	TTD	PORTLAND-TROUTDALE	PORTLAND
1728	OR	UAO	AURORA STATE	AURORA
1729	PA	1N9	ALLENTOWN QUEEN CITY MUNI	ALLENTOWN
1730	PA	29D	GROVE CITY	GROVE CITY
1731	PA	2G9	SOMERSET COUNTY	SOMERSET
1732	PA	40N	CHESTER COUNTY G O CARLSON	COATESVILLE
1733	PA	6G1	TITUSVILLE	TITUSVILLE
1734	PA	8G2	CORRY-LAWRENCE	CORRY
1735	PA	8G7	ZELIENOPLE MUNI	ZELIENOPLE
1736	PA	ABE	LEHIGH VALLEY INTERNATIONAL	ALLENTOWN
1737	PA	AFJ	WASHINGTON COUNTY	WASHINGTON
1738	PA	AGC	ALLEGHENY COUNTY	PITTSBURGH
1739	PA	AOO	ALTOONA-BLAIR COUNTY	ALTOONA
1740	PA	AVP	WILKES-BARRE/SCRANTON INTL	WILKES-BARRE/SCRANTON
1741	PA	AXQ	CLARION COUNTY	CLARION
1742	PA	BFD	BRADFORD REGIONAL	BRADFORD
1743	PA	BTP	BUTLER COUNTY/K W SCHOLTER FIELD	BUTLER
1744	PA	BVI	BEAVER COUNTY	BEAVER FALLS
1745	PA	CXY	CAPITAL CITY	HARRISBURG
1746	PA	ERI	ERIE INTL/TOM RIDGE FIELD	ERIE

1747	PA	FIG	CLEARFIELD-LAWRENCE	CLEARFIELD
1748	PA	FKL	VENANGO REGIONAL	FRANKLIN
1749	PA	FWQ	ROSTRAVER	MONONGAHELA
1750	PA	GKJ	PORT MEADVILLE	MEADVILLE
1751	PA	HMZ	BEDFORD COUNTY	BEDFORD
1752	PA	HZL	HAZLETON MUNI	HAZLETON
1753	PA	IDI	INDIANA COUNTY/JIMMY STEWART FLD/	INDIANA
1754	PA	IPT	WILLIAMSPORT RGNL	WILLIAMSPORT
1755	PA	JST	JOHN MURTHA JOHNSTOWN-CAMBRIA CO	JOHNSTOWN
1756	PA	LBE	ARNOLD PALMER REGIONAL	LATROBE
1757	PA	LHV	WILLIAM T. PIPER MEMORIAL	LOCK HAVEN
1758	PA	LNS	LANCASTER	LANCASTER
1759	PA	LOM	WINGS FIELD	PHILADELPHIA
1760	PA	MDT	HARRISBURG INTERNATIONAL	HARRISBURG
1761	PA	MPO	POCONO MOUNTAINS MUNI	MOUNT POCONO
1762	PA	N27	BRADFORD COUNTY	TOWANDA
1763	PA	N68	FRANKLIN COUNTY REGIONAL	CHAMBERSBURG
1764	PA	N70	PENNRIDGE	PERKASIE
1765	PA	N79	NORTHUMBERLAND COUNTY	SHAMOKIN
1766	PA	OYM	ST MARYS MUNI	ST MARYS
1767	PA	PHL	PHILADELPHIA INTL	PHILADELPHIA
1768	PA	PIT	PITTSBURGH INTERNATIONAL	PITTSBURGH
1769	PA	PNE	NORTHEAST PHILADELPHIA	PHILADELPHIA
1770	PA	PSB	MID-STATE	PHILIPSBURG
1771	PA	PTW	POTTSTOWN LIMERICK	POTTSTOWN
1772	PA	RDG	READING REGIONAL/CARL A SPAATZ FIELD	READING
1773	PA	RVL	MIFFLIN COUNTY	REEDSVILLE
1774	PA	SEG	PENN VALLEY	SELINSGROVE
1775	PA	THV	YORK	YORK
1776	PA	UCP	NEW CASTLE MUNI	NEW CASTLE
1777	PA	UNV	UNIVERSITY PARK	STATE COLLEGE
1778	PA	VVS	CONNELLSVILLE	CONNELLSVILLE
1779	PA	WAY	GREENE COUNTY	WAYNESBURG

1780	PA	WBW	WILKES-BARRE WYOMING VALLEY	WILKES-BARRE
1781	PA	ZER	SCHUYLKILL COUNTY /JOE ZERBEY/	POTTSVILLE
1782	RI	OQU	QUONSET STATE	NORTH KINGSTOWN
1783	RI	PVD	THEODORE FRANCIS GREEN STATE	PROVIDENCE
1784	RI	SFZ	NORTH CENTRAL STATE	PAWTUCKET
1785	RI	WST	WESTERLY STATE	WESTERLY
1786	SC	34A	LAURENS COUNTY	LAURENS
1787	SC	47J	CHERAW MUNI/LYNCH BELLINGER FIELD	CHERAW
1788	SC	50J	BERKELEY COUNTY	MONCKS CORNER
1789	SC	51J	LAKE CITY MUNI CJ EVANS FIELD	LAKE CITY
1790	SC	73J	BEAUFORT COUNTY	BEAUFORT
1791	SC	88J	ALLENDALE COUNTY	ALLENDALE
1792	SC	9A6	CHESTER CATAWBA REGIONAL	CHESTER
1793	SC	AIK	AIKEN MUNI	AIKEN
1794	SC	AND	ANDERSON RGNL	ANDERSON
1795	SC	BBP	MARLBORO COUNTY JETPORT - H.E. AVENT FIELD	BENNETTSVILLE
1796	SC	BNL	BARNWELL COUNTY	BARNWELL
1797	SC	CAE	COLUMBIA METROPOLITAN	COLUMBIA
1798	SC	CDN	WOODWARD FIELD	CAMDEN
1799	SC	CEU	OCONEE COUNTY REGIONAL	CLEMSON
1800	SC	CHS	CHARLESTON AFB/INTL	CHARLESTON
1801	SC	CKI	WILLIAMSBURG REGIONAL	KINGSTREE
1802	SC	CRE	GRAND STRAND	NORTH MYRTLE BEACH
1803	SC	CUB	COLUMBIA OWENS DOWNTOWN	COLUMBIA
1804	SC	DYB	SUMMERVILLE	SUMMERVILLE
1805	SC	FDW	FAIRFIELD COUNTY	WINNSBORO
1806	SC	FLO	FLORENCE REGIONAL	FLORENCE
1807	SC	GGE	GEORGETOWN COUNTY	GEORGETOWN
1808	SC	GMU	GREENVILLE DOWNTOWN	GREENVILLE
1809	SC	GRD	GREENWOOD COUNTY	GREENWOOD
1810	SC	GSP	GREENVILLE SPARTANBURG INTL -	GREER
1811	SC	GYH	DONALDSON CENTER	GREENVILLE
1812	SC	HVS	HARTSVILLE RGNL	HARTSVILLE

