Parallel Computing for DoD Airlift Allocation

Steven Baker[1], Udatta Palekar[3], Gagan Gupta[2], Laxmikant Kale[2], Akhil Langer[2], Mark Surina[1], and Ramprasad Venkataraman[2]

[1]The MITRE Corporation

[2]Department of Computer Science, University of Illinois at Urbana-Champaign

[3]Department of Business Administration, University of Illinois at Urbana-Champaign

March 9, 2012

Abstract

This research demonstrates the applicability of parallel computing to Air Mobility Command's aircraft allocation problem. The Tanker Airlift Control Center (TACC) is responsible for worldwide tasking of DoD airlift in a manner that is robust against random future events. Our work seeks to improve this process using a parallel stochastic mathematical program that allocates aircraft to various mission sets during a planning cycle. To reconcile real-world mission changes, we incorporate random realizations of the subsequent cycle's execution, accounting for humanitarian/wartime contingencies, cargo fluctuations, aircraft breakdowns, and short-notice special missions.

The technical approach employs a two-stage stochastic mixed-integer program with complete recourse. The first stage allocates aircraft by home base and aircraft type

using information provided from second-stage schedules of 100+ possible future outcomes. Parallel processors compute these second stage realizations, each of which is a moderately sized linear program. The first and parallelized second stages provide mutual feedback to convergence using multi-cut Benders' decomposition.

Results on several prototype problems indicate a parallel processing speedup of over eight times the single-processor computation time; parallelizing the first stage holds greater promise for more detailed modeling. Additionally, we demonstrate a significant benefit from a small allocation adjustment: preserving a modest airlift reserve capacity incurs minor additional expense, but is enormously cost-effective when the future requirements turn out to be higher than expected. These results provide the prelude for additional research that will scale problem complexity and realism to the realm of operational TACC allocations.

1 Introduction

Serial computing techniques, the current state-of-the-practice in the DoD logistics decision-making environment, reach practical limits quickly, so contemporary decision support tools provide approximate solutions at best. This lack of rigor results in operational cost inefficiencies and a need for significant human intervention during planning, tasking, and execution. Parallel computing techniques show promise to overcome these shortcomings, are rapidly evolving in the non-federal sector, and are becoming practical for application to DoD problems; yet in the mobility community there has been little exploration of the feasibility and utility of such techniques.

The most dynamic element of the DoD logistics domain involves aircraft, whose missions are subject to enormous uncertainty based on worldwide tensions and commitments. Accommodating this uncertainty into operational tasking is a largely manual process based on subject

matter expertise. To complicate the picture, DoD has a fleet of over 1300 aircraft (AMC, 2009) to manage and operate globally, frequently with little notice, to destinations lacking infrastructures but populated with hostile forces. It is the world's largest "airline," but one that operates in a uniquely challenging environment. Nonetheless, significant improvement could be attained by leveraging methodologies used by commercial air carriers, coupled with the massive capability available from parallel computing.

The onus for operational management of DoD airlift rests with the 618th Tanker-Airlift Control Center (TACC), an element of the Airlift Mobility Command (AMC)(TACC, 2011). Despite the uncertainty of future delivery requirements, the TACC must provide airlift wings with periodic operations plans (WOPs) that reconcile the stability of sound planning with the inevitability of disruption. Hence, these plans must be robust; they must be sufficiently flexible to address myriad random changes in both cost- and mission-effective manners. They must also be sufficiently detailed to account for the unique details of military logistics.

The WOP cycle is nominally one month, but is effectively much shorter due to changing conditions. WOPs specify what mission types will be flown by which aircraft types using which airlift wings. These plans primarily allocate aircraft to missions; details such as aircraft assignment, crew pairing, and mission routing are addressed subsequently. The WOP must incorporate sufficient buffer between aircraft allocated and aircraft available to address disruptions such as changes in cargo requirements, weather delays, aircraft breakdowns, and myriad other factors. When allocations are insufficient to meet these anomalies, the TACC must choose between delaying deliveries or potentially chartering civilian aircraft to fill the void.

This paper describes an analytical approach to replicate the TACC's planning cycle. The associated model is a parallel stochastic mixed integer program that allocates aircraft to three of the primary mission types flown by AMC: 1) channel missions, which are regularly scheduled missions between the US and overseas locations, 2) contingency missions, which are irregularly

scheduled missions that deliver cargo to an international "hot spot," and 3) special assignment airlift missions (SAAMs), in which military aircraft are chartered by organizations for a specific purpose. Aircraft are allocated by airlift wing, aircraft type, and mission type; daily allocation fidelity is a subject of ongoing research. The time horizon considered is 30 days, consistent with the WOP cycle.

We model the allocation process as a two-stage stochastic mixed integer program with complete recourse. Aircraft are allocated in the first stage; the second stage subproblems conduct more detailed planning with probability-weighted mission and cargo demands over dozens of scenarios. Recourse is assured by allowing missions to be not flown.

We employ parallel processing in the second stage using Benders' decomposition; many processors resolve dozens of subproblems for a given first stage allocation (Benders, 1962). In turn, dual information is returned to the first stage for reallocation until a convergence threshold is met. Feasibility is maintained due to the recourse formulation, and integrality is mandated in the first stage.

There is a wealth of literature addressing optimal strategies for cargo and passenger movement in the commercial sector; see Barnhart et al. (2003) for an overview. Baker et al. (2002) discuss a military application of airlift optimization; Oak Ridge National Labs (2011) describe a model that is currently used to manage US military airlift. Morton et al. (2009) employ stochastic programming for military sealift management. Parallel processing for stochastic programming was proposed by Dantzig and Glynn (1990), and has since been employed by Gondzio and Kouwenberg (2001) and Avriel et al. (1989), among others.

2 Stochastic Program Formulation

The intended output of this model is the vector of aircraft allocations produced in stage 1 of a stochastic mixed integer program. Realizations of future outcomes in stage 2 influence

this output. The second stage realizations consist of aircraft capacity constraints, aircraft flow constraints, cargo demand satisfaction constraints, and limits on usage of allocated aircraft. Currently, only cargo and aircraft demand fluctuations are modeled as random variables; incorporating random mission durations is ongoing. The objective function seeks to minimize the probability-weighted costs of aircraft operation and leasing, plus the cost of late and undelivered cargo.

2.1 Mission Types

AMC missions vary widely. Excluding training, which are largely external to TACC allocations, this research considers the three largest mission types. Any of these may require aerial refueling and transshipment of cargo, which are also incorporated.

