Integrated Arrival-Departure-Surface-Enroute Air Traffic Optimization

Parikshit Dutta[1], Prasenjit Sengupta[2], Jason Kwan[3] and P.K. Menon[4]
Optimal Synthesis Inc., Los Altos, CA, 94022.

This paper presents the application of Traffic Flow Management (TFM) optimizer to the integrated arrival-departure-surface-enroute (IADSE) optimization problem. The optimizer uses the Bertsimas-Lulli-Odoni (BLO) model, which can be used model flights with multiple possible routes in a given volume of airspace using large-scale Mixed-Integer Linear Programs (MILP). Furthermore a novel method is proposed, by which the separation between each aircraft is converted into convex constraints, such that they can be used in the BLO framework.

By leveraging the strong linear programming relaxation and by using the Dantzig-Wolfe (DW) decomposition technique, the large-scale MILP is decomposed into a large number of sub-problems and one master problem. The number of sub-problems is equal to the number of flights and can be solved independently of each other, whereas the master problem consists of a significantly smaller number of constraints than the original MILP. The sub-problems are therefore amenable to solution in parallel, resulting in significant acceleration over a solution to the original, monolithic problem.

The proposed algorithm is implemented to schedule traffic in the New York N90 Terminal Radar Approach Control. The results obtained from the proposed optimizer was compared with that obtained from a first-come-first-served scheduler. It was found that the proposed algorithm performed better in minimizing aircraft delays than the baseline scheduler.

Nomenclature

ATC	Air Traffic Control
ATM	Air Traffic Management
BSP	Bertsimas and Stock-Patterson
BLO	Bertsimas-Lulli-Odoni
DW Decomposition	Dantzig-Wolfe Decomposition
FAA	Federal Aviation Administration
LP	Linear Program
TFM	Traffic Flow Management
TRACON	Terminal Radar Approach Control
SJC	Norman K. Mineta San Jose International Airport
JFK	John F. Kennedy International Airport
SFO	San Francisco International Airport

I. Introduction

AIR traffic demand growth in response to economic activity has led to congestion and increased delays at the busiest airports in the National Airspace System (NAS). NASA's Airspace Systems Program, through its Technical Challenge 2, aims to significantly increase the throughput at airports and metroplexes while reducing sensitivity to off-nominal behavior (increased robustness) and minimizing environmental impact. Similar objectives have been identified for airport surface operations. NASA and the Federal Aviation Administration (FAA) have been involved in extensive efforts to develop advanced concepts, technologies, and procedures for the Next

[1] Research Scientist, 95 First Street, Suite 240, AIAA Member.
[2] Currently, Staff Scientist, Swift Navigation, San Francisco, AIAA Associate Fellow.
[3] Research Engineer, 95 First Street, Suite 240.
[4] Chief Scientist, 95 First Street, Suite 240, AIAA Fellow.

Generation Air Transportation System (NextGen). The objective of these research efforts has been to improve the capacity, efficiency, and safety in the next-generation National Airspace System (NAS). Improvements can come in the form advanced communication, navigation, and surveillance capabilities, fuel-efficient vehicle designs, and through improved air-traffic operations realized using sophisticated automation systems. A significant portion of the NextGen research is aimed at (i) developing ground-side automation systems to assist controllers in strategic planning operations such as scheduling flights in terminal area, (ii) developing tactical controller decision support tools to separate and space the traffic, and (iii) developing flight-deck-side automation to assist pilots in accomplishing airborne merging and spacing operations.

A relatively newer effort is the integration of arrival, departure, surface, and enroute (IADSE) operations in order to further improve efficiency in resource utilization. The objective of NAS-wide integrated arrival, departure, surface and enroute optimization is to create an optimal gate-to-gate traffic flow management solution based on capacity constraints and separation rules. For example, airport surface optimization should consider taxiway and runway capacities and gate occupancy constraints. Minimum separation rules between taxiing aircraft and successive departures must also be enforced. Compliance of route, speed and altitude constraints, and separation rules for different classes of aircraft arriving and departing from airports within metroplexes is required. In addition to surface and terminal airspace capacities and separation rules, the optimization procedure needs to comply with sector capacity constraints and enroute airspace separation standards. The optimization procedure should seek to output controls that can be practically implemented by air traffic control such that delays are minimized and equitably distributed between aircraft operators, thereby maximizing the surface and airspace utilization. Some of the controls currently used by air traffic control are- ground-hold, miles-in-trail (MIT) or minutes-in-trail (MINIT), route-change, speed-change, altitude-change, and required time of arrival (RTA). For example, Traffic Management Advisor (TMA) creates a meter-list that specifies a scheduled time of arrival (STA) for arrivals at a certain distance from the airport for an arrival meter-fix or meter-arc. The optimal solution could create such meter-lists at chosen control locations in the airspace for air traffic control. It could also compute controls for aircraft such as the changes in speed within aircraft performance limits for use by the air traffic control. TFM programs manage the imbalance between the traffic demand and the available airspace capacity using Traffic Management Initiatives (TMIs) such as Ground Stops, Ground Delay Programs (GDPs), Mile-In-Trail (MIT), Rerouting, Airborne Holding, Airspace Flow Programs (AFP), Sequencing Programs and Fix Balancing.

A method that has been applied successfully to this class of problems is the Bertsimas and Stock-Patterson (BSP)[1] optimization procedure accelerated using the Dantzig-Wolfe decomposition. The binary integer program formulated by Bertsimas and Stock-Patterson approach is a well-understood and often-used[2,3] optimization approach to the aircraft-level TFM problem, as is done by Bertsimas Lulli and Odoni (BLO)[4]. This model solves the TFM problem with multiple airports and deterministic sector capacities. However possible extensions of BSP/ BLO framework to solve air traffic control problems in terminal area or surface is missing in the literature[5].

The current work proposes a solution wherein the TFM problem in the BSP framework can be formulated to solve an IADSE problem. The basic difference between the TFM and IADSE problem is that the former uses sector capacities as constraints and is developed in a control-volume formalism. On the other hand the IADSE problem uses the separation between aircrafts as constraints and uses a trajectory-based formalism. Hence, the current work proposes an innovative method to convert the separation constraints into convex capacity constraints such that the IADSE problem can be solved in the BSP/BLO framework. Furthermore, the current work also develops a parallel methodology such that the Dantzig-Wolfe based decomposition can be efficiently implemented into multicore integrated hardware architectures.

The rest of the paper is organized as follows: Section II describes the problem briefly and derives the constraints in the BLO framework, Section III describes the Dantzig-Wolfe (DW) solution methodology, Section IV deals with the implementation of the DW algorithm in many-core integrated architecture, Section V presents result of implementation of the current algorithm to New York TRACON, finally Section VI concludes the current work.

II. Problem Description and Constraint Formulation

In this section, constraint generation using the BLO formalism is derived[4,6]. The BLO model formulates the TFM problem in terms of the following optimization problem:

$$\begin{aligned} &\max \boldsymbol{c}^{\top}\boldsymbol{x} \\ &\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ &\boldsymbol{x} \in \{0,1\} \end{aligned} \tag{1}$$

To obtain the decision variables $\boldsymbol{x}$ a binary integer programming problem is solved. It can be shown that the problem has a strong linear programming relaxation which make it amenable to be solved using conventional LP techniques such as the simplex method[7].

A. Decision Variable and Data Sets

The variable of interest in IP formulation is denoted by $x_{f,XY}^{t}$, which is a binary variable, i.e. $x_{f,XY}^{t} \in \{0,1\}$. A value of 1 indicates that flight $f \in \mathcal{F}$, where $\mathcal{F}$ is the set of all the fights currently in the system, reaches Node Y from Node X, by time interval $t \in \mathcal{T}$ using an arc connecting the two nodes. Nodes X and Y belong to set $\mathcal{N}_f$ that is composed of all nodes on the route(s) of flight f. The sets $\mathcal{K}^d$ and $\mathcal{K}^a$ denote the set of nodes corresponding to departure and arrival airports, respectively. Since an airport in general can be a departure as well as an arrival airport, $\mathcal{K}^d \cap \mathcal{K}^a \neq \emptyset$. Let $k_f^d \in \mathcal{K}^d$ and $k_f^a \in \mathcal{K}^a$ denote departure and arrival airport nodes for flight f, respectively. Node $q(k_f^d)$ denotes the departure airport boundary, and $p(k_f^a)$ denotes the arrival airport boundary. The distinction between an airport node and its boundary node is required in order to model ground holds and runway delays. Both are expressed in terms of the number of time units required by a flight to reach the airport boundary from the airport node.

The concept of 'airport' and 'airport boundary' are only introduced in order to develop equations while following the BLO formalism. In reality, the first node of an aircraft's trajectory in a given simulation can be considered the airport node. For example, if the first appearance of an aircraft is at the TRACON boundary, the first STAR can be considered the airport node. The BLO model does not distinguish between arriving and departing traffic, and therefore, modeling of shared resource such as waypoints, meterfixes and taxiways is straightforward. Arriving and departing traffic can be penalized separately. In fact, individual flights can be penalized separately. For instance, variables can be fixed beyond a certain horizon.

The arc XY is a member of set $\mathcal{A}_f = \{XY | X, Y \in \mathcal{N}_f\}$ that is composed of all arcs on the route(s) of flight f. The set $\Gamma_f^+(X) = \{Y | XY \in A_f\}$ and $\Gamma_f^-(X) = \{Y | YX \in A_f\}$ are the set of nodes that have arcs from Node X and leading into Node Y respectively, for flight f.