1813	SC	HXD	HILTON HEAD	HILTON HEAD ISLAND
1814	SC	HYW	CONWAY-HORRY COUNTY	CONWAY
1815	SC	JZI	CHARLESTON EXECUTIVE	CHARLESTON
1816	SC	LKR	LANCASTER COUNTY-MC WHIRTER FIELD	LANCASTER
1817	SC	LQK	PICKENS COUNTY	PICKENS
1818	SC	LRO	EAST COOPER	MOUNT PLEASANT
1819	SC	MAO	MARION COUNTY	MARION
1820	SC	MNI	SANTEE COOPER REGIONAL	MANNING
1821	SC	MYR	MYRTLE BEACH INTL	MYRTLE BEACH
1822	SC	OGB	ORANGEBURG MUNI	ORANGEBURG
1823	SC	RBW	LOWCOUNTRY RGNL	WALTERBORO
1824	SC	S19	MC CORMICK COUNTY	MC CORMICK
1825	SC	SMS	SUMTER	SUMTER
1826	SC	SPA	SPARTANBURG DOWNTOWN MEMORIAL	SPARTANBURG
1827	SC	UDG	DARLINGTON COUNTY JETPORT	DARLINGTON
1828	SC	UZA	ROCK HILL/YORK CO/BRYANT FIELD	ROCK HILL
1829	SD	0D8	GETTYSBURG MUNI	GETTYSBURG
1830	SD	9V9	CHAMBERLAIN MUNI	CHAMBERLAIN
1831	SD	ABR	ABERDEEN REGIONAL	ABERDEEN
1832	SD	ATY	WATERTOWN REGIONAL	WATERTOWN
1833	SD	BKX	BROOKINGS REGIONAL	BROOKINGS
1834	SD	BTN	BRITTON MUNI	BRITTON
1835	SD	FSD	JOE FOSS FIELD	SIOUX FALLS
1836	SD	HON	HURON REGIONAL	HURON
1837	SD	HSR	HOT SPRINGS MUNI	HOT SPRINGS
1838	SD	LEM	LEMMON MUNI	LEMMON
1839	SD	MBG	MOBRIDGE MUNI	MOBRIDGE
1840	SD	MDS	MADISON MUNI	MADISON
1841	SD	MHE	MITCHELL MUNI	MITCHELL
1842	SD	PHP	PHILIP	PHILIP
1843	SD	PIR	PIERRE REGIONAL	PIERRE
1844	SD	RAP	RAPID CITY REGIONAL	RAPID CITY
1845	SD	SFD	BOB WILEY FIELD	WINNER

1846	SD	SPF	BLACK HILLS-CLYDE ICE FIELD	SPEARFISH
1847	SD	VMR	HAROLD DAVIDSON FIELD	VERMILLION
1848	SD	YKN	CHAN GURNEY MUNI	YANKTON
1849	TN	0A3	SMITHVILLE MUNI	SMITHVILLE
1850	TN	0M1	SCOTT FIELD	PARSONS
1851	TN	0M2	REELFOOT LAKE	TIPTONVILLE
1852	TN	0M3	JOHN A BAKER FLD	HOHENWALD
1853	TN	0M4	BENTON COUNTY	CAMDEN
1854	TN	0M5	HUMPHREYS COUNTY	WAVERLY
1855	TN	1A3	MARTIN CAMPBELL FIELD	COPPERHILL
1856	TN	1A7	JACKSON COUNTY	GAINESBORO
1857	TN	1M5	PORTLAND MUNI	PORTLAND
1858	TN	2A0	MARK ANTON	DAYTON
1859	TN	2A1	JAMESTOWN MUNI	JAMESTOWN
1860	TN	2M2	LAWRENCEBURG-LAWRENCE COUNTY	LAWRENCEBURG
1861	TN	2M8	CHARLES W. BAKER	MILLINGTON
1862	TN	3M3	COLLEGEDALE MUNI	COLLEGEDALE
1863	TN	3M7	LAFAYETTE MUNI	LAFAYETTE
1864	TN	6A4	JOHNSON COUNTY	MOUNTAIN CITY
1865	TN	8A3	LIVINGSTON MUNI	LIVINGSTON
1866	TN	APT	MARION COUNTY-BROWN FIELD	JASPER
1867	TN	BGF	WINCHESTER MUNI	WINCHESTER
1868	TN	BNA	NASHVILLE INTL	NASHVILLE
1869	TN	CHA	LOVELL FIELD	CHATTANOOGA
1870	TN	CKV	OUTLAW FIELD	CLARKSVILLE
1871	TN	CSV	CROSSVILLE MEMORIAL-WHITSON FIELD	CROSSVILLE
1872	TN	DKX	KNOXVILLE DOWNTOWN ISLAND	KNOXVILLE
1873	TN	DYR	DYERSBURG REGIONAL	DYERSBURG
1874	TN	FYE	FAYETTE COUNTY	SOMERVILLE
1875	TN	FYM	FAYETTEVILLE MUNI	FAYETTEVILLE
1876	TN	GCY	GREENEVILLE-GREENE COUNTY MUNI	GREENEVILLE
1877	TN	GHM	CENTERVILLE MUNI	CENTERVILLE
1878	TN	GKT	GATLINBURG-PIGEON FORGE	SEVIERVILLE

1879	TN	GZS	ABERNATHY FIELD	PULASKI
1880	TN	HDI	HARDWICK FIELD	CLEVELAND
1881	TN	HZD	CARROLL COUNTY	HUNTINGDON
1882	TN	JAU	CAMPBELL COUNTY	JACKSBORO
1883	TN	JWN	JOHN C TUNE	NASHVILLE
1884	TN	LUG	ELLINGTON	LEWISBURG
1885	TN	M01	GENERAL DEWITT SPAIN	MEMPHIS
1886	TN	M02	DICKSON MUNI	DICKSON
1887	TN	M04	COVINGTON MUNI	COVINGTON
1888	TN	M08	WILLIAM L. WHITEHURST FIELD	BOLIVAR
1889	TN	M15	PERRY COUNTY	LINDEN
1890	TN	M29	HASSELL FIELD	CLIFTON
1891	TN	M31	ARNOLD FIELD	HALLS
1892	TN	M33	SUMNER COUNTY REGIONAL	GALLATIN
1893	TN	M52	FRANKLIN WILKINS	LEXINGTON
1894	TN	M53	HUMBOLDT MUNI	HUMBOLDT
1895	TN	M54	LEBANON MUNI	LEBANON
1896	TN	M91	SPRINGFIELD ROBERTSON COUNTY	SPRINGFIELD
1897	TN	M93	HOUSTON COUNTY	MC KINNON
1898	TN	MBT	MURFREESBORO MUNI	MURFREESBORO
1899	TN	MEM	MEMPHIS INTL	MEMPHIS
1900	TN	MKL	MC KELLAR-SIPES REGIONAL	JACKSON
1901	TN	MMI	MCMINN COUNTY	ATHENS
1902	TN	MNV	MONROE COUNTY	MADISONVILLE
1903	TN	MOR	MOORE-MURRELL	MORRISTOWN
1904	TN	MQY	SMYRNA	SMYRNA
1905	TN	MRC	MAURY COUNTY	COLUMBIA/MOUNT PLEASANT
1906	TN	NQA	MILLINGTON REGIONAL JETPORT	MILLINGTON
1907	TN	PHT	HENRY COUNTY	PARIS
1908	TN	RKW	ROCKWOOD MUNI	ROCKWOOD
1909	TN	RNC	WARREN COUNTY MEMORIAL	MC MINNVILLE
1910	TN	RVN	HAWKINS COUNTY	ROGERSVILLE
1911	TN	SCX	SCOTT MUNI	ONEIDA

