Parallel allocation-mission optimization of a 128-route network

John T. Hwang [*] and Joaquim R. R. A. Martins [†]

Department of Aerospace Engineering, University of Michigan, Ann Arbor, Michigan, 48109, United States

In aircraft design, a simultaneous optimization of the airline allocation, mission profile, and the airframe-propulsion design can be advantageous because it can capture how design changes might affect the aircraft utilization at the airline level. As a first step towards this goal, previous research solved the allocation-mission optimization problem by taking a modular adjoint approach, but with only a 3-route network to simplify the problem. This paper extends the previous work to solve a 128-route allocation-mission optimization problem using a parallel computational framework. The computation of the constraint gradients can be parallelized by extending the framework to support linear algebra with multiple right-hand sides. The 128-route optimization problem contains roughly 6,000 design variables and 23,000 constraints, and it converges about 3 orders of magnitude in optimality and feasibility in approximately 8 hours on 128 processors.

I. Introduction

The growing interest in next-generation aircraft concepts motivates the development of computational tools with tighter integration between disciplines. For instance, the truss-braced wing (TBW) concept uses struts to enable thinner wings with higher aspect ratios, but coupled aerostructural analysis and optimization are required to accurately quantify the weight and drag savings. Operational changes are being explored for future aircraft as well, such as continuous descent approach and morphing wings, and these necessitate analysis of the full mission profile. Concepts such as the TBW are also being designed for low cruise Mach numbers, impacting how the aircraft may be utilized within an airline's network.

Because of the coupling between the disciplines, integrating all of these changes in a single concept would require simultaneously considering the airline allocation, mission analysis, and the design of the airframe and propulsion system. Typically, design optimization is performed at a single or a small number of representative conditions. For better accuracy, especially when considering morphing wings, a potential design candidate should be analyzed across the full mission profile. Since design decisions and the performance on the mission profiles affect which routes may be optimal to fly, it would be desirable to simultaneously consider the aircraft allocation problem as well, so that the metric is the airline-level profit or industry-level fuel burn.

This simultaneous problem, which we call *allocation-mission-design optimization*, is ambitious and large in scope. Compared to design-only optimization, this problem requires tens of thousands more evaluations of the aircraft performance model because of the number of mission points, the number of iterations required to converge the coupled mission analysis, and the number of routes in the airline network. A previous study outlined a method for making this problem tractable and focused on *allocation-mission optimization* as a starting point [1]. The proposed method would use aerostructural and propulsion surrogate models that are retrained every optimization iteration, and these surrogate models would be used for the mission analysis at much lower cost.

In that paper, an allocation-mission optimization algorithm was presented as an initial step towards the goal of allocation-mission-design optimization. That is; for a fixed airframe and engine design, the next-generation aircraft were allocated to hypothetical routes in competition with existing aircraft models, while simultaneously optimizing the mission profile on each route. The mission profiles were analyzed using a surrogate model for the aerodynamic performance, but the training points were pre-computed and did not change throughout the optimization. This simplified problem presented several challenges that are also present in the allocation-mission-design optimization problem.

The first challenge is the large number of design variables resulting from the parametrizations of the altitude profiles requiring $\mathcal{O}(10)$ design variables per route. This was addressed by using gradient-based optimization in conjunction with the adjoint method to compute derivatives, enabling the optimization problem to be solved at a cost of roughly $\mathcal{O}(10^2)$ evaluations of the model.

[*] Postdoctoral Research Fellow, AIAA Member

[†] Associate Professor, AIAA Associate Fellow

The second challenge is the code complexity due to combining several disciplines together and requiring the efficient computation of derivatives involving multiple coupled disciplines. The complexity is mitigated by integrating the airline models, mission analysis equations, and surrogate models within a computational framework that solves the coupled multidisciplinary system and centrally computes derivatives using the adjoint method. This framework, which uses the *modular analysis and unified derivatives* (MAUD) architecture [2], greatly simplified the code implementation and solution of the optimization problem. MAUD is now implemented in OpenMDAO [3], and it has been applied to the solution of allocation [1], mission [4], satellite [5], and wind turbine [6] design optimization problems.

The third challenge is the combinatorial nature of the allocation problem since a finite number of aircraft are allocated to discrete routes. This was addressed by formulating a continuous problem and applying the branch-and-bound (B&B) method to ensure integer solutions. What was unique about this particular B&B implementation is that the mixed-integer nonlinear allocation-mission optimization was initialized using the solution of the allocation-only problem as a starting point. Since the allocation-only problem is linear, its solution is the global optimum of the linear problem, so the B&B algorithm solves the mixed-integer allocation-mission optimization problem very efficiently, taking advantage of warm starts to solve each continuous optimization in much less time than that of a full optimization.

While the solutions to these 3 challenges were successful, the major limitation in the results from Hwang et al. [1] is that the problem considered a network with only 3 routes because the algorithm scaled poorly for larger networks. Realistic networks representative of airlines contain $\mathcal{O}(100)$ routes.

The objective of this paper is to extend the methods presented in Hwang et al. [1] to solve allocation-mission optimization problems on larger networks by exploiting parallel computing. The mixed-integer nature of the allocation-mission optimization problem is kept out of the scope of this work for simplicity. The justification for this is that the previous results showed that the cost of each node evaluation in the B&B algorithm is much less than a full continuous optimization thanks to warm starts. Therefore, the rationale is that embedding the parallel allocation-mission optimization with a large number of routes in the B&B algorithm would not increase the computation time by multiple orders of magnitude.