Whereas sets $\mathcal{N}_f$ and $\mathcal{A}_f$ denote all possible nodes and all possible arcs for flight f, the sets $\mathcal{N}_f^* \subseteq \mathcal{N}_f$ and $\mathcal{A}_f^* \subseteq \mathcal{A}_f$ denote the nodes and arcs corresponding to the scheduled route of flight f. If only one route is modeled for a flight, then $\mathcal{N}_f^* = \mathcal{N}_f$ and $\mathcal{A}_f^* = \mathcal{A}_f$. Sets $\mathcal{N}_f$ and $\mathcal{A}_f$ can consist of nodes and links that are not currently used but can possibly be used if, for example, runway or TRACON configuration changes.

The variables $l_{f,XY}$, r_f, and d_f denote the travel time (number of time periods) for flight f over arc XY, the scheduled departure time period, and the scheduled arrival time period, respectively. It is noted in Ref. 6 that $l_{f,k_f^d,q(k_f^d)} = l_{f,p(k_f^a),k_f^a} = 0, \forall k_f^d \in \mathcal{K}^d, k_f^a \in \mathcal{K}^a$. In other words, a flight reaches the departure airport boundary immediately after leaving the departure airport node, and a flight reaches the arrival airport node immediately after leaving the arrival airport boundary. It also follows that

$$r_f = d_f + \sum_{XY \in A_f^*} l_{f,XY} \tag{2}$$

In other words, the scheduled arrival time of the flight is given by the sum of the departure time and flight times along scheduled route segments. Furthermore, $\bar{l}_{f,XY}$ and $\underline{l}_{f,XY}$ denote the maximum and minimum number of time segments for flight f on arc XY.

The binary BLO variables can be used to determine quantities of interest for TFM. For instance, the time segment in which the flight f reaches node n is denoted by $T_{f,n}$ and given by the following summation:

$$T_{f,n} = \sum_{X \in \Gamma_f^-(Y)} \sum_{t \in \mathcal{T}} t\left(x_{f,XY}^{t} - x_{f,XY}^{t-1}\right) \tag{3}$$

It follows that given the time of entry at a node and the time of entry at a preceding node, the number of time intervals required to travel on the arc connecting the nodes can be calculated. Similarly the number of aircraft which reach a node n at a given time t is given by the following summation:

$$S_{n,t} = \sum_{f \in \mathcal{F}} \sum_{X \in \Gamma_f^-(n)} \left(x_{f,Xn}^{t} - x_{f,Xn}^{t-1}\right) \tag{4}$$

Additional quantities such as sector counts (given the arcs belonging to a sector) can also be calculated, as detailed in Ref. 4.

B. Constraint Formulation

The variables are linked with constraints resulting from the spatio-temporal representation of the graph. The so-called flight structure constraints define the continuity in time and space for a flight. The temporal continuity constraints[6] are represented by the following linear inequalities and equalities:

$$\begin{aligned} x_{f,XY}^{t-1} &\le x_{f,XY}^{t}, \qquad t \in \mathcal{T}_{f,XY}^{*}, (X,Y) \in \mathcal{A}_f, f \in \mathcal{F} \\ x_{f,XY}^{t-1} &= x_{f,XY}^{t}, \qquad t \in \mathcal{T}_{f,XY} \setminus \mathcal{T}_{Xf,Y}^{*}, (X,Y) \in \mathcal{A}_f, f \in \mathcal{F} \end{aligned} \tag{5}$$

where $\mathcal{T}_{f,XY}^{*}$ is the set of feasible time units in which a flight f can reach Node Y from Node X over the arc connecting the two nodes, and $\mathcal{T}_{f,XY}$ is the smallest set of consecutive time intervals that contains $\mathcal{T}_{f,XY}^{*}$. These constraints state that if a flight was in node X by time period t, then this must also hold true for any later time period $t' > t$.

The spatial continuity constraints[6] are given by the following inequalities:

$$\begin{aligned} \sum_{Z \in \Gamma_f^{+}(Y)} x_{f,YZ}^{t+\underline{l}_{f,YZ}} \le \sum_{X \in \Gamma_f^{-}(Y)} x_{f,XY}^{t} \le \sum_{Z \in \Gamma_f^{+}(Y)} x_{f,YZ}^{t+\bar{l}_{f,YZ}}, \\ t \in \mathcal{T}_{f,Y}, Y \in \mathcal{N}_f \backslash \{k_f^d, k_f^a\}, f \in \mathcal{F} \end{aligned} \tag{6}$$

In the foregoing, $\mathcal{T}_{f,Y}$ denotes the set of all times units by which a flight f can reach Node Y from any other node along the route of that flight. Spatial continuity constraints force connectivity through a node.

The third set of constraints is composed of those that are derived from airspace capacity. To formulate the problem with capacity constraints, the sets $\mathcal{N}_f^{j+}$ and $\mathcal{N}_f^{j-}$ are defined for a flight f in the jth sector, as the set of nodes entering and leaving the jth Sector. The sector capacity constraints are given by the following[6]:

$$\sum_{f \in \mathcal{F}} \left[\sum_{Y \in \mathcal{N}_f^{j+}} \sum_{X \in \Gamma_f^{-}(Y)} x_{f,XY}^{t} - \sum_{Y \in \mathcal{N}_f^{j-}} \sum_{X \in \Gamma_f^{-}(Y)} x_{f,XY}^{t} \right] \le S_j^t, \qquad t \in \mathcal{T}, j \in \mathcal{J} \tag{7}$$

The foregoing equation counts the number of flights entering Sector j at time t, and subtracts from it, the number flights leaving the Sector at that time. This number is constrained to be less than the Sector capacity at that time, S_j^t, for a Sector $j \in \mathcal{J}$. Similar capacity constraints can be derived for airport arrival and departure capacity, but were not used in this work. Mechanisms to include arrival and departure capacity constraints are described in Ref. [1], which can also explicitly model scenarios where the arrival and departure capacity constraints are dependent on each other due to simultaneous operation on the same runways.

C. Cost Function

In the BLO model, the cost J has contributions from different components, depending on the modeling requirements of the problem. A comprehensive list is presented in Ref. 6, which not only includes the components presented in Ref. 4, but also introduces additional terms for greater flexibility in formulating TFM problems. In this work, the number of cancelled flights, overall flight ground delays, and airborne delays were penalized. These three cost function components, denoted by J_{cancel}, J_{ground}, and J_{airborne}, are listed as follows:

$$J_{\text{cancel}} = \sum_{f \in \mathcal{F}} c_{f,cancel} x_{f,k_f^d,q\left(k_f^d\right)}^{t}, \qquad t = \max \mathcal{T}_{f,k_f^d,q\left(k_f^d\right)} \tag{8}$$

$$J_{\text{ground}} = -\sum_{f \in \mathcal{F}} \sum_{t \in \mathcal{T}_{f,k_f^d,q\left(k_f^d\right)}} c_{f,G}(t) \left(x_{f,k_f^d,q\left(k_f^d\right)}^{t} - x_{f,k_f^d,q\left(k_f^d\right)}^{t-1} \right) \tag{9}$$

$$J_{\text{airborne}} = -\sum_{f\in\mathcal{F}}\left[\sum_{t\in\mathcal{T}_{f,p\left(k_f^a\right),k_f^a}} c_{f,T}(t)\left(x^{t}_{f,p\left(k_f^a\right),k_f^a} - x^{t-1}_{f,p\left(k_f^a\right),k_f^a}\right)\right.$$
$$\left. - \sum_{t\in\mathcal{T}_{f,k_f^d,q\left(k_f^d\right)}} c_{f,G}(t)\left(x^{t}_{f,k_f^d,q\left(k_f^d\right)} - x^{t-1}_{f,k_f^d,q\left(k_f^d\right)}\right)\right] \tag{10}$$

where $c_{f,T}(t) = c_T \cdot \left(t - r_f\right)$ and $c_{f,G}(t) = c_G \cdot \left(t - d_f\right)$ (with constant c_T and c_G) are coefficients such that each additional unit of delay from scheduled arrival and departure has a proportionately heavier penalty. Alternative formulations include the so-called superlinear cost function Ref. 4 with additional penalty on larger delays. Computational experiments described in Ref. 4 have shown that this results in a more equitable distribution of delays over the set of flights. It should be noted that although the cost function coefficients can be functions of the time unit, they are not functions of the decision variables, and consequently, the resulting cost function is still linear as shown in Equation (1). It should also be noted that by virtue of the fact that the present work uses a *maximization* approach (see Equation (1)), all of the cost functions shown in Equations (8) through (10) are multiplied by -1.

D. Example Problem Description

In order to motivate a better understanding of the IADS problem formalism using BLO, the problem is approached by using an example to develop the constraint equations. The simplicity of the example results in a relatively small number of linear constraints that can be solved using enumeration, and insights can be gained on the impact of constraints on the solution.

For the example problem, two flights are assumed. For demonstration purposes, only the surface routes (gate to runway threshold) of the aircrafts are assumed. The technique presented here and be extended to include the arrival, departure and enroute operations, without loss of generality. The link-node model for both flights (shown in red and green) is depicted in Figure 1.

