Genetic Algorithm Based Ground Delay Program Computations for Sector Density Control

Monish D. Tandale* and P. K. Menon†
Optimal Synthesis Inc., Palo Alto, CA 94303-4622

A Monitor Alert is triggered whenever the number of aircraft in an Air Traffic Control sector increases beyond its capacity limit. Departure rates at the airports feeding the airspace can be decreased to reduce the sector density. This paper presents a genetic algorithm for determining the optimal departure delays to be imposed on aircraft to maintain the sector counts below the Monitor Alert levels while minimizing the total delay in the system. The predicted state of the airspace is obtained using a fast-time airspace simulation system, which is then used to compute the quality of the resulting departure rates in the genetic algorithm. An interactive computing environment is used for implementing the optimization algorithm. Optimal aircraft departure results are presented for 2-airport and 10-airport example problems.

I. Introduction

Air traffic management in the US National Airspace employs a fixed airspace structure tied to geographical locations within the National Airspace (NAS) and can be termed as *Fixed Airspace Operations*. In this fixed airspace structure, the NAS is partitioned into Centers, which are further sub-divided into sectors. From the considerations of safe separation between aircraft, each air traffic control sector has a specified capacity, or Monitor Alert Parameter (MAP). A Monitor Alert is triggered whenever the number of aircraft in an Air Traffic Control sector is projected to increases beyond its capacity. In the congested northeast region of the United States, these imbalances in the sector demand and capacity are resolved through the implementation of miles-in-trail (MIT) restrictions. Because of the limited airspace available in Centers, such as New York, the delays associated with these restrictions typically manifest themselves as departure control restrictions at major airports, such as Newark and La Guardia. To date, only a limited number of studies have attempted to model and explore the impact of these restrictions on air traffic operations in the northeast (see for example, References 1 and 2).

The objective of this paper is to illustrate a general methodology for calculating near-optimal airport departure rates that minimize system delays while satisfying en route capacity constraints, such as sector MAP values. To accomplish this, a novel genetic algorithm[3-6] based approach is employed that systematically varies the chromosomes (i.e., airport departure rates) and assesses the fitness (i.e., quality) of each successive generation to (1) reduce the traffic demand below the sector capacities, and (2) minimize the system delay. For each successive generation of new chromosomes, NASA's *F*uture *A*TM *C*oncepts *E*valuation *T*ool (FACET)[7] is used to calculate the predicted sector occupancy counts which are in turn used to assess the fitness values of these chromosomes. A more detailed discussion of the genetic algorithm used in this study is reserved for Section II.A. To demonstrate this approach, a scenario involving departure controls at two major airports in the New York area are demonstrated, and another involving ten major airports in the northeast region of the U.S. are examined.

This paper is organized as follows. Section II provides an overview of genetic algorithms, and presents the integrated software framework for the optimization problem using genetic search methods. Results for a 2-airport and a 10–airport optimization problem are given in section III. Section IV discusses the issues in generating sufficiently rapid solutions to enable the use of the present methodology as a decision aid. Conclusions and recommendations for future work are in Section V.

---

* Research Scientist, 868 San Antonio Road, Member AIAA.
† Chief Scientist and President, 868 San Antonio Road, Associate Fellow AIAA.

II. Solution Methodology

The following subsections will provide an overview of genetic algorithms, the application of genetic algorithms in this study, and the software framework for conducting the optimization.

A. Brief Overview of Genetic Algorithms

A genetic algorithm (or GA) is a numerical technique useful for finding approximate solutions to optimization and search problems. The main premise behind these techniques is that the successful evolutionary processes observed in nature and described by Charles Darwin[8] in 1859 can be simulated on the computer to generate a population of possible answers for complex optimization problems.

In the genetic search paradigm, each candidate answer or 'individual' is considered to belong to a population. The population evolves towards better solutions with successive iterations or 'generations' through a series of simulated genetic operations. Each member is represented using a coding methodology that represents it as a 'chromosome'. For instance, each member can be represented as a binary number or character string encoding its characteristics. Associated with each member is a 'fitness value' that describes the suitability of the member with respect to the conditions describing the solution being sought. In each generation multiple individuals (or 'parents') are selected stochastically based on their fitness values and simulated genetic operations such as crossover and mutation are carried out to create new members ('or off-spring'). Periodically, the members showing poor fitness are decimated to keep the population from becoming too large. These simulated biological processes of selection-creation-destruction are repeated for a certain number of generations. The genetic algorithm is terminated after a specified number of generations or when the maximum fitness value ceases to show any improvement. The member with the maximum (minimum) fitness in the population then represents the answer.

Due to their lack of restrictions on the type of operations and functions that can be used, genetic search techniques are suitable for implementing a wide variety of search processes. This approach has been found to be exceedingly effective in problems that lack continuity, smoothness or linearity in the performance index and constraints. However, this flexibility comes at a price. Thus, genetic search methods are computationally intensive and are much less efficient in problems involving smooth, well-behaved functions wherein traditional search methods can exploit additional information such as gradients or Hessian to compute efficient search directions.

B. Application of Genetic Search Algorithm to the ATM Problem

For the problem discussed in this paper, the aircraft departure rates at the airports under consideration are as coded as 5-bit, fixed-length binary chromosomes. Predicted airspace densities are derived using FACET. FACET is a fast-time airspace simulation environment, developed by NASA for investigating next-generation air traffic management (ATM) concepts. These airspace densities are then used to compute the fitness values for each chromosome in the population. The fitness function includes the total aircraft delays and the monitor alert trigger events. The genetic search procedure described in the previous section is implemented to obtain the solution to the optimization problem. Although alternative methodologies based on nonlinear programming are conceivable, it was found during the present research that genetic algorithms provide sufficiently accurate solutions.