1912	TN	SNH	SAVANNAH-HARDIN COUNTY	SAVANNAH
1913	TN	SRB	UPPER CUMBERLAND RGNL	SPARTA
1914	TN	SYI	BOMAR FIELD-SHELBYVILLE MUNI	SHELBYVILLE
1915	TN	SZY	ROBERT SIBLEY	SELMER
1916	TN	TGC	GIBSON COUNTY	TRENTON
1917	TN	THA	TULLAHOMA REGIONAL ARPT/WM NORTHERN FIELD	TULLAHOMA
1918	TN	TRI	TRI-CITIES RGNL TN/VA	BRISTOL/JOHNSON/KINGSPORT
1919	TN	TYS	MC GHEE TYSON	KNOXVILLE
1920	TN	UCY	EVERETT-STEWART	UNION CITY
1921	TX	07F	GLADEWATER MUNI	GLADEWATER
1922	TX	11R	BRENHAM MUNI	BRENHAM
1923	TX	20R	CRYSTAL CITY MUNI	CRYSTAL CITY
1924	TX	21F	JACKSBORO MUNI	JACKSBORO
1925	TX	2F5	LAMESA MUNI	LAMESA
1926	TX	3R0	BEEVILLE MUNI	BEEVILLE
1927	TX	3T5	FAYETTE REGIONAL AIR CENTER	LA GRANGE
1928	TX	45R	HAWTHORNE FIELD	KOUNTZE/SILSBEE
1929	TX	4F2	PANOLA COUNTY-SHARPE FIELD	CARTHAGE
1930	TX	4T2	KENNETH COPELAND	FORT WORTH
1931	TX	50R	LOCKHART MUNI	LOCKHART
1932	TX	5T9	MAVERICK COUNTY MEML INTL	EAGLE PASS
1933	TX	67R	RIO GRANDE CITY MUNI	RIO GRANDE CITY
1934	TX	6R3	CLEVELAND MUNI	CLEVELAND
1935	TX	6R6	TERRELL COUNTY	DRYDEN
1936	TX	7F3	CADDO MILLS MUNI	CADDO MILLS
1937	TX	7T3	GOLIAD COUNTY INDUSTRIAL AIRPARK	BERCLAIR
1938	TX	84R	SMITHVILLE CRAWFORD MUNI	SMITHVILLE
1939	TX	ABI	ABILENE REGIONAL	ABILENE
1940	TX	ACT	WACO REGIONAL	WACO
1941	TX	ADS	ADDISON	DALLAS
1942	TX	AFW	FORT WORTH ALLIANCE	FORT WORTH
1943	TX	ALI	ALICE INTL	ALICE
1944	TX	AMA	RICK HUSBAND AMARILLO INTL	AMARILLO

1945	TX	AQO	LLANO MUNI	LLANO
1946	TX	ARM	WHARTON RGNL	WHARTON
1947	TX	ASL	HARRISON COUNTY	MARSHALL
1948	TX	AUS	AUSTIN-BERGSTROM INTL	AUSTIN
1949	TX	AXH	HOUSTON-SOUTHWEST	HOUSTON
1950	TX	BAZ	NEW BRAUNFELS MUNI	NEW BRAUNFELS
1951	TX	BBD	CURTIS FIELD	BRADY
1952	TX	BFE	TERRY COUNTY	BROWNFIELD
1953	TX	BGD	HUTCHINSON COUNTY	BORGER
1954	TX	BKD	STEPHENS COUNTY	BRECKENRIDGE
1955	TX	BKS	BROOKS COUNTY	FALFURRIAS
1956	TX	BMQ	BURNET MUNI KATE CRADDOCK FIELD	BURNET
1957	TX	BMT	BEAUMONT MUNI	BEAUMONT
1958	TX	BPG	BIG SPRING MC MAHON-WRINKLE	BIG SPRING
1959	TX	BPT	SOUTHEAST TEXAS REGIONAL	BEAUMONT/PORT ARTHUR
1960	TX	BRO	BROWNSVILLE/SOUTH PADRE ISLAND INT'L	BROWNSVILLE
1961	TX	BWD	BROWNWOOD RGNL	BROWNWOOD
1962	TX	BYY	BAY CITY MUNI	BAY CITY
1963	TX	CDS	CHILDRESS MUNI	CHILDRESS
1964	TX	CFD	COULTER FIELD	BRYAN
1965	TX	CLL	EASTERWOOD FIELD	COLLEGE STATION
1966	TX	CNW	TSTC WACO	WACO
1967	TX	COM	COLEMAN MUNI	COLEMAN
1968	TX	COT	COTULLA-LA SALLE COUNTY	COTULLA
1969	TX	CPT	CLEBURNE MUNI	CLEBURNE
1970	TX	CRP	CORPUS CHRISTI INTL	CORPUS CHRISTI
1971	TX	CRS	C DAVID CAMPBELL FIELD-CORSICANA MUNI	CORSICANA
1972	TX	CXO	LONE STAR EXECUTIVE	HOUSTON
1973	TX	CZT	DIMMIT COUNTY	CARRIZO SPRINGS
1974	TX	DAL	DALLAS LOVE FIELD	DALLAS
1975	TX	DFW	DALLAS/FORT WORTH INTERNATIONAL	DALLAS-FORT WORTH
1976	TX	DHT	DALHART MUNI	DALHART
1977	TX	DKR	HOUSTON COUNTY	CROCKETT

1978	TX	DRT	DEL RIO INTL	DEL RIO
1979	TX	DTO	DENTON MUNI	DENTON
1980	TX	DUX	MOORE COUNTY	DUMAS
1981	TX	DWH	DAVID WAYNE HOOKS MEMORIAL	HOUSTON
1982	TX	E01	ROY HURD MEMORIAL	MONAHANS
1983	TX	E11	ANDREWS COUNTY	ANDREWS
1984	TX	E38	ALPINE-CASPARIS MUNICIPAL	ALPINE
1985	TX	E42	SPEARMAN MUNICIPAL	SPEARMAN
1986	TX	E48	UPTON COUNTY	MC CAMEY
1987	TX	EBG	EDINBURG INTL	EDINBURG
1988	TX	EFD	ELLINGTON FIELD	HOUSTON
1989	TX	ELP	EL PASO INTL	EL PASO
1990	TX	ERV	KERRVILLE MUNI/LOUIS SCHREINER FIELD	KERRVILLE
1991	TX	F00	JONES FIELD	BONHAM
1992	TX	F05	WILBARGER COUNTY	VERNON
1993	TX	F12	RUSK COUNTY	HENDERSON
1994	TX	F17	CENTER MUNI	CENTER
1995	TX	F21	MEMPHIS MUNI	MEMPHIS
1996	TX	F49	SLATON MUNI	SLATON
1997	TX	F83	ABERNATHY MUNI	ABERNATHY
1998	TX	FST	FORT STOCKTON-PECOS COUNTY	FORT STOCKTON
1999	TX	FTW	FORT WORTH MEACHAM INTL	FORT WORTH
2000	TX	FWS	FORT WORTH SPINKS	FORT WORTH
2001	TX	GGG	EAST TEXAS RGNL	LONGVIEW
2002	TX	GKY	ARLINGTON MUNI	ARLINGTON
2003	TX	GLE	GAINESVILLE MUNI	GAINESVILLE
2004	TX	GLS	SCHOLES INTL AT GALVESTON	GALVESTON
2005	TX	GNC	GAINES COUNTY	SEMINOLE
2006	TX	GPM	GRAND PRAIRIE MUNI	GRAND PRAIRIE
2007	TX	GTU	GEORGETOWN MUNI	GEORGETOWN
2008	TX	GVT	MAJORS	GREENVILLE
2009	TX	GYB	GIDDINGS-LEE COUNTY	GIDDINGS
2010	TX	GYI	GRAYSON COUNTY	SHERMAN/DENISON