The structure of the paper is as follows. Sec. II begins with an overview of the general approach: the allocation-mission optimization problem, the adjoint-based optimization strategy, and the MAUD framework. In Sec. III, we present the equations for the allocation and mission models and in Sec. IV, we describe the parallel computational framework implementing the MAUD architecture. Finally, Sec. V presents the results for a 128-route allocation-mission optimization problem.

II. Overview

The allocation-mission optimization problem we wish to solve in this paper is presented below (NLP-a-m). There are n_{rt} routes and n_{ac} types of aircraft, with n_{nac} of them being 'next-generation' types for which we perform mission analyses. For the remaining types of aircraft, we use pre-computed fuel-burn and block-time values since these will be fixed in allocation-mission-design optimization. The altitude profile is parametrized using a B-spline curve with n_{cp} B-spline control points and n_{pt} points in the discretized mission profile. The objective function being maximized is profit and the design variables are the number of passengers per flight $\text{pax\_flt}_{i,j}$ and the number of flights per day $\text{flt\_day}_{i,j}$, for route i and aircraft type j.

In (NLP-a-m), the first two inequality constraints ensure the total number of passengers being flown on a route in a given day is no more than the demand and that the total usage for each aircraft type is allowable given the number of that type of aircraft that the airline owns. The altitude control points at the start and end of the mission profile are fixed to zero, Kreisselmeier–Steinhauser (KS) functionals [7] are used to aggregate the minimum and maximum thrust constraints across the mission profile to single values, and the slope of the altitude profile (climb angle) is limited to $35°$ to avoid unphysical altitude profiles.

$$
\begin{array}{rlr}
\text{maximize} & \text{profit} & \\
\text{with respect to} & \text{pax\_flt}_{i,j} \in [0, \text{ac\_capacity}_j] \ , & 1 \le i \le n_{rt} \ , \ 1 \le j \le n_{ac} \\
 & \text{flt\_day}_{i,j} \in [0, \infty) \ , & 1 \le i \le n_{rt} \ , \ 1 \le j \le n_{ac} \\
 & \text{altitude}_{i,j,k} \in [0, \text{max\_altitude}] \ , & 1 \le i \le n_{rt} \ , \ 1 \le j \le n_{nac} \\
 & & 1 \le k \le n_{cp} \\
\text{subject to} & \text{total\_pax}_i \le \text{demand}_i \ , & 1 \le i \le n_{rt} \\
 & \text{total\_usage}_j \le \text{num\_ac}_j \ , & 1 \le j \le n_{ac} \\
 & \text{altitude}_{i,j,1} = 0 \ , & 1 \le i \le n_{rt} \ , \ 1 \le j \le n_{nac} \\
 & \text{altitude}_{i,j,n_{cp}} = 0 \ , & 1 \le i \le n_{rt} \ , \ 1 \le j \le n_{nac} \\
 & \text{idle\_throttle} \le \text{throttle}_{i,j}(\text{KS}) \ , & 1 \le i \le n_{rt} \ , \ 1 \le j \le n_{nac} \\
 & \text{throttle}_{i,j} \le \text{max\_throttle (KS)} \ , & 1 \le i \le n_{rt} \ , \ 1 \le j \le n_{nac} \\
 & \text{min\_slope} \le \text{slope}_{i,j,k} \le \text{max\_slope} \ , & 1 \le i \le n_{rt} \ , \ 1 \le j \le n_{nac} \\
 & & 1 \le k \le n_{pt}
\end{array}
\qquad \text{(NLP-a-m)}
$$

As previously mentioned, the first challenge to address is the large number of design variables in the optimization problem. Typically, we use 20-50 control points and 100-250 discretized points for each mission profile, so the total number of design variables in even an allocation-problem with only $\mathcal{O}(10)$ route is in the hundreds. Therefore, our approach is to use gradient-based optimization with the derivatives computed using the adjoint method.

However, the mission analysis for each route involves the solution of a coupled system of equations incorporating multiple disciplines and components. The need to compute derivatives in addition to that makes the development, integration, and management of the code even more difficult and error-prone.

To manage the resulting code complexity, we use a computational framework that enables us to develop the multidisciplinary model component by component. If each component follows a specified interface, the framework provides many features such as centrally solving the nonlinear systems of equations that arise. This framework, which implements the *modular analysis and unified derivatives* (MAUD) architecture, is different from other frameworks because it uses a unique mathematical formulation.

Once the user defines a set of components that map inputs to outputs, the framework concatenates the full list of variables as a single vector (in conceptual terms). The framework implicitly formulates all the components as a single nonlinear system of equations,

$$R(u) = 0 \quad , \quad u = [x_1, \ldots, x_n, y_1, \ldots, y_m, f_1, \ldots, f_p]^T \tag{1}$$

where $x_1, \ldots, x_n$ are the input variables (design variables and parameters), $y_1, \ldots, y_m$ are the intermediate variables (state variables and coupling variables), and $f_1, \ldots, f_p$ are the output variables (objective, constraints, and other quantities of interest). Therefore, evaluating the multidisciplinary model translates to simply solving this nonlinear system. Martins and Hwang [8] showed that the various methods for computing derivatives are unified by the equation,

$$\frac{\partial R}{\partial u}\frac{\mathrm{d}u}{\mathrm{d}r} = \mathcal{I} = \frac{\partial R}{\partial u}^T \frac{\mathrm{d}u}{\mathrm{d}r}^T , \tag{2}$$

so that the task of computing derivatives in the MAUD formulation reduces to solving a common, unified linear system, whether the chain rule, the adjoint method, or any other method is desired.

