



# Multidisciplinary Design Optimization Approach to Integrated Space Mission Planning and Spacecraft Design

Masafumi Isaji\*, Yuji Takubo†, and Koki Ho‡  
*Georgia Institute of Technology, Atlanta, GA, 30332, USA*

Space mission planning and spacecraft design are tightly coupled and need to be considered together for optimal performance; however, this integrated optimization problem results in a large-scale Mixed-Integer Nonlinear Programming (MINLP) problem, which is challenging to solve. In response to this challenge, this paper proposes a new solution approach to this MINLP problem by iterative solving a set of coupled subproblems via the augmented Lagrangian coordination approach following the philosophy of Multi-disciplinary Design Optimization (MDO). The proposed approach leverages the unique structure of the problem that enables its decomposition into a set of coupled subproblems of different types: a Mixed-Integer Quadratic Programming (MIQP) subproblem for mission planning and one or more Nonlinear Programming (NLP) subproblem(s) for spacecraft design. Since specialized MIQP or NLP solvers can be applied to each subproblem, the proposed approach can efficiently solve the otherwise intractable integrated MINLP problem. An automatic and effective method to find an initial solution for this iterative approach is also proposed so that the optimization can be performed without the need for a user-defined initial guess. In the demonstration case study, a human lunar exploration mission sequence is optimized with a subsystem-level parametric spacecraft design model. Compared to the state-of-the-art method, the proposed formulation can obtain a better solution with a shorter computational time even without parallelization. For larger problems, the proposed solution approach can also be easily parallelizable and thus is expected to be further advantageous and scalable.

## I. Nomenclature

$\mathcal{A}$	= Set of arcs
$a_{vijt}$	= Cost coefficient matrix of commodity
$a'_{vijt}$	= Cost coefficient matrix of spacecraft
$d_{it}$	= demand vector
$e_v$	= Spacecraft design variable vector
$\mathcal{F}(-)$	= Spacecraft sizing function
$f$	= Objective function (subproblem)
$g$	= Inequality constraint
$h$	= Equality constraint
$H_{vij}$	= Concurrency matrix
$\mathcal{J}$	= Objective function
$k$	= Decomposed subproblem index
$L$	= Number of subsystems in the dry mass
$M$	= Number of subproblems in a quasi-separable MDO problem
$m$	= Mass of spacecraft subsystems
$m_d$	= Spacecraft dry mass
$m_f$	= Spacecraft propellant capacity
$m_p$	= Spacecraft payload capacity
$N$	= Number of types of spacecraft
$\mathcal{N}$	= Set of nodes

\*Ph.D. Student, Daniel Guggenheim School of Aerospace Engineering, Atlanta, GA, AIAA Student Member.

†Undergraduate Student, Daniel Guggenheim School of Aerospace Engineering, Atlanta, GA, AIAA Student Member.

‡Assistant Professor, Daniel Guggenheim School of Aerospace Engineering, Atlanta, GA, AIAA Senior Member.

$n$	= dimension of variables
$Q_{vijt}$	= Commodity transformation matrix
$q$	= Iteration count
$\mathcal{T}$	= Set of time steps
$t_{mis}$	= Mission length
$\Delta t_{ij}$	= Time of Flight (ToF)
$u_{vijt}$	= Spacecraft flow variable
$\mathcal{V}$	= Set of spacecraft
$W_{ij}$	= Launch time window
$x_{vijt}$	= Commodity flow variable
$y$	= Shared variables
$z$	= Local variables
$\zeta$	= Propellant type
$\phi$	= Penalty function
<i>Subscript</i>	
$i$	= Node index (departure)
$j$	= Node index (arrival)
$k$	= Subproblem index
$l$	= Subsystem index
$t$	= Time index
$v$	= Vehicle index

## II. Introduction

As we pursue sustainable presence in space, a framework to optimize large-scale, long-term space missions efficiently is imperative. A number of studies on space logistics that incorporates the transportation network in large-scale space mission design have been developed, including SpaceNet [1], the interplanetary logistics model [2], and the extensive literature on space logistics optimization frameworks based on the generalized multicommodity network flow [3–5]. Utilizing the linear nature of such space logistics or transportation network optimization problems, researchers have developed frameworks that can efficiently optimize the mission design as Mixed-Integer Linear Programming (MILP) problems [6–9]. However, due to the nonlinear nature of spacecraft design, a naive integration of spacecraft design into space mission/campaign planning (a transportation scheduling or resource distribution) would result in a large-scale Mixed-Integer Nonlinear Programming (MINLP) problem, which is oftentimes computationally prohibitive. Since the concurrent optimization of space mission planning and spacecraft design is highly desired in practice, each community took different approaches to bridge these two domains.

In the space logistics community, spacecraft design has been considered as a high-level nonlinear sizing model and has been integrated into mission planning either by separating the nonlinear part from the mission planning optimization or by piecewise linearization of the spacecraft model. Taylor [10] developed a parametric spacecraft sizing model which determines the spacecraft dry mass from its payload capacity and propellant capacity. Based on this model, Simulated Annealing (SA) or a similar metaheuristic optimization algorithm optimizes the spacecraft design variables, while the linear programming (LP) or MILP solver evaluates the constraints and determines transportation flow variables. In this way, the LP or MILP solver is embedded into SA, and thus it was called the embedded optimization methodology. Using the same spacecraft sizing model, Chen and Ho [6] employed piecewise linear (PWL) approximation of the nonlinear model to approximate the entire MINLP problem as a MILP problem that can be efficiently solvable. However, this approach is an approximation model, and the resulting solution is not guaranteed to be feasible nor optimal in the original nonlinear problem.

On the other hand, aerospace vehicle design has been tackled by the Multidisciplinary Design Optimization (MDO) community. Despite various optimization and sizing methods that can deal with the high-dimensional nonlinear design of aircraft or spacecraft [11], few studies integrated the mission-level analysis or optimization. One of the few studies that tackled the integrated mission planning and spacecraft design is Ref. [12] by Beauregard et al., which proposed an MDO architecture for a lunar lander design with a lunar mission sequence architecture analysis. This architecture connects the mission planning and spacecraft design problem using a sequential procedure without a feedback structure (i.e., the mission architecture is first chosen and fixed, then the lunar lander MDO is performed); therefore, strictly speaking, the

mission and spacecraft are not simultaneously optimized and spacecraft design is neglected when selecting the mission architecture. In addition, the candidates of the mission architectures are given *a priori* and discrete (combinatory). These two factors limit the design space and make this approach not suitable for the integrated space mission design.

This paper proposes an efficient decomposition-based optimization scheme for integrated space mission planning and spacecraft design. The key idea is to decompose the integrated MINLP problem into multiple coupled subproblems of different types: the Mixed-Integer Quadratic Programming (MIQP) subproblem for space mission planning and the Nonlinear Programming (NLP) subproblem(s) for spacecraft design. Since specialized efficient MIQP or NLP optimizers (e.g., Gurobi [13] for MIQP; IPOPT [14] for NLP) can be utilized to solve each subproblem, the proposed method can solve the otherwise intractable integrated MINLP problem efficiently. The iterative coordination between each subproblem can be achieved using an MDO approach [11, 15]. Specifically, the Augmented Lagrangian Coordination (ALC) approach [16] with the Analytical Target Cascading (ATC) structure [17, 18] is chosen for the proposed method. This architecture fits our problem well because (1) it allows us to decompose the original complex problem into the subproblems with different and simpler types (MIQP or NLP), each of which can be efficiently solvable with specialized solvers; (2) it has a robust convergence property; and (3) it allows the complex hierarchical structure for the spacecraft design subproblem(s) and can be easily parallelizable (and thus scalable) if needed. Since the nonlinear optimization solvers generally require a good initial guess, we further develop an automated initial guess generation method based on PWL approximation to the MINLP problem so that no user-defined initial guess is needed for the optimization.