2011	TX	HBV	JIM HOGG COUNTY	HEBBRONVILLE
2012	TX	HDO	HONDO MUNI	HONDO
2013	TX	HHF	HEMPHILL COUNTY	CANADIAN
2014	TX	HOU	WILLIAM P HOBBY	HOUSTON
2015	TX	HQZ	MESQUITE METRO	MESQUITE
2016	TX	HRL	VALLEY INTL	HARLINGEN
2017	TX	HRX	HEREFORD MUNI	HEREFORD
2018	TX	HYI	SAN MARCOS MUNI	SAN MARCOS
2019	TX	IAH	GEORGE BUSH INTERCONTINENTAL/HOUSTON	HOUSTON
2020	TX	IKG	KLEBERG COUNTY	KINGSVILLE
2021	TX	ILE	SKYLARK FIELD	KILLEEN
2022	TX	INK	WINKLER COUNTY	WINK
2023	TX	IWS	WEST HOUSTON	HOUSTON
2024	TX	JCT	KIMBLE COUNTY	JUNCTION
2025	TX	JSO	CHEROKEE COUNTY	JACKSONVILLE
2026	TX	JWY	MID-WAY RGNL	MIDLOTHIAN/WAXAHACHIE
2027	TX	LBB	LUBBOCK PRESTON SMITH INTL	LUBBOCK
2028	TX	LBX	BRAZORIA COUNTY	ANGLETON/LAKE JACKSON
2029	TX	LFK	ANGELINA COUNTY	LUFKIN
2030	TX	LHB	HEARNE MUNI	HEARNE
2031	TX	LLN	LEVELLAND MUNI	LEVELLAND
2032	TX	LNC	LANCASTER	LANCASTER
2033	TX	LRD	LAREDO INTL	LAREDO
2034	TX	LZZ	LAMPASAS	LAMPASAS
2035	TX	MAF	MIDLAND INTERNATIONAL	MIDLAND
2036	TX	MDD	MIDLAND AIRPARK	MIDLAND
2037	TX	MFE	MC ALLEN MILLER INTL	MC ALLEN
2038	TX	MKN	COMANCHE COUNTY-CITY	COMANCHE
2039	TX	MNZ	HAMILTON MUNI	HAMILTON
2040	TX	MRF	MARFA MUNI	MARFA
2041	TX	MWL	MINERAL WELLS	MINERAL WELLS
2042	TX	OCH	A L MANGHAM JR. REGIONAL	NACOGDOCHES
2043	TX	ODO	ODESSA-SCHLEMEYER FIELD	ODESSA

2044	TX	ONY	OLNEY MUNI	OLNEY
2045	TX	ORG	ORANGE COUNTY	ORANGE
2046	TX	OSA	MOUNT PLEASANT RGNL	MOUNT PLEASANT
2047	TX	OZA	OZONA MUNI	OZONA
2048	TX	PEQ	PECOS MUNI	PECOS
2049	TX	PEZ	PLEASANTON MUNI	PLEASANTON
2050	TX	PIL	PORT ISABEL-CAMERON COUNTY	PORT ISABEL
2051	TX	PKV	CALHOUN COUNTY	PORT LAVACA
2052	TX	PPA	PERRY LEFORS FIELD	PAMPA
2053	TX	PRX	COX FLD	PARIS
2054	TX	PSN	PALESTINE MUNI	PALESTINE
2055	TX	PSX	PALACIOS MUNI	PALACIOS
2056	TX	PVW	HALE COUNTY	PLAINVIEW
2057	TX	PWG	MC GREGOR EXECUTIVE	WACO
2058	TX	PYX	PERRYTON OCHILTREE COUNTY	PERRYTON
2059	TX	RBD	DALLAS EXECUTIVE	DALLAS
2060	TX	RBO	NUECES COUNTY	ROBSTOWN
2061	TX	RKP	ARANSAS CO	ROCKPORT
2062	TX	RPH	GRAHAM MUNI	GRAHAM
2063	TX	SAT	SAN ANTONIO INTL	SAN ANTONIO
2064	TX	SEP	CLARK FIELD MUNI	STEPHENVILLE
2065	TX	SGR	SUGAR LAND RGNL	HOUSTON
2066	TX	SJT	SAN ANGELO REGIONAL/MATHIS FIELD	SAN ANGELO
2067	TX	SLR	SULPHUR SPRINGS MUNI	SULPHUR SPRINGS
2068	TX	SNK	WINSTON FIELD	SNYDER
2069	TX	SPS	SHEPPARD AFB/WICHITA FALLS MUNI	WICHITA FALLS
2070	TX	SSF	STINSON MUNI	SAN ANTONIO
2071	TX	SWI	SHERMAN MUNI	SHERMAN
2072	TX	SWW	AVENGER FIELD	SWEETWATER
2073	TX	T17	NEW GULF	NEW GULF
2074	TX	T24	PINELAND MUNI	PINELAND
2075	TX	T41	LA PORTE MUNI	LA PORTE
2076	TX	T43	T P MC CAMPBELL	INGLESIDE