In Sec. IV, we present some of the details of a parallel implementation of the MAUD architecture which yields a framework that enables us to efficiently solve the allocation-mission optimization problem in parallel with a high level of modularity and automation.

III. Allocation and mission models

In this section, we present the equations in the mission analysis and the models in the airline allocation problem.

A. Mission analysis

The mission analysis formulation used here follows the approach of Kao et al. [4] For a given route, the goal of the mission analysis is to compute the performance given the altitude and Mach number profiles. In this paper, we do not allow the Mach number profile to vary because of limitations in the aerodynamic surrogate model; only the altitude profile is optimized, using a B-spline parametrization.

In reality, some altitude profiles cannot be flown, but when a model fails to solve, this can cause problems for the optimizer. Thus, within each optimization iteration, our approach is to compute the required thrust at each point in the mission regardless of whether that much thrust is available with the given engines. The optimizer is given the responsibility of ensuring that a physically realizable mission profile will be found by giving it constraints that the thrust at each mission point is between idle and maximum.

The mission analysis discretizes the profile and solves the equilibrium equations, interrogating the surrogate model for aircraft performance as required. The full list of variables modeled in the mission analysis and their dependency graph are shown in Fig. 1. The blue variables are the input variables, evident by the fact that they do not depend on any other variable. The B-spline variables are computed as a function of the B-spline control points, indicated by the tildes, and the atmospheric properties are computed in sequence in terms of previous variables. The variables in purple capture the coupling in the mission analysis, and the outputs are vectors or scalars computed in terms of earlier variables.