C. Software Framework

The optimization algorithm is configured around FACET and MATLAB® [9] software packages coupled together through CARAT#[10,11]. Genetic algorithm is implemented using the Genetic Search Toolbox[12] software developed by Optimal Synthesis Inc. for the Mathworks® product MATLAB. This software provides an integrated environment for performing all aspects of a genetic search, such as functions for selection, mutation-crossover operations, fitness evaluation and decimation. The genetic search toolbox unifies approaches used in the disciplines of genetic algorithms, genetic programming and evolutionary programming.

Figure 1 shows the software components employed in the present research. Important functional elements of the genetic search are: a population with an associated fitness evaluation methodology, one or more selection method, creation using genetic operation strategies and a decimation strategy, as shown in this figure.

The fitness evaluation can be carried out by any function or simulation within MATLAB, Simulink® or any of the associated toolboxes. For the present research, fitness evaluation requires the prediction of the future state of the airspace to determine if any monitor alerts have been triggered. As stated in the previous section, the airspace state is propagated in to the future using FACET. FACET is capable of modeling system-wide operations over the national airspace system. Airspace models such as Center/Sector boundaries, airways, Special Use Airspace, navigation aids /fixes and airports are available together with the performance models of several different aircraft types. Weather models including the ambient winds, temperature, severe weather cells are also available. A core capability of

FACET is the accurate computation of aircraft trajectories. Using spherical-earth kinematic equations, the aircraft can be flown along either along great circle flight plan routes, rhumb line navigation or wind-optimal routes as they climb, cruise and descend according to their individual aircraft-type performance data. Aircraft performance parameters such as climb/descent rates and cruise speeds are obtained from data table lookups. Heading and airspeed dynamics are also modeled.

The fitness evaluation requires access to the FACET airspace state prediction functionality from within MATLAB. This can be achieved through the CARAT# ("Carat-Sharp", Configurable Airspace Research and Analysis Tool – Scriptable) software[10,11], which allows access to the FACET functionality through Java programs, MATLAB and Jython© [13]. Using the CARAT# software, FACET can be integrated into any MATLAB application by instantiating a CARAT# object from within MATLAB.

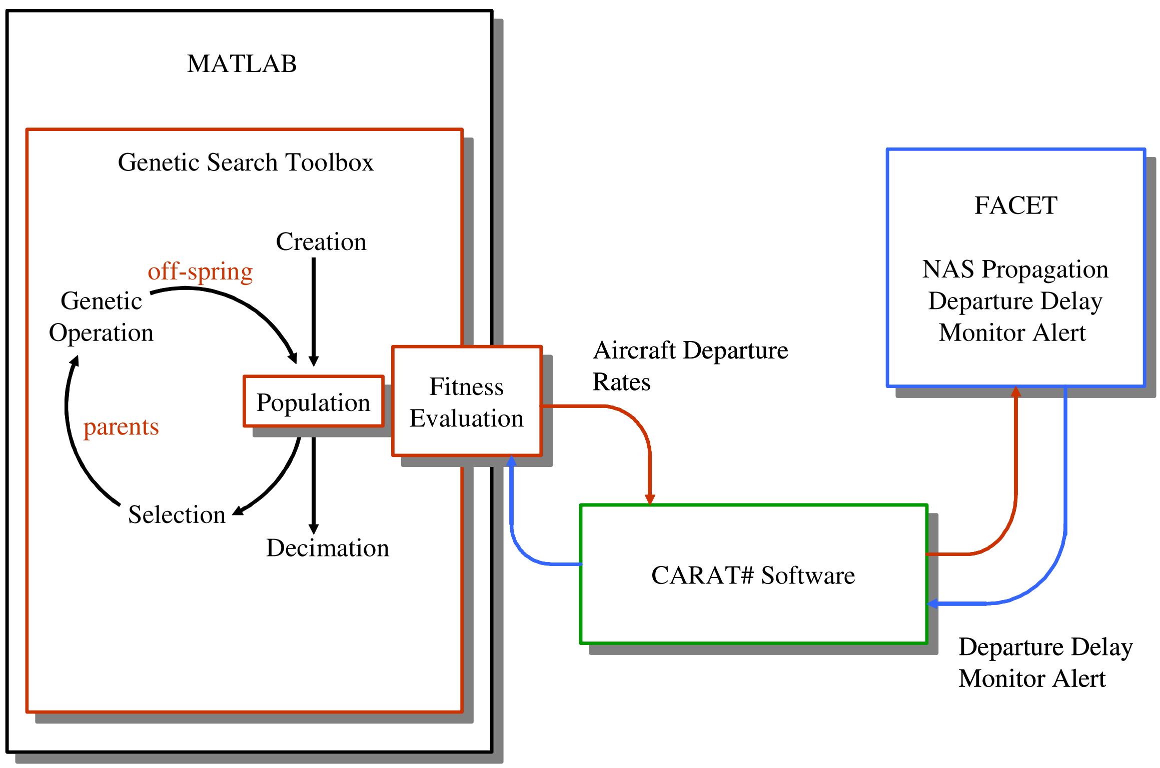

Figure 1. Functional Components of the Software Architecture

III. Optimization Results

This section presents a 2-airport and a 10-airport optimal delay computation problem. The following subsections will describe the set-up of the genetic search procedure and will present the results of the optimization procedure.

A. 2-Airport Optimization Problem

In order to serve as a lead-in to a more complex flow control problem, departures from the Newark (EWR) and LaGuardia (LGA) airports in the New York Center (ZNY) are considered first. The objective of the optimization is to minimize the total delay for the aircraft departing from EWR and LGA, subject to the constraint that the Monitor Alerts are not triggered in any of the Sectors in ZNY. The aircraft departure rates (ADR) from the two airports EWR and LGA are the 2 dimensions in this optimization problem. The purpose of formulating a simplified 2-dimensional optimization problem is to examine the efficacy of the genetic search process, since an exhaustive search solution can be obtained in this simplified setting. This solution can be used as a benchmark to assess the performance of the optimization algorithm. A description of the exhaustive search algorithm is provided in the following section.