The remainder of this paper proceeds as follows. In Section III, the problem definition of the integrated space mission planning and spacecraft design as an all-in-one optimization problem formulation is described. Section IV illustrates the solution procedure for the proposed problem based on the decomposition-based method. Section V introduces a case study of human lunar exploration missions and compares the computational efficiency of the proposed method and existing method. Finally, Section VI states the conclusion.

### III. Problem Definition: Integrated Space Mission Planning and Spacecraft Design

The goal of this research is to optimize the transportation scheduling (referred to as space mission planning) and vehicle design (referred to as spacecraft design) for a long-term space campaign that can potentially comprise multiple missions. This section introduces the formulation for this integrated space mission planning and spacecraft design problem (referred to as the all-in-one formulation). The idea behind this formulation is to consider space mission planning as a transportation network optimization problem for which the design of vehicles is also part of the decision variables. In the network, the nodes correspond to the orbital or surface locations and the arcs correspond to the trajectories connecting the nodes. The decision variables include both the commodities that flow over the network and the design parameters for the vehicles that carry these commodities. The optimization formulation is listed as follows, and the list of variables and parameters is included in Table 1.

$$\min \quad \mathcal{J} = \sum_{t \in \mathcal{T}} \sum_{(v, i, j) \in \mathcal{A}} (\mathbf{a}_{vijt}^T \mathbf{x}_{vijt} + a'_{vijt}^T m_{dv} u_{vijt}) \quad (1)$$

subject to

$$\sum_{(v, j) : (v, i, j) \in \mathcal{A}} \begin{bmatrix} \mathbf{x}_{vijt} \\ m_{dv} u_{vijt} \end{bmatrix} - \sum_{(v, j) : (v, i, j) \in \mathcal{A}} Q_{vijt} \begin{bmatrix} \mathbf{x}_{vji(t-\Delta t_{ji})} \\ m_{dv} u_{vji(t-\Delta t_{ji})} \end{bmatrix} \leq \mathbf{d}_{it} \quad \forall t \in \mathcal{T} \quad \forall i \in \mathcal{N} \quad (2)$$

$$H_{vij} \mathbf{x}_{vijt} \leq \mathbf{e}_v u_{vijt} \quad \forall t \in \mathcal{T} \quad \forall (v, i, j) \in \mathcal{A} \quad (3)$$

$$\begin{cases} \mathbf{x}_{vijt} \geq \mathbf{0}_{p \times 1} & \text{if } t \in W_{ij} \\ \mathbf{x}_{vijt} = \mathbf{0}_{p \times 1} & \text{otherwise} \end{cases} \quad \forall (v, i, j, t) \in \mathcal{A} \quad (4)$$

$$m_{dv} = \mathcal{F}(\mathbf{e}_v, \zeta_v) \quad \forall v \in \mathcal{V} \quad (5)$$

$$\mathbf{x}_{vijt} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}_{vijt}, \quad \begin{array}{ll} x_n \in \mathbb{R}_{\geq 0} & \forall n \in C_c \\ x_n \in \mathbb{Z}_{\geq 0} & \forall n \in C_d \end{array} \quad \forall (v, i, j, t) \in \mathcal{A} \quad (6)$$

$$u_{vijt} \in \mathbb{Z}_{\geq 0} \quad \forall (v, i, j, t) \in \mathcal{A} \quad (7)$$

$$\mathbf{e}_v = \begin{bmatrix} m_p \\ m_f \end{bmatrix}_v, \quad m_{p_v}, m_{\zeta_v}, m_{d_v} \in \mathbb{R}_{\geq 0}, \quad \forall v \in \mathcal{V} \quad (8)$$

**Table 1** Variables and parameters used in the space transportation scheduling problem

Name	Description
<i>Variables</i>	
$x_{vijt}$	Commodity flow variable, or the quantity of the commodity delivered from node $j$ to $i$ at time $t$ by spacecraft $v$ . $x_{vijt} \geq 0$ . Each component of this variable can contain either continuous variables ( $C_c$ ) or discrete variables ( $C_d$ ). This vector will be $p \times 1$ vector if the total commodity variation is $p$ .
$u_{vijt}$	Spacecraft flow variable, which indicates the number of spacecraft type $v$ moving from node $i$ to $j$ at time $t$ . This variable is integer scalar.
$\mathbf{e}_v$	Spacecraft design variables and parameters. In this problem, it includes payload capacity $m_p$ and propellant capacity $m_f$ .
$m_d$	Spacecraft dry mass.
<i>Parameters</i>	
$a_{vijt}$	Cost coefficient matrix of commodity.
$a'_{vijt}$	Cost coefficient of spacecraft.
$d_{it}$	Demands/supplies of different commodities and spacecraft at node $i$ at time $t$ .
$Q_{vijt}$	Transformation matrix.
$H_{vij}$	Concurrency constraint matrix.
$W_{ij}$	Launch window vector, which indicates the available launch window of spacecraft.
$\mathcal{F}(-)$	Spacecraft sizing function. This illustrates the nonlinear relationship of the spacecraft design variables and design parameters.
$\Delta t_{ij}$	Time of Flight (ToF) from node $i$ to $j$ .
$\zeta_v$	Propellant type for each spacecraft (predetermined).
<i>Sets</i>	
$\mathcal{A}(\mathcal{V}, \mathcal{N}, \mathcal{N}, \mathcal{T})$	Set of arcs realized by spacecraft.
$\mathcal{N}$	Set of nodes.
$\mathcal{T}$	Set of time steps.
$\mathcal{V}$	Set of spacecraft (vehicles).

Equation (1) indicates the objective function, which can be the lifecycle cost or launch mass, depending on the application context. In this research, we set the coefficients  $a_{vijt}$  and  $a'_{vijt}$  so that the objective function corresponds to the sum of initial mass at low-earth orbit (IMLEO).

Equations (2)-(4) are the constraints for space mission planning. First, Eq. (2) is the mass balance constraint that guarantees that the inflow (supply) of the commodity is larger than the sum of the outflow and demand.  $Q_{vijt}$  is the transformation matrix, which indicates the transformation of the commodity during the spaceflight; for example, the relationship of impulsive propellant consumption can be illustrated using this constraint. Next, Eq. (3) is the concurrency constraint. This indicates that the commodity loaded on each spacecraft is constrained by the dimension of the spacecraft. Specifically in this paper, the payload and propellant flow is limited: the amount of propellant is lower than the propellant capacity of the spacecraft, and the sum of other payloads is lower than the payload capacity. Finally, Eq. (4) is the time window constraints. The commodity flow is allowed only if the time  $t$  belongs to the launch window vector  $W_{ij}$ , and for the remaining time steps, the commodity flow is conserved to be zero.