2077	TX	T47	KICKAPOO DOWNTOWN AIRPARK	WICHITA FALLS
2078	TX	T74	TAYLOR MUNI	TAYLOR
2079	TX	T77	PRESIDIO LELY INTL	PRESIDIO
2080	TX	T78	LIBERTY MUNI	LIBERTY
2081	TX	T82	GILLESPIE COUNTY	FREDERICKSBURG
2082	TX	T89	CASTROVILLE MUNI	CASTROVILLE
2083	TX	T90	CHAMBERS COUNTY-WINNIE STOWELL	WINNIE/STOWELL
2084	TX	TKI	COLLIN COUNTY REGIONAL AT MC KINNEY	MC KINNEY
2085	TX	TPL	DRAUGHON-MILLER CENTRAL TEXAS REGIONAL	TEMPLE
2086	TX	TRL	TERRELL MUNI	TERRELL
2087	TX	TYR	TYLER POUNDS RGNL	TYLER
2088	TX	UTS	HUNTSVILLE MUNI	HUNTSVILLE
2089	TX	UVA	GARNER FIELD	UVALDE
2090	TX	VCT	VICTORIA REGIONAL	VICTORIA
2091	TX	VHN	CULBERSON COUNTY	VAN HORN
2092	UT	1L9	PAROWAN	PAROWAN
2093	UT	36U	HEBER CITY MUNI - RUSS MCDONALD FIELD	HEBER
2094	UT	41U	MANTI-EPHRAIM	MANTI
2095	UT	74V	ROOSEVELT MUNI	ROOSEVELT
2096	UT	BCE	BRYCE CANYON	BRYCE CANYON
2097	UT	BDG	BLANDING MUNI	BLANDING
2098	UT	BMC	BRIGHAM CITY	BRIGHAM CITY
2099	UT	CDC	CEDAR CITY REGIONAL	CEDAR CITY
2100	UT	CNY	CANYONLANDS FIELD	MOAB
2101	UT	DTA	DELTA MUNI	DELTA
2102	UT	ENV	WENDOVER	WENDOVER
2103	UT	HVE	HANKSVILLE	HANKSVILLE
2104	UT	KNB	KANAB MUNI	KANAB
2105	UT	LGU	LOGAN-CACHE	LOGAN
2106	UT	MLF	MILFORD MUNI/BEN AND JUDY BRISCOE FIELD	MILFORD
2107	UT	OGD	OGDEN-HINCKLEY	OGDEN
2108	UT	PUC	CARBON COUNTY	PRICE
2109	UT	PVU	PROVO MUNI	PROVO

2110	UT	RIF	RICHFIELD MUNI	RICHFIELD
2111	UT	SGU	ST GEORGE MUNI	ST GEORGE
2112	UT	SLC	SALT LAKE CITY INTL	SALT LAKE CITY
2113	UT	TVY	BOLINDER FIELD-TOOELE VALLEY	TOOELE
2114	UT	U14	NEPHI MUNI	NEPHI
2115	UT	U19	FILLMORE MUNI	FILLMORE
2116	UT	U34	GREEN RIVER MUNI	GREEN RIVER
2117	UT	U42	SALT LAKE CITY MUNI 2	SALT LAKE CITY
2118	UT	U52	BEAVER MUNI	BEAVER
2119	UT	U55	PANGUITCH MUNI	PANGUITCH
2120	UT	U77	SPANISH FORK-SPRINGVILLE	SPANISH FORK
2121	UT	VEL	VERNAL	VERNAL
2122	VA	0VG	LEE COUNTY	JONESVILLE
2123	VA	6V3	TAZEWELL COUNTY	RICHLANDS
2124	VA	AKQ	WAKEFIELD MUNI	WAKEFIELD
2125	VA	AVC	MECKLENBURG-BRUNSWICK REGIONAL	SOUTH HILL
2126	VA	BCB	VIRGINIATECH/MONTGOMERY EXECUTIVE	BLACKSBURG
2127	VA	BKT	ALLEN C PERKINSON/BAAF	BLACKSTONE
2128	VA	CHO	CHARLOTTESVILLE-ALBEMARLE	CHARLOTTESVILLE
2129	VA	CJR	CULPEPER REGIONAL	CULPEPER
2130	VA	CPK	CHESAPEAKE REGIONAL	NORFOLK
2131	VA	DAN	DANVILLE REGIONAL	DANVILLE
2132	VA	EMV	EMPORIA-GREENSVILLE REGIONAL	EMPORIA
2133	VA	FCI	CHESTERFIELD COUNTY	RICHMOND
2134	VA	FKN	FRANKLIN MUNI-JOHN BEVERLY ROSE	FRANKLIN
2135	VA	FRR	FRONT ROYAL-WARREN COUNTY	FRONT ROYAL
2136	VA	FVX	FARMVILLE REGIONAL	FARMVILLE
2137	VA	FYJ	MIDDLE PENINSULA RGNL	WEST POINT
2138	VA	HEF	MANASSAS REGIONAL/HARRY P. DAVIS FIELD	MANASSAS
2139	VA	HLX	TWIN COUNTY	GALAX HILLSVILLE
2140	VA	HSP	INGALLS FIELD	HOT SPRINGS
2141	VA	JYO	LEESBURG EXECUTIVE	LEESBURG
2142	VA	LKU	LOUISA COUNTY/FREEMAN FIELD	LOUISA

2143	VA	LNP	LONESOME PINE	WISE
2144	VA	LYH	LYNCHBURG REGIONAL/PRESTON GLENN FLD	LYNCHBURG
2145	VA	MFV	ACCOMACK COUNTY	MELFA
2146	VA	MKJ	MOUNTAIN EMPIRE	MARION/WYTHEVILLE
2147	VA	MTV	BLUE RIDGE	MARTINSVILLE
2148	VA	OFP	HANOVER COUNTY MUNI	RICHMOND/ASHLAND
2149	VA	OKV	WINCHESTER REGIONAL	WINCHESTER
2150	VA	OMH	ORANGE COUNTY	ORANGE
2151	VA	ORF	NORFOLK INTL	NORFOLK
2152	VA	PHF	NEWPORT NEWS/WILLIAMSBURG INTERNATIONAL	NEWPORT NEWS
2153	VA	PSK	NEW RIVER VALLEY	DUBLIN
2154	VA	PTB	DINWIDDIE COUNTY	PETERSBURG
2155	VA	RIC	RICHMOND INTERNATIONAL	RICHMOND
2156	VA	RMN	STAFFORD REGIONAL	STAFFORD
2157	VA	ROA	ROANOKE REGIONAL/WOODRUM FIELD	ROANOKE
2158	VA	SFQ	SUFFOLK EXECUTIVE	SUFFOLK
2159	VA	SHD	SHENANDOAH VALLEY REGIONAL	STAUNTON/WAYNESBORO/HARRISONBURG
2160	VA	VJI	VIRGINIA HIGHLANDS	ABINGDON
2161	VA	W45	LURAY CAVERNS	LURAY
2162	VA	W66	WARRENTON-FAUQUIER	WARRENTON
2163	VA	W78	WILLIAM M TUCK	SOUTH BOSTON
2164	VA	W96	NEW KENT COUNTY	QUINTON
2165	VT	BTV	BURLINGTON INTL	BURLINGTON
2166	VT	DDH	WILLIAM H. MORSE STATE	BENNINGTON
2167	VT	EFK	NEWPORT STATE	NEWPORT
2168	VT	MPV	EDWARD F KNAPP STATE	BARRE/MONTPELIER
2169	VT	MVL	MORRISVILLE-STOWE STATE	MORRISVILLE
2170	VT	RUT	RUTLAND STATE	RUTLAND
2171	VT	VSF	HARTNESS STATE (SPRINGFIELD)	SPRINGFIELD
2172	WA	0S9	JEFFERSON COUNTY INTL	PORT TOWNSEND
2173	WA	3W7	GRAND COULEE DAM	ELECTRIC CITY
2174	WA	ALW	WALLA WALLA REGIONAL	WALLA WALLA
2175	WA	AWO	ARLINGTON MUNI	ARLINGTON