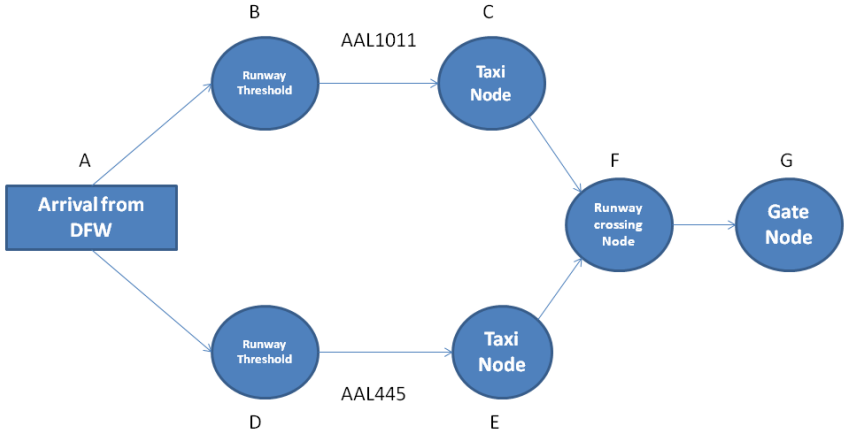


Figure 1. Node-Link Layout of the Two Flights

1. Constraint Generation using BLO Formalism

Let $\mathcal{F} = \{\text{AAL1011}\,, \text{AAL445}\,\}$ be the set of flights in the simulation; $f_1 = \{\text{AAL1011}\}$ and $f_2 = \{\text{AAL445}\}$ will be referred to as Flight 1 and Flight 2 respectively. The set of nodes for Flight 1 is given by $\mathcal{N}_1 = \{A, B, C, F, G\}$ and for Flight 2 is given by $\mathcal{N}_2 = \{A, D, E, F, G\}$ (see Figure 1). The trajectories for Flight 1 and Flight 2 are defined by the set of arcs connecting nodes that constitute their path. In other words, $\mathcal{A}_1 = \{(A,B),(B,C),(C,F),(F,G)\}$ and $\mathcal{A}_2 = \{(A,D),(D,E),(E,F),(F,G)\}$. Since each flight has only one possible route, the scheduled route sets are given by $\mathcal{A}_1^* = \mathcal{A}_1$ and $\mathcal{A}_2^* = \mathcal{A}_2$. Keeping in mind the BLO formalism, the first node / airport boundary for both

flights is Node A shown in Figure 1. Forward connectivity is given by $\Gamma_1^+(A) = B$, $\Gamma_1^+(B) = C$, $\Gamma_1^+(C) = F$, $\Gamma_1^+(F) = G$, $\Gamma_1^+(G) = \emptyset$, $\Gamma_2^+(A) = D$, $\Gamma_2^+(D) = E$, $\Gamma_2^+(E) = F$, $\Gamma_2^+(F) = G$, $\Gamma_2^+(G) = \emptyset$. Similarly, reverse connectivity is given by $\Gamma_1^-(A) = \emptyset$, $\Gamma_1^-(B) = A$, $\Gamma_1^-(C) = B$, $\Gamma_1^-(F) = C$, $\Gamma_1^-(G) = F$, $\Gamma_2^-(A) = \emptyset$, $\Gamma_2^-(D) = A$, $\Gamma_2^-(E) = D$, $\Gamma_2^-(F) = E$, $\Gamma_2^-(G) = F$.

The transit time for both flights along the links which correspond to their respective routes are equal to 1 time unit, except for Flight 2 in Arc AD, which requires 2 time units. Departure times (i.e. arrival at Node A) are both at 1 time unit, i.e. $d_1 = d_2 = 1$. The transit times are then given as follows:

$$\begin{aligned}
l_{1,AB} = \bar{l}_{1,AB} = \underline{l}_{1,AB} = 1 \\
l_{1,BC} = \bar{l}_{1,BC} = \underline{l}_{1,BC} = 1 \\
l_{1,CF} = \bar{l}_{1,CF} = \underline{l}_{1,CF} = 1 \\
l_{1,FG} = \bar{l}_{1,FG} = \underline{l}_{1,FG} = 1 \\
l_{2,AD} = \bar{l}_{2,AD} = \underline{l}_{2,AD} = 2 \\
l_{2,DE} = \bar{l}_{2,DE} = \underline{l}_{2,DE} = 1 \\
l_{2,EF} = \bar{l}_{2,EF} = \underline{l}_{2,EF} = 1 \\
l_{2,FG} = \bar{l}_{2,FG} = \underline{l}_{2,FG} = 1
\end{aligned} \tag{11}$$

Flight 1 and Flight 2 are allowed to delay their departure by up to 2 time units. Any additional delay will result in flight cancellation. Given that $d_1 = 1$, the set of feasible time units during which Flight 1 can be at Node A is given by {1,2,3}. In other words, if a Flight 1 has not departed from Node A by the time unit 3, then it is considered canceled. Feasible time units for arcs AB, BC, CF, and CG and nodes B, C, F, and G, can be obtained using the initial feasible set and the transit times shown in Equations (11).

$$\begin{aligned}
\mathcal{T}_{1,A}^* &= \{1,2,3\} \\
\mathcal{T}_{1,AB}^* &= \left\{t + \underline{l}_{1,AB} | t \in \mathcal{T}_{1,A}^*\right\} \cup \left\{t + l_{1,AB} | t \in \mathcal{T}_{1,A}^*\right\} \cup \left\{t + \bar{l}_{1,AB} | t \in \mathcal{T}_{1,A}^*\right\} = \{2,3,4\} \\
\mathcal{T}_{1,B}^* &= \mathcal{T}_{1,AB}^* = \{2,3,4\} \\
\mathcal{T}_{1,BC}^* &= \left\{t + \underline{l}_{1,BC} | t \in \mathcal{T}_{1,B}^*\right\} \cup \left\{t + l_{1,BC} | t \in \mathcal{T}_{1,B}^*\right\} \cup \left\{t + \bar{l}_{1,BC} | t \in \mathcal{T}_{1,B}^*\right\} = \{3,4,5\} \\
\mathcal{T}_{1,C}^* &= \mathcal{T}_{1,BC}^* = \{3,4,5\} \\
\mathcal{T}_{1,CF}^* &= \left\{t + \underline{l}_{1,CF} | t \in \mathcal{T}_{1,C}^*\right\} \cup \left\{t + l_{1,CF} | t \in \mathcal{T}_{1,C}^*\right\} \cup \left\{t + \bar{l}_{1,CF} | t \in \mathcal{T}_{1,C}^*\right\} = \{4,5,6\} \\
\mathcal{T}_{1,F}^* &= \mathcal{T}_{1,CF}^* = \{4,5,6\} \\
\mathcal{T}_{1,FG}^* &= \left\{t + \underline{l}_{1,FG} | t \in \mathcal{T}_{1,F}^*\right\} \cup \left\{t + l_{1,FG} | t \in \mathcal{T}_{1,F}^*\right\} \cup \left\{t + \bar{l}_{1,FG} | t \in \mathcal{T}_{1,F}^*\right\} = \{5,6,7\}
\end{aligned} \tag{12}$$

In the foregoing equation, the sets of feasible times are used to define the BLO variables which represent Flight 1. In other words, the complete trajectory of Flight 1 can be specified using 12 binary variables $x_{1,AB}^2, x_{1,AB}^3, x_{1,AB}^4, x_{1,BC}^3, x_{1,BC}^4, x_{1,BC}^5, x_{1,CF}^4, x_{1,CF}^5, x_{1,CF}^6, x_{1,FG}^5, x_{1,FG}^6, x_{1,FG}^7$. Similarly feasible time set for Flight 2 can be found which is completely specified by 12 binary variables whch are given by, $x_{2,AD}^3, x_{2,AD}^4, x_{2,AD}^5, x_{2,DE}^4, x_{2,DE}^5, x_{2,DE}^6, x_{2,EF}^5, x_{2,EF}^6, x_{2,EF}^7, x_{2,FG}^6, x_{2,FG}^7, x_{2,FG}^8$.

The set of time units over which optimization takes place is $\mathcal{T} = \{1,2,3,4,5,6,7,8\}$. Keeping in mind the BLO formalism, set $\mathcal{T}_{f,XY}$ is defined for a flight f and node pair X and Y, which is the smallest continuous interval containing set $\mathcal{T}_{f,XY}^*$. Similarly, set $\mathcal{T}_{f,X}$ is the smallest continuous interval containing $\mathcal{T}_{f,X}^*$ for a node X. In the example considered in this section, the two sets are identical because each flight has only one possible route. The use of feasible time sets is equivalent to the process of constraint elimination performed by sophisticated linear programming codes. For example, the knowledge that a variable cannot have non-zero value at any time instant prior to the earliest feasible time instant is implicitly recognized in this approach.