- **Channel**

  These missions originate and terminate at an aircraft's home base, making several enroute stops to pick up and drop off cargo and passengers at major aerial ports. The routes are regularly scheduled based on forecast cargo demand patterns. A realization consists of random cargo and passenger draws for each day along each route (outbound and inbound). The routes are fixed, but the frequency and aircraft used may be varied for the purposes of this model. Non-delivery penalties occur if channel cargo is undelivered for more than seven days.

- **Contingency**

  These are similar to channel missions in that they require cargo and passengers to be carried between specified locations, generally from the US to an overseas region. However, they differ from channel missions due to high demand variance and localized destinations. A realization consists of a Bernoulli draw for each type of contingency, each of which

requires between a few and (potentially) hundreds of sorties. Late and non-delivery penalties are similar to those used for channel missions.

- **Special Assignment Airlift Missions (SAAM)**

  These aircraft are chartered for a specific time frame by a military organization for their exclusive use. A realization consists of daily aircraft required, aircraft type, mission routing, and mission duration. Demand is aircraft, not cargo centric, and is of moderate variance. There are opportunities for SAAM missions to carry channel cargo while positioning to or from the customer's specified location. The unmet missions are penalized above the short-term rental rate of the associated aircraft.

2.2 Indices and Index Sets

- $i \in \mathcal{I}$: Cargo demand identifiers. This index is overloaded to accommodate the precise definitions of the different mission types. For channel missions i represents two-way *origin-destination* (OD) pairs with daily cargo delivery demands. For contingency missions i represents a specific cargo delivery demand between OD pairs at a specific time. For SAAM missions, i serves as an index of aircraft charter demands.

- $j \in \mathcal{J}$: Aircraft (jet) types, civilian and military. Aircraft types differ in their infrastructure requirements, capacity, operating cost, etc.

- $k \in \mathcal{K}$: Cargo types. We generalize cargo into four types, consistent with AMC nomenclature. Bulk cargo consists of small items consolidated into aircraft pallets that fit on all cargo aircraft. Oversize cargo consists of items such as rolling stock that fit on some civilian and most military aircraft. Outsize cargo consists of items such as tanks or helicopters, which fit only on wide-body military aircraft. Passengers may be carried on all aircraft equipped with seating. This set is overloaded with and additional index 'sam'

that denotes a SAAM demand, which is independent of cargo type. Unless explicitly indicated, 'sam' is excluded from summation and domain expressions.

- $l \in \mathcal{L}$: Locations. These may be aircraft home bases, cargo origin or destination bases, enroute bases, or aerial locations used for inflight refueling.

- $m \in \mathcal{M}$: Mission types. Channel, contingency, and SAAM missions are indexed with 'ch,' 'co,' and 'sa,' respectively. Each mission type is subject to uncertainty, realized either by cargo demand (channel, contingency) or aircraft needed (SAAM).

- $r \in \mathcal{R}$: Routes. Each cargo route begins at an aircraft home base and transits a cargo origin and destination or transshipment location (both inbound and outbound). It may also transit one or more enroute locations for refueling. For example, a route may be of the type: home base — origin — enroute — destination — origin — home base, or home base — origin — air refueling — destination — home base. In some cases the home base and origin are co-located. Air refueling routes begin at an aircraft home base and visit an air refueling location to deliver fuel to another aircraft.

- $t \in \mathcal{T}$: Time periods. We discretize time into days.

- $s \in \mathcal{S}$: Stage 2 scenarios.

- $J_{mil}, J_{civ} \subset \mathcal{J}$: Subsets of military and civilian aircraft, respectively.

- $G_1, G_2 \subset \mathcal{J}$: Subsets of aircraft that are allocated in stage 1 and stage 2, respectively. Note that only civilian aircraft (short-notice rentals) are elements of G_2.

- $JT \subset \mathcal{J}$: Subset of aircraft that can serve as tankers or airlifters.

- $JS_i \subset \mathcal{J}$: Subset of aircraft j requested by SAAM i.

- $K_j \subset \mathcal{K}$: Subset of cargo types that can be carried by aircraft type j.

- $LA \subset \mathcal{L}$: Subset of locations that are air refueling locations.
- $O_r \subset \mathcal{L}$: Home base location l (origin) of route r (1 element per subset).
- $S_i \subset \mathcal{R}$: Subset of routes serving i.
- $S1_i \subset \mathcal{R}$: Subset of routes which constitute the first portion of a transshipped delivery for i. These routes are flown by civilian aircraft.
- $S2_i \subset \mathcal{R}$: Subset of routes which constitute the second portion of a transshipped delivery for i. These routes are flown by military aircraft.
- $\mathcal{TS} \subset \mathcal{I}$: The subset of $\mathcal{I}$ that requires transshipment, which can occur when civilian aircraft cannot be flown into regions of conflict.
- $Q1_l \subset \mathcal{R}$: Subset of airlift routes transiting air refueling location l.
- $Q2_l \subset \mathcal{R}$: Subset of tanker routes servicing air refueling location l.

2.3 Data

- $A_{j,t}$: Number of hours an aircraft j is available for flying in period t.
- $C_{r,j}$: Capacity of aircraft j when flying route r. $C_{r,j}$ will depend on the distance of each leg, fuel requirements and other factors.
- $C^k_{r,j}$: Capacity of aircraft j for carrying cargo type k when flying route r. This accommodates different space requirements of bulk cargo, oversize cargo, outsize cargo, and passengers.
- $\tilde{C}_{r,j}$: Surplus capacity on aircraft j from a SAAM mission to carry channel cargo in time period t.

- $D^m_{i,k,t}(s)$: Demand of cargo type k for requirement i of mission m in period t as realized in scenario s. The demand is modeled as a random variable. The units for the demand are tons for channel and contingency missions, and aircraft for SAAM missions.

- $E_{r,j}$: Operational expenses incurred flying aircraft j on route r.

- $H_{i,k,t}$: Maximum unmet channel cargo demand for requirement i, cargo type k at time t. Unmet demand greater than H is penalized at a higher rate.

- $Opt_s(\mathbf{y})$: Optimal stage 2 objective function value for scenario s, given allocation vector $\mathbf{y}$; $\mathbf{y}^*$ denotes an incumbent solution.

- $P^1_{k,m}$ ($P^2_{k,m}$): Penalty per unit weight for late (very late) delivery of a cargo of type k for mission m.

- R_j: Per period cost of (civilian) aircraft j if leased well in advance. Used in stage 1.

- $\hat{R}_j$: Per period rental cost of (civilian) aircraft j if rented on short notice. Used in stage 2.

- RDD_i: The required delivery date for contingency requirement i.

- $T^m_{r,j,t,t'}(s)$: Hours required in period t to complete route r with aircraft j when launched in period t' for mission m as realized in scenario s.

- $T'^m_{r,j}$: Flying hours of aircraft j to complete route r while flying mission type m.