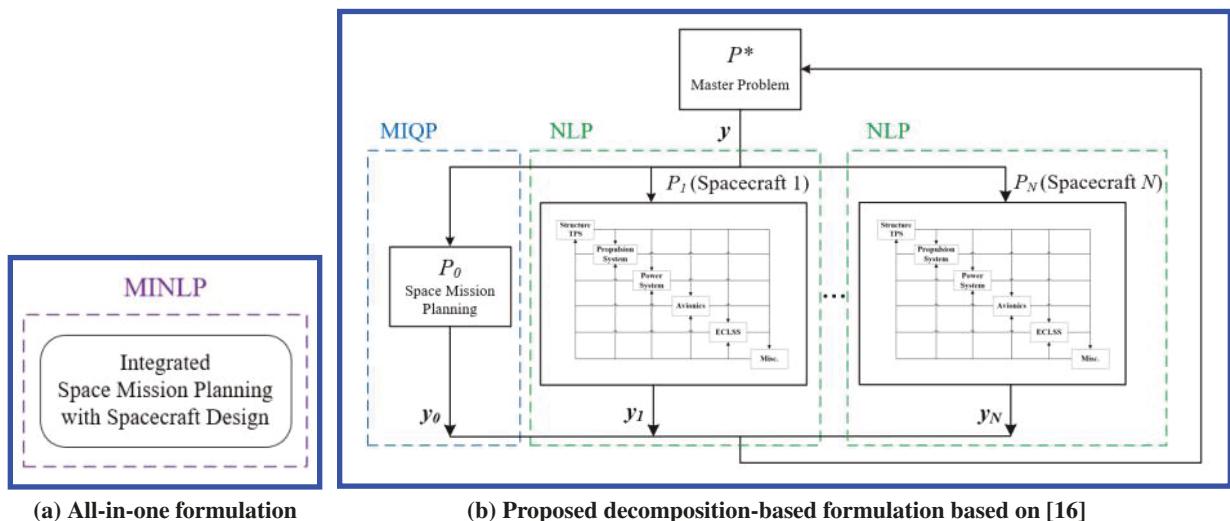
Equation (5) indicates an abstract representation of the spacecraft design constraints, which describes the constraints between the properties of the vehicle. This can take a wide range of complexity, including an explicit or implicit relationship of the subsystems or design parameters of the spacecraft; when the spacecraft requires multiple disciplines or multiple subsystems, an MDO problem can be embedded in this constraint.

Along with Table 1, Eqs. (6), (7), and (8) show the definitions and domains of commodity flow variables, spacecraft flow variables, and spacecraft design variables, respectively.

This integrated mission planning and spacecraft design problem results in a constrained MINLP problem, one of the most challenging optimization problem types to solve. Namely, this problem contains both discrete and continuous variables as well as both linear and nonlinear constraints. Specifically, the discrete variables represent the definition of the commodity flow and the number of spacecraft on the mission planning side of the problem. In addition, the nonlinearity appears in two ways: (1) the spacecraft design relationship in Eq. (5); (2) the quadratic terms in the mass balance constraint (Eq. 2) and concurrency constraint (Eq. 3) for mission planning (Note: both  $e_v$  and  $u_{vijt}$  are variables). Fortunately, this second nonlinearity can be converted into an equivalent linear relationship through the big-M method, as explained in Ref. [6], so that the nonlinearity only exists on the spacecraft design side of the problem. Therefore, as a result, the problem contains two coupled problems: one for space mission planning which is linear with integer variables, and the other for spacecraft design which is nonlinear with continuous variables. Our approach leverages this unique structure of the problem and proposes a new approach to solve this problem efficiently.

#### IV. Proposed Approach: Decomposition-Based Optimization with Augmented Lagrangian Coordination

Decomposition-based optimization is often used to decompose an MDO problem in terms of disciplines or subsystems. Leveraging the unique feature of the integrated space mission planning and spacecraft design problem, we apply this approach to decompose the large-scale MINLP problem (Fig. 1a) into coupled MIQP and NLP subproblems (Fig. 1b), each of which is significantly easier to solve with specialized solvers compared to the original MINLP problem. The space mission planning subproblem can be solved using a MIQP solver, and the spacecraft design subproblem can be solved using an NLP solver without any integer variables. The coupled subproblems are solved iteratively using the ALC-based coordination until convergence is reached. To enable the optimization without a user-defined initial guess, an automated and effective initial solution generation approach is also proposed.



**Fig. 1** Solution strategy for integrated space mission planning and spacecraft design.

##### A. Derivation of Decomposed Problems with Augmented Lagrangian Coordination

We first start with deriving the formulations of the decomposed problems with ALC. ALC tackles complex MDO optimization problems that are quasi-separable and thus can be decomposed into a set of coupled subproblems. ALC is attractive because of (1) its ability to break down our MINLP problem into MIQP and NLP problems; (2) its robust

convergence property; and (3) its flexibility with the hierarchical structure of the problems. For an extensive discussion on ALC, refer to Ref. [16].

The formulation for the quasi-separable MDO problem with  $M$  subproblems is given as follows:

$$\begin{aligned} & \min_{\mathbf{y}, \mathbf{z}_0, \dots, \mathbf{z}_{M-1}} \sum_{k=0}^{M-1} f_k(\mathbf{y}, \mathbf{z}_k) \\ \text{subject to } & \mathbf{g}_k(\mathbf{y}, \mathbf{z}_k) \leq \mathbf{0} \quad k = 0, \dots, M-1 \\ & \mathbf{h}_k(\mathbf{y}, \mathbf{z}_k) = \mathbf{0} \quad k = 0, \dots, M-1 \end{aligned} \quad (9)$$

where  $\mathbf{y} \in \mathbb{R}^{n^y}$  indicates the shared variables,  $\mathbf{z}_k \in \mathbb{R}^{n_k^z}$  indicates the local variables for subproblem  $k$ . The shared variables  $\mathbf{y}$  can be common variables over multiple subproblems.  $f_k : \mathbb{R}^{n_k} \mapsto \mathbb{R}$  indicates the local objective function,  $\mathbf{g}_k$  and  $\mathbf{h}_k$  indicate the equality and inequality constraints for each subproblem. The dimension of the total design variable  $\mathbf{s} = [\mathbf{y}^T, \mathbf{z}_0^T, \dots, \mathbf{z}_{M-1}^T]^T$ ,  $\mathbf{s} \in \mathbb{R}^n$  is  $n = n^y + \sum_{k=0}^{M-1} n_k^z$ . The dimension of the local design variable is  $n_j = n^y + n_k^z$ .

The decomposition-based approach for this problem follows the following steps. First, we introduce the auxiliary variables and consistency constraints so that the local constraints,  $\mathbf{g}_k$  and  $\mathbf{h}_k$ , are only dependent on the auxiliary variables  $\mathbf{y}_k$  and independent of the shared variables  $\mathbf{y}$ .

$$\begin{aligned} & \min_{\mathbf{y}, \mathbf{y}_0, \mathbf{z}_0, \dots, \mathbf{y}_{M-1}, \mathbf{z}_{M-1}} \sum_{k=0}^{M-1} f_k(\mathbf{y}_k, \mathbf{z}_k) \\ \text{subject to } & \mathbf{g}_k(\mathbf{y}_k, \mathbf{z}_k) \leq \mathbf{0} \quad k = 0, \dots, M-1 \\ & \mathbf{h}_k(\mathbf{y}_k, \mathbf{z}_k) = \mathbf{0} \quad k = 0, \dots, M-1 \\ & \mathbf{c}_k(\mathbf{y}, \mathbf{y}_k) = \mathbf{0} \quad k = 0, \dots, M-1 \end{aligned} \quad (10)$$