For the present problem, the temporal constraints for Flight 1 and 2 are given by the following two sets of equations:

$$\begin{aligned} x_{1,AB}^{2} \le x_{1,AB}^{3} \le x_{1,AB}^{4} \\ x_{1,BC}^{3} \le x_{1,BC}^{4} \le x_{1,BC}^{5} \\ x_{1,CF}^{4} \le x_{1,CF}^{5} \le x_{1,CF}^{6} \\ x_{1,FG}^{5} \le x_{1,FG}^{6} \le x_{1,FG}^{7} \end{aligned} \tag{13}$$

$$\begin{aligned} x_{2,AD}^{3} \le x_{2,AD}^{4} \le x_{2,AD}^{5} \\ x_{2,DE}^{4} \le x_{2,DE}^{5} \le x_{2,DE}^{6} \\ x_{2,EF}^{5} \le x_{2,EF}^{6} \le x_{2,EF}^{7} \\ x_{2,FG}^{6} \le x_{2,FG}^{7} \le x_{2,FG}^{8} \end{aligned} \tag{14}$$

The spatial continuity constraints for Flight 1 are given by,

$$\begin{aligned} x_{1,AB}^{i} = x_{1,BC}^{j}, \forall (i,j) \in \{(2,3),(3,4),(4,5)\} \\ x_{1,BC}^{i} = x_{1,CF}^{j}, \forall (i,j) \in \{(3,4),(4,5),(5,6)\} \\ x_{1,CF}^{i} = x_{1,FG}^{j}, \forall (i,j) \in \{(4,5),(5,6),(6,7)\} \end{aligned} \tag{15}$$

For Flight 2 similarly,

$$\begin{aligned} x_{2,AD}^{i} = x_{2,DE}^{j}, \forall (i,j) \in \{(3,4),(4,5),(5,6)\} \\ x_{2,DE}^{i} = x_{2,EF}^{j}, \forall (i,j) \in \{(4,5),(5,6),(6,7)\} \\ x_{2,EF}^{i} = x_{2,FG}^{j}, \forall (i,j) \in \{(5,6),(6,7),(7,8)\} \end{aligned} \tag{16}$$

Using Equations (13), (14), (15), (16), it may be observed that the entire problem is determined by 6 variables $x_{1,AB}^{2}, x_{1,AB}^{3}, x_{1,AB}^{4}, x_{2,AD}^{3}, x_{2,AD}^{4}, x_{2,AD}^{5}$ and the following constraints:

$$\begin{aligned} x_{1,AB}^{2} - x_{1,AB}^{3} \le 0 \\ x_{1,AB}^{3} - x_{1,AB}^{4} \le 0 \\ x_{2,AD}^{3} - x_{2,AD}^{4} \le 0 \\ x_{2,AD}^{4} - x_{2,AD}^{5} \le 0 \\ x_{1,AB}^{2} \in \{0,1\} \\ x_{1,AB}^{3} \in \{0,1\} \\ x_{1,AB}^{4} \in \{0,1\} \\ x_{2,AD}^{3} \in \{0,1\} \\ x_{2,AD}^{4} \in \{0,1\} \\ x_{2,AD}^{5} \in \{0,1\} \end{aligned} \tag{17}$$

2. Linear Relaxation of the BLO Formulation

For the example studied, it is straightforward to demonstrate that the linear relaxation is valid. In other words, constraints for the first flight, given by $x_{1,AB}^{2} - x_{1,AB}^{3} \le 0$, $x_{1,AB}^{3} - x_{1,AB}^{4} \le 0$, together with $x_{1,AB}^{2}, x_{1,AB}^{3}, x_{1,AB}^{4} \in \{0,1\}$ are relaxed into the following inequalities:

$$
\begin{aligned}
x_{1,AB}^2 - x_{1,AB}^3 &\leq 0 \\
x_{1,AB}^3 - x_{1,AB}^4 &\leq 0 \\
0 \leq x_{1,AB}^2 &\leq 1 \\
0 \leq x_{1,AB}^3 &\leq 1 \\
0 \leq x_{1,AB}^4 &\leq 1
\end{aligned}
\tag{18}
$$

The bounds on the variables define a 3-dimensional cube as shown in Figure 2. The relaxed problem by is bounded by construction, therefore, an unbounded solution obtained from any solver (e.g. simplex) is not expected. The extreme points are clearly the 8 corners of the cube, viz. $(0,0,0), (0,0,1), \ldots, (1,1,1)$.

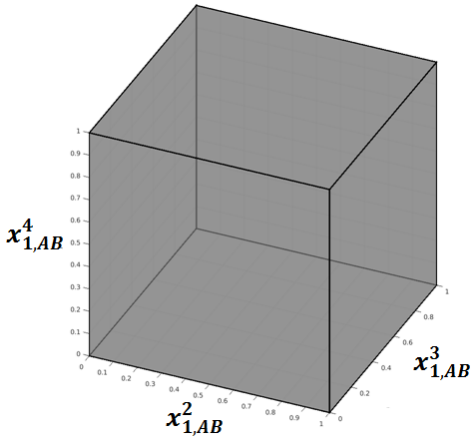


Figure 2. Feasible Region obtained from BLO Variable Bounds

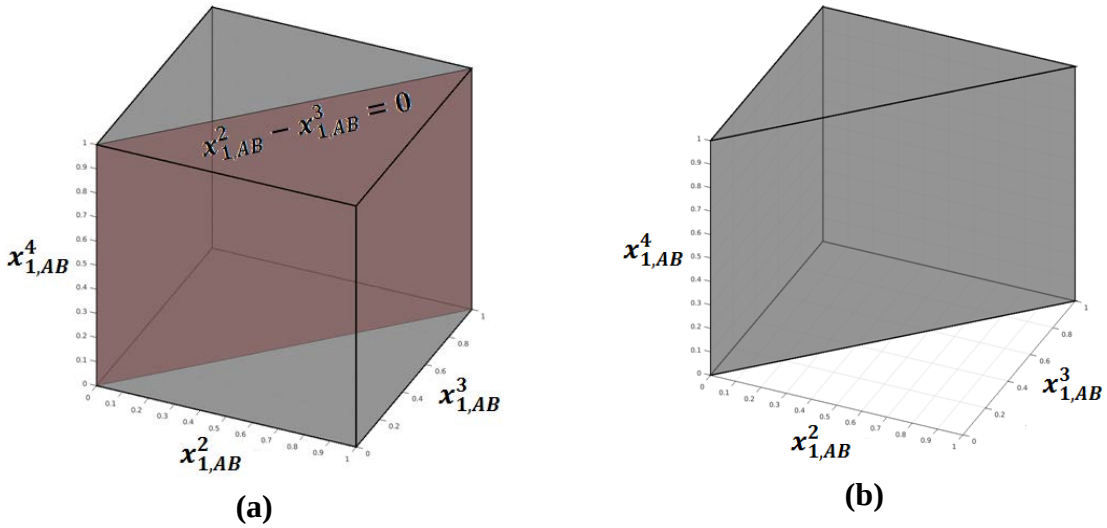


Figure 3. Modification of Feasibility Space with $x_{1,AB}^2 - x_{1,AB}^3 \leq 0$

The constraint $x_{1,AB}^2 - x_{1,AB}^3 \leq 0$ is now applied to the feasibility space. The plane $x_{1,AB}^2 - x_{1,AB}^3 = 0$ is shown in red in Figure 3(a). Any point to the left of the plane satisfies the constraint, and the modified feasibility space is shown in Figure 3(b). Application of this constraint removes 2 extreme points: $(1,0,0)$ and $(1,0,1)$.

After processing all the constraints, the final feasible region has 4 extreme points: $(0,0,0)$, $(0,0,1)$, $(0,1,1)$, and $(1,1,1)$ and the relaxation of the binary problem into a bounded LP is valid. In general, the relaxation holds for large problems composed of BLO variables. If there are k BLO variables, the initial feasibility region is a k-dimensional hypercube since each BLO variable is bounded in the range $[0,1]$ after relaxation. Every constraint in an individual

flight's problem is of the form $a^\top x \leq 0$ or $a^\top x = 0$. In computational experience, the constraint coefficients are $+1$ or -1, which results in hyperplanes which pass through one of the corner points of the hypercube. However, it is possible that coupling constraints between flights can result in new corner points which have a non-binary component. Bertsimas and Stock-Patterson[1] have noted that the linear relaxation holds "almost everywhere", and Rios and Ross[3] have noted that the relaxation strength increases when decomposition methods are used.

3. Cost Function

For the example problem, the cancellation cost coefficient is set to a large value (100) to ensure that delays are preferred to flight cancellations. It follows that,

$$J_{\text{cancel}} = 100x_{1,AB}^4 + 100x_{2,AD}^5 \tag{19}$$

The foregoing cost function favors scenarios where the flight has departed by its last feasible time unit of departure. It is possible to enforce non-cancellation by using an equality constraint of the form $x_{1,AB}^4 = 1$, however, this approach can impact the availability of a basic feasible solution to the LP, since all the BLO variables cannot be arbitrarily set to 0 in order to ensure coupling constraints are not violated.

In the present example, the airborne delay cost is not used. This is because the minimum, maximum, and scheduled transit times in a link are identical and a flight will not be able to add additional delays beyond the first link. Therefore, all delays are captured in the 'ground' segment of the problem, i.e. AB for Flight 1, and AD for Flight 2. Ground delay cost with $c_G = 1$ is given as follows:

$$\begin{aligned} J_{\text{ground}} = &- \sum_{t\in\{2,3,4\}} (t-2)\left(x_{1,AB}^t - x_{1,AB}^{t-1}\right) - \sum_{t\in\{3,4,5\}} (t-3)\left(x_{2,AD}^t - x_{2,AD}^{t-1}\right) \\ &= -\left(x_{1,AB}^3 - x_{1,AB}^2\right) - 2\left(x_{1,AB}^4 - x_{1,AB}^3\right) - \left(x_{2,AD}^4 - x_{2,AD}^3\right) \\ &- 2\left(x_{2,AD}^5 - x_{2,AD}^4\right) \\ &= -2x_{1,AB}^4 + x_{1,AB}^3 + x_{1,AB}^2 - 2x_{2,AD}^5 + x_{2,AD}^4 + x_{2,AD}^3 \end{aligned} \tag{20}$$

The total cost function, given by adding the cancellation and ground delay cost components obtained from Equation (19) and (20) is given by

$$J = 98x_{1,AB}^4 + x_{1,AB}^3 + x_{1,AB}^2 + 98x_{2,AD}^5 + x_{2,AD}^4 + x_{2,AD}^3 \tag{21}$$