- $TR_{r,j,l}$: Tankers required (baselined by KC10 equivalents, which is a large tanker) by aircraft j flying route r at air refueling location l

- μ_j: Permissible flying time utilization of aircraft of type j.

- $\Delta_r^{i,j}$: Time periods needed to reach the destination or transshipment base for requirement i on route r using aircraft type j (channel and contingency). For SAAMs it denotes the time periods required to reach the initial SAAM mission location.

- Δ_r^j : time periods needed to complete route r using aircraft type j. As used in the air refueling constraint, it denotes the number of lag periods between an airlift mission launch on route r by aircraft j, and the air refueling event.

- $Y_{j,l}$: Number of aircraft of type j available for allocation at location l.

- π_s: Probability of scenario s.

2.4 Variables

All variables are continuous, non-negative, and used in stage 2 unless otherwise noted.

- $u^1_{i,k,m,t}, u^2_{i,k,m,t}$: Unmet demand of cargo type k for requirement i of mission m in period t. The penalty for unmet demand grows linearly when cargo has not been delivered for $t_{i,m}$ days. Beyond that the penalty is increased. To model this for channel missions, we divide the unmet demand into two parts. $u^1_{i,k,ch,t}$ is the unmet demand that is less than $t_{i,ch}$ days old and $u^2_{i,k,ch,t}$ is the unmet demand that is more than $t_{i,ch}$ days old. We define the threshold H as follows:

  $$H_{i,k,t} = \sum_{t-t_{i,ch}}^{t} D^{ch}_{i,k,t}(s)$$

  Late cargo for contingency missions is defined using the parameter RDD_i. Cargo delivered on or before the RDD_i is unpenalized. Cargo delivered one to $t_{i,co}$ days late is penalized at rate $P^1_{k,co}$. Thereafter, cargo is penalized at rate $P^2_{k,m}$. Late SAAM missions are disallowed: SAAMs are either flown on the requested day or a penalty is imposed.

- $x^m_{r,j,t}$: Number of type j aircraft launched on route r in time t supporting mission m.

- $y_{j,l,m}$ (general integer, stage 1 variable): Number of stage 1 aircraft j allocated to (base) location l for mission m. This is the principal output of the program. Stage 1 allocations include all military aircraft and civilian aircraft on advanced (more than one month prior) lease. We use $\mathbf{y}$ to denote the allocation vector. Section 4 describes ongoing efforts to incorporate a t index on this variable.

- $\hat{y}_{j,l,m,t}$: Number of stage 2 rented aircraft (short-notice, high-cost) j allocated at (base) location l for mission m at time t. This is also a principal output of the program, and is approximated with a linear variable.

- $z^m_{i,j,k,r,t}$: Tons of of type k cargo for requirement i transported on aircraft j using route r in period t.

- θ_s (stage 1 variable): Stage 2 cut support for scenario s.

- $v^s_{\cdot}$: Optimal dual variables for scenario 's' obtained by solving stage 2 of the model. The (.) subscript denotes the constraint block number and variable domain.

2.5 Stage One Formulation

The first stage determines the values of $y_{j,l,m}$, the allocation of aircraft by type, location, and mission. The problem is decomposed to facilitate parallel processing of the stochastic realizations, necessitating stage 2 cuts in the formulation. Stage 2 cuts are added successively until a convergence tolerance is met.

$$\min \quad \sum_{j \in J_{Civ} \cap G_1, l, m} R_j y_{j,l,m} + \sum_s \pi_s \theta_s$$

s.t.

$$\textit{Feasible Allocation}: \quad \sum_m y_{j,l,m} \leq Y_{j,l} \qquad \forall j \in G_1, l$$

$$\textit{Stage 2 Cuts}: \quad \theta_s \geq \; Opt_s(\mathbf{y}^*) + \sum_{j \in G_1, l, m} (A_{j,t} v^s_{7,j,l,m} + \mu_j v^s_{9,j,l,m})(y_{j,l,m} - y^*_{j,l,m,t})$$
$$+ \sum_{j \in G_1, l, m} A_{j,t}(v^s_{7,j,l,t} + \mu_j v^s_{9,j,l}) \qquad \forall s$$

$$y_{j,l,m} \in \{0, 1, 2, ...\} \qquad \forall j, l, m$$

The objective function seeks to minimize the cost of civilian aircraft allocation, plus the probability-weighted sum of stage 2 cuts. Military aircraft are excluded from the first term because they do not incur allocation costs. The feasible allocation constraints limit the total allocated aircraft to the number available at each base. The stage 2 cuts represent the dual costs from the mission and flight time constraints as affected by the stage 1 y variables. The values of v are fixed in the stage 1 problem.

2.6 Stage Two Formulation

The second stage models the execution of channel, contingency, and SAAM missions for a large number of stochastic realizations. These are approximated by linear programs. The values of $y_{j,l,m,t}$ generated from stage 1 are used as inputs to this program.

$$Opt_s(\mathbf{y}) = \min \sum_{j \in J_{civ}, l, m, t} \hat{R}_j \hat{y}_{j,l,m,t} + \sum_{i,k,t} P^1_{k,ch} u^1_{i,k,ch,t} + \sum_{i,k,t=RDD_i}^{RDD_i + t_{i,co}} P^1_{k,co} u^1_{i,co,t} + \sum_{i,k,m,t} P^2_{k,m} u^2_{i,k,m,t}$$
$$+ \sum_{r,j \in J_{mil}, m, t} E_{r,j} x^m_{r,j,t}$$

s.t.

Channel

$$\textit{Demand}\ (1a): \quad -(u^1_{i,k,ch,t-1} + u^2_{i,k,ch,t-1}) + \sum_{r\in S_i}\sum_{j} z^{ch}_{i,j,k,r,t} + u^1_{i,k,ch,t} + u^2_{i,k,ch,t} = D^{ch}_{i,k,t}(s) \quad \forall i \notin \mathcal{TS}, k, t$$

$$\textit{TS Demand}\ (2a): \quad -(u^1_{i,k,ch,t-1} + u^2_{i,k,ch,t-1}) + \sum_{r\in S_i}\sum_{j\in J_{Mil}} z^{ch}_{i,j,k,r,t} + \sum_{r\in S1_i}\sum_{j\in J_{Civ}} z^{ch}_{i,j,k,r,t} + u^1_{i,k,ch,t} + u^2_{i,k,ch,t} = D^{ch}_{i,k,t}(s) \quad \forall i \in \mathcal{TS}, k, t$$