With the consistency constraints  $\mathbf{c}_k$ , which ensures that the auxiliary variables  $\mathbf{y}_k$  are the same as the shared variables  $\mathbf{y}$ , the shared variables are separated from the local variables while representing the same problem as the original one. Next, the relaxation of the consistency constraints is introduced with the local Lagrangian penalty function:

$$\begin{aligned} & \min_{\mathbf{y}, \mathbf{y}_0, \mathbf{z}_0, \dots, \mathbf{y}_{M-1}, \mathbf{z}_{M-1}} \sum_{k=0}^{M-1} f_k(\mathbf{y}_k, \mathbf{z}_k) + \sum_{k=0}^{M-1} \phi_k(\mathbf{c}_k(\mathbf{y}, \mathbf{y}_k)) \\ \text{subject to } & \mathbf{g}_k(\mathbf{y}_k, \mathbf{z}_k) \leq \mathbf{0} \quad k = 0, \dots, M-1 \\ & \mathbf{h}_k(\mathbf{y}_k, \mathbf{z}_k) = \mathbf{0} \quad k = 0, \dots, M-1 \end{aligned} \quad (11)$$

The augmented Lagrangian penalty function for subproblem  $k$ ,  $\phi_k$ , is defined as follows.

$$\phi_k(\mathbf{c}_k(\mathbf{y}, \mathbf{y}_k)) = \mathbf{v}_k^T(\mathbf{y} - \mathbf{y}_k) + \|\mathbf{w}_k \circ (\mathbf{y} - \mathbf{y}_k)\|_2^2 \quad (12)$$

where  $\mathbf{v}$  is the vector of Lagrange multiplier estimates, and  $\mathbf{w}$  is the vector of penalty weights. Here,  $\circ$  represents the element-wise product of matrices or vectors, also known as the Hadamard product. By moving the consistency constraints into the local objective functions, the local subproblems can be completely separated. The bi-level decomposition-based problem is now formulated by establishing the master problem above the subproblems. The master problem minimizes the penalty function and updates the shared variables  $\mathbf{y}$ . Note that even though the bi-level formulation is employed here, the ALC has the capability to handle multi-level hierarchical formulation as well.

(1) Master Problem

$$\min_{\mathbf{y}} \sum_{k=0}^{M-1} \phi_k(\mathbf{c}_k(\mathbf{y}, \mathbf{y}_k)) \quad (13)$$

(2) Subproblem  $k$

$$\begin{aligned} & \min_{\mathbf{y}_k, \mathbf{z}_k} f_k(\mathbf{y}_k, \mathbf{z}_k) + \phi_k(\mathbf{c}_k(\mathbf{y}, \mathbf{y}_k)) \\ \text{subject to } & \mathbf{g}_k(\mathbf{y}_k, \mathbf{z}_k) \leq \mathbf{0} \\ & \mathbf{h}_k(\mathbf{y}_k, \mathbf{z}_k) = \mathbf{0} \end{aligned} \quad (14)$$

Adopting the above approach to our problem of the integrated space mission planning and spacecraft design with  $N$  vehicle types, Fig. 1b represents the decomposition-based optimization architecture. We have one space mission

planning subproblem (Subproblem 0) and multiple spacecraft design subproblems (Subproblems 1, ...,  $N$ ), where  $N$  is the number of spacecraft types; thus, we have  $N + 1$  subproblems in total (i.e.,  $M = N + 1$ ). The shared variables among them include the vehicle design parameters  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_N^T]^T$ , where  $\mathbf{y}_v = [m_{p_v}, m_{f_v}, m_{d_v}]^T$  for each vehicle  $v$  where  $m_p, m_f, m_d$  respectively represent payload capacity, propellant (fuel) capacity, and dry mass of the spacecraft.

First, the space mission planning problem ( $P_0$  in Fig. 1b) is different from the all-in-one formulation outlined in Section III with respect to the following two points: the nonlinear vehicle sizing constraint (Eq. (5)) is not included, and the quadratic penalty function is added to the objective function as Eq. (15) shows. Due to the quadratic objective function, this subproblem is a MIQP problem.

$$\begin{aligned} \min_{\mathbf{x}_{vijt}, u_{vijt}, \mathbf{y}_0} \quad & \sum_{t \in \mathcal{T}} \sum_{(v, i, j) \in \mathcal{A}} (\mathbf{a}_{vijt}^T \mathbf{x}_{vijt} + a_{vijt}^T m_{d_v} u_{vijt}) + \phi_0(\mathbf{c}_0(\mathbf{y}, \mathbf{y}_0)) \\ \text{subject to} \quad & \text{Eqs. (2)–(4) and (6)–(8)} \\ \text{where} \quad & \mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_N^T]^T \quad \text{and} \quad \mathbf{y}_v = [m_{p_v}, m_{f_v}, m_{d_v}]^T \end{aligned} \quad (15)$$

Next, for the spacecraft design subproblems ( $P_v$  in Fig. 1b), the penalty function is minimized and the vehicle sizing constraint ( $m_{d_v} = \mathcal{F}(m_{p_v}, m_{f_v})$ ) is enforced. This subproblem contains various interacting subsystems and a hierarchical structure can be used to provide detailed subsystem-level design if needed. The subproblem for  $v$ -th type of vehicle can be expressed as Eq. (16). Due to the nonlinear constraint, the subproblem is an NLP problem and can be solved by an NLP solver.

$$\begin{aligned} \min_{\mathbf{y}_v} \quad & \phi_v(\mathbf{c}_v(\mathbf{y}, \mathbf{y}_v)) \\ \text{subject to} \quad & m_{d_v} = \mathcal{F}(m_{p_v}, m_{f_v}) \\ \text{where} \quad & \mathbf{y}_v = [m_{p_v}, m_{f_v}, m_{d_v}]^T \end{aligned} \quad (16)$$

## B. Solution Algorithm and Iteration Scheme

This subsection introduces the iterative solution algorithm for the decomposition-based algorithm introduced in Section IV.A. The formulated decomposed optimization problems with ALC can be solved iteratively in two loops: the outer loop updates the augmented Lagrangian penalty parameters ( $\mathbf{v}, \mathbf{w}$ ), while the inner loop solves the master problem and each subproblem to update the variables. The iteration continues until the convergence (i.e., all subproblems are consistent, or  $\mathbf{c}_k$  is near zero, within a tolerance). The following describes the details of each loop.

For the updates for the outer loop, the solution from the inner loop is used [16]. Specifically, at  $q$ -th iteration,  $\mathbf{v}$  is updated as follows:

$$\mathbf{v}^{q+1} = \mathbf{v}^q + 2\mathbf{w}^q \circ \mathbf{w}^q \circ \mathbf{c}^q \quad (17)$$

In addition, for  $r$ -th consistency constraint  $c_r$ , the corresponding penalty weight  $w_r$  is updated as follows:

$$w_r^{q+1} = \begin{cases} w_r^q & \text{if } |c_r^q| \leq \gamma_2 |c_r^{q-1}| \\ \gamma_1 w_r^q & \text{if } |c_r^q| > \gamma_2 |c_r^{q-1}| \end{cases} \quad (18)$$

where  $\gamma_1 > 1$  and  $0 < \gamma_2 < 1$ . The initial penalty parameter values can take  $\mathbf{v}^1 = \mathbf{0}$  and  $\mathbf{w}^1 \approx \mathbf{1}$ .