E. Coupling Constraints: Separation and Capacity

The case is now considered where a separation is desired between the two flights at the merge point F. Let t_1 and t_2 denote the time units at which Flight 1 and Flight 2 arrive at Node F along their respective trajectories. From Equation (3) it can be shown that:

$$\begin{aligned} t_1 = \cdots + 3(0-0) + 4\left(x_{1,CF}^4 - 0\right) + 5\left(x_{1,CF}^5 - x_{1,CF}^4\right) + 6\left(x_{1,CF}^6 - x_{1,CF}^5\right) \\ + 7\left(x_{1,CF}^6 - x_{1,CF}^6\right) + \cdots \end{aligned} \tag{22}$$

In the foregoing, since the set of feasible times at which Flight 1 can arrive at Node F is given by $\mathcal{T}_{1,F}^* = \{4,5,6\}$, $x_{1,CF}^t = 0$ when $t < 4$ (Flight 1 cannot arrive at F before time unit 4), and $x_{1,CF}^t = x_{1,CF}^6$ when $t > 6$ (if Flight 1 has/has not arrive at F by time unit 6, it can/not have arrived by any time after that). Using variable elimination discussed in Section II.D.1, the foregoing equation can be simplified to the following:

$$t_1 = 6x_{1,AB}^4 - x_{1,AB}^3 - x_{1,AB}^2 \tag{23}$$

Similarly, time of arrival of Flight 2 at Node F can be written as follows:

$$t_2 = 7x_{2,AD}^5 - x_{2,AD}^4 - x_{2,AD}^3 \tag{24}$$

1. Reformulation of Separation Constraint as a Capacity Constraint

If a separation of 2 time units is desired, then the following constraint is required:

$$\begin{aligned} |t_1 - t_2| \geq 2 \\ \text{or, } \left|\left(6x_{1,AB}^4 - x_{1,AB}^3 - x_{1,AB}^2\right) - \left(7x_{2,AD}^5 - x_{2,AD}^4 - x_{2,AD}^3\right)\right| \geq 2 \end{aligned} \tag{25}$$

However, the foregoing equation is a non-convex constraint and cannot be implemented as a linear program. Therefore a sequencing heuristic must be imposed in order to make the constraint convex. In other words, only one of $t_1 - t_2 \geq 2$ and $t_2 - t_1 \geq 2$ can be imposed, and this implies that *a priori* knowledge of whether Flight 1 arrives

before or after Flight 2 is available. This poses a huge disadvantage and may lead to suboptimal solutions. A more serious shortcoming of the foregoing constraint is the loss of the validity of the linear relaxation, which can be easily seen if the foregoing problem is solved using enumeration. However, solving the linear relaxation using either simplex or interior point methods shows that multiple solutions exist for this problem, with identical costs. For instance, $x^2_{1,AB} = 1$, $x^3_{1,AB} = 1$, and $x^4_{1,AB} = 1$, and $x^3_{2,AD} = t$, $x^4_{2,AD} = 1 - t$, and $x^5_{2,AD} = 1$, with $t \in [0,0.5]$ are all valid solutions to the foregoing problem, each with a cost of $J = 199$.

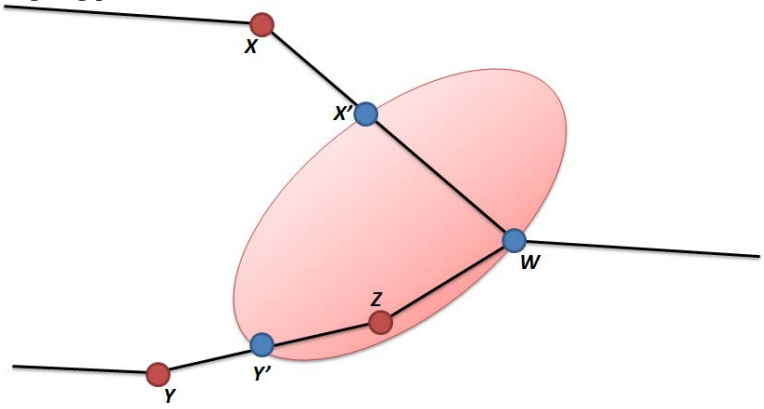


Figure 4. Modeling Separation Constraint as a Capacity Constraint

An alternative approach is to impose the separation constraint as a capacity constraint, which is achieved by modeling a sector / volume of airspace superimposed on the flight trajectories. Separation can be imposed on a single route or at merge points, and the second case is considered for generality. A merge point for two routes is depicted in Figure 4. The two routes merge at Node W, and the preceding node for the first route is X, and that for the second route is Z. Consider a desired separation of t_{sep} time units. It is assumed without loss of generality that the transit time for Flight 1 on Route 1 on the arc connecting Node X and W is greater than t_{sep}. Conversely, Flight 2 on Route 2 requires a flight time from Node Z to Node W which is less than t_{sep}. A sector (depicted in red) is now created such that the point of entry into the sector for Flight 1 is given by Node X′ and that for Flight 2 is given by Node Y′. It should be noted that

$$\begin{gathered} l_{1,X'W} = \underline{l}_{1,X'W} = \bar{l}_{1,X'W} = t_{sep} \\ l_{1,XX'} = l_{1,XW} - t_{sep} \\ l_{2,Y'Z} = t_{sep} - l_{2,ZW} \\ l_{2,YY'} = l_{2,YZ} - t_{sep} + l_{2,ZW} \end{gathered} \tag{26}$$

To simplify analysis, it is assumed that the maximum and minimum transit times are identical to the nominal transit time inside the sector, and the maximum and minimum transit times for the partial segment outside the sector is adjusted such that they are at least zero.

In the present example, $l_{1,CF}$ and $l_{2,EF}$ are both equal to 1, therefore, Nodes C, E, and F constitute the boundary nodes for the sector. In line with the BLO formalism, let S denote this sector, and $\mathcal{N}^S = \{C, E, F\}$ denote the set of nodes which belong to this sector. The sets $\mathcal{N}_1^S = \{C, F\}$ and $\mathcal{N}_2^S = \{E, F\}$ denote the set of nodes in sector S for Flight 1 and Flight 2, respectively. The sets of incoming and outgoing border nodes are defined as follows:

$$\begin{aligned} \mathcal{N}_f^{j+} &= \left\{m | n \in \mathcal{N}_f^j, m = \Gamma_f^-(n) \notin \mathcal{N}_f^j\right\} \\ \mathcal{N}_f^{j-} &= \left\{n | m' \in \mathcal{N}_f^j, m' = \Gamma_f^+(n) \notin \mathcal{N}_f^j\right\} \end{aligned} \tag{27}$$

In the present example, $\mathcal{N}_1^{S+} = \{C\}$, $\mathcal{N}_1^{S-} = \{G\}$, $\mathcal{N}_2^{S+} = \{E\}$, $\mathcal{N}_2^{S-} = \{G\}$. The sector capacity constraint is given by the following expression:

$$\sum_{f\in\mathcal{F}}\left[\sum_{Y\in\mathcal{N}_f^{j+}}\sum_{X\in\Gamma_f^-(Y)}x_{f,XY}^t-\sum_{Y\in\mathcal{N}_f^{j-}}\sum_{X\in\Gamma_f^-(Y)}x_{f,XY}^t\right]\le S_j^t,\qquad t\in\mathcal{T},j\in\mathcal{J} \tag{28}$$

Where S_j^t is the capacity of the j^{th} sector at time t in the simulation. In order to enforce separation at Node F, only one flight is permitted in the sector at all times, therefore $S_j^t=1$. The sector capacity inequalities for $\mathcal{T}=\{1,2,3,4,5,6,7,8\}$ are as follows:

$$x_{1,BC}^t-x_{1,FG}^t+x_{2,DE}^t-x_{2,FG}^t\le 1, t\in[1,8] \tag{29}$$

Using the feasible time interval information (given in Equation (12) for Flight 1 and similarly can be calculated for Flight 2), the foregoing constraints reduce to the following:

$$\begin{gathered}x_{1,BC}^3\le 1\\ x_{1,BC}^4+x_{2,DE}^4\le 1\\ x_{1,BC}^5-x_{1,FG}^5+x_{2,DE}^5\le 1\\ x_{1,BC}^5-x_{1,FG}^6+x_{2,DE}^6-x_{2,FG}^6\le 1\\ x_{1,BC}^5-x_{1,FG}^7+x_{2,DE}^6-x_{2,FG}^7\le 1\\ x_{1,BC}^5-x_{1,FG}^7+x_{2,DE}^6-x_{2,FG}^8\le 1\end{gathered} \tag{30}$$

The first of the foregoing inequalities is always satisfied due to the linear relaxation. By utilizing variable elimination, Equation (30) simplifies to the following:

$$\begin{gathered}x_{1,AB}^3+x_{2,AD}^3\le 1\\ x_{1,AB}^4-x_{1,AB}^2+x_{2,AD}^4\le 1\\ x_{1,AB}^4-x_{1,AB}^3+x_{2,AD}^5-x_{2,AD}^3\le 1\\ x_{1,AB}^4-x_{1,AB}^4+x_{2,AD}^5-x_{2,AD}^4\le 1\\ x_{1,AB}^4-x_{1,AB}^4+x_{2,AD}^5-x_{2,AD}^5\le 1\end{gathered} \tag{31}$$

In the foregoing, the fourth constraint reduces to a variable bound which is always satisfied due to the linear relaxation, and the fifth constraint is redundant. Therefore, the final form of the sector capacity constraint is the following:

$$\begin{gathered}x_{1,AB}^3+x_{2,AD}^3\le 1\\ x_{1,AB}^4-x_{1,AB}^2+x_{2,AD}^4\le 1\\ x_{1,AB}^4-x_{1,AB}^3+x_{2,AD}^5-x_{2,AD}^3\le 1\end{gathered} \tag{32}$$

It should be noted that the non-binary solution observed when posing the separation constraint as a $\geq$-type constraint, given by $x_{1,AB}^2=x_{1,AB}^3=x_{1,AB}^4=x_{2,AD}^5=1,\ x_{2,AD}^3=t,\ x_{2,AD}^4=1-t$ satisfies the second and third inequalities of the foregoing, but only $t=0$ satisfies the first inequality.