2176	WA	BFI	BOEING FIELD/KING COUNTY INTL	SEATTLE
2177	WA	BLI	BELLINGHAM INTL	BELLINGHAM
2178	WA	BVS	SKAGIT REGIONAL	BURLINGTON/MOUNT VERNON
2179	WA	CLM	WILLIAM R FAIRCHILD INTL	PORT ANGELES
2180	WA	CLS	CHEHALIS-CENTRALIA	CHEHALIS
2181	WA	DEW	DEER PARK	DEER PARK
2182	WA	EAT	PANGBORN MEMORIAL	WENATCHEE
2183	WA	ELN	BOWERS FIELD	ELLENSBURG
2184	WA	EPH	EPHRATA MUNI	EPHRATA
2185	WA	FHR	FRIDAY HARBOR	FRIDAY HARBOR
2186	WA	GEG	SPOKANE INTL	SPOKANE
2187	WA	HQM	BOWERMAN	HOQUIAM
2188	WA	KLS	KELSO-LONGVIEW	KELSO
2189	WA	MWH	GRANT CO INTL	MOSES LAKE
2190	WA	OLM	OLYMPIA	OLYMPIA
2191	WA	OMK	OMAK	OMAK
2192	WA	PAE	SNOHOMISH COUNTY (PAINE FLD)	EVERETT
2193	WA	PSC	TRI-CITIES	PASCO
2194	WA	PUW	PULLMAN/MOSCOW REGIONAL	PULLMAN/MOSCOW -ID
2195	WA	PWT	BREMERTON NATIONAL	BREMERTON
2196	WA	RLD	RICHLAND	RICHLAND
2197	WA	RNT	RENTON MUNI	RENTON
2198	WA	S50	AUBURN MUNI	AUBURN
2199	WA	S52	METHOW VALLEY STATE	WINTHROP
2200	WA	S98	VISTA FIELD	KENNEWICK
2201	WA	SEA	SEATTLE-TACOMA INTL	SEATTLE
2202	WA	SFF	FELTS FIELD	SPOKANE
2203	WA	SHN	SANDERSON FIELD	SHELTON
2204	WA	TDO	ED CARLSON MEMORIAL FIELD - SOUTH LEWIS CO	TOLEDO
2205	WA	TIW	TACOMA NARROWS	TACOMA
2206	WA	UIL	QUILLAYUTE	QUILLAYUTE
2207	WA	YKM	YAKIMA AIR TERMINAL/MCALLISTER FIELD	YAKIMA
2208	WI	3CU	CABLE UNION	CABLE

2209	WI	3WO	SHAWANO MUNI	SHAWANO
2210	WI	4R5	MADELINE ISLAND	LA POINTE
2211	WI	57C	EAST TROY MUNI	EAST TROY
2212	WI	82C	MAUSTON-NEW LISBON UNION	NEW LISBON
2213	WI	8D1	NEW HOLSTEIN MUNI	NEW HOLSTEIN
2214	WI	AHH	AMERY MUNI	AMERY
2215	WI	AIG	LANGLADE COUNTY	ANTIGO
2216	WI	ARV	LAKELAND/NOBLE F. LEE MEMORIAL FIELD	MINOCQUA-WOODRUFF
2217	WI	ASX	JOHN F KENNEDY MEMORIAL	ASHLAND
2218	WI	ATW	OUTAGAMIE COUNTY REGIONAL	APPLETON
2219	WI	AUW	WAUSAU DOWNTOWN	WAUSAU
2220	WI	BCK	BLACK RIVER FALLS AREA	BLACK RIVER FALLS
2221	WI	BUU	BURLINGTON MUNI	BURLINGTON
2222	WI	C02	GRAND GENEVA RESORT	LAKE GENEVA
2223	WI	C29	MIDDLETON MUNI - MOREY FIELD	MIDDLETON
2224	WI	C35	REEDSBURG MUNI	REEDSBURG
2225	WI	C59	LAKE LAWN	DELAVAN
2226	WI	CLI	CLINTONVILLE MUNI	CLINTONVILLE
2227	WI	CMY	SPARTA/FORT MC COY	SPARTA
2228	WI	CWA	CENTRAL WISCONSIN	MOSINEE
2229	WI	DLL	BARABOO WISCONSIN DELLS	BARABOO
2230	WI	EAU	CHIPPEWA VALLEY REGIONAL	EAU CLAIRE
2231	WI	EFT	MONROE MUNI	MONROE
2232	WI	EGV	EAGLE RIVER UNION	EAGLE RIVER
2233	WI	ENW	KENOSHA REGIONAL	KENOSHA
2234	WI	ETB	WEST BEND MUNI	WEST BEND
2235	WI	FLD	FOND DU LAC COUNTY	FOND DU LAC
2236	WI	GRB	AUSTIN STRAUBEL INTERNATIONAL	GREEN BAY
2237	WI	HYR	SAWYER COUNTY	HAYWARD
2238	WI	ISW	ALEXANDER FIELD SOUTH WOOD COUNTY	WISCONSIN RAPIDS
2239	WI	JVL	SOUTHERN WISCONSIN REGIONAL	JANESVILLE
2240	WI	LNL	KINGS LAND O' LAKES	LAND O' LAKES
2241	WI	LNR	TRI-COUNTY REGIONAL	LONE ROCK

2242	WI	LSE	LA CROSSE MUNI	LA CROSSE
2243	WI	LUM	MENOMONIE MUNICIPAL-SCORE FIELD	MENOMONIE
2244	WI	MDZ	TAYLOR COUNTY	MEDFORD
2245	WI	MFI	MARSHFIELD MUNI	MARSHFIELD
2246	WI	MKE	GENERAL MITCHELL INTERNATIONAL	MILWAUKEE
2247	WI	MRJ	IOWA COUNTY	MINERAL POINT
2248	WI	MSN	DANE COUNTY REGIONAL-TRUAX FIELD	MADISON
2249	WI	MTW	MANITOWOC COUNTY	MANITOWOC
2250	WI	MWC	LAWRENCE J TIMMERMAN	MILWAUKEE
2251	WI	OCQ	J. DOUGLAS BAKE MEML.	OCONTO
2252	WI	OEO	L O SIMENSTAD MUNI	OSCEOLA
2253	WI	OSH	WITTMAN REGIONAL	OSHKOSH
2254	WI	OVS	BOSCOBEL	BOSCOBEL
2255	WI	PBH	PRICE COUNTY	PHILLIPS
2256	WI	PCZ	WAUPACA MUNI	WAUPACA
2257	WI	PDC	PRAIRIE DU CHIEN MUNI	PRAIRIE DU CHIEN
2258	WI	PVB	PLATTEVILLE MUNICIPAL	PLATTEVILLE
2259	WI	RAC	JOHN H BATTEN	RACINE
2260	WI	RCX	RUSK COUNTY	LADYSMITH
2261	WI	RHI	RHINELANDER-ONEIDA COUNTY	RHINELANDER
2262	WI	RNH	NEW RICHMOND REGIONAL	NEW RICHMOND
2263	WI	RPD	RICE LAKE REGIONAL - CARL'S FIELD	RICE LAKE
2264	WI	RRL	MERRILL MUNI	MERRILL
2265	WI	RYV	WATERTOWN MUNI	WATERTOWN
2266	WI	RZN	BURNETT COUNTY	SIREN
2267	WI	SBM	SHEBOYGAN COUNTY MEMORIAL	SHEBOYGAN
2268	WI	STE	STEVENS POINT MUNI	STEVENS POINT
2269	WI	SUE	DOOR COUNTY CHERRYLAND	STURGEON BAY
2270	WI	SUW	RICHARD I BONG	SUPERIOR
2271	WI	TKV	TOMAHAWK REGIONAL	TOMAHAWK
2272	WI	UBE	CUMBERLAND MUNI	CUMBERLAND
2273	WI	UES	WAUKESHA COUNTY	WAUKESHA
2274	WI	UNU	DODGE COUNTY	JUNEAU