$$\textit{Transshipment}\ (3a): \quad \sum_{r\in S1_i}\sum_{j\in J_{Civ}} z^{ch}_{i,j,k,r,(t-\Delta^{i,j}_r)} - \sum_{r\in S2_i}\sum_{j\in J_{Mil}} z^{ch}_{i,j,k,r,t} = 0 \qquad \forall i \in \mathcal{TS}, k, t$$

$$\textit{Aggregate capacity}\ (4a): \quad \sum_{k\in K_j} z^{ch}_{i,j,k,r,(t+\Delta^{i,j}_r)} - C_{r,j}x^{ch}_{r,j,t} - \tilde{C}_{r,j}x^{sa}_{r,j,t} \leq 0 \quad \forall i, j, r, t$$

$$\textit{Specific capacity}\ (5a): \quad z^{ch}_{i,j,k,r,(t+\Delta^{i,j}_r)} - C^k_{r,j}x^{ch}_{r,j,t} \leq 0 \quad \forall i, j, k, r, t$$

$$\textit{Price Break}\ (6): \quad 0 \leq u^1_{i,k,1,t-1} \leq H_{i,k,t} \qquad \forall i, k, t$$

Contingency

$$\textit{Demand}\,(1b): \quad -(u^1_{i,k,co,t-1}+u^2_{i,k,co,t-1})+\sum_{r\in S_i}\sum_{j} z^{co}_{i,j,k,r,t} +u^1_{i,k,co,t}+u^2_{i,k,co,t} = D^{co}_{i,k,t}(s) \quad \forall i\notin \mathcal{TS},k,t$$

$$\textit{TS Demand}\,(2b): \quad -(u^1_{i,k,co,t-1}+u^2_{i,k,co,t-1})+\sum_{r\in S_i}\sum_{j\in J_{Mil}} z^{co}_{i,j,k,r,t} +\sum_{r\in S1_i}\sum_{j\in J_{Civ}} z^{co}_{i,j,k,r,t}+u^1_{i,k,co,t}+u^2_{i,k,co,t} = D^{co}_{i,k,t}(s) \quad \forall i\in \mathcal{TS},k,t$$

$$\textit{Transshipment}\,(3b): \quad \sum_{r\in S1_i}\sum_{j\in J_{Civ}} z^{co}_{i,j,k,r,(t-\Delta^{i,j}_r)}-\sum_{r\in S2_i}\sum_{j\in J_{Mil}} z^{co}_{i,j,k,r,t} = 0 \quad \forall i\in \mathcal{TS},k,t$$

$$\textit{Aggregate capacity}\,(4b): \quad \sum_{k\in K_j}\sum_{i:r\in S_i} z^{co}_{i,j,k,r,(t+\Delta^{i,j}_r)}-C_{r,j}x^{co}_{r,j,t} \le 0 \quad \forall j,r,t$$

$$\textit{Specific capacity}\,(5b): \quad \sum_{i:r\in S_i} z^{co}_{i,j,k,r,(t+\Delta^{i,j}_r)}-C^k_{r,j}x^{co}_{r,j,t} \le 0 \quad \forall j,k,r,t$$

SAAM

$$\textit{Demand}\,(1c): \quad \sum_{r\in S_i} x^{sa}_{r,j,(t-\Delta^{i,j}_r)}+u^2_{i,sam,sa,t} = D^{sa}_{i,j,t}(s) \quad \forall i,j\in JS_i,t$$

Aircraft usage

Mission times $(7, 8)$:

$$\sum_{t'} \sum_{r:l \in O_r} T^m_{j,r,t,t'}(s) x^m_{r,j,t'} \leq A_{j,t} y^*_{j,l,m} \quad \forall j \in G_1, l, m, t$$

$$\sum_{t'} \sum_{r:l \in O_r} T^m_{j,r,t,t'}(s) x^m_{r,j,t'} - A_{j,t} \hat{y}_{j,l,m,t} \leq 0 \quad \forall j \in G_2, l, m, t$$

Flying times (9) :

$$\sum_{t} \sum_{r:l \in O_r} T'^m_{r,j} x^m_{r,j,t} \leq \sum_t \mu_j y^*_{j,l,m} \quad \forall j \in G_1, l, m$$

Air Refueling(10) :

$$\sum_{r \in Q1_l} \sum_{j} TR_{r,j,l} x^m_{r,j,(t-\Delta^j_r)} = \sum_{r \in Q2_l} \sum_{j \in JT} x^m_{r,j,t} \quad \forall l \in LA, t, m$$

$$u^1_{i,k,m,t},\ u^2_{i,k,m,t} \geq 0 \quad \forall i, k, m, t; \quad x^m_{r,j,t} \geq 0 \quad \forall r, j, m, t;$$

$$z^m_{i,j,k,r,t} \geq 0 \quad \forall i, j, k, m, r, t; \quad \hat{y}_{j,l,m,t} \geq 0 \quad \forall j, l, m, t$$

The stage 2 objective function minimizes the sum of short-term rental costs, the cost of late channel and contingency cargo, the cost of very late or undelivered cargo, and the cost of aircraft operations.

The demand constraints are represented as cargo inventories for each requirement across time periods; the difference between previous and current inventories, adjusted for deliveries, equals demand. There are separate constraints for cargo that must be directly delivered, and cargo that may be either directly delivered or transshipped. The transshipment constraints ensure cargo delivered in the first leg of a transshipment equals cargo delivered in the second leg. The aggregate capacity constraints limit cargo deliveries by the cumulative capacity of the aircraft assigned for delivery. The specific capacity constraints are similar, but constrain the

individual cargo types separately to account for their unique loading characteristics. The price break constraint enforces limits on late cargo.

Mission time constraints ensure that no more aircraft are away from home base than have been allocated. Similarly, the flying time constraint limits aircraft flight hours throughout the model time horizon to their historical maximum. Finally the air refueling constraint ensures that tankers are flown in support of aircraft sent along routes that transit air refueling locations.

2.7 Solution Technique

A typical *deterministic* problem instance consists of approximately $44,000$ rows, $61,000$ columns, and $225,000$ non-zeros, of which $1,300$ are general integers. Problems are generated and written to MPS format using GAMS (Brooke et al., 1992), then decomposed and augmented with up to 120 stochastic scenarios. This increases problem size by almost two orders of magnitude. Once parallelized, the first and second stage problems are solved using the Gurobi commercial solver (Gurobi Corp., 2011).

To parallelize the second stage, we create a master thread which forks multiple workers, each of which solves stage 2 problems. Each slave solves a different scenario and sends back multiple cuts to the master—see Birge and Louveaux (1988) for a detailed development of the multi-cut method. The master adds these cuts to the stage 1 model, solves it optimally to determine the values of $\mathbf{y}$ which is communicated to the workers. This process continue until the master converges to a unique solution.