B. Exhaustive Search Solution

The search solution can be obtained using CARAT# functions as follows:

1) The traffic scenario for the study is from the FACET data set, for a time window of 90 minutes from 12:22 UTC to 13:52 UTC on 3/18/1999. With this data, some of the Sectors in the ZNY Center were found to saturate even when the aircraft departure rates at both the airports were zero. The exact cause of this problem is currently under investigation, but is likely due to unrealistically low sector capacity values for select sectors in or near the New York Terminal Area Approach Control (TRACON). In order to create a feasible region with this data for the present study, the Sector capacities of all the Sectors are multiplied by 1.9.
2) Aircraft departure rate constraints are enforced at the airports using CARAT# methods, with the aircraft departure rates taking on the values between zero and fifty aircraft/hour. Thus a two dimensional matrix of test points can be obtained with EWR and LGA taking on the values [0, 2, 4,…50] aircraft/hour.
3) For each test point, the FACET traffic simulation is executed to determine the total delay for the aircraft departing from both the airports.
4) Next, the boundary of the feasible region is found by finding the least LGA departure rates that trigger a monitor alert for a given EWR aircraft departure rate.
5) The smallest total departure delay required to maintain feasibility is then the optimal flow rate, obtained by visual inspection in this 2-airport example.

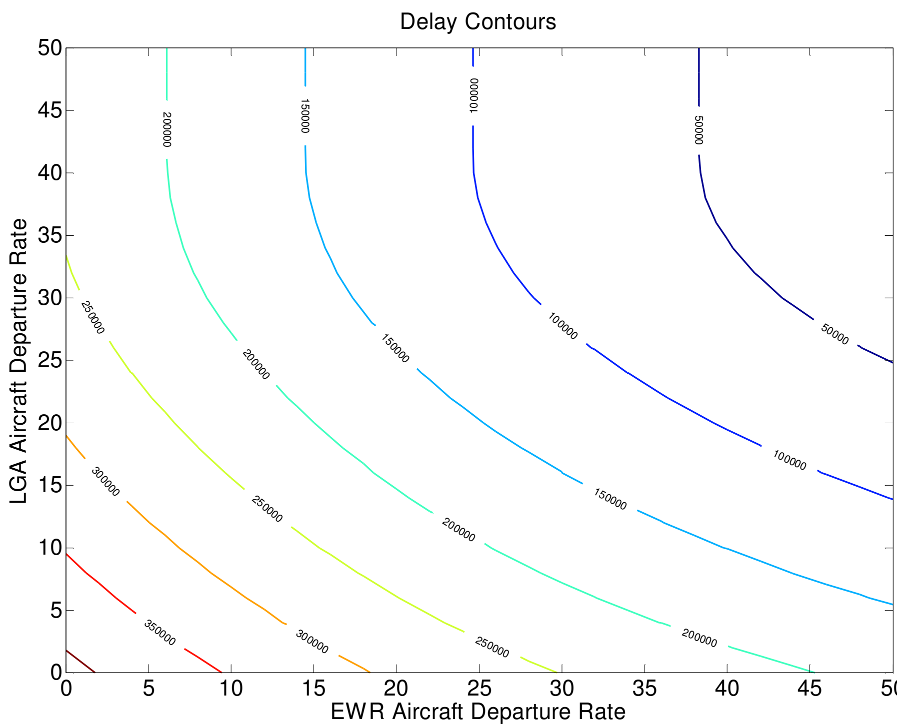

Figure 2. Contour Plot of the EWR-LGA Cumulative Delays in seconds as Functions of Hourly Aircraft Departure Rates

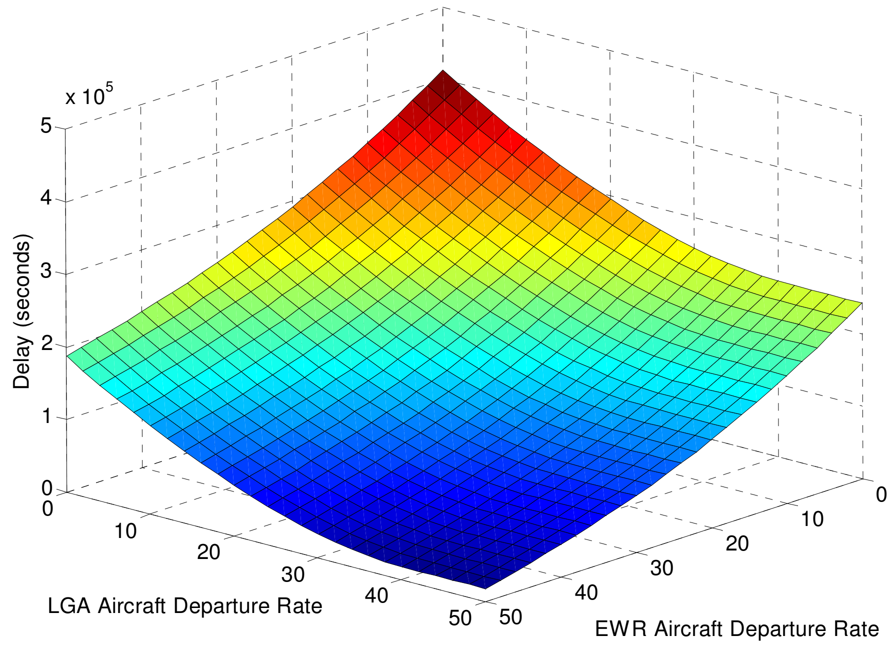

Figure 3. Surface plot of the Hourly EWR-LGA Aircraft Departure Rates with Cumulative Delays in seconds as Functions of Departure Rates