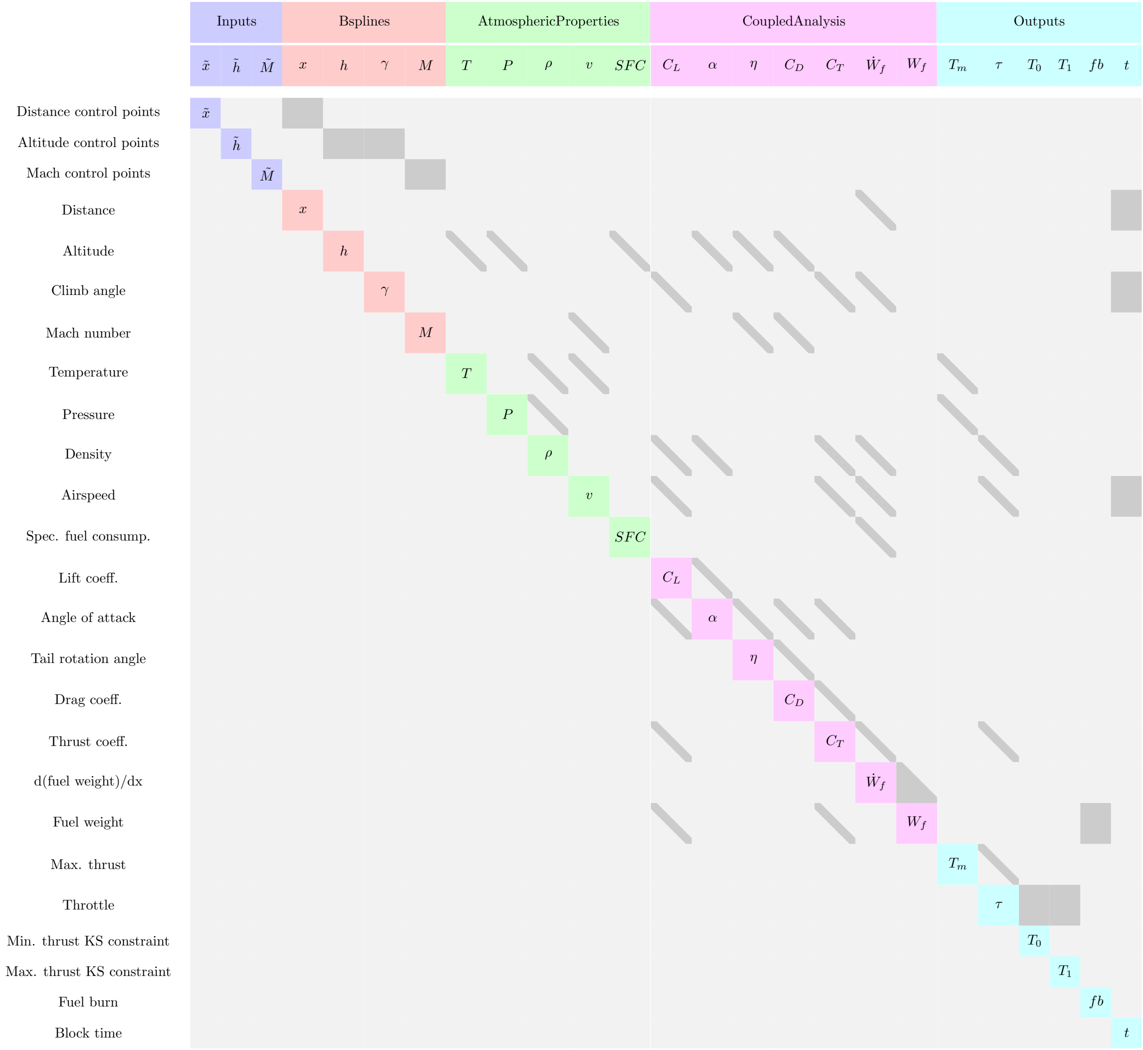


Figure 1: The variables in the mission analysis and their dependencies (shown in dark grey).

The variables in the coupled analysis group capture the following sequence of relationships. First, the lift coefficient can be explicitly computed by rearranging the vertical equilibrium equation to get

$$C_L = \frac{W \cos\gamma}{\frac{1}{2}\rho v^2 S} - C_T \sin\alpha. \tag{3}$$

Next, the angle of attack, α, is implicitly computed by solving

$$\tilde{C}_L(\alpha, \eta, M, h) - C_L = 0, \tag{4}$$

and similarly, the tail rotation angle, η, is implicitly computed by solving

$$\tilde{C}_m(\alpha, \eta, M, h) = 0. \tag{5}$$

The drag coefficient can be explicitly evaluated from

$$C_D = \tilde{C}_D(\alpha, \eta, M, h), \tag{6}$$

from which the thrust coefficient can be computed by rearranging the horizontal equilibrium equation to get

$$C_T = \frac{C_D}{\cos\alpha} + \frac{W \sin\gamma}{\frac{1}{2}\rho v^2 S \cos\alpha}. \tag{7}$$

Finally, the fuel weight ODE is given by

$$\dot{W}_f = \frac{\{SFC\} C_T \frac{1}{2}\rho v^2 S}{v \cos\gamma}, \tag{8}$$

which is integrated to get the fuel weight at each point in the mission. Since the fuel weight contributes to the total aircraft weight which affects the lift coefficient, this completes the feedback loop.

The training points for the aerodynamic surrogate model are generated using TriPan, a 3-D panel code. For this paper, simplified engine models are used; the thrust-specified fuel consumption is modeled to decrease linearly with altitude and maximum thrust is modeled as

$$T_m = T_{m,SL} \frac{P}{P_{SL}} \sqrt{\frac{T_{SL}}{T}}, \tag{9}$$

where SL stands for sea level.

B. Allocation model

The allocation problem formulation we use has two sets of design variables; the number of flights per day and the number of passengers per flight, for a given type of aircraft on a given route [9]. To simplify the problem, we ignore all routing and scheduling considerations and we make the assumption that all flights in a day for a given route and aircraft type have the same number of passengers.

Profit is modeled as

$$\text{profit} = \sum_{i}^{n_{rt}} \sum_{j}^{n_{ac}} \left[\text{price\_pax}_{i,j} \cdot \text{pax\_flt}_{i,j} \cdot \text{flt\_day}_{i,j} \right] \tag{10}$$

$$- \sum_{i}^{n_{rt}} \sum_{j}^{n_{ac}} \left[\left(\text{cost\_flt}_{i,j} + \text{cost\_fuel} \cdot \text{fuel\_flt}_{i,j} \right) \cdot \text{flt\_day}_{i,j} \right], \tag{11}$$

where price_pax$_{i,j}$ is the ticket price per flight, cost_flt$_{i,j}$ is the total cost of operating a flight minus fuel, cost_fuel is the cost per unit fuel, and fuel_flt is the total fuel burn on a flight.

The allocation problem contributes two inequality constraints to the allocation-mission optimization. The first ensures that the total number of people that fly on a given route on a given day is less than the total demand for that route, and it is given by

$$\text{total\_pax}_i = \sum_{j}^{n_{ac}} \left[\text{pax\_flt}_{i,j} \cdot \text{flt\_dat}_{i,j} \right] \leq \text{demand}_i \ , \quad 1 \leq i \leq n_{rt}. \tag{12}$$

The second inequality constraint takes into consideration how many aircraft of a given type are actually owned by the airline, and it is given by

$$\text{total\_usage}_j = \sum_{i}^{n_{rt}} \left[\text{flt\_day}_{i,j} \cdot \left(\text{time\_flt}_{i,j}(1 + \text{maint}_j) + \text{turn\_flt} \right) \right] \leq 12\text{hr} \cdot \text{num\_ac}_j \ , \quad 1 \leq j \leq n_{ac}, \tag{13}$$

where time_flt$_{i,j}$ is the block time for a flight, maint$_j$ is the maintenance time required as a multiple of block time, turn_flt is the turnaround time between flights, and num_ac$_j$ is the number of aircraft available for a given type.

IV. Parallel computational framework

The parallel computational framework implements the MAUD architecture with the additional requirements of parallel data handling and parallel solvers, leveraging the Portable, Extensible Toolkit for Scientific Computation (PETSc) [10]. One of the motivations is to exploit the opportunity for more modularity and automation in the parallel setting by centrally handling the logistics of assigning a route to each processor. The primary objective is to keep memory and computational costs manageable so that the allocation-mission optimization algorithm scales efficiently to large network sizes.