2275	WI	Y72	BLOYER FIELD	TOMAH
2276	WV	3I2	MASON COUNTY	POINT PLEASANT
2277	WV	6L4	LOGAN COUNTY	LOGAN
2278	WV	BKW	RALEIGH COUNTY MEMORIAL	BECKLEY
2279	WV	BLF	MERCER COUNTY	BLUEFIELD
2280	WV	CKB	HARRISON/MARION REGIONAL	CLARKSBURG
2281	WV	CRW	YEAGER	CHARLESTON
2282	WV	EKN	ELKINS-RANDOLPH CO-JENNINGS RANDOLPH FLD	ELKINS
2283	WV	HLG	WHEELING OHIO CO	WHEELING
2284	WV	HTS	TRI-STATE/MILTON J. FERGUSON FIELD	HUNTINGTON
2285	WV	I18	JACKSON COUNTY	RAVENSWOOD
2286	WV	LWB	GREENBRIER VALLEY	LEWISBURG
2287	WV	MGW	MORGANTOWN MUNI-WALTER L. BILL HART FLD	MORGANTOWN
2288	WV	MRB	EASTERN WV REGIONAL/SHEPHERD FLD	MARTINSBURG
2289	WV	PKB	MID-OHIO VALLEY REGIONAL	PARKERSBURG
2290	WV	W22	UPSHUR COUNTY RGNL	BUCKHANNON
2291	WV	W99	GRANT COUNTY	PETERSBURG
2292	WY	82V	PINE BLUFFS MUNI	PINE BLUFFS
2293	WY	BPI	BIG PINEY-MARBLETON	BIG PINEY
2294	WY	BYG	JOHNSON COUNTY	BUFFALO
2295	WY	COD	YELLOWSTONE REGIONAL	CODY
2296	WY	CPR	NATRONA COUNTY INTL	CASPER
2297	WY	CYS	CHEYENNE REGIONAL/JERRY OLSON FIELD	CHEYENNE
2298	WY	DGW	CONVERSE COUNTY	DOUGLAS
2299	WY	EAN	PHIFER AIRFIELD	WHEATLAND
2300	WY	ECS	MONDELL FIELD	NEWCASTLE
2301	WY	EMM	KEMMERER MUNI	KEMMERER
2302	WY	EVW	EVANSTON-UINTA COUNTY BURNS FIELD	EVANSTON
2303	WY	GCC	GILLETTE-CAMPBELL COUNTY	GILLETTE
2304	WY	GEY	SOUTH BIG HORN COUNTY	GREYBULL
2305	WY	JAC	JACKSON HOLE	JACKSON
2306	WY	LAR	LARAMIE REGIONAL	LARAMIE
2307	WY	LND	HUNT FIELD	LANDER

2308	WY	LSK	LUSK MUNI	LUSK
2309	WY	PNA	RALPH WENZ FIELD	PINEDALE
2310	WY	POY	POWELL MUNI	POWELL
2311	WY	RIW	RIVERTON REGIONAL	RIVERTON
2312	WY	RKS	ROCK SPRINGS-SWEETWATER COUNTY	ROCK SPRINGS
2313	WY	RWL	RAWLINS MUNI/HARVEY FIELD	RAWLINS
2314	WY	SAA	SHIVELY FIELD	SARATOGA
2315	WY	SHR	SHERIDAN COUNTY	SHERIDAN
2316	WY	THP	HOT SPRINGS CO-THERMOPOLIS MUNI	THERMOPOLIS
2317	WY	TOR	TORRINGTON MUNI	TORRINGTON
2318	WY	U68	NORTH BIG HORN COUNTY	COWLEY/LOVELL/BYRON
2319	WY	W43	HULETT MUNICIPAL	HULETT
2320	WY	WRL	WORLAND MUNI	WORLAND

APPENDIX G AIRCRAFT TYPE AND WEIGHT

CLASS LIST

Type Code	Weight Class	Aircraft Type
007	S	Aero Commander 200
008	S	Aero Macchi Al-60
009	S	Aeronca 7-Ac
010	S	Beech Bonanza 35a/C/D/E/G/H/J/K/S/V/ 36a
024	S	Beech B-23 Musketeer
029	S	Cessna 150/152
030	S	Cessna 180
031	S	Cessna 180a/B
032	S	Cessna 180c/D/E/F
033	S	Cessna 185a/B/C Skywagon
035	S	Cessna C206/207/209/210 Stationair
036	S	Cessna 172 Skyhawk
037	S	Cessna 195
038	S	Cessna 177 Cardinal
039	S	Cessna 182 Skylane
040	S	Dehavilland Dhc2 Beaver
041	S	Cessna 205
042	L	Dehavilland Dhc3 Otter
079	S	Piper Pa-32 (Cherokee 6)
080	S	Piper Pa-18 (Super-Cub)
081	S	Piper Pa-14 (Family-Cruiser)
082	S	Piper Pa-22 (Tri-Pacer)
083	S	Piper Pa-24 (Comanche)
084	S	Piper Pa-28 (Cherokee)
086	S	Piper Pa-12 (Supercruiser)
105	S	Beech C-50 (Twin Bonanza)
111	S	Beech King Air C-90
113	S	Beech B-95 (Travelair)
117	S	Beech Baron (55 Series)
121	S	Cessna C-421
122	S	Cessna C-310 Series
125	S	Cessna C-402/402a
128	S	Cessna 404
131	S	Pilatus Britten-Norman Bn2/A Islander
133	S	Beech 65/65a-80/65b-80 (Queen Air)
140	L	Convair Cv-240
143	L	Convair Cv-340/440
148	S	Cessna C-337 (Super Sky Master)
149	S	Cessna C-401
152	S	Cessna C-411