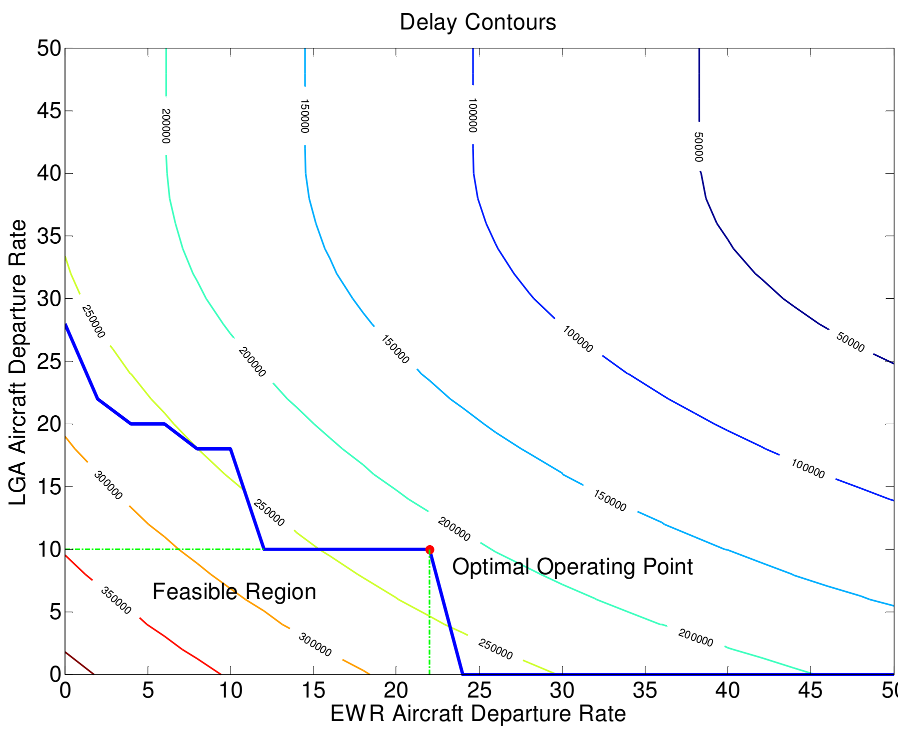

Figure 4. Cumulative Delay Contours in seconds with the Feasible Region and Optimal Operating Point Depicted for the Hourly EWR-LGA Aircraft Departure Rates

Figure 2 and Figure 3 show the contour and surface plots of the EWR-LGA cumulative delays (in seconds) as functions of aircraft departure rates. The region of Figure 4 that is labeled “Feasible Region” represents those values of the EWR and LGA hourly departure rates for which the demand for the sectors in New York Center were below the capacities, or MAP values. The optimal operating point can be obtained visually by locating a point on the boundary with the least delay as shown in Figure 4. This operating point represents the optimal solution as it provides the least delay without triggering any monitor alerts.

C. Genetic Algorithm Optimization Solution

Although the approximate optimal operating point can be picked-out by inspection in Figure 4, a numerical search algorithm must be employed in a more general, higher-dimensional case. However, since the constraint boundary contains corners (see for example, the boundary illustrated by the thick, blue line in Figure 4), and the solution for the aircraft departure rates must belong to the set of non-negative integers, it is unlikely that methods such as Gradient Projection[14] will yield useful results for this problem, since the boundary is not convex. On the other hand, Genetic search algorithms are useful for solving optimization problems that may involve complex constraints, multiple objective functions, discontinuities and non-convex performance indices. Unlike the conventional optimization algorithms, genetic search methods do not require good initial guesses or smoothness. However, depending on the parameterization of the problem, they may be good only for generating near-optimum results. In some situations, it may be possible to use conventional optimization methods to refine the results.

In view of this, a genetic search algorithm is next set up to determine the optimal operating point. The genetic search algorithm is implemented using the software described in Reference 12.

In the genetic search process, each candidate solution is coded as a *chromosome* that can be manipulated using biologically inspired genetic operations such as mutation and crossover. Resulting offspring are then decoded to determine their *fitness* or the performance index. For the present problem, the aircraft departure rate at an airport is coded as a 5-bit, fixed-length binary chromosome. Thus each chromosome can take on values, from 0 to 31. For instance, a chromosome 11001 represents a departure rate of 25. Two separate initial populations are created, each representing the departure rates at the two airports under consideration. Likewise for the 10-dimensional example presented in section III- 0, 10 initial populations are created, each representing the departure rates at the ten airports under consideration.

The fitness function for the genetic search is defined as a linear combination of the total delay, and a large number corresponding to the triggering of the Monitor Alerts.

$$Fitness = Total\ Delay + Monitor\ Alert\ Trigger \times 10^{10} \quad (1)$$

The *Monitor Alert Trigger* parameter takes on the values of 1 or 0 in the above equation. Note that the fitness function penalizes the chromosomes that violate the Monitor Alert constraint.

In order to start the search process, 30 chromosomes are randomly generated to form the initial members of both the populations. Figure 5 shows the individuals in the initial population. The fitness function for each individual in the population is obtained by running a FACET simulation to calculate the total delay and to determine if any Monitor Alerts were triggered. New generation of members is derived using a proportional selection[3] methodology in conjunction with the crossover operations. In the proportional selection methodology, the probability of selecting any pair of individuals is proportional to their fitness. The computational algorithm for the genetic search is illustrated in Figure 6.

The genetic search is run for 550 generations and the results are presented in Figure 7 through Figure 9. It may be observed from Figure 7 that the optimal solution was reached approximately by the 80th generation. Beyond that, the search process did not produce any improvement in the fitness of the population. Figure 8 shows the location of the fittest individuals in the search space, with the progress of the generations of the genetic algorithm. The numbers 1-24, 25-35, 35-62 etc indicate the generations corresponding to the solution. Figure 9 shows a comparison between the optimal solutions obtained by visual inspection and that computed using the genetic search.