A key feature of the MAUD architecture is that it hierarchically decomposes the problem for managing the parallel data passing and for its nonlinear and linear solvers. It was discussed earlier that the flat list of variables in Fig. 1 are grouped into inputs, B-splines, atmospheric properties, coupled analysis, and ouputs. As illustrated in Fig. 2, these are in turn grouped into a single mission system, and the n mission systems are grouped into a 'mission analyses' system. The 'mission analyses' system is grouped with the allocation inputs and allocation outputs systems under the top-level system.

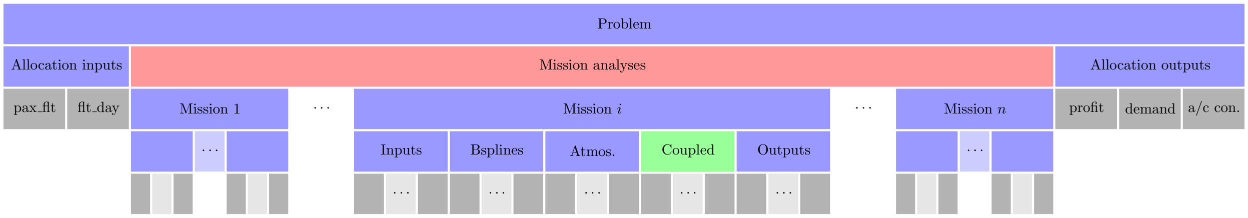


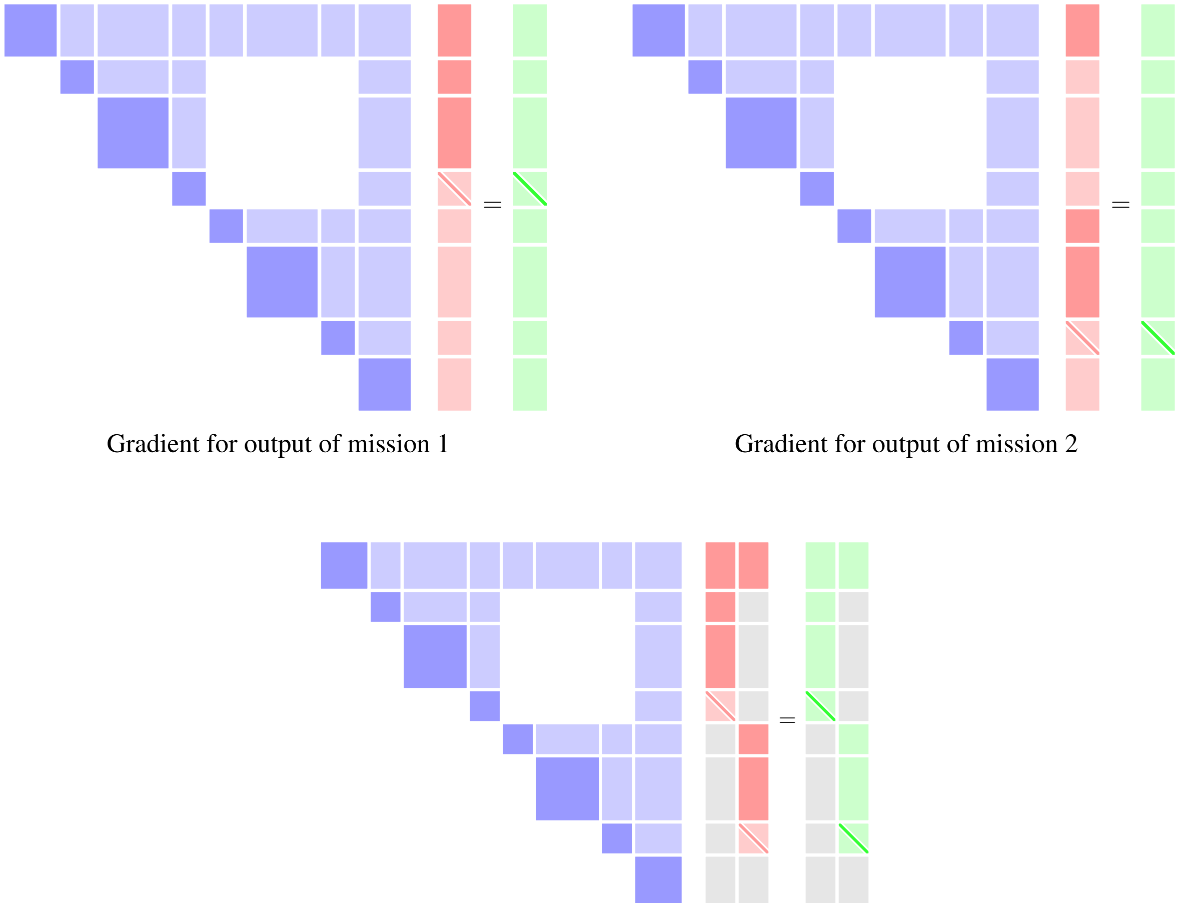
Figure 2: Hierarchy tree showing how the allocation-mission model is constructed. The boxes in grey are the variables, the blue boxes group systems together in series, and the red box groups its systems in parallel. The coupled analysis box is shown in green because its variables are coupled, so it uses Newton's method and GMRES.