153	S	Dehavilland Dhc4 Caribou
160	S	Mcdonnell Douglas Dc-3/A/C,c-47/B
166	S	Cessna C-336
190	S	Piper Pa-23-250 (Aztec/Apache)
194	S	Piper Pa-31 (Navajo)/T-1020
201	S	Pilatus Britten-Norman Bn2a Trislander
210	L	Mcdonnell Douglas Dc-4(C54/C54a/C54b/C54e)
216	L	Mcdonnell Douglas Dc-6
218	L	Mcdonnell Douglas Dc-6a
220	L	Mcdonnell Douglas Dc-6b
225	L	Mcdonnell Douglas Dc-7a/B
228	L	Mcdonnell Douglas Dc-7c
403	S	Beech 99 Airliner
404	S	Beech C99
405	S	Beech 1900 A/B/C
406	S	Beech 200 Super Kingair
407	L	British Aerospace (Hawker-Siddeley) Bae-748
408	L	British Aerospace Bae-Atp
411	S	Beech King Air 65-A90 Queen Air
416	S	Cessna 208 Caravan
417	S	Cessna 406 Caravan Ii
418	S	Cessna C-441
420	L	Convair Cv-540
430	L	Convair Cv-580
435	L	Convair Cv-600
440	L	Convair Cv-640
441	L	Aerospatiale/Aeritalia Atr-42
442	L	Aerospatiale/Aeritalia Atr-72
443	L	Antonov 12
445	L	Convair Cv-660
448	S	Dornier 228
449	S	Dornier 328
450	L	Fokker Friendship F-27/Fairchild F-27/A/B/F/J
452	L	Fokker 50
456	S	Saab-Fairchild 340/B
457	S	Beech King Air C-90
458	S	Beechcraft Super King Air
459	S	Saab-Fairchild 340/A
461	S	Embraer Emb-120 Brasilia
462	S	Swearingen Metro Merlin
464	S	Embraer Emb-110 Bandeirante
466	S	Swearingen Metro Ii
467	S	Swearingen Metro Iii
469	S	British Aerospace Jetstream 31
471	S	British Aerospace Jetstream 41
473	S	Gulfstream I-Commander
476	S	Piper Pa-31t Cheyenne Ii Xl

482	L	Dehavilland Dhc8-400 Dash-8
483	L	Dehavilland Dhc8-100 Dash-8
484	L	Dehavilland Dhc8-300 Dash 8
485	S	Dehavilland Twin Otter Dhc-6
486	S	Shorts Harland Sc-7 Skyvan
487	S	Shorts 330
488	L	Carstedt Cj-600a
489	S	Shorts 360
491	L	Dehavilland Dhc8-200q Dash-8
507	L	Antonov 12
520	L	Canadair Cl-44d
601	L	Fokker F28-1000 Fellowship
602	L	Fokker F28-4000/6000 Fellowship
603	L	Fokker 100
604	L	Fokker 70
608	L	Boeing 717-200
612	L	Boeing 737-700/700lr
614	L	Boeing 737-800
615	L	Boeing 737-5/600lr
616	L	Boeing 737-500
617	L	Boeing 737-400
618	L	Boeing 737-300lr
619	L	Boeing 737-300
620	L	Boeing 737-100/200
621	L	Boeing 737-200c
622	757	Boeing 757-200
623	757	Boeing 757-300
624	H	Boeing 767-400
625	H	Boeing 767-200/200er
626	H	Boeing 767-300/300er
627	H	Boeing 777
628	L	Canadair Rj-100/Rj-100er
629	L	Canadair Rj-200er /Rj-440
630	L	Mcdonnell Douglas Dc-9-10
631	L	Canadair Rj-700
632	S	Dornier 328 Jet
633	L	Boeing 737-600
634	L	Boeing737-900
635	L	Mcdonnell Douglas Dc-9-15f
636	S	Cessna Citation Ii
637	S	Cessna Citation Ii
638	L	Canadair Rj 900
639	S	Cessna Citationjet/Cj1/Cj
640	L	Mcdonnell Douglas Dc-9-30
644	L	Airbus Industrie A-318
645	L	Mcdonnell Douglas Dc-9-40
646	S	Cessna Citation Iii Model 650

650	L	Mcdonnell Douglas Dc-9-50
654	L	Mcdonnell Douglas Dc9 Super 87
655	L	Mcdonnell Douglas Dc9 Super 80/Md81/2/3/7/8
656	L	Mcdonnell Douglas Md-90
667	L	Gulfstream V/ G-V Exec/ G-5/ 550
668	L	Canadair (Bombardier) Challlenger 601
672	L	Embraer Erj-145 Xr
673	L	Embraer Erj-175
674	L	Embraer-135
675	L	Embraer-145
676	L	Embraer-140
677	L	Embraer 170
678	L	Embraer 190
681	S	Dassault-Breguet Mystere-Falcon
687	H	Airbus A330-300
689	H	Airbus Industrie A330-600st (Beluga)
690	H	Airbus Industrie A300b/C/F-100/200
691	H	Airbus Industrie A300-600/R/Cf/Rcf
692	H	Airbus Industrie A310-200c/F
693	H	Airbus Industrie A310-300
694	L	Airbus Industrie A320-100/200
695	H	Airbus Industrie A300-B2
696	H	Airbus Industrie A330-200
697	H	Airbus Industrie A340
698	L	Airbus Industrie A319
699	L	Airbus Industrie A321
710	L	Boeing 727-100
711	L	Boeing 727-100c/Qc
715	L	Boeing 727-200/231a
730	H	Mcdonnell Douglas Dc-10-10
731	H	Mcdonnell Douglas Dc-10-20
732	H	Mcdonnell Douglas Dc-10-30
733	H	Mcdonnell Douglas Dc-10-40
735	H	Mcdonnell Douglas Dc-10-30cf
740	H	Mcdonnell Douglas Md-11
760	H	Lockheed L-1011-1/100/200
765	H	Lockheed L-1011-500 Tristar
770	L	Dassault Falcon 900
771	S	Dassault Falcon 50
780	L	Tupolev Tu-154
800	H	Boeing 707-100
802	H	Boeing 707-100b
804	H	Boeing 707-200
806	H	Boeing 707-300
808	H	Boeing 707-300b
809	H	Boeing 707-300c
810	H	Boeing 707-400

812	L	Boeing 720-000
814	L	Boeing 720-000b
816	H	Boeing 747-100
817	H	Boeing 747-200/300
818	H	Boeing 747c
819	H	Boeing 747-400
820	H	Boeing 747f
822	H	Boeing 747sp
830	L	Convair 990 Coronado (Cv-30)
835	L	Avroliner Rj85
840	H	Mcdonnell Douglas Dc-8-10
842	H	Mcdonnell Douglas Dc-8-20
844	H	Mcdonnell Douglas Dc-8-30
846	H	Mcdonnell Douglas Dc-8-40
848	H	Mcdonnell Douglas Dc-8-50
850	H	Mcdonnell Douglas Dc-8-50f
851	H	Mcdonnell Douglas Dc-8-61
852	H	Mcdonnell Douglas Dc-8-63f
854	H	Mcdonnell Douglas Dc-8-62
856	H	Mcdonnell Douglas Dc-8-63
860	H	Mcdonnell Douglas Dc-8-71
862	H	Mcdonnell Douglas Dc-8-72
864	H	Mcdonnell Douglas Dc-8-73
865	H	Mcdonnell Douglas Dc-8-73f
866	L	British Aerospace Bae-146-100/Rj70
867	L	British Aerospace Bae-146-200
868	L	British Aerospace Bae-146-300
871	H	Airbus Industrie A340-300
872	H	Airbus Industrie A340-500
873	H	Airbus Industrie A340-200
874	H	Airbus Industrie A340-600
875	H	Aerospatiale/British Aerospace Concorde
876	H	Ilyushin 62
877	H	Ilyushin 76/Td
878	H	Ilyushin 86
879	H	Ilyushin 96
880	H	Antonov 124
890	H	Antonov 225 (6 Engine)