3 Implementation and Results

The data used to implement this model have a variety of pedigrees. Aircraft characteristics, costing, basing, and routing are based on historical patterns and publically available informa-

tion. Channel and SAAM demands are historically based; contingency demands and locations are derived from a commonly used analytical data set. TACC's tradeoff between leasing additional aircraft and delaying cargo is subject to a variety of conditions, but we generalize as follows: the maximum penalty for a planeload of non-delivered cargo is 10 percent higher than the cost of the most appropriate short-notice leased aircraft, multiplied by the duration of a typical mission length for that aircraft.

As with most parallel applications sound processor management proves key to performance speedup of this model. We facilitate processor management with the use of Charm++ (Univ. of Il., 2011), which gives the user considerable insight into processor load balancing and other issues. The following methods help improve processor utilization:

- Dedicated processor for workflow management. Charm++ assigns parallelized work with one of the available processors. Additional work requests (assignment of additional scenarios) can be delayed if this "communications" processor is busy with its own scenario solve. Consequently, we assign a separate processor to conduct these operations without burdening it with mathematical computation.

- Scenario buffering. Simultaneous assignment of multiple scenarios to each processor reduces communications delays, thus increasing processing time. One must be careful not to over-buffer using this technique, since it can result in waiting on an over-subscribed processor's work completion near the end of the run.

- Selective use of advanced bases. Scenarios with similar coefficients (demand patterns) can be grouped and assigned to specific processors. This facilitates use of advanced bases, which can markedly increase similar scenarios' solution time.

- Staggered stage 1 recomputations. A Benders' "round" is complete when all of the subproblem scenario cuts have returned; this normally initiates a new stage 1 solve with new

allocation vectors passed to stage 2. Premature initiation of stage 1 forgoes information provided by the remaining cuts, but can markedly reduce idle time associated with many processors waiting on a few processors' remaining work.

- Speculative rounds. This is a complement to staggered recomputations; it is used when stage 2 processors are idle. While waiting for a new stage 1 allocation, these processors solve stage 2 problems using previously computed but sub-optimal stage 1 allocations. The resulting cuts are valid and may prove useful as the decomposition converges.

Incorporating these techniques yields considerable reduction in solution time; Figure 1 depicts a representative instance of that decrease. The techniques used vary somewhat based on the number of cores, for example, the two core run does not dedicate a separate processor for communications.

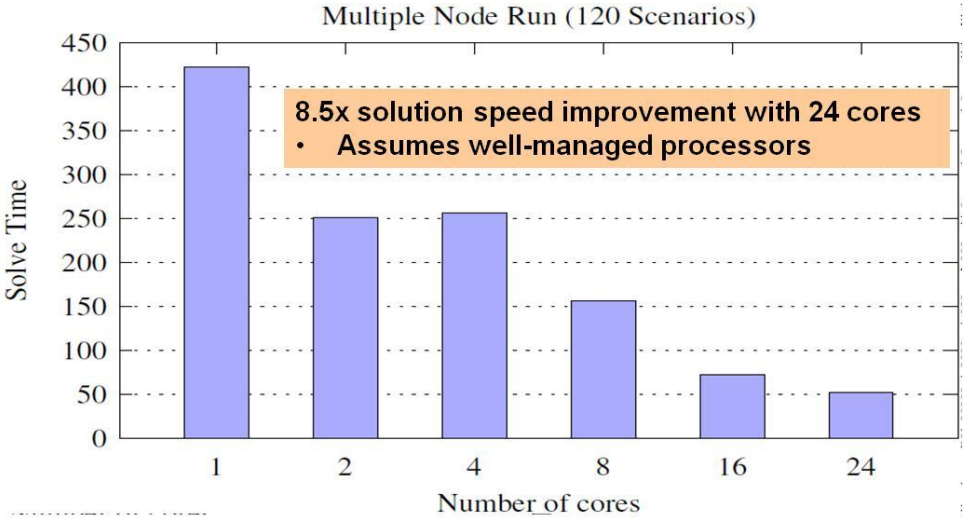

Figure 1: Solution times vary markedly as a function of parallel processing cores. In this example, a single processor solution requires 425 seconds, where a 24-core solution requires just over 50 seconds.

In addition to the speedup of parallel processing, stochastic programming provides a "hedge" against future uncertainty by reducing the costs of mission perturbations. Figure 2 illustrates a simplified 6-scenario example. It shows that deterministic optimization of average demands

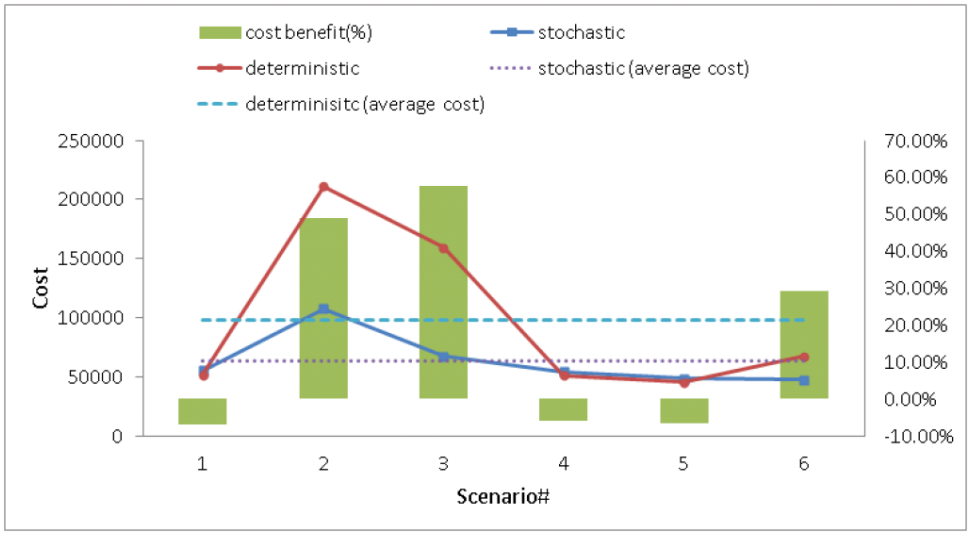

Figure 2: Hedging against future uncertainty can play an significant cost-saving role. In this simplified 6-scenario example, red circles correspond to each scenario's cost using deterministic optimization of expected demands; the horizontal average is depicted as a dashed line. In contrast, the blue squares correspond to stochastic optimization costs for each scenario (with the dashed-line average). Histobars depict the percentage cost savings of stochastic optimization. Stochastic optimization yields only slightly more costly solutions when actual demands are low (scenarios 1, 4, and 5), but are much less costly when demands are elevated (scenarios 2, 3, and 6). The average cost savings in this example is approximately 10 percent.

yields low cost solutions when actual cargo and aircraft demands are small or average, but perform very badly when actual demands are elevated. In contrast, stochastic modeling incorporates hedging against uncertainty and yields much improved solutions when actual demands are elevated. The cost of hedging is approximately 5 percent, but reduces overall costs by as much as 57 percent and an average of 10 percent. Additionally, the cost variance is reduced by 66 percent. This finding supports the planners' goal of finding not a point solution at an unstable minimum, but a stable "trough" on the solution surface that balances cost savings with re-planning needs, while minimizing disruption to the existing plan. When implemented, this methodology could realize a significant reduction in cost, or a significant increase in timely mission accomplishment.