The updates for the inner loop is performed by alternating between solving the master problem and the subproblems with the fixed penalty parameters. While each subproblem can be solved using the specialized numerical optimizer for MIQP or NLP, the master problem can be solved analytically as follows.

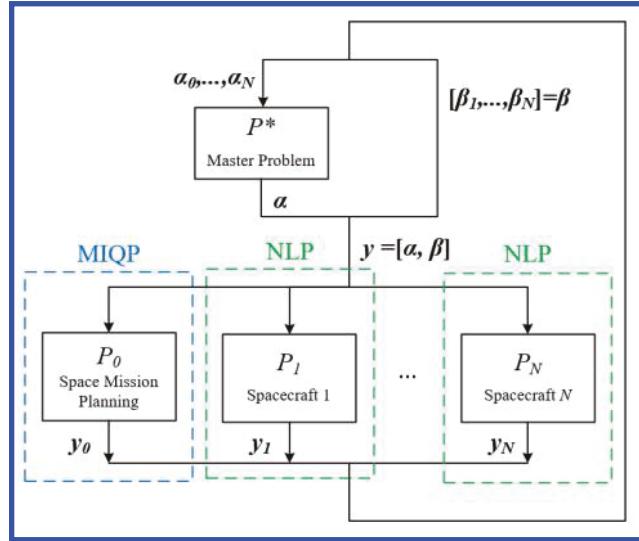
$$\mathbf{y} = \underset{\mathbf{y}}{\operatorname{argmin}} \sum_{k=0}^N \phi_k(\mathbf{c}_k(\mathbf{y}, \mathbf{y}_k)) = \frac{\sum_{k=0}^N (\mathbf{w}_k \circ \mathbf{w}_k \circ \mathbf{y}_k) - \frac{1}{2} \sum_{k=0}^N \mathbf{v}_k}{\sum_{k=0}^N (\mathbf{w}_k \circ \mathbf{w}_k)} \quad (19)$$

For our problem, we make an additional heuristics-based modification to the master problem to facilitate the convergence. Namely, the aforementioned master problem updates all the shared variables at the same time at every iteration, but this approach does not work effectively in our problem. This is because, the space mission planning, with no knowledge of the constraints behind the spacecraft design, can return an aggressive or infeasible spacecraft design, which can deteriorate the convergence performance. Therefore, we propose to only update the spacecraft payload capacity and the propellant capacity in the master problem, while passing the spacecraft dry mass from the

spacecraft design subproblem directly to the next iteration, as shown in Fig. 2. Mathematically, we separate the shared variables  $\mathbf{y}$  into the regular shared variables  $\boldsymbol{\alpha} = [m_{p_1}, m_{f_1}, \dots, m_{p_N}, m_{f_N}]$  and the prioritized shared variables  $\boldsymbol{\beta} = [m_{d_1}, \dots, m_{d_N}]$  (i.e.,  $\mathbf{y} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$ ), and only  $\boldsymbol{\alpha}$  is updated in the master problem.

$$\min_{\boldsymbol{\alpha}} \sum_{k=0}^N \phi_k(\mathbf{c}_k(\boldsymbol{\alpha}, \boldsymbol{\alpha}_k)) \quad (20)$$

Note that, in the space mission planning subproblem, the spacecraft dry mass remains a variable, not a fixed parameter, and is subject to the penalty function. It indicates that the resultant dry mass  $\beta_0$  is not used in the entire optimization architecture but only used to facilitate the convergence of the whole optimization problem.



**Fig. 2** Proposed decomposition-based optimization architecture with prioritized shared variables.

### C. Automatic Initial Solution Generation

For the above iterative algorithm to perform effectively, a good initial guess of the shared variable is necessary. Thus, there is a need to develop an automatic and effective method that does not require a user-defined initial guess. To this end, we propose to use the PWL approximation of the nonlinear optimization spacecraft design problem, and convert the entire MINLP into a MILP problem, which can be solved using a specialized solver [6]. Although the PWL approximation does not necessarily return an optimal or even feasible solution to the original MINLP problem, the returned shared variables can be used as a good initial guess for the iterative approach. Another advantage is that the MILP problem can be solved to the global optimum for the approximated nonlinear model [6]. Thus, the MILP-based initial guess is not only automatically generated but also likely to be close to the nonlinear global optimum.

Specifically, in our problem, nonlinearity exists on the spacecraft design side of the problem. Thus, we choose a series of equally-spaced "mesh" points over the feasible ranges of the spacecraft design parameters and use them as breakpoints for the PWL function generation. Note that since the dry mass is an (implicit) function of the payload capacity and propellant capacity, we only used the latter two for breakpoint generation. The breakpoint increment (or the number of breakpoints) is a key hyperparameter; a smaller increment or more breakpoints would lead to a more accurate initial guess, but it will also require a longer computational time.

### V. Case Study: Human Lunar Exploration Campaign

To demonstrate the effectiveness of the proposed approach, we perform a demonstration case study and compare our approach with the state-of-the-art method. We first introduce the case study settings, followed by the results and the computational performance analysis.

### A. Case Study Settings

A human lunar exploration with two missions is considered here for the case study. The mission network model, parameters, commodity demand and supply used in this case study are presented in Fig. 3, Table 2, and Table 3, respectively. Note that only one type of spacecraft, which is a single-stage lunar lander, is considered for simplicity. It means that the lander sizing constraint is applied to other vehicles such as in-space transfer vehicles. As landers are typically heavier than other spacecraft due to their landing structure, the optimization result might represent a conservative design. In addition, in-situ resource utilization (ISRU) is also considered as an option in the formulation, although it is never chosen by the optimizer in this case study due to the considered short time horizon.

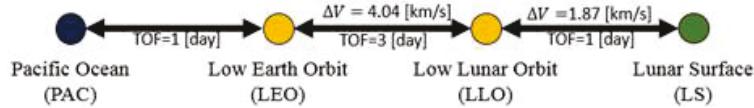


Fig. 3 Lunar campaign network model [6].

Table 2 Parameters used in the case study problem

Parameters	Assumed values
Spacecraft Propellant type	LH2/LOX
Propellant $I_{sp}$ , s	420
Propellant density $\rho_f$ , kg/m <sup>3</sup>	360
Spacecraft miscellaneous mass fraction $c_{misc}$ (see Eq. (21))	0.05
Type of spacecraft designed	1
Number of vehicles for each type	6
Crew mass (including space suit), kg/person	100
Crew consumption, kg/day/person	8.655
Spacecraft maintenance, structure mass/flight	1%

Table 3 Lunar campaign commodity demand and supply

Payload Type	Node	Time [days]	Supply/Demand
Outbound to the Moon			
Crew	Earth	0, 365	4
Habitat, Equipment, and Propellant, kg	Earth	0, 365	$\infty$
Crew	Moon	5, 370	-4
Habitat & Equipment, kg	Moon	5, 370	-2000
Inbound to the Earth			
Crew	Moon	8, 373	4
Returned mass, kg	Moon	8, 373	1000
Crew	Earth	13, 378	-4
Returned mass, kg	Earth	13, 378	-1000

The subsystem-level spacecraft model used as the spacecraft design constraint in Eq. (5) in this study is developed by the least square curve fitting to the data from the lunar lander design database in Ref. [19, 20]. The following set of equations shows the spacecraft model used in the case study.