While the capacity constraint can be imposed on the optimization directly, it is often beneficial to pose the inequality as a "soft" constraint with an associated penalty, so that flights are canceled only if the penalty of constraint violation is very high. This allows flexibility in order to model scenarios where a small amount of constraint violation can be tolerated under extreme circumstances which would otherwise cancel flights. Moreover, in the specific example of linear programs which are generated from TFM scenarios, a basic feasible solution consisting of all variables set to zero exists, which corresponds to the case where all flights are canceled. However, this solution can violate the coupling / separation constraints when posed in "hard" form.

When a basic feasible solution to the LP is not available by intuition, a common technique is to use a two-phase simplex algorithm where the first phase penalizes artificial variables in order to generate a feasible solution which

satisfies the constraints of the problem with no regard for optimality. Thereafter, the second phase iterates upon the feasible solution to obtain an optimal solution.

The second approach, known as the "big M" method is the approach followed here. In this approach, artificial variables are added to each constraint, and penalized heavily in comparison with the other variables in the problem. Let s_{11}, s_{12}, and s_{13} denote surplus variables and s_{14}, s_{15}, s_{16} denote slack variables for the capacity constraint which can be converted to equalities:

$$\begin{aligned} x^3_{1,AB} + x^3_{2,AD} - s_{11} + s_{14} &= 1 \\ x^4_{1,AB} - x^2_{1,AB} + x^4_{2,AD} - s_{12} + s_{15} &= 1 \\ x^4_{1,AB} - x^3_{1,AB} + x^5_{2,AD} - x^3_{2,AD} - s_{13} + s_{16} &= 1 \end{aligned} \tag{33}$$

When $x^3_{1,AB} + x^3_{2,AD} \leq 1$, $s_{11} = 0$ and $s_{14} = 1 - \left(x^3_{1,AB} + x^3_{2,AD}\right)$. If $x^3_{1,AB} + x^3_{2,AD} > 1$, then $s_{11} = \left(x^3_{1,AB} + x^3_{2,AD}\right) - 1$, and $s_{14} = 0$. Therefore, if the surplus variable is penalized, the capacity violation will be minimized. Similar conclusions can be drawn for the second and third capacity constraints. Consequently, let $J_{capacity} = -c_{cap}(s_{11} + s_{12} + s_{13})$ denote the cost of capacity violation and let the total cost be modified to $J' = J + J_{capacity} = J_{cancel} + J_{ground} + J_{capacity}$. It can be easily observed using enumeration that if $c_{cap} = 200$ then the optimal cost $J' = 199$, and the optimal solution is given by, $x^2_{1,AB} = 1, x^3_{1,AB} = 1, x^4_{1,AB} = 1, x^3_{2,AD} = 0, x^4_{2,AD} = 1$ and $x^5_{2,AD} = 1$. Furthermore, because the separation constraint has been imposed as a capacity constraint, the linear relaxation is still valid[4] and a simplex-based solver will result in binary integer solutions. Moreover, a sequencing heuristic is no longer required.

III. Solution Methodology

Prior work described in Ref. [3] shows that the BSP model exhibits the so-called primal block structure[8], which is shown Figure 5. This is also known as the Dantzig-Wolfe decomposition of the constraint space[9]. This structure is also exhibited by the BLO model. The implementation details on parallel and high-performance computers are discussed in the following subsections.

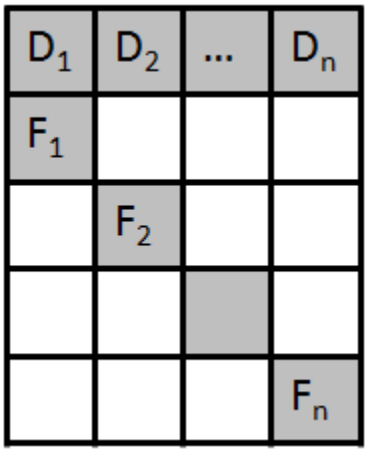


Figure 5. Primal Block Angular Structure of the TFM Constraint Matrix

A. Dantzig-Wolfe Structure

With reference to Figure 5, the constraints of the LP can be divided into master problem blocks D_1 through D_n, and sub-problem blocks F_1 through F_n, where n is the number of flights in the simulation. In other words, the LP can be written in the following form:

$$\begin{aligned} &\min c_1^\top x_1 + c_2^\top x_2 + \cdots c_n^\top x_n \\ &D_1 x_1 + D_2 x_2 + \cdots + D_n x_n = b_D \\ &F_1 x_1 = b_{F_1} \\ &F_2 x_2 = b_{F_2} \end{aligned} \tag{34}$$

$$\vdots$$

$$F_n x_n = b_{F_n}$$

$$x_1, x_2, \ldots, x_n \geq 0$$

where the variable set x_f consist of all variables $x_{f,XY}^t$ for a flight $f \in \mathcal{F}$ in the simulation. The master problem constraints composed of block matrices D_1 through D_n consists of constraints that relate the variables of multiple flights, and are composed of capacity constraints shown in Equation (7). The sub-problem constraints composed of block matrices F_1 through F_n consist of spatio-temporal constraints of individual flights and only relate variables associated with a single flight each, as shown in Equations (5) and (6). Note that in the foregoing equation the general LP form is assumed for the optimization problem, in which all constraints are equality constraints and inequalities are converted to equalities using slack, surplus, and artificial variables.

Let $P_f = \{x_f | F_f x_f = b_f, x_f \geq 0\}$. This set denotes the feasible values for the variables for the fth flight and can be rewritten as a convex combination of its extreme points and rays. In the context of the BLO formulation, the feasible region is bounded since all variables are bounded in the region [0,1], and as a consequence, P_f can be expressed as a convex combination of extreme points only. Let the extreme points of P_f be denoted by x_f^j, where $j \in \mathcal{J}_f$ and $\mathcal{J}_f$ is the set of indices iterating over the extreme points. The monolithic problem shown in Equation (34) is rewritten as the so-called master program:

$$\begin{aligned}
&\min \sum_{f \in \mathcal{F}} \sum_{j \in \mathcal{J}_f} \lambda_f^j c_f^\top x_f^j \\
&\sum_{f \in \mathcal{F}} \sum_{j \in \mathcal{J}_f} \lambda_f^j D_f x_f^j = b_D \\
&\sum_{j \in \mathcal{J}_f} \lambda_f^j = 1, \qquad \forall f \in \mathcal{F} \\
&\lambda_f^j \geq 0, \qquad \forall j \in \mathcal{J}, f \in \mathcal{F}
\end{aligned} \tag{35}$$

where λ_f^j are decision variables.

Let the number of rows in the master problem blocks $D_{1,\ldots,n}$ be denoted by m_0 and the number of rows in each sub-problem block be denoted by $m_{1,\ldots,n}$. Then, the number of constraints in the monolithic form of the problem shown in Eq. (34) is equal to $m_0 + \sum_{f \in \mathcal{F}} m_f$, whereas the master program shown in Eq. (35) consists of $m_0 + n$ constraints. The BLO formulation constraints are such that m_0 is significantly smaller than the number of monolithic constraints; however this is achieved by using an exponentially larger number of variables which correspond to the extreme points of all the sub-problems.

The advantage of the DW decomposition is that in spite of a large number of variables, at any given simplex iteration of the master program, a vast majority of the variables λ_f^j are zero and their corresponding columns in the master program simplex tableau are not used. By utilizing a process called *delayed column generation*, only potentially useful columns are added to the master program iteration. Details of the algorithm can be found in Refs. [9] and [10] and are summarized here.

Dual variables are associated with the master problem constraints, of which m_0 dual variables, denoted by σ, correspond to the capacity constraints, and n dual variables, denoted by $\pi_{1,\ldots,n}$, correspond to the convexity constraints for the n sub-problems. In the revised simplex tableau, the dual variables are obtained directly from the row of reduced costs. The fth sub-problem consists of the LP $\min \left(c_f^\top - \sigma^\top D_f\right) x_f, x_f \in P_f$. If the optimal cost to this problem is less than π_f, then the optimal solution is an extreme point, and a column $\left[\left(D_f x_f^j\right)^\top \; e_f^\top\right]^\top$ is generated, where e_f is a vector of length n consisting of zeros everywhere except for the fth entry, which is equal to 1. If the optimal cost is no less than π_f, no column is generated. Since the BLO variables are bounded, the optimal cost is always finite. The master problem reaches optimality when no columns are generated by any sub-problem.

Of importance is the fact that the fth sub-problem consists only of constraints given by the matrix F_f, and the solution to one sub-problem does not depend on the solutions to the other sub-problems. This motivates the use of parallel implementations of the DW decomposition.