While the example in figure 2 was a simplified scenario used to demonstrate the potentially large benefits that are obtainable from using stochastic programming, the example in figure 3 is

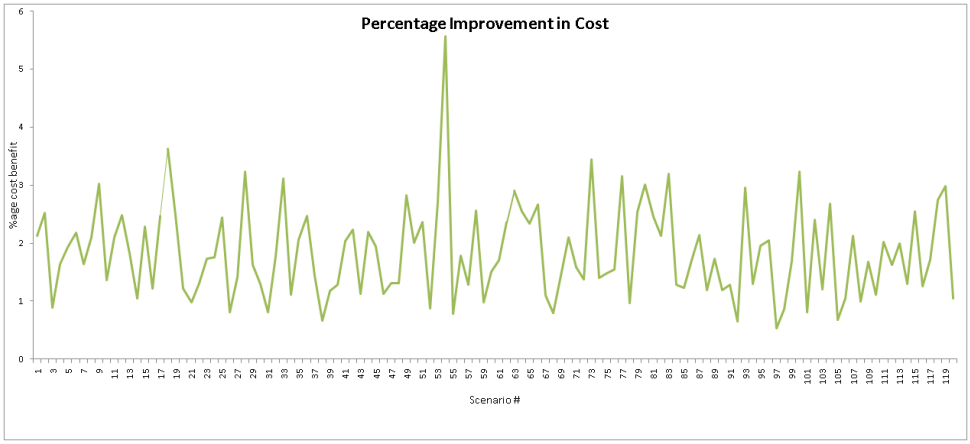

Figure 3: For randomly generated scenarios, hedging against future uncertainty usually leads to cost saving. In this 120-scenario example, the stochastic programming solution has a lower cost in all scenarios. The average cost savings in this example are 3 percent

more typical. Here we randomly generate 120 scenarios which are used by to obtain the stochastic programming solution. The figure shows the percent improvement of the stochastic solution over the corresponding deterministic solution obtained using average values for demand. Notice that in all cases the stochastic solution does better than the deterministic solution albeit the differences are not as significant as in figure 2. Because the scenarios are randomly generated, the average scenario is not one of the chosen scenarios. Moreover, in random scenarios, the chance that all demands will be higher or lower than their expected demand is very small. Accordingly, the benefits obtained from stochastic programming are somewhat smaller - averaging approximately 3 percent and with a maximum of 6 percent. The stochastic program performs better in al cases because it provides higher allocations to hedge against the chace that there will be high demands especially for those demands for which there is no recourse but to leave the demand unsatisfied. Thus, it incurs higher costs of short term leases for demands that can be satisfied by commercial aircraft but ensures that missions that require military only aircraft are suitably covered.

4 Ongoing Research

There are two areas of ongoing research. The foremost involves parallelization of the first stage. Incorporating daily allocations, i.e., a time index on the y variable, requires a factor of 30 increase in the number of integer variables for a problem with a one month time horizon. Using the techniques described below we have solved problem instances with 5 time periods in less than 1 minute, and can link multiple 5-period solves with Lagrangian relaxation or other temporal decomposition. However, our goal is to minimize such linkages.

The second area of ongoing research involves incorporation of aircrews into the allocation. Crews often represent the binding constraint in current airlift operations, and increasing their numbers is unlikely in today's constrained budget environment. This issue only applies to military aircrews, since civilian aircrews are managed by the corporate carrier as part of a lease agreement. Nonetheless, military crew allocation remains a significant issue for the TACC.

4.1 Stage One Parallelization

In the implementation of sections 2 and 3, each stage 1 problem is solved as an integer program with a master-worker parallel structure as described earlier. Such parallelization successfully reduces the time taken to evaluate an allocation across all scenarios. However, this makes the serialization during the master phase (stage 1 IP) increasingly prominent as we scale to more processors. We mitigated this problem with the use of speculative rounds that harness the idle worker processors to generate additional cuts while the stage 1 IP is solved. This helps reduce the number of rounds required to accumulate a collection of cuts that would characterize the stage 2 LP and hence reduces the time to solution.

However, problems that involve allocation decisions for multiple time periods are significantly larger and more complex to solve. Such problems require many more rounds to converge, and thus, automatically build a large collection of cuts. Speculative rounds in this context are

actually counter-productive as they inflate the size of the stage 1 model, leading to even longer stage 1 solve times. Instead, large problems with multiple time periods actually require judicious management of the collection of cuts held in the stage 1 instance. We implemented a common approach to cut management, which retires cuts generated in earlier rounds that have not been binding for a period of time. Despite such measures, the stage 1 IP remains a significant serial bottleneck that impacts the scalability of the problem.

The solution of integer programs in each round has the further disadvantage that each round repeats the enumerative search inherent in a branch-and-bound strategy. To overcome this redundant and repetitive search, we have changed the solution strategy.

To address the serial bottleneck described above, the problem is initially relaxed to form a stochastic linear program (SLP) by linearizing the stage 1 integer variables. The SLP requires significantly less time than the IP form, thereby reducing the stage 1 serial bottleneck. This relaxed SLP is solved at the root vertex and then a single branch and bound enumerative search is conducted. At each vertex of this branch-and-bound tree we solve a stochastic linear program. This strategy has several advantages. First, we eliminate the repetitive enumeration inherent in the previous approach. Second, we use the fact that cuts generated in one part of the tree are valid at all vertices of the search tree. Thus at each vertex of the tree, there is a sufficiently rich set of valid cuts which can be used to establish an initial bound at that vertex.

Because we now control the branch and bound process (in contrast with the previous approach of allowing Gurobi to solve the integer programs) we have more opportunity to parallelize computation and effectively use a larger number of processors. Thus there are two available sources of parallelism: simultaneous optimization of vertices (generated by parallel branching) in the branch-and-bound tree and scenario parallelization within each vertex.