$$m_d = \sum m_{sub} = m_{str} + m_{prop} + m_{power} + m_{avi} + m_{ECLSS} + m_{misc}$$

where

$$\begin{aligned}
 m_{str} &= n_{stg}^{-0.6705} (0.3238 m_d + 693.7 m_p^{0.04590}) \\
 m_{prop} &= 0.1648 (m_d + m_p) + 20.26 \left( \frac{m_f}{\rho_f} \right) \\
 m_{power} &= 7.277 \cdot 10^{-8} m_d^{2.443} + 137.0 \\
 m_{avi} &= 1.014 m_{power}^{0.8423} + 22.33 t_{mis} \\
 m_{ECLSS} &= 0.004190 n_{crew} t_{mis} m_d^{0.9061} n_{stg}^{0.7359} + 434.7 \\
 m_{misc} &= c_{misc} m_d
 \end{aligned} \tag{21}$$

Note that, beyond the payload capacity and propellant capacity, there are some additional parameters in these equations:  $n_{stg}$  is the number of stages (either 1 or 2),  $\rho_f$  is the propellant density in  $\text{kg/m}^3$ ,  $t_{mis}$  is the surface time of the lunar mission in days,  $n_{crew}$  is the number of crew,  $c_{misc}$  is the miscellaneous mass fraction. The miscellaneous mass fraction  $c_{misc}$  represents how much of the dry mass is categorized as the miscellaneous mass. It can range from 0 to 0.15, meaning 0% to 15% of the dry mass is the miscellaneous mass. The higher  $c_{misc}$ , the heavier and more conservative the vehicle design becomes. All mass properties are defined in kg.

As shown in Eq. (21), the model captures the subsystem-level interactions to return the relationship between the payload capacity, propellant capacity, and dry mass of the spacecraft. Particularly, the subsystem interactions are captured *through* the dry mass. For instance, an increase in any subsystem mass will raise the dry mass. Since each subsystem mass is dependent on the dry mass, their mass should increase as well, which will further raise the dry mass. The 'balanced' dry mass with such subsystem circular references can be found by solving Eq. (21) for the dry mass,  $m_d$ . More details on this model can be found in Appendix A.

In the case study, the computational time for all problems is measured on a platform with Intel Core i7-10700 (8 Core at 2.9 GHz). In the proposed decomposition-based method, Gurobi 9.1 solver [13] is used for the initial MILP problem and MIQP subproblem, and IPOPT [14] is chosen for the NLP subproblem.

## B. Optimization Results by the Proposed Decomposition-Based Formulation

This subsection introduces the optimization results by the proposed decomposition-based formulation. Since the performance of the proposed method is dependent on the breakpoint increment (or the number of breakpoints) for the PWL approximation of the MILP-based initial solution generation, five different increments are tested. The results are shown in Table 4. Although the optimizer's computational time involves some randomness depending on the individual problems, there are several general trends that can be observed. First, when the increment is too large (too few breakpoints, e.g., 10,000 kg increment with 13 mesh points), the initial solution quality becomes poor, and thus the final solution IMLEO is also poor. Second, the computational time to solve the initial MILP problem rapidly increases when the increment is too small (too many breakpoints, e.g., 625 kg increment with 1,595 mesh points), resulting in a long total computational time. In summary, we can observe the expected trend that a smaller increment (more breakpoints) leads to a better initial guess at the cost of computational time. Thus, the most efficient strategy is to use an increment that can generate a reasonably accurate initial solution and leave the rest to the proposed decomposition-based optimization. Although this hyperparameter needs to be chosen for the proposed algorithm, it is also worth noting that the computational performance is not very sensitive against the choice of its value except for the extreme cases. Also, note that, theoretically speaking, if we reduce the increment to zero (an infinite number of breakpoints), the solution would match with the global optimum; however, this is impractical as it requires infinite computational time. The proposed decomposition-based formulation can take the reasonable approximate solution by PWL formulation and offer a better computational efficiency to achieve a high-quality solution.

**Table 4 Optimization results by the proposed decomposition-based formulation**

Breakpoint increment for PWL, kg	10,000	5,000	2,500	1,250	625
Number of mesh points for PWL	13	36	120	425	1,595
Initial IMLEO via PWL, kg	741,115	700,684	677,035	677,343	677,315
Initial solution generation time, s	5.617	4.277	3.620	23.83	495.9
Decomposition-based optimization time, s	28.34	19.53	12.65	12.91	12.94
Solution IMLEO, kg	727,558	695,271	676,930	677,203	677,179
Total optimization time, s	33.96	23.81	16.27	36.75	508.8

### C. Benchmarking with State-of-the-Art Method: Embedded Optimization

Although our formulation of the integrated mission planning and subsystem-level spacecraft design has not been directly tackled in the literature, we can extend straightforwardly a state-of-the-art approach for a similar problem as a benchmark to evaluate our newly proposed method.

The identified state-of-the-art approach is the embedded optimization method by Taylor [10], which was demonstrated to be more efficient than directly solving the original integrated MINLP problem using a global optimizer. With the embedded optimization method, the spacecraft variables are separated from the whole problem and determined by a metaheuristics algorithm. At every iteration, the metaheuristics algorithm picks the payload and propellant capacity of  $N$  vehicles, and the corresponding spacecraft dry mass is then calculated as a function of them, following the spacecraft subproblem procedure. After obtaining the feasible vehicle design, these values are fed to the space mission planning problem, which is solved by the MILP optimizer. Unlike the all-in-one formulation, the vehicle parameters are fixed within the space mission planning part. Then, the corresponding objective function value is returned to the metaheuristic optimizer for the evaluation for the next iteration. As a result, the metaheuristics only handles an optimization problem with  $2N$  variables (i.e., the payload capacity and propellant capacity for each spacecraft), where the evaluation of the constraints and the determination of the remaining variables are handled by the embedded MILP solver. The problem to be optimized by the metaheuristic solver is expressed as Eq. (22).

$$\min_{\alpha} \text{IMLEO}(\alpha, \mathcal{F}(\alpha)) \quad (22)$$

where  $\alpha = [m_{p_1}, m_{f_1}, \dots, m_{p_N}, m_{f_N}]$ ,  $\alpha \in \mathbb{R}^{2N}$

Since the performance of the embedded optimization would depend on the choice of the metaheuristics algorithm, three different metaheuristics algorithms are tested: the extended Ant Colony Optimization (ACO) [21], the Genetic Algorithm (GA) [22], and the Particle Swarm Optimization (PSO) [23]. The optimization is terminated when a predefined number of generations are populated; different termination generation numbers are tested for each algorithm to explore the tradeoff between the computational time and accuracy. Furthermore, due to the random nature of the metaheuristic optimizers, the optimization is run three times with the same algorithm and generation number.

Table 5 shows each algorithm's best results with 10, 50, and 100 generations. Note that 'inf' indicates that no feasible solutions can be found. The complete set of results is given in Table 7 in Appendix. In many cases, especially with low numbers of generations, the optimizers fail to even reach a feasible IMLEO solution. As the number of generations increases, the computational time increases, a feasible solution is more likely to be found, and the solution tends to be better, although such trends might not always hold due to the random nature of the metaheuristic algorithms.