The parallelization occurs in the mission analyses system. During initialization, each serial group (each blue box) passes its MPI communicator (group of processors) to each of its children, but the parallel group (the red box) splits its MPI communicator and assigns each of the n mission systems a single processor in this case. In terms of solvers, the mission analyses system applies block nonlinear Jacobi and block linear Jacobi, while the serial groups apply block nonlinear Gauss–Seidel and block linear Gauss–Seidel on its child systems. The exception among the serial groups is the coupled analysis system; since its children are coupled, as shown in Fig. 1, its nonlinear solver is the Newton iteration and its linear solver is the generalized minimal residual (GMRES) method.

There are two key developments in the design of the parallel framework that make it scale well in computation time and memory usage. The first enables the parallel computation of independent constraints. Let us consider the problem of finding the gradient of the min or max thrust KS constraint with respect to the altitude control points in a two-mission problem. To solve the optimization problem (NLP-a-m), we must compute these gradients for both missions, but we can take advantage of the fact that each mission's KS constraints are only affected by its altitude control points and not those of other missions. As shown in Fig. 3, a naive approach would solve one mission's gradients at a time, but the parallel framework allows for multiple gradients to be solved simultaneously by supporting linear algebra with multiple right-hand sides. The second key aspect of the parallel framework's design is that it analyzes the dependency graph as an initialization step and determines which variables are involved in the computation of each gradient, given the inputs of interest. The vectors are then allocated to contain only the parts that are relevant in the computation to minimize memory usage, but the remaining parts of the vector that are actually allocated are concatenated and stored in a contiguous way in memory so that the vector operations and parallel data transfers are efficient. Figure 3 illustrates this as well—the grey portions of the solution and right-hand side vectors are not allocated.

V. Results

In this section, we solve a 128-route allocation-mission optimization problem to demonstrate the efficiency of the parallel framework and the parallel allocation-mission optimization algorithm. The routes vary in range from roughly 200 nmi to 7600 nmi, and we have chosen a diverse fleet. The existing aircraft are the Embraer 170, Boeing 737-800, Boeing 777-200ER, and the Boeing 747-400, and the next-generation aircraft is an advanced conventional aircraft based on the common research model (CRM) [11]. The cost and performance data for the existing aircraft are obtained from the FLEET simulation tool [12].