B. Application of Dantzig-Wolfe Formulation

The example problem with capacity constraints can be rewritten as follows[10]:

$$\begin{gathered} \max c_1^\top x_1 + c_2^\top x_2 + c_3^\top x_3 \\ D_1 x_1 + D_2 x_2 + D_3 x_3 = b_0 \\ F_1 x_1 = b_1 \\ F_2 x_2 = b_2 \\ x_1, x_2, x_3 \geq 0 \end{gathered} \tag{36}$$

where, $x_1 = \left\{x_{1,AB}^2, x_{1,AB}^3, x_{1,AB}^4, s_1, s_2, s_3, s_4, s_5\right\}^\top$ and $x_2 = \left\{x_{2,AD}^3, x_{2,AD}^4, x_{2,AD}^5, s_6, s_7, s_8, s_9, s_{10}\right\}^\top$ where s_1, s_2, s_6, and s_7 are the slack variables for the inequality constraints and s_3, s_4, s_5, s_8, s_9, and s_{10} denote the slack variables for the linear relaxation. Let $x_3 = \{s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}\}^\top$ denote the vector of slack/surplus variables defined in Section II. The matrices D_1, D_2, and D_3 denote the master problem constraint matrices which couple the states x_1, x_2, and x_3, where D_1 and D_2 are the contributions from each flight, and, and D_3 is an auxiliary master problem block matrix in order to convert the coupling inequality constraints into equality constraints. The vector b_0 denotes the master problem capacities. The matrices are given by the following

$$\begin{gathered} D_1 = \begin{bmatrix} & 1 & & & & & \\ -1 & & 1 & & & & \\ & -1 & 1 & & & & \end{bmatrix} \\ D_2 = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ -1 & & 1 & & & & \end{bmatrix} \\ D_3 = \begin{bmatrix} -1 & & & 1 & & \\ & -1 & & & 1 & \\ & & -1 & & & 1 \end{bmatrix} \\ b_0 = [1 \quad 1 \quad 1]^\top \end{gathered} \tag{37}$$

Cost function components are given by the following:

$$\begin{gathered} c_1 = [1 \quad 1 \quad 98 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^\top \\ c_2 = [1 \quad 1 \quad 98 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^\top \\ c_3 = [-200 \quad -200 \quad -200 \quad 0 \quad 0 \quad 0]^\top \end{gathered} \tag{38}$$

Cost function vector c_3 is used to penalize any violation of the capacity constraint. Note that for coupling constraints of the form $Dx \geq b$ where $b > 0$, the slack variable should be penalized.

The sub-problems are defined as the constraints for variables that do not include other variables in the monolithic LP. The BLO model lends itself naturally to a decomposition of unique sub-problems, as shown below:

$$F_1 = \begin{bmatrix} 1 & -1 & & 1 & & & \\ & 1 & -1 & & 1 & & \\ 1 & & & & & 1 & \\ & 1 & & & & & 1 \\ & & 1 & & & & \end{bmatrix}$$

$$F_2 = \begin{bmatrix} 1 & -1 & & 1 & & & \\ & 1 & -1 & & 1 & & \\ 1 & & & & & 1 & \\ & 1 & & & & & 1 \\ & & 1 & & & & \end{bmatrix} \qquad (39)$$

$$b_1 = [0 \quad 0 \quad 1 \quad 1 \quad 1]^\top$$
$$b_2 = [0 \quad 0 \quad 1 \quad 1 \quad 1]^\top$$

It should be noted that the auxiliary sub-problem does not consist of any constraints.

C. Solution using Delayed Column Generation

Dantzig-Wolfe decomposition is predicated upon the idea that every sub-problem can be reformulated in terms of extreme points and extreme rays. Extreme points are vertices of the feasibility region, as shown in Figure 6.

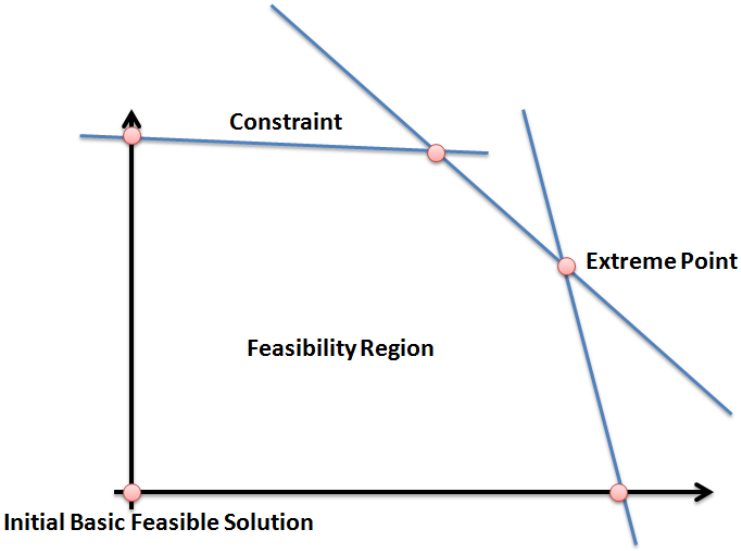


Figure 6. Extreme Points in a Linear Program

In TFM and consequently in the IADSE examples, the sub-problems are bounded by construction (the linear relaxation ensures that every variable is bounded between 0 and 1). Therefore, there are no extreme rays and every sub-problem solution can be written as a linear combination of its extreme points. Such a form of the master problem and the sub-problems can be derived for the example at hand. This would result in the reduction in the size of the master problem. For the example problem, it can be easily seen that the reduced master problem consists of 14 variables (4 convex multipliers each for 2 sub-problems, and 1 slack and surplus variable for each constraint) and 5 constraints (3 master problem constraints and 1 convexity constraint each for 2 sub-problems). In comparison, the original, monolithic problem consists of 22 variables (8 variables for each sub-problem and 1 slack and surplus variable for each coupling constraint) and 11 constraints (5 constraints for each sub-problem and 3 coupling constraints). The computational magnitude is approximately the same regardless of whether one attempts to solve the monolithic problem or the decomposed problem. In the present example, the extreme points can be identified easily, but in realistic cases the extreme points for each sub-problem may be too numerous to list exhaustively. In this scenario, the DW decomposition results in a problem with a relatively small number of constraints (sum of the number of coupling constraints and one convex constraint for each sub-problem) but a potentially exponential

number of variables (one convex multiplier for each extreme point in a sub-problem). The power of DW decomposition arises from its use of *delayed column generation*[10] because not all columns in the simplex tableau for the reduced master problem are required at any given point; in fact a large number of them can be ignored when the corresponding convex multipliers are very small. In other words, knowledge of all the extreme points for a sub-problem is not usually required and the required extreme points can be generated when necessary, by solving sub-problems independently (and in parallel) with a modified cost function. It can be shown that the solution obtained using primal simplex and delayed column generation are identical[5].

As noted in the preceding paragraph, the power of DW decomposition is predicated on the idea that the number of columns in the reduced master problem for realistic cases can be too numerous, and a relatively small number of columns are required at any iteration. For the example discussed, the simplex tableau is able to reach optimality within 3 iterations, so only 3 columns were required. The use of a limited number of columns motivates the revised simplex algorithm with compact basis. The difference between the revised simplex algorithm and 'regular' (primal) simplex algorithm is that any given iteration, the revised simplex tableau consists only of the inverse of the current basis matrix. The same iterations which are used to transform B to I (identity matrix) in the regular simplex method, will transform I to B^{-1} in the revised simplex method. For the same LP, revised simplex requires less memory because fewer columns are stored. However, the required number of computations increases because candidate entering columns must be calculated at each simplex iteration using $B^{-1}a_j$ where j is a non-basic variable, whereas in the regular simplex tableau, these columns are already available in the current simplex tableau. It can be shown that, for the current problem the revised simplex method requires 4 iterations to achieve optimality[5]. However, for the sake of brevity wee have omitted it in the current work.

IV. Parallelization and Performance Tests

Due to the massive parallelizable opportunity that the current problem poses, when DW decomposition is done, it was appropriate that its implementation in multi-core machines be investigated. The current optimization problem was hence implemented on Intel® Xeon Phi™ processors[11]. In this section we benchmark the sub-problem solver (bounded simplex solver) on both the host and the coprocessor with various problem sizes and logical cores. A comparison of load balancing and task scheduling is also provided. Table 1 provides the results.

Table 1. Execution Time vs Number of Logical Cores on Host and Coprocessor for 1000 Flights

Number of logical cores	Execution time on CPU (sec) Intel Xeon E5-2620 @ 2.00GHz	Execution time on Xeon Phi (sec) Intel Xeon Phi 7120P @ 1.00GHz
1	14.3790	139.4403
16	2.6429	11.7328
64	N/A	4.5658
120	N/A	2.7172
240	N/A	4.8757

Note that the peak performance is generally reached by reducing a few cores from the maximum number of cores. Both the host and the coprocessor needs a few cores to run system application, loading them with task will quench performance. Also note that the coprocessor reaches maximum performance at 120 logical core (61 cores – 1 core for system * 2 way hyper-threading = 120 logical cores). Increasing logical cores—hence increasing hyperthreading capacity will also quench performance due to maximum bandwidth limitations.

In this test, we time the execution time for 50, 100, 1000, and 4000 flights using the best result obtained in Table 1: 22 logical cores on host and 120 logical cores on the device. Table 2 lists the results.

Table 2. Problem Size vs Host and Coprocessor Best Execution Time

Problem Size	Execution time on CPU (sec) Intel Xeon E5-2620 @ 2.00GHz	Execution time on Xeon Phi (sec) Intel Xeon Phi 7120P @ 1.00GHz
50	0.2600	0.4997
100	0.3127	0.7124

1000	2.32511	2.7172
4000	15.6016	14.8421

V. Application of the IADS Solver to New York TRACON

The current algorithm was implemented to schedule flights in the New York N90 TRACON. Only the traffic in the terminal area is considered. To standardize the arrival routes, each flight is assumed to follow a certain route. The standardized routes are given Standard Terminal Arrival Routes (STARs) and Standard Instrument Departures (SIDs) for arrivals and departure respectively.