The solution (EWR-ADR, LGA-ADR) according to the genetic search is (23,8), which implies that departure rate from EWR is 23 aircraft per hour and the rate from LGA is 8 aircraft per hour, with a total departure delay of 223920 seconds (3732 minutes). This means that with the current flight plans, the departure rate of 23 and 8 aircraft per hour, from the airports EWA and LGA respectively will lead to the least delay of 3732 minutes without triggering any monitor alerts. The visual solution is (22,10) with a total departure delay of 216,510 seconds (3608 minutes). This point is labeled as the “Optimal Operating Point” in Figure 4, since these rates give the best results for the fitness function defined by Eq. 1.

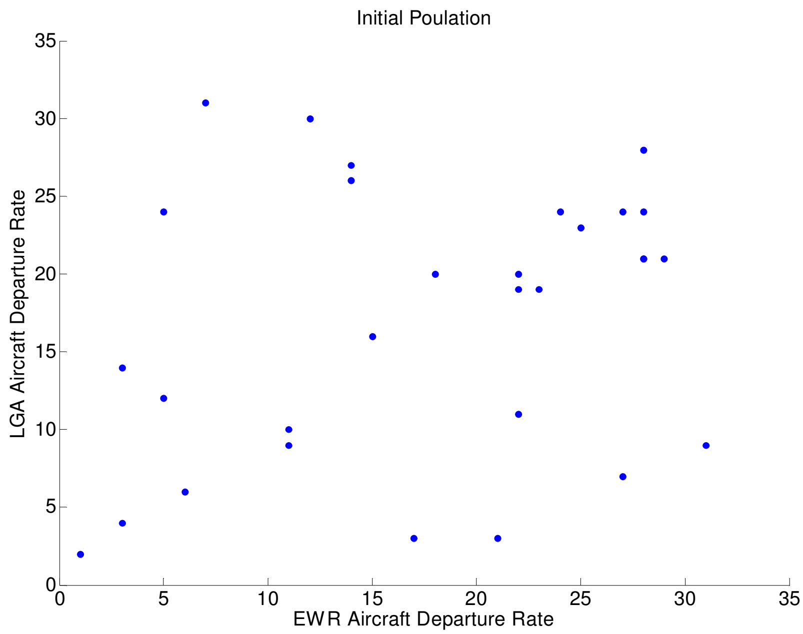

Figure 5. Initial Population Used for Starting the Genetic Search

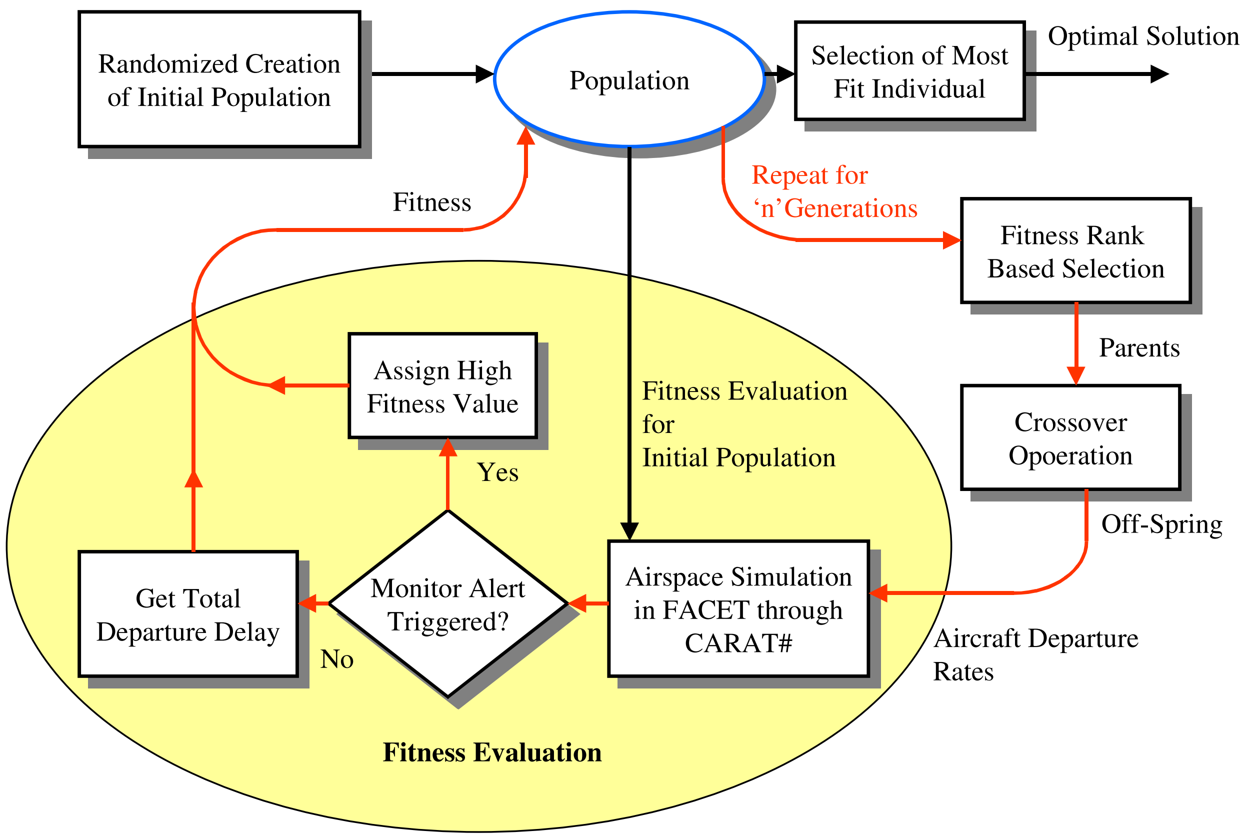

Figure 6. Computational Flowchart of the Optimal Delay Optimization Algorithm for Sector Density Control Using the Genetic Search Methodology