Figure 3 shows a schematic of a branch-and-bound tree. Notice that the root vertex is an SLP and can be solved using the same ideas as in section 2.7. However, stage 1 is a linear

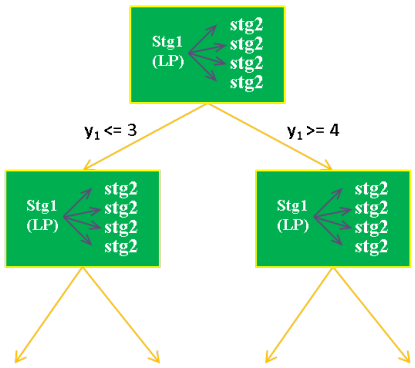

Figure 4: Two sources of parallelism apply to this research. Within each branch-and-bound vertex (box) a stochastic program is parallelized using LP relaxation in stage 1. In turn, each vertex of the resulting branch-and-bound tree can be parallelized to converge toward the stage 1 integer solution.

program and thus requires significantly less time than an integer program thereby reducing the stage 1 serial bottleneck. Once the root vertex SLP converges to a desired level of accuracy, a branch-and-bound tree is formed by branching on suitable fractional variables in the root vertex SLP. At this stage a second level of parallelism is possible. By successively branching on several variables it is possible to enlarge the tree to create a number of search vertices within it. These vertices can now be solved in parallel. While this is a relatively apparent parallelism, there is a chance that prematurely branching on variables can lead to the exploration of parts of the search tree that could otherwise have been pruned due to bounds. This requires a judicious management of the branch and bound enumeration to balance processor workload and unnecessary creation of stage 2 solver workload.

4.2 Parallel Branch and Bound Architecture

The complexity inherent in our multiple-parallelization scheme drives a need for a specialized computing architecture; Figure 4 illustrates this design. Available processors are divided into four different entities: one stage 1 load balancer (S1LB), one stage 2 manager (S2M) and rest of the processors act either in the capacity of a stage 1 solver or a stage 2 solver.

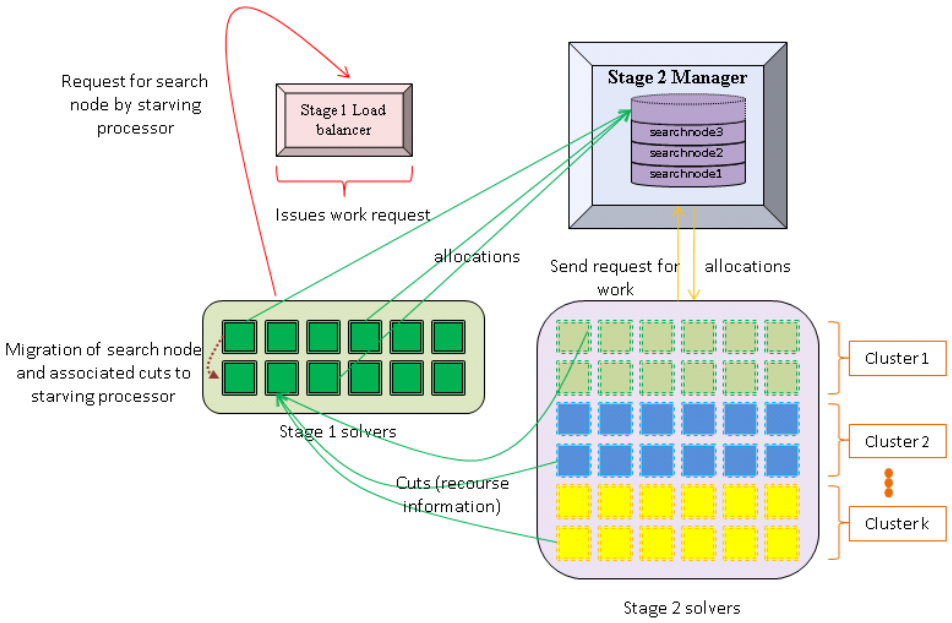

Figure 5: Multiple levels of parallelization requires a specialized architecture. Each stage has a balancer to keep processors busy, and each has a bank of processors grouped by cargo demand pattern in the case of stage 2

Each stage 1 solver is assigned a set of search vertices and is responsible for performing stage 1 optimizations when stage 2 cuts become available for the specific vertex. The S1LB coordinates with the stage 1 solvers to provide work to idle stage 1 solvers, and maintains a priority queue to prefer high priority vertices for optimization. The S1LB also tracks the load (the number of queued vertices) at each stage 1 solver. Each stage 1 solver dynamically updates S1LB with its load status whenever there is a change in its load. Whenever the load at a stage 1 solver becomes zero, the S1LB issues a work request to the solver with the highest load, which then relinquishes half of its queued load to the starving stage 1 solver.

The stage 2 solvers process stage 2 of the stochastic optimization. Each stage 2 solver receives a scenario and an allocation from the S2M. It optimizes the stage 2 problem for the given scenario and allocation and returns the resulting cut to the appropriate stage 1 solver. To ensure that the memory resident advanced basis is beneficially used, each stage 2 solver is assigned a cluster of scenarios (clustered by cargo demand similarity).

The S2M maintains priority queues of allocations received from stage 1 solvers. When requested for work from stage 2 solvers, it fairly distributes the queued allocations by cycling over the array of stage 1 queues.

With this implementation, we are able to solve 5 time period problem instances in less than a minute. Our current efforts are aimed at scaling the approach to solve larger problem instances with up to 30 time periods and to effectively use more processors. This requires careful management of both processor allocation ad scheduling, and branch and bound tree exploration. We are investigating the use of dynamic workload balancing and adaptive search strategies for the branch and bound enumeration.

4.3 Crew Allocation

Allocating military aircrews can be viewed as an extension of aircraft allocations. Military aircraft are authorized between 1.5 and 3.0 crews per aircraft. Assuming this ratio holds for each aircraft allocated by the TACC, crew allocations are implied by aircraft allocations as represented with the first stage y variables.

The work to date includes modeling aircrews deterministically; stochastic implementation is ongoing. Crew modeling requires the following additional notation and constraints:

- LC_j: Subset of locations that serve as crew stations or rest bases for aircraft j. Rest bases exist along each route to allow crews to take a required amount of time off. Aircraft arriving at these bases are re-manned by crews that have completed a rest period.
- RCR_l: Subset of routes that transit crew station or rest base l.
- CR_j: Authorized crew-to-aircraft ratio for aircraft j (military only)
- $\lambda^1_{j,r,l}$: Periods required for aircraft j flying outbound on route r to depart crew rest location l.

- $\lambda^2_{j,r,l}$: Periods required for aircraft j flying inbound on route r to depart crew rest location l.

- $\Lambda^1_{j,r,l}$: Periods required for crew of aircraft type j headed outbound on route r to be rested for departure from crew rest location l.

