GENERATION OF REALISTIC AIR TRAFFIC SCENARIOS USING A GENETIC ALGORITHM

Robert D. Oaks, Signal Corporation

Mike Paglione, Federal Aviation Administration, William J. Hughes Technical Center, Atlantic City International Airport, NJ 08405

Abstract

Traffic flow management decision support tools such as the User Request Evaluation Tool (URET), developed by the MITRE Center for Advanced Aviation Systems Development, and the Center-TRACON Automation System (CTAS), developed by the National Aeronautics and Space Administration/Ames Research Center, use simulation as a tool for development, technical assessment, and field evaluation. Air traffic scenarios based on recorded live data are used to test these decision support tools. Frequently the scenarios need to be modified in order to create aircraft-to-aircraft encounters and conflicts that are not present in the live data. This paper presents an implementation of a genetic algorithm that is being used to time shift the flights within an air traffic scenario to create encounters with specific constrained characteristics. These constraints are the distributions of the horizontal and vertical closest points of approach, the encounter angle at the closest point of horizontal approach, and the vertical type of encounter. This paper describes how the genetic algorithm was implemented, including a description of the solution chromosome and of the fitness function used to measure the potential solutions. After describing the implementation a specific example of its use is presented.

Introduction

Both the User Request Evaluation Tool (URET), developed by the MITRE Center for Advanced Aviation Systems Development, and the Center-TRACON Automation System (CTAS), developed by the National Aeronautics and Space Administration/Ames Research Center, are decision support tools (DSTs) that support en route air traffic controllers. Each has a conflict probe function that predicts aircraft-to-aircraft and aircraft-to-airspace conflicts.

In 1996 the Federal Aviation Administration's Traffic Flow Management Branch (ACT-250) established the Conflict Probe Assessment Team (CPAT) to evaluate the accuracy of the conflict probes in these DSTs. In 2002, CPAT became a part of the Simulation and Modeling Group (ACB-330). Over the past six years CPAT has measured the conflict prediction accuracy of URET [1], measured the trajectory modeling accuracy of both URET and CTAS [2], and assisted in the accuracy testing of URET Current Capability Limited Deployment (CCLD) [3, 4], which is the operational implementation of URET.

Air Traffic Scenarios

For each of these tasks CPAT used air traffic scenarios, which are data files describing the flow of aircraft traffic over a period of time. The files contain time-stamped planning and advisory information and track data. The planning and advisory information describe the aircraft's planned flight; which includes its flight plan and flight plan amendments, interim altitude clearances, and hold information. The track data represents the aircraft's actual flight path. It consists of several fields including the flight's time-stamped horizontal coordinates and altitude.

Encounters and Conflicts

An aircraft-to-aircraft encounter is an instance when the relative spatial distance between two aircraft is less than some parametric value. This distance is usually specified in two dimensions: its projection onto a horizontal plane and its projection onto a vertical axis. The values for defining an encounter in this paper are 25 nautical miles (nm) in the horizontal plane and 5000 feet vertically.

An aircraft-to-aircraft conflict is an aircraft-to-aircraft encounter for which these horizontal and vertical distances also violate published air traffic control standards. In en route airspace the horizontal separation standard is 5 nm and the vertical separation standard is either 2000 feet if both aircraft are above FL290 or 1000 feet if one or both aircraft are below FL290. Since encounters and conflicts, as defined in this paper, differ only with regards to distance parameters, the terms are used interchangeably.

Time Shifting

Two specific requirements for the URET CCLD accuracy testing were that the air traffic scenarios had to be based on recorded field data and that these scenarios had to contain a specified minimum number of encounters and conflicts [5].

Recorded field data will contain aircraft-to-aircraft encounters, but under normal operating conditions this data will not contain aircraft-to-aircraft conflicts. In order to meet the URET CCLD accuracy test requirement, CPAT time shifted the flights in the recorded field data.

This time shifting consisted of determining a flight specific time increment that was added to all the events associated with the flight. This caused each flight to follow it's recorded flight profile, but at a different time. This caused aircraft-to-aircraft encounters and conflicts to occur in the scenarios that did not exist in the field.

For the URET CCLD accuracy scenarios CPAT developed software that calculated these time increments using time compression and random time adjustment. For time compression the time increment is derived by multiplying a constant times the difference between a flight's start time and a base time that precedes all the start times in the scenario. For random time adjustment the time increment is randomly selected. A more detailed description of these techniques and an overview of CPAT's scenario generation process are presented in Reference [6].

This approach was satisfactory for the URET CCLD accuracy testing, but CPAT realized that the distribution of key encounter parameters (e.g., encounter angle) was not controlled by these techniques. In order to control these parameters CPAT investigated the feasibility of using a genetic algorithm to determine a set of delta times (i.e., flight specific time increments) that can be applied to the flights in a scenario so that the distribution of aircraft-to-aircraft encounters and conflicts meets user defined distribution constraints. The results of this investigation are documented in Reference [7].

The Genetic Algorithm

The genetic algorithm (GA) was invented by John Holland at the University of Michigan in the 1960s and 1970s. GAs are a specific case of a broad class of algorithms called *Random Heuristic Search* [8] algorithms and are considered the most prominent example of evolutionary programming. Comprehensive information regarding the history, study, application, and theory of GAs can be found in the literature. Most of CPAT's implementation of a GA is based on material gleaned from References [9], [10], and [11].

GAs derive their behavior from a metaphor of the biological processes associated with evolution. There is no specific GA; instead a GA is an approach to solving a problem. But all GA approaches have the following traits in common: a population of chromosomes, selection according to fitness, crossover to create new offspring, and random mutation.

CPAT implemented a GA in a program named *Cat*,[1] which was developed using:

- *gcc* Version 2.7.2.3, the GNU C/C++ compiler
- *libg+* Version 2.7.2, the GNU C/C++ libraries
- *Pro*C/C++* Version 8.1.6, the Oracle preprocessor that provides a software interface to tables within an Oracle Version 8.1.6 relational database

The goal of *Cat* is to find a set of delta times that can be applied to the flights in a scenario so that the distribution of parameters characterizing aircraft-to-aircraft conflicts meets user defined

[1] Cat was named for the character Cat on the British television series Red Dwarf. Cat is a humanized feline; the result of 3,000,000 years of evolution on the space ship Red Dwarf after all but one of its crew were killed by a radiation leak.