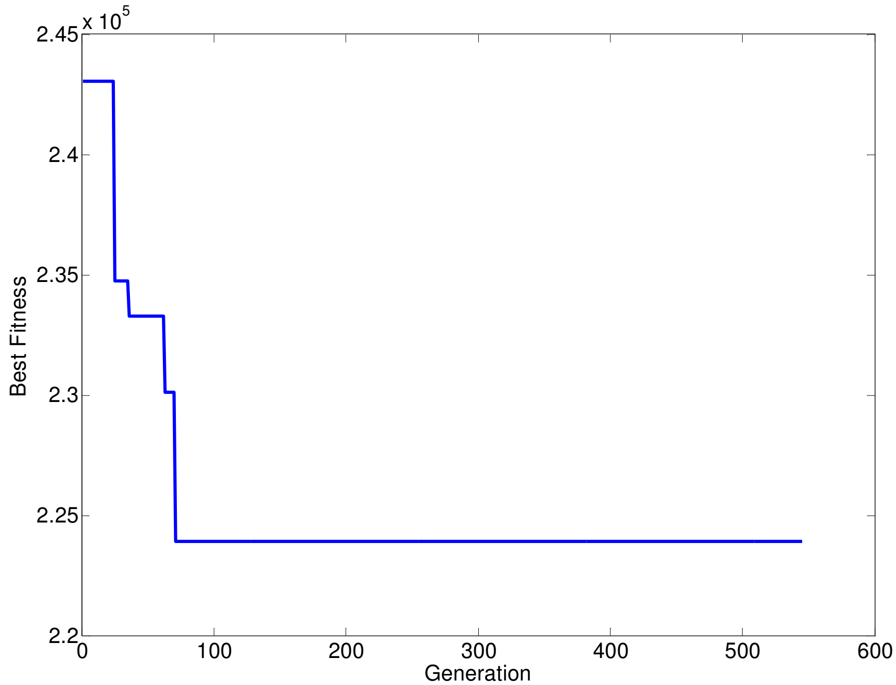

Figure 7. Progression of the Fitness of the Best Individual during the Genetic Search

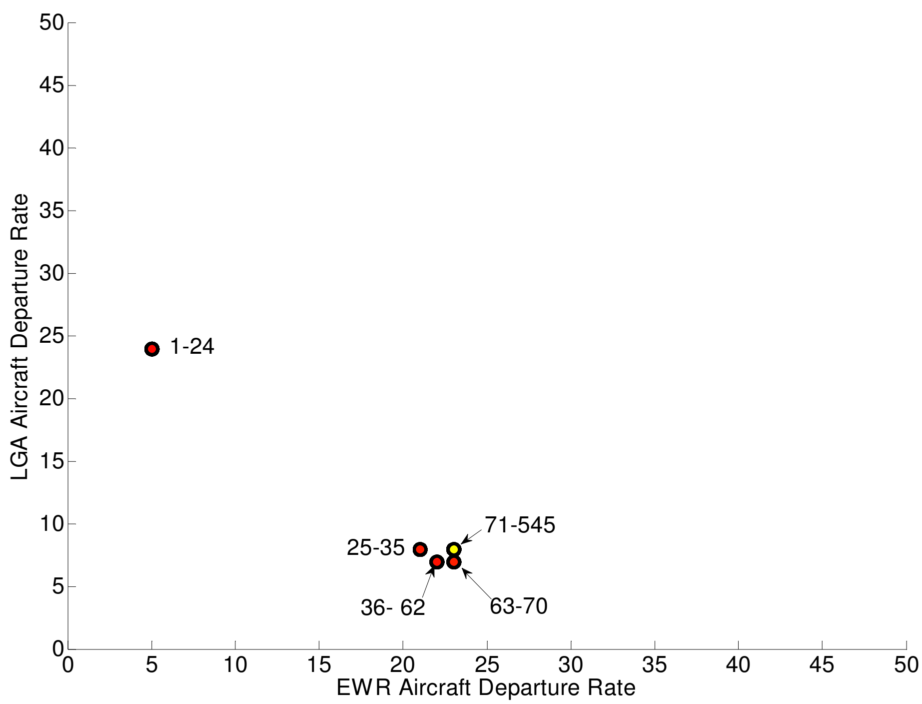

Figure 8. Departure Rates for the Best Individuals during the Genetic Search

(Yellow Dot denotes the Optimal Solution)