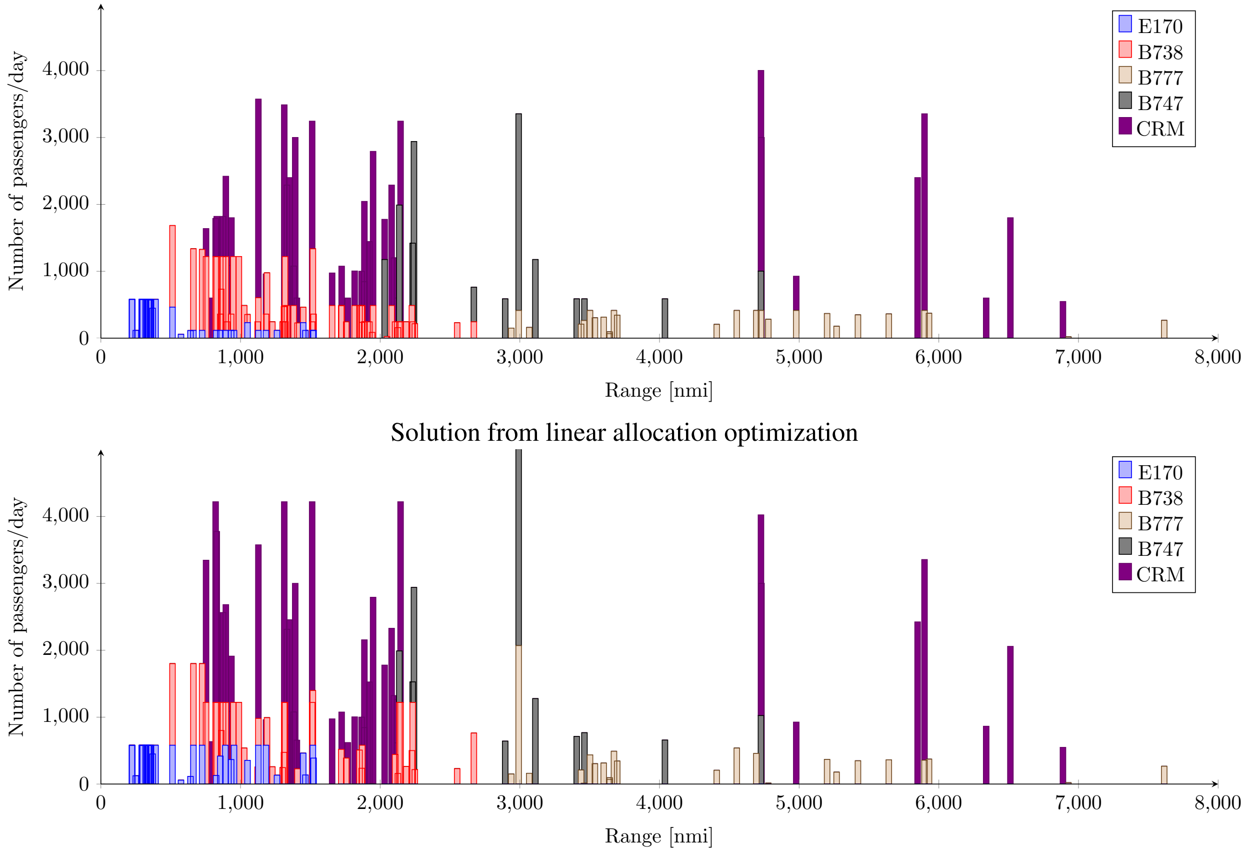
Gradient for output of mission 1

Gradient for output of mission 2

Gradients for outputs of both missions

Figure 3: The computation of the gradients of the min and max thrust KS constraints with respect to the altitude control points in a two-route allocation-mission optimization problem. At the top, the naive approach solves for the two gradient sequentially, meaning one processor idles while the other solves for the gradient. The bottom system shows how the parallel framework is implemented as it is able to solve linear systems with multiple right-hand sides simultaneously.

The parallel allocation-mission optimization was initialized by first optimizing each mission profile individually, and solving the linear allocation-only optimization problem to obtain good starting points for the allocation design variables. Figure 4 shows the optimization solutions from the allocation-only optimization and the parallel allocation-mission optimization. An interesting result is that the CRM, which is similar in size to the Boeing 777, a long-range jet, is actually used for many short-range missions, likely because its overall efficiency outweighs the fact that it is designed for longer-range routes.



Solution from linear allocation optimization

Solution from parallel allocation-mission optimization

Figure 4: Optimization results from the 128-route problem.

The 128-route optimization problem is presented in Tab. 1 and Fig. 5 shows the convergence history. It converged in about 8 hours. It is solved using SNOPT [13] via a modified version of the Python interface provided in the pyOpt optimization library [14]. Because of the accuracy that analytic derivative computation provides, the optimization is able to converge several orders of magnitude in both optimality and feasibility. The merit function does not show a significant increase because the initial values for passengers per flight and flights per day design variables are obtained from the solution of the allocation-only optimization problem. Since the parallel allocation-mission optimization solves the relaxed problem, it is not surprising that the final profit of $29.83 million is much higher than the profit after the mixed-integer linear optimization of $23.61 million.

Figure 6 plots run times for the evaluation of the allocation-mission model and the computation of all the adjoints. For each data point in Fig. 6, the results are obtained by running the computation on the same number of processors as routes. The model evaluation time shows very good scaling with the problem size up to 128 routes. The derivative computation time shows weaker scaling; however, the plot is of the total solve time for all the adjoints. Part of the increasing trend can be attributed to the fact that as the number of routes increases, the number of constraints increases, in addition to the sizes of the adjoint linear systems increasing.

	Variable	Quantity
maximize	profit	
with respect to	passengers per flight for each aircraft type and route	5 × 128
	flights per day for each aircraft type and route	5 × 128
	altitude control points from each route	4575
	Total number of design variables	5855
subject to	demand constraint for each route	128
	total flight time constraint for each aircraft	5
	idle thrust KS constraint for each route	128
	max thrust KS constraint for each route	128
	linear climb angle bounds for each mission	22875
	Total number of constraints	23264

Table 1: The 128-route allocation-mission optimization problem.

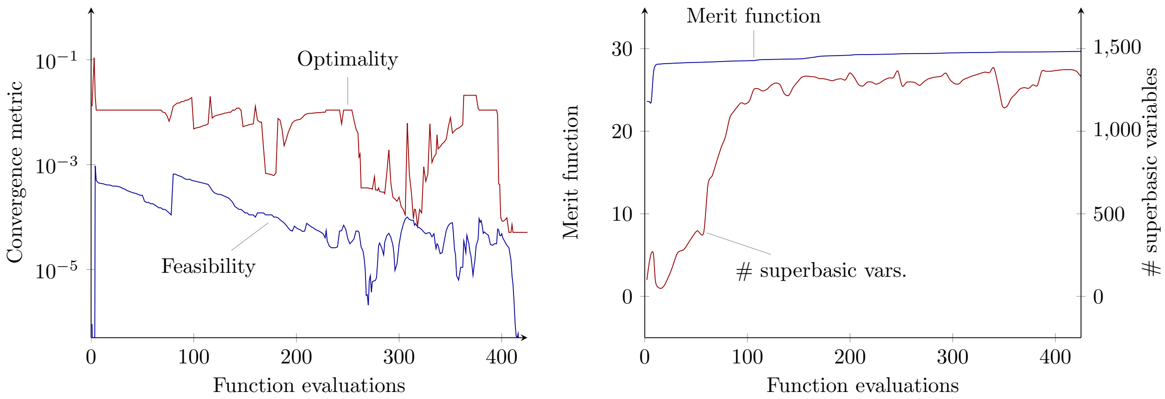


Figure 5: Optimization convergence history.