**Table 5 Results of the state-of-the-art embedded optimization**

Algorithm	Ant Colony Optimization			Genetic Algorithm			Particle Swarm Optimization		
Number of generations	10	50	100	10	50	100	10	50	100
Best Solution IMLEO, kg	723,090	747,398	689,287	764,301	746,702	728,831	inf	677,659	677,221
Optimization Time, s	85.03	391.7	776.0	86.18	413.6	836.6	84.69	397.9	793.9

#### D. Performance Analysis and Discussions

As we compare the optimization results by the proposed decomposition-based optimization method in Table 4 and the state-of-the-art embedded optimization method in Table 5, it is clear that the proposed method can achieve a better solution (lower IMLEO) at a less computational time. Even with respect to the best embedded optimization case in Table 5, which is the PSO case with 100 generations (IMLEO 677,221kg, computational time 793.9s), the proposed decomposition-based optimization can achieve a better solution with a substantially shorter computational time (676,930kg, computational time 16.27s). Note that the computational time by the two methods is measured without any parallelization under a fair setting.

Beyond the numerical solution comparison, one substantial advantage of the proposed method is its deterministic and thus repeatable performance. This is in contrast to the metaheuristics that returns different results every run, varying from near-optimal results to infeasible results. The proposed formulation can consistently achieve better solutions than those that the metaheuristics optimizers would find "by chance."

Another advantage of the proposed formulation is that more complicated spacecraft design problems, such as models with more constraints or even MDO problems, can be integrated in a scalable way. Namely, if more subproblems are considered, they can be parallelized to further reduce the computational time. When complex MDO problems are included as subproblems, a multi-level hierarchical ALC formulation can also be utilized [18].

Overall, the case study demonstrates that the higher computational performance of the proposed method compared to the state-of-the-art embedded optimization method. The proposed formulation can consistently obtain a better solution in a shorter computational time. It also has greater room for potential improvement and extension, such as parallelization and MDO subproblem integration.

## VI. Conclusion

This paper tackles the challenging problem of integrated space mission planning and spacecraft design. The all-in-one formulation is presented as an MINLP problem, and an efficient solution approach is developed leveraging the unique structure of the problem and following the philosophy of MDO. Namely, the all-in-one MINLP problem is decomposed into the space mission planning subproblem (MIQP) and the spacecraft design subproblem(s) (NLP) so that they can be solved iteratively using the ALC approach to find the optimal solution for the original MINLP problem. Furthermore, an automatic and effective approach for finding an initial solution for this iterative process is proposed leveraging a piecewise linear approximation of the nonlinear vehicle model, so that no user-defined initial guess is needed. The case study results demonstrate that the proposed method achieves a better result in less time compared to the state-of-the-art embedded optimization method. The combination of the unique problem structure, the iterative algorithms for shared variables, and the efficient initial solution generation method leads to this computational efficiency even without parallelization. The parallelizable nature of the algorithm is expected to make the proposed method even more advantageous for large-scale problems. Due to the flexibility of the ALC method, the proposed formulation can also integrate more complex vehicle design models, which is left for future work.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1942559.

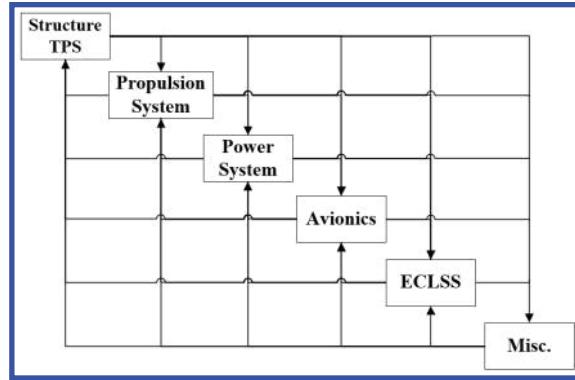
## Appendix A: Spacecraft Design Model

This appendix provides more details on the parametric sizing model for the spacecraft used in the case study. In the considered model, the subsystems of single-stage landers and their relations to the dry mass are defined as Eq. (23).

$$m_d = \sum m_{sub} = m_{str} + m_{prop} + m_{power} + m_{avi} + m_{ECLSS} + m_{misc} \quad (23)$$

where  $m_{sub}$  indicates the mass of subsystem.  $m_{str}$  indicates the structure and thermal protection system (TPS), which includes all subsystems that are attached to support or connect other components. This is not limited but includes landing legs and truss, TPS for the reentry to the earth, and docking mechanism.  $m_{prop}$  is the propulsion system, such as propellant tanks, reaction control system (RCS), and hardware of engines.  $m_{power}$  is the power system, which contains batteries, fuel cells, solar panels, or other electrical systems.  $m_{avi}$  indicates the avionics, and  $m_{ECLSS}$  indicates the environmental and life control system (ECLSS) that supports the crew's lives such as consumables (food, water, air) or related piping and tankage. Finally, we also consider other miscellaneous required components, expressed as  $m_{misc}$ .

Through the dry mass, each subsystem interacts with every other subsystem, and this relation is visualized in Fig. 4 as an  $N^2$  diagram.



**Fig. 4 Relationship of domains in a single-stage lunar lander.**

For the defined subsystems, mass estimation relationships (MERs) are developed as functions of payload capacity, propellant capacity, propellant type  $f$ , and some other known parameters. If the propellant type is fixed, the subsystems MERs and dry mass are dependent on the payload capacity and propellant capacity only, and thus serves as the vehicle sizing constraint (Eq. (5),  $m_d = \mathcal{F}(m_p, m_f)$ ). Each subsystem MER is developed by the least square curve fitting to the data from the lunar lander design database in Ref. [19, 20], which includes both existing and elaborated conceptual design. The form of each subsystem's MER is manually determined to be a sufficiently simple yet accurate form. The resultant MERs are shown in Eq. (21).

Table 6 summarizes the independent variables, the  $R^2$  values for curve fitting, number of data points used for curve fitting ( $N_{data}$ ), average errors against the data points, and the maximum errors. Note that only a small number of data points are used for the propulsion system MER since two-stage lander data are excluded as their propulsion systems with staging are too distinct from those of single-stage ones. One may also see that relatively poor correlations are obtained for the power systems and avionics mass as they simply might not be strong functions of the dry mass or vehicle size. However, since they typically account for small portions of the dry mass, the poor correlation does not have a significant effect on the validation process.

The limitation of this sizing model should also be noted. Because the MERs are developed from the existing data points, a solution for vehicles that are significantly heavier than the ones in the database would either be a low-fidelity model or infeasible. In other words,  $m_d$  that satisfies Eq. (23) might not exist for certain weight classes. Specifically, the upper bound of the dry mass is approximately 23,000 kg. When  $t_{mis}$  is 3 days,  $n_{crew}$  is 4,  $c_{misc}$  is 0.05, and the propellant is LH2/LOX, the upper bound are found at 500 kg payload and 75,500 kg propellant, or at 10,000 kg payload and 45,500 kg propellant.

**Table 6 Summary of subsystem MERs**

Subsystem	Notation	Independent Variables	$R^2$	$N_{data}$	Avg. Error	Max. Error
Structure + TPS	$m_{str}$	$m_d, n_{stg}, m_p$	0.9254	17	7.379%	24.31%
Propulsion System	$m_{prop}$	$m_d, m_p, \rho_p$	0.9279	8	7.429%	11.16%
Power System	$m_{power}$	$m_d$	0.7182	13	16.24%	36.68%
Avionics	$m_{avi}$	$m_{power}(m_d), t_{mis}$	0.6204	22	36.42%	75.94%
ECLSS	$m_{ECLSS}$	$m_d, n_{crew}, n_{stg}, t_{mis}$	0.9293	12	11.93%	38.09%
Miscellaneous	$m_{misc}$	$m_d$	-	-	-	-

## Appendix B: Summary of the Embedded Optimization Results

Table 7 includes the full results obtained from the embedded optimization.

