An Efficient Algorithm for Solving an Air Traffic Management Model of the National Airspace System (NAS) – E Andrew Boyd, Rusty Burlingame, Kenneth Lindsay, INFORMS Journal of Computing, Fall 1998

Introduction: The NAS is the infrastructure that supports the 60,000 daily flights over the United States. The hub and spoke operations of various airlines has changed traffic patterns and contributed significantly to congestion and delay in the NAS. Growth of air traffic, matching or exceeding growth in capacity and occasional bad weather, has added to this congestion and delay. Air traffic delay has been a subject of research for many years. In this paper, a large integer-programming model for minimizing air traffic delay has been presented. Solving problem instances arising from this model involves use of preprocessing, constraint strengthening, and a carefully designed computer implementation.

Modeling Approaches: Apart from categorizing modeling approaches as linear, integer, and mixed integer programming models, the paper highlights a further classification of the model into categories based on the types of decision variables used. The flight event timing models contain real-valued variables denoting the clock times that events such as flight departures and landings are to occur. These models at times make use of sequencing constraints that make the related integer programs really hard to solve. The aggregate flow models contain integer variables denoting the number of flights traveling between a particular origin – destination pair. Much similar to aggregate flow models are the third category of models – time-interval assignment model (TAM). These models also discretize time into set of fixed intervals in the same way, but the decision variables are binary and indicate whether or not a particular event such as a flight departure should occur during a particular time interval. As a result, these models are large-scale 0-1 integer programs.

The Model and Constraints: The model proposed in this paper is a TAM. It seeks to assign the time intervals during which each flight is to be at given fixes in its flight path. The assignment is made to minimize the total cost of delaying flights throughout the system. A fix is a point in space such as an arrival airport, a departure airport, or a point on aircraft’s flight path. A 0-1-decision variable is then introduced for each combination of flight, time period, and fix. Flights cannot depart the initial fix before a set time and cannot travel from one fix to the next in less than a minimal specified time, but flights may depart late or travel at a reduced speed between fixes at a penalty. The objective function ascribes a cost whenever a flight from one fix to the next immediate fix requires more than the minimum time to fly between those fixes. The objective is subject to a number of constraints. There is a limit on the number of flights passing through a given fix during a given time interval. Each flight has to pass through each fix in its flight path exactly once. The time required for a flight to travel between two consecutive fixes is no less than the minimum travel time between those fixes. These constraints also guarantee that each flight traverses the fixes in its flight path in the appropriate order. The model also incorporates itinerary constraints, which guarantee that the layover time of an aircraft at an airport is no less than the minimum turnaround time.
Optimization:

The algorithm consists of a problem reformulation phase in which preprocessing and constraint generation is applied in an effort to strengthen the problem formulation. The problem is then passed to a general purpose branch and bound algorithm. Preprocessing consists of a variety of techniques for reducing the size of a problem by permanently fixing variables and removing unnecessary constraints based on logical considerations. Preprocessing methods are normally exceptionally fast and often have the effect of substantially reducing problem size without affecting the optimality of the obtained solution. However for solving large problems preprocessing alone proved inadequate to solve. Constraint generation and cutting planes were required for large problem sizes. Cutting planes were added to deal with additional constraints iteratively, solve the linear program after appending some subset of constraints, determine if there are any violated constraints, append some other violated constraints, and repeat this fashion until all constraints are satisfied. This process was slower than solving a single large linear program. Once the linear program is solved and some of the variables were found to be fractional then a branch and cut procedure was adopted. The speed of the algorithm was determined by \( m \) - the number of potential time intervals during which a flight event can occur and thus was an important factor in the implementation of the algorithm. A compromise solution is to solve the problem with increasingly larger values of \( m \) and to terminate when the next larger value of \( m \) yields no decrease in objective function. This approach was observed to be a practically viable approach. It was observed that almost all of the time spent solving the larger problems is spent in solving the linear programming relaxation. Because of this, some effort was made to reduce this time, most notably by experimenting with alternative linear programming options and developing heuristics. The heuristic found to generate the best initial bases most consistently was motivated by the observation that over 90% of the variables in the problems being solved were basic at optimality, and relatedly, that most of the slack variables were non basic.

Re-optimization: When taking real world into consideration, variability in all its forms remains a significant topic in the discussion of model formulation. One way to deal with variability is to solve the initial problem before beginning of the day and then to incorporate changes and re-solve the problem as necessary. Another approach to solving problems that arise throughout the day is to modify the problem data and re-solve the problem using the simplex algorithm with a warm start. This approach was found to be effective especially when sudden weather changes were modeled.

Conclusions: Integer programs formulated from real data and containing over one million variables have been solved to optimality on a desktop workstation. Solution methodology was clearly presented and appeared to be robust. Though the work presented in the paper is encouraging, more emphasis can be given on finer levels of details on how events should be sequenced within an interval or across bordering intervals. Another issue relates to comparing results from various models based on different decision variables. Last but not the least, a myriad of practical difficulties need to taken into account to manage air traffic in real time.