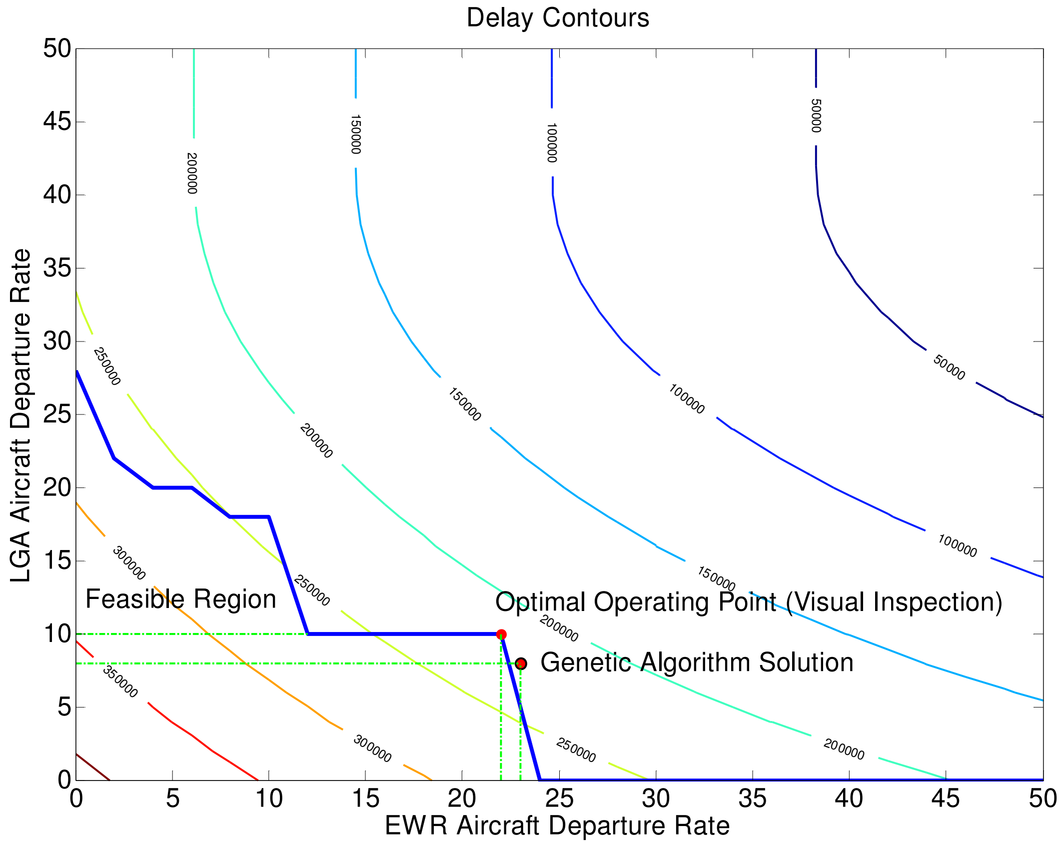

Figure 9. Genetic Search Solution for the Optimal Operating Point and Comparison with the Visual Solution

D. Extension of the Optimal Departure Rate Problem to 10 Airports

The genetic search methodology is next scaled-up to include ten airports on the eastern seaboard. The airports considered in this example are:

1) EWR - Newark Liberty International Airport
2) LGA - La Guardia Airport
3) JFK - John F Kennedy International Airport
4) PHL - Philadelphia International Airport
5) BOS – Boston General Edward Lawrence Logan International Airport
6) DCA - Ronald Reagan Washington National Airport
7) IAD - Washington Dulles International Airport
8) MDT - Harrisburg International Airport
9) PVD – Providence Theodore Francis Green State Airport
10) BWI – Baltimore/Washington International Thurgood Marshall Airport

Monitor Alerts are verified for all the sectors within the following Air Traffic Control Centers

1) ZNY – New York Center
2) ZOB - Cleveland Center
3) ZDC – Washington Center
4) ZBW – Boston Center

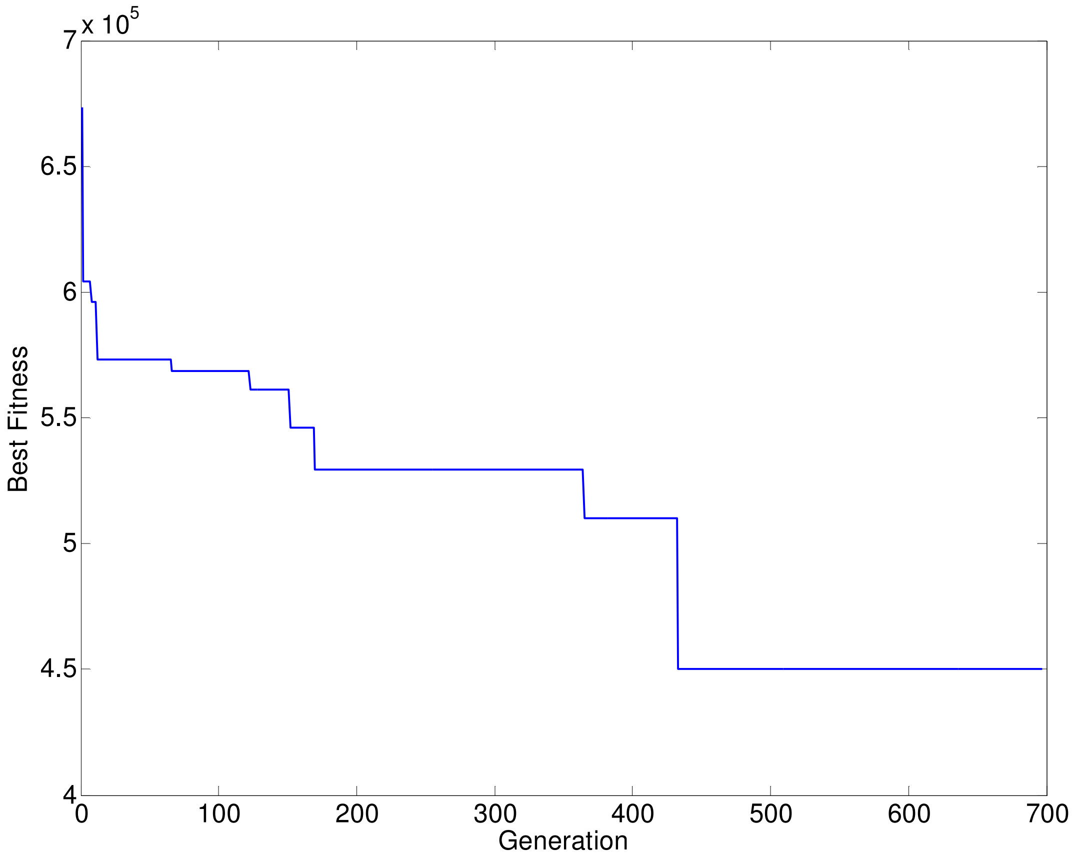

Figure 10. Progression of the Fitness of the Best Individual during the Genetic Search (10-Airport Example)