**Table 7 Summary of the embedded optimization results**

Algorithm	Number of generations	Optimization time, s	IMLEO, kg
Ant Colony Optimization	5	45.35	820891
		45.06	inf
		45.77	681302
	10	83.19	806153
		85.03	723090
		83.81	793031
	20	161.4	714202
		161.2	801618
		162.3	710166
Genetic Algorithm	50	389.8	761514
		391.2	765182
		391.7	747398
	100	776.0	689287
		771.9	732297
		775.9	691874
	5	46.50	inf
		45.71	inf
		46.09	inf
Particle Swarm Optimization	10	85.38	inf
		86.18	764301
		83.97	inf
	20	161.9	inf
		172.4	761116
		161.7	inf
	50	394.7	inf
		413.6	746702
		394.6	inf
	100	836.6	728831
		788.3	875069
		777.7	inf
Ant Colony Optimization	5	46.54	inf
		46.57	inf
		46.26	711382
	10	84.75	inf
		84.69	inf
		85.22	inf
	20	162.5	726951
		162.1	688182
		161.6	inf
Genetic Algorithm	50	392.3	inf
		397.9	677659
		394.6	714169
	100	788.9	677754
		793.9	677221
		791.2	677316

## References

[1] Shull, S. A., “Integrated Modeling and Simulation of Lunar Exploration Campaign Logistics,” Master’s thesis, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA, 2007.

[2] Taylor, C., Song, M., and Klabjan, D., “A Mathematical Model for Interplanetary Logistics,” *Logistics Spectrum*, Vol. 41, No. 1, 2007, pp. 22–33.

►[3] Ishimatsu, T., De Weck, O. L., Hoffman, J. A., and Ohkami, Y., “Generalized Multicommodity Network Flow Model for the Earth-Moon-Mars Logistics System,” *Journal of Spacecraft and Rockets*, Vol. 53, No. 1, 2016, pp. 25–38. <https://doi.org/10.2514/1.A33235>.

[4] Ho, K., De Weck, O. L., Hoffman, J. A., and Shishko, R., “Dynamic Modeling and Optimization for Space Logistics Using Time-Expanded Networks,” *Acta Astronautica*, Vol. 105, No. 2, 2014, pp. 428–443. <https://doi.org/10.1016/j.actaastro.2014.10.026>.

[5] Ho, K., De Weck, O. L., Hoffman, J. A., and Shishko, R., “Campaign-Level Dynamic Network Modelling for Spaceflight Logistics for the Flexible Path Concept,” *Acta Astronautica*, Vol. 123, 2016, pp. 51–61. <https://doi.org/10.1016/j.actaastro.2016.03.006>.

►[6] Chen, H., and Ho, K., “Integrated Space Logistics Mission Planning and Spacecraft Design with Mixed-Integer Nonlinear Programming,” *Journal of Spacecraft and Rockets*, Vol. 55, No. 2, 2018, pp. 365–381. <https://doi.org/10.2514/1.A33905>.

►[7] Chen, H., Lee, H. W., and Ho, K., “Space Transportation System and Mission Planning for Regular Interplanetary Missions,” *Journal of Spacecraft and Rockets*, Vol. 56, No. 2, 2019, pp. 12–20. <https://doi.org/10.2514/1.A34168>.

[8] Chen, H., Sarton du Jonchay, T., Hou, L., and Ho, K., “Integrated In-Situ Resource Utilization System Design and Logistics for Mars Exploration,” *Acta Astronautica*, Vol. 170, 2019, pp. 80–92. <https://doi.org/10.1016/j.actaastro.2020.01.031>.

[9] Takubo, Y., Chen, H., and Ho, K., “Hierarchical Reinforcement Learning Framework for Stochastic Spaceflight Campaign Design,” *Journal of Spacecraft and rockets*, 2021. (Accepted).

[10] Taylor, C., “Integrated Transportation System Design Optimization,” Ph.D. thesis, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA, 2007.

[11] Sobiesczanski-Sobieski, J., and Haftka, R. T., “Multidisciplinary aerospace design optimization: survey of recent developments,” *Structural optimization*, Vol. 14, No. 1, 1997, pp. 1–23. <https://doi.org/10.1007/BF01197554>.

[12] Beauregard, A., Urbano, A., S, L.-D., and Morlier, J., “Multidisciplinary Design and Architecture Optimization of a Reusable Lunar Lander,” *Journal of Spacecraft and Rockets*, 2021, pp. 1–14. <https://doi.org/10.2514/1.A34833>.

[13] Gurobi Optimization, LLC, “Gurobi Optimizer Reference Manual,” , 2021. URL <https://www.gurobi.com>.

[14] Wächter, A., and Biegler, L. T., “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57. <https://doi.org/10.1007/s10107-004-0559-y>.

►[15] Martins, J. R. R. A., and Lambe, A. B., “Multidisciplinary Design Optimization: A Survey of Architectures,” *AIAA Journal*, Vol. 51, No. 9, 2013, pp. 2049–2075. <https://doi.org/10.2514/1.J051895>.

[16] Tosserams, S., Etman, L., and Rooda, J., “An augmented Lagrangian decomposition method for quasi-separable problems in MDO,” *Structural and Multidisciplinary Optimization*, Vol. 34, No. 3, 2007, pp. 211–217. <https://doi.org/10.1007/s00158-006-0077-z>.

[17] Kim, H. M., Rideout, D. G., Papalambros, P. Y., and Stein, J. L., “Analytical Target Cascading in Automotive Vehicle Design ,” *Journal of Mechanical Design*, Vol. 125, No. 3, 2003, pp. 481–489. <https://doi.org/10.1115/1.1586308>.

[18] Tosserams, S., Etman, L. F. P., and Rooda, J. E., “Augmented Lagrangian coordination for distributed optimal design in MDO,” *International Journal for Numerical Methods in Engineering*, Vol. 73, No. 13, 2008, pp. 1885–1910. <https://doi.org/10.1002/nme.2158>.

[19] Isaji, M., Maynard, I., and Chudoba, B., “Surface Access Architecture Modeling: Trend Analysis and Classification from a Lander Database,” *AIAA SPACE and Astronautics Forum and Exposition 2018*, AIAA, 2018. <https://doi.org/10.2514/6.2018-5131>.

[20] Isaji, M., Maynard, I., and Chudoba, B., “A New Sizing Methodology for Lunar Surface Access Systems,” *AIAA Scitech 2020 Forum*, AIAA, 2020. <https://doi.org/10.2514/6.2020-1775>.

- [21] Schlüter, M., Egea, J. A., and Banga, J. R., “Extended ant colony optimization for non-convex mixed integer nonlinear programming,” *Computers & Operations Research*, Vol. 36, No. 7, 2009, pp. 2217–2229. <https://doi.org/https://doi.org/10.1016/j.cor.2008.08.015>.
- [22] Holland, J. H., “Genetic Algorithms,” *Scientific American*, Vol. 267, No. 1, 1992, pp. 66–73. URL <http://www.jstor.org/stable/24939139>.
- [23] Kennedy, J., and Eberhart, R., “Particle swarm optimization,” *Proceedings of ICNN'95 - International Conference on Neural Networks*, Vol. 4, 1995, pp. 1942–1948 vol.4. <https://doi.org/10.1109/ICNN.1995.488968>.