- $\Lambda^2_{j,r,l}$: Periods required for crew of aircraft type j headed inbound on route r to be rested for departure from crew rest location l.

- $\Lambda'_{j,l,l'}$: Periods required to relocate a crew for aircraft type j from base l to l' (deadhead).

- $\mathcal{I}(\cdot)$: Indicator function; 1 if argument is true and 0 otherwise.

- $v^+_{j,l,t}$ ($v^-_{j,l,t}$): Elastic variable for positive (negative) deviation from crew constraint.

- $w_{j,l,t}$: Variable number of crews (workers) for aircraft j available at rest base l during time t (integer variable relaxed to linear).

- $wd_{j,l,l',t}$: Variable number of crews (workers) for aircraft j relocating (deadheading) from location l to l' in time period t (integer variable relaxed to linear). The variable is not defined for $l = l'$.

$$
\begin{aligned}
\textit{Crews}: \quad w_{j,l,t+1} = w_{j,l,t} + v^+_{j,l,t} - v^-_{j,l,t} &+ \sum_m \sum_{r \in RCR_l} x^m_{j,r,(t-\Lambda^1_{j,r,l})} \cdot \mathcal{I}(\Lambda^1_{j,r,l} \neq 0) \\
&+ \sum_m \sum_{r \in RCR_l} x^m_{j,r,(t-\Lambda^2_{j,r,l})} \cdot \mathcal{I}(\Lambda^1_{j,r,l} \neq \Lambda^2_{j,r,l}) - \sum_m \sum_{r \in RCR_l} x^m_{j,r,(t-\lambda^1_{j,r,l})} \\
&- \sum_m \sum_{r \in RCR_l} x^m_{j,r,(t-\lambda^2_{j,r,l})} \cdot \mathcal{I}\left((\lambda^1_{j,r,l} \neq \lambda^2_{j,r,l}) \wedge (\lambda^2_{j,r,l} \neq 0)\right) \\
+ \sum_{l' \in LC_j} wd_{j,l',l,(t-\Lambda'_j)} - \sum_{l' \in LC_j} wd_{j,l,l',t} &\quad \forall j \in J_{mil}, l \in LC_j, t : |t| \neq |\mathcal{T}|
\end{aligned}
$$

$$
\sum_{l \in LC_j} w_{j,l,t} = CR_j \sum_l \sum_m y_{j,l,m} \quad \forall j \in J_{mil}, t = 1
$$

Additionally, the objective function is modified to include a large penalty term on v^+ and v^-, and the stage 1 cuts are modified to include the duals for these constraints.

In this formulation, aircrews flow in a manner similar to aircraft, but they are delayed at select rest bases along the route. Upon return to home base, they are further delayed to allow post-mission rest. The constraint assures that the crews available in ensuing time periods equals: 1) the crews available in the current time period, 2) plus and minus adjustments for constraint elasticity, 3) plus any change in allocated crews at home bases between the current and former periods, 4) plus outbound crews (superscript 1) completing rest from the previous mission leg (except at home base where there are no previous legs), 5) plus inbound crews completing rest from the previous mission leg (without double counting offload location crews), 6) minus crews required to fly outbound leg missions, 7) minus crews required to fly inbound missions (without double counting offload location crews, and excepting inbound missions returning to home base where there is no subsequent mission leg), 8) plus relocated crews from other bases, and 9) minus relocating crew to other bases. The second crew constraint allows an initial crew dispersal throughout the bases.

5 Summary

This research has shown the viability of parallel stochastic programming for a DoD airlift problem. We have formally incorporated randomness into the aircraft allocation problem; current operational methods accommodate random system perturbations in an ad-hoc manner at best. Results suggest considerable cost savings or performance improvements could be realized using this technique. These improvements are facilitated by the speedup leveraged by parallel computing.

Ongoing research will extend the effort into the realm of massive parallel computing. In so doing, we will allow optimal allocations with greater time and other modeling fidelity without

the use of heuristics. This will provide a cost-savings technique for a logistics problem of critical national importance.

References

Air Mobility Command, Air Mobility Command Almanac 2009. Retrieved 12 Sep 2011, http://www.amc.af.mil/shared/media/document/AFD-090609-052.pdf.

Avriel, M., Dantzig, G., Glynn, P., "Decomposition and Parallel Processing for Large-Scale Electric Power System Planning Under Uncertainty," Proceedings of the Workshop on Resource Planning Under Uncertainty for Electric Power Systems, Jan 21-22 1989, Stanford University.

Baker, S., Morton, D., Rosenthal, R., Williams, L., "Optimizing Military Airlift," *Operations Research* 50 #4 (Jul-Aug 2002), pp. 582-602.

Barnhart, C., Belobaba, P., Odoni, A., "Applications of Operations Research in the Air Transport Industry," *Transportation Science* 37 #4 (Nov 2003), pp. 368-391.

Benders, J., "Partitioning Procedures for Solving Mixed Variables Programming Problems," *Numerische Mathematik* 4 (1962), pp. 238-252.

Birge, J., and Louveaux, F., "A Multicut Algorithm for Two-Stage Stochastic Linear Programs," *European Journal of Operations Research* 34 (1988) pp. 384-392.

Brooke, A., Kendrick, D., and Meerhaus, A., *GAMS, a User's Guide*, The Scientific Press, South San Francisco, 1992.

Dantzig, G., Glynn, P., "Parallel Processors for Planning Under Uncertainty," *Annals of Operations Research* 22 (1990) pp. 1-21.

Gondzio, J., Kouwenberg, R., "High-Performance Computing for Asset-Liability Management," *Operations Research* 49 #6 (Nov-Dec 2001), pp. 879-891.

Gurobi Optimization Corporation, Gurobi Optimizer Version 4.5. Retrieved 13 Sep 2011, http://www.gurobi.com.

Kale, L., "Some Essential Techniques for Developing Efficient Petascale Applications," *Proceedings of SciDAC* July 2008.

Morton, D, Salmeron, J. and Wood, R., 2009, "A Stochastic Program for Optimizing Military Sealift Subject to Attack," *Military Operations Research* 14 #2 (2009), pp. 19-39.

Oak Ridge National Laboratory, Consolidated Air Mobility Planning System (CAMPS): An Air Mobility Planning and Scheduling System, Research Brief, Center for Transportation Analysis, http://cta.ornl.gov/, 27 Aug 2011.

Tanker Airlift Control Center, 618th Air and Space Operations Center (TACC) fact sheet. Retrieved 28 Aug 2011, http://www.618tacc.amc.af.mil.

University of Illinois, UIUC Parallel Programming Laboratory. Retrieved 12 Sep 2011, http://charm.cs.uiuc.edu/.