As in the previous example, the Sector capacities are multiplied by 1.9 to ensure the existence of feasible solutions. After random creation, the individuals with fitness less than 7×10^5 were deliberately deleted to create an initial population of 142 members. This was done to ensure that the search begins with bad initial guesses and to demonstrate that the genetic algorithm can provide solutions starting with crude initial guesses. After 697 generations, the genetic search produced the following optimal hourly departure rates at the 10 airports.

$$[\text{EWR LGA JFK PHL BOS DCA IAD MDT PVD BWI}]^T = [6\ 25\ 16\ 25\ 29\ 18\ 16\ 26\ 7\ 10]^T \quad (2)$$

Total departure delay corresponding to the optimal departure rates is found to be 450150 seconds. The evolution of the fitness of the best individual in a population during the genetic search is given in Figure 10

IV. Computational Considerations for Implementation

The previous sections illustrated the genetic search based methodology for maintaining the traffic density by departure rate control. In order to be useful as a decision aid, it should be possible to carry out the departure rate computations in a specified time window.

The most time-consuming part of the genetic search process is the computation of fitness of the members of the population. As presently configured, the fitness evaluation for an individual requires a single run of the FACET over the 90 minute simulation time window. At a simulation step size of 30 seconds, this run takes about 2.5 minutes on a Pentium IV, 2 GHz computer running the Windows® XP operating system. The 2-airport example presented in this paper required a run time of almost 48 hours for 550 generations on a single machine. For the 10-airport example, the trajectory propagation took almost 100 hours for 700 generations.

These experiments show that alternate computing architectures or methods for faster trajectory propagation will be essential for implementing the proposed methodology as a decision support system. Due to its basic nature, the genetic search method readily lends itself to cluster computing[‡] architectures. Consequently, if a computing cluster of sufficient size can be configured, it is entirely feasible to employ the proposed algorithm in decision support systems.

V. Conclusions

This paper discussed a methodology useful for determining the optimal aircraft departure rates to ensure that the airspace Sector counts remain below the Monitor Alert levels. The optimality criterion was the minimization of the total departure delay in the system. The proposed methodology is applied to compute departure rates in a 2-airport example and a 10-airport example. In the case of the 2-airport example, the optimal operating point obtained using genetic search algorithm was shown to match closely with the optimal solution obtained by exhaustive search. The approach was shown to easily generalize to a 10-airport example. No difficulties are seen in extending the methodology to larger number of airports.

Computational issues in developing a decision support system based on the present methodology were briefly examined. While single computer implementations are not feasible for real-time usage, high-performance cluster computing architectures can readily provide the desired speed. This and other issues will be of future interest.

Acknowledgments

This research was supported under Contract No. NNA04BE64C from NASA Ames Research Center. Dr. Banavar Sridhar and Dr. Shon R. Grabbe of NASA Ames Research Center were the first to formulate the optimization problem discussed in this paper, and have assisted the authors through its current implementation. We are grateful for their contributions to this research.

References

[1]Grabbe, S. and Sridhar, B., "Modeling and Evaluation of Miles-in-Trail Restrictions in the National Air Space," AIAA Guidance, Navigation and Control Conference, Austin, TX, August 11-13, 2003.

[2]Kopardekar, P. and Green, S., "Airspace Restriction Planner for Sector Congestion Management," AIAA Aviation Technology, Integration and Operations Conference, Arlington, VA, Sep. 26-28, 2005.

[3]Holland, J. H., *Adaptation in Natural and Artificial Systems*, The MIT Press, Cambridge, MA, 1993.

[4]Fogel, D. B., *Evolutionary Computation*, IEEE Press, Piscataway, NJ, 1995.

[5]Goldberg, D. E., *Genetic Algorithms*, Addison-Wesley, Reading, MA, 1989.

[6]Koza, J. R., *Genetic Programming*, The MIT Press, Cambridge, MA, 1992.

[7]Bilimoria, K. D., Sridhar, B., Chatterji, G. B., Sheth, G., and Grabbe, S., "FACET: Future ATM Concepts Evaluation Tool," *3rd USA/Europe Air Traffic Management R&D Seminar*, Naples, Italy, June 2000. Also appeared in *Air Traffic Control Quarterly,* Vol. 9, No. 1, March 2001, pp. 1–20.

[8]Darwin C., *The Origin of Species,* G. Surioano, Ed., New York, NY: Grammercy, 1998.

[9]Anon, *Using MATLAB®*, The MathWorks, Inc., Natick, MA, 2007.

[10]Menon, P. K., Diaz, G. M., Tandale, M. D., and Kwan, J., "CARAT# - A Rapid Prototyping Software for Developing Next-Generation Air Traffic Management Algorithms," Final Report Prepared under NASA Phase II SBIR Contract No. NNA05BE64C, Vol. I, Optimal Synthesis Inc, Palo Alto, CA, November 21, 2006.

[11]Menon, P. K., Diaz, G. M., Vaddi, S. S., and Grabbe, S. R., "A Rapid Prototyping Environment for En Route Air Traffic Management Research", *AIAA Guidance, Navigation and Control Conference*, August 15-18 2005, San Francisco, CA.

[12]Menon, P. K., Cheng, V. H. L., Lam, T., Crawford, L. S., Sweriduk, G. D., and Dewell, L. D., *Genetic Search Toolbox*, Optimal Synthesis Inc, Palo Alto, CA, 2007.

[13] Pedroni, S., and Rappin, N., *Jython Essentials*, O'Reilley, Sebastopol, CA, 2002.

[14] Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, New York, NY, 1981.

[‡] http://en.wikipedia.org/wiki/Computer_cluster#High-performance_computing_.28HPC.29_clusters, July 6, 2007.