For the surface, only arrivals and departures to JFK airport is considered. The surface in JFK is represented as runway nodes, taxi nodes, ramp nodes and gate nodes. Typical surface traffic in JFK is seen in Figure 7.

Figure 7. Surface Traffic at JFK

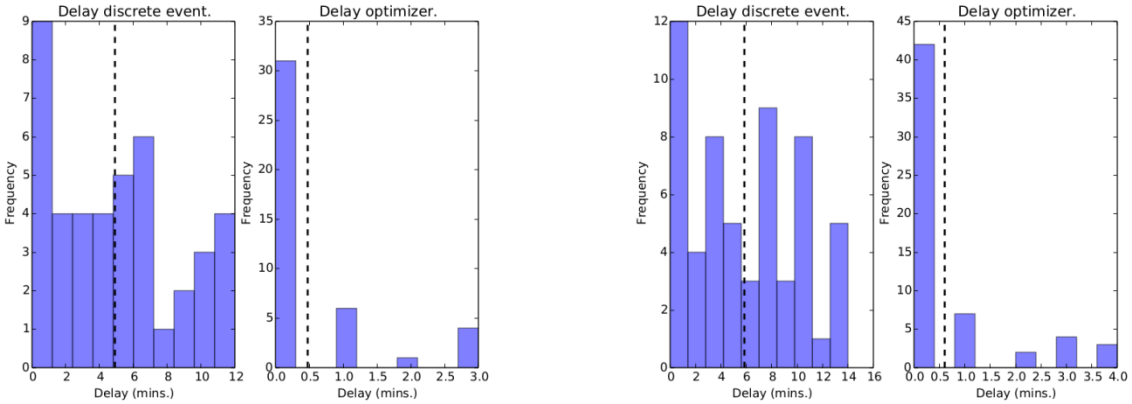
A. Numerical Results

It is assumed that the traffic data is streamed to the BLO scheduler. It means that the BLO scheduler was run in cycles. Each cycles consisted of a collection of aircraft which is in the terminal area but has not yet reached the gate. If an aircraft reaches the gate it is removed from the problem. Furthermore the results of the BLO based optimizer is compared with a discrete-event simulation-based (DES-based) deconflicting algorithm. The discrete event simulation algorithm is based on first come first serve basis.

Two scenarios are constructed; in the first scenario, separation requirement for all flights is 30 seconds at merge points, whereas in the second scenario separation requirement is 60 seconds at merge points. These scenarios reflect airport operations impacted by arrival capacity, i.e. when operations are limited by factors such as weather, it is necessary to increase the spacing between arrival flights.

In each scenario, schedules are obtained using the DES as well as the optimizer, using streaming data. The first 'snapshot' consists of flights in the system at T=15 minutes, and the second 'snapshot' consists of flights in the system at T=30 minutes. It is important to note that the streaming data is from historical sources, and the TFM directives (delays) obtained by the optimizer or DES are ignored.

1.1.1. Separation of 30 Seconds

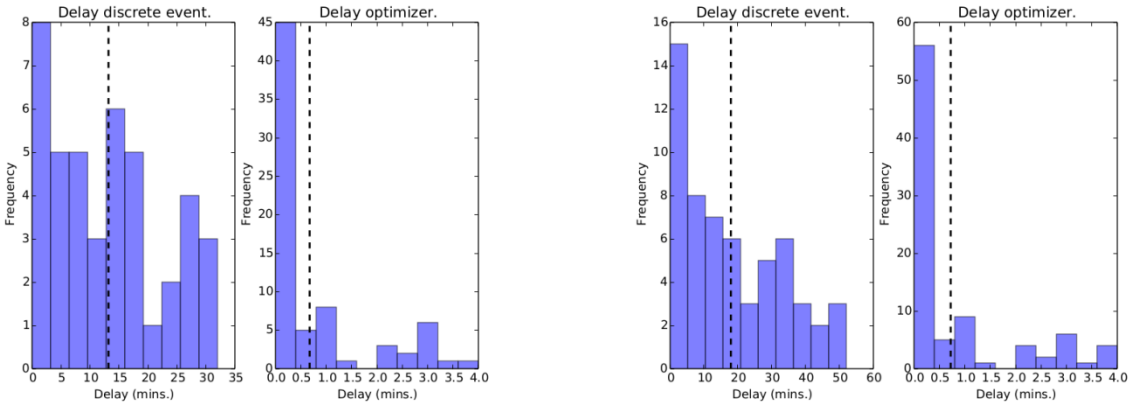


(a) T=15 Minutes **(b) T=30 minutes**

Figure 8. System Delays Obtained using DES and Optimizer for Separation of 30 Seconds

Delays obtained for flights in the simulation for a terminal-area separation of 30 seconds is shown in Figure 8(a) and (b). The delays obtained from the optimizer are significantly smaller than those obtained from the DES. At a given iteration of the streaming data, there are approximately 60 flight operations in the TRACON. The mean of the delays is shown by the broken line, however, due to the small number of flights, the standard error associated with the mean is significant and the mean itself is not necessarily a good indicator of the performance. However, the x-axis on both figures show that the maximum delay obtained from the optimizer is 4 minutes, whereas the DES shows delays of up to 12 minutes. Figure 8(a) shows the result of the scheduler when the streaming data is at time stamp of 15 minutes, and Figure 8(b) shows the result at time stamp 30 minutes. As time progresses, the increase in the mean delay is less than 1 minute in both cases, although the number of flights with larger delays increases slightly using both methods.

1.1.2. Separation of 60 Seconds



(a) T=15 Minutes **(b) T=30 minutes**

Figure 9. System Delays Obtained using DES and Optimizer for Separation of 60 Seconds

Delay effects are more pronounced when the separation requirement is increased to 60 seconds, as shown in Figure 9(a) and Figure 9(b). In comparison with Figure 8, delays obtained DES increase by approximately 2 ×. The increase in the delays from the optimizer are modest; although the maximum delay does not increase significantly, the proportion of flights which experience larger delays increases. Again, as noted before, the results from two different points in time show that the delays increase as time progresses.

VI. Conclusion

The current work implements an integrated arrival departure surface enroute optimization algorithm based on the Bertsimas-Lulli Odoni model, for scheduling aircraft. A novel method of converting separation constraints between two aircraft into airspace capacity constraints was proposed, such that it was consistent with the BLO formalism. The problem was decomposed into a number of sub-problems and a master problem by using Dantzig-Wolfe (DW) decomposition and strong LP relaxation. Due to DW decomposition the problem was amenable to be solved using parallel machines.

The parallelized BLO solver was implemented to schedule flights in New York N90 TRACON with the TRACON situation being constantly updated using data streams. The method was then compared with baseline first-come-first-serve methodology, implemented using discrete event simulation. It was found that the BLO based optimizer performed better in minimizing delays and thus increasing throughput.

Acknowledgments

This research was supported under the NASA Contract No. NNX14CA65P, with Dr. Kee Palopo serving as the Technical Monitor/COTR. The authors wish to thank the Dr. Kee Palopo for his interest and support of this work. Thanks are due to Dr. Gano Chatterjee, Yun Zheng, and Chok Lai of UC Santa Cruz for several discussions and data for carrying out this research.

References

[1]Bertsimas D., and Patterson S. S., "The Air Traffic Flow Management Problem with Enroute Capacities," *Operations Research*, Vol. 46, No. 3, May-June 1998.

[2]Rios J., and Ross K., "Parallelization of the Traffic Flow Management Problem," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, Aug. 18-21, 2008. AIAA-2008-6519

[3]Rios J., and Ross K., "Massively Parallel Dantzig-Wolfe Decomposition Applied to Traffic Flow Scheduling," *Journal of Aerospace Computing, Information, and Communication*, Vol.7, No.1, pp 32-45, Jan 2010.

[4]Bertsimas, D., Lulli, G., and Odoni, A., "An Integer Optimization Approach to Large-Scale Air Traffic Flow Management," *Operations Research*, Vol. 59, No. 1, pp. 211-227, January-February 2011.

[5]Sengupta, P., Dutta, P. , Kwan, J. S., Tsai, J., and Menon, P. K., "National Airspace System Scale Integrated Arrival-Departure-Surface-Enroute Optimization," Final Report prepared under NASA SBIR Phase III Contract No. NNX14CA65P, Oct., 2015.

[6]Agustin, A., Alonso-Ayuso, A., Escudero, L. F., and Pizarro, C., "On Air Traffic Flow Management with Rerouting. Part I: Deterministic Case," *European Journal of Operational Research*, Vol. 219, No. 1, pp. 156-166, May 2012.

[7]Sengupta, P., Kwan, J., and Menon, P. K., "Optimal Traffic Flow Scheduling Using High-Performance Computing," *Journal of Aerospace Information Systems*, Vol. 12, No. 10, pp. 661-671, 2015.

[8]Tebboth, J. R., *A Computational Study of the Dantzig-Wolfe Decomposition*, Ph.D. Dissertation, University of Buckingham, United Kingdom, 2001.

[9]Dantzig G B., and Wolfe P., "Decomposition Principle for Linear Programs," *Operations Research*, Vol. 8, No. 1, pp. 101-111, Jan-Feb 1960.

[10]Bertsimas D., & Tsitsiklis J. N., "Introduction to Linear Optimization," *Athena Scientific*, Nashua NH, 1997.

[11]Jeffers, J., & Reinders, J., "Intel Xeon Phi coprocessor high-performance programming," *Elsevier*, Boston, MA, 2013.