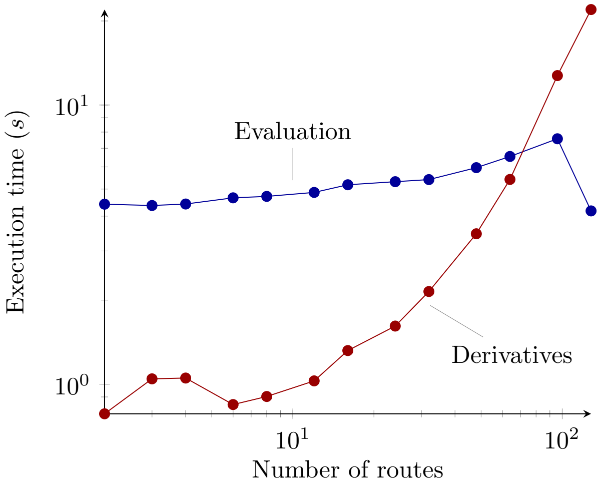


Figure 6: Scaling of model evaluation and derivative computation time with number of routes. The results are run using the same number of processors as routes.

VI. Conclusion

In this paper, we demonstrated parallel allocation-mission optimization on a representative 128-route network. Following previous work, we implemented the aerodynamic surrogate model, mission analysis models, and airline allocation models within a computational framework that automates derivative computation using the adjoint method. We extended the parallel framework to support linear algebra with multiple right-hand sides to enable parallel constraint gradient computation, which is a key, enabling component of the algorithm. The vectors are sparse at the discipline- or component-level for better scaling of memory usage. We demonstrated that the algorithm scales well up to at least 128 processors by solving a 128-route allocation-mission optimization problem.

There are several topics for future work as a continuation of this research effort. The first is to add design into the problem to perform allocation-mission-design optimization. With the parallel framework at least partially validated, it is ready to add a CFD solver as a component to dynamically re-train the surrogate model during optimization to allow aircraft design variables to be included. Another topic for future work is to improve the propulsion model, potentially using a surrogate as well to allow engine design variables to be included in the future. The final major avenue for future work is to re-introduce the mixed-integer aspect of the problem once again and incorporate the parallel allocation-mission-design optimization within the B&B algorithm.

VII. Acknowledgments

The authors gratefully acknowledge support from NASA through grant number NNX14AC73A—technical monitor Justin S. Gray. The authors also thank Satadru Roy from Purdue University for providing the network data.

References

[1] Hwang, J. T., Roy, S., Kao, J. Y., Martins, J. R. R. A., and Crossley, W. A., "Simultaneous aircraft allocation and mission optimization using a modular adjoint approach," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Jan 2015. doi:10.2514/6.2015-0900.

[2] Hwang, J. T., *A modular approach to large-scale design optimization of aerospace systems*, Ph.D. thesis, University of Michigan, 2015.

[3] Heath, C. and Gray, J., "OpenMDAO: Framework for Flexible Multidisciplinary Design, Analysis and Optimization Methods," *Proceedings of the 53rd AIAA Structures, Structural Dynamics and Materials Conference*, Honolulu, HI, April 2012, AIAA-2012-1673.

[4] Kao, J. Y., Hwang, J. T., Martins, J. R. R. A., Gray, J. S., and Moore, K. T., "A modular adjoint approach to aircraft mission analysis and optimization," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Jan 2015. doi:10.2514/6.2015-0136.

[5] Hwang, J. T., Lee, D. Y., Cutler, J. W., and Martins, J. R. R. A., "Large-Scale Multidisciplinary Optimization of a Small Satellite's Design and Operation," *Journal of Spacecraft and Rockets*, Vol. 51, No. 5, September 2014, pp. 1648–1663. doi:10.2514/1.A32751.

[6] Gray, J., Hearn, T., Moore, K., Hwang, J. T., Martins, J. R. R. A., and Ning, A., "Automatic Evaluation of Multidisciplinary Derivatives Using a Graph-Based Problem Formulation in OpenMDAO," *Proceedings of the 15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, June 2014. doi:10.2514/6.2014-2042.

[7] Kreisselmeier, G. and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *International Federation of Active Controls Syposium on Computer-Aided Design of Control Systems, Zurich, Switzerland*, 1979.

[8] Martins, J. R. R. A. and Hwang, J. T., "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, Vol. 51, No. 11, November 2013, pp. 2582–2599. doi:10.2514/1.J052184.

[9] Ferguson, A. R. and Dantzig, G. B., "The allocation of aircraft to routes-an example of linear programming under uncertain demand," *Management science*, Vol. 3, No. 1, 1956, pp. 45–73.

[10] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, Birkhäuser Press, 1997, pp. 163–202.

[11] Vassberg, J. C., DeHaan, M. A., Rivers, S. M., and Wahls, R. A., "Development of a Common Research Model for Applied CFD Validation Studies," 2008, AIAA 2008-6919.

[12] Govindaraju, P. and Crossley, W. A., "Profit Motivated Airline Fleet Allocation and Concurrent Aircraft Design for Multiple Airlines," *Aviation Technology, Integration, and Operations Conference*, Aug 2013, AIAA 2013-4391.

[13] Gill, P., Murray, W., and Saunders, M., "SNOPT: An SQP algorithm for large–scale constraint optimization," *SIAM Journal of Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[14] Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., "pyOpt: a Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization," *Structural and Multidisciplinary Optimization*, Vol. 45, No. 1, January 2012, pp. 101–118. doi:10.1007/s00158-011-0666-3.