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 problem based on decomposition-based optimization via augmented Lagrangian coordination. 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 a user-defined initial guess. The demonstration case study shows that, compared to the state-of-the-art method, the proposed formulation converges substantially faster and the converged solution is also at least the same or better given the same computational time limit.

I. Nomenclature

$\mathcal{A}$	=	Set of arcs
$\boldsymbol{a}_{vijt}$	=	Cost coefficient vector of commodity
a'_{vijt}	=	Cost coefficient of spacecraft
C_c	=	set of continuous commodity flow variables
C_d	=	set of discrete commodity flow variables
$\boldsymbol{d}_{it}$	=	Demand vector
$\boldsymbol{e}_v$	=	Spacecraft design variable vector

[*] This paper is a revised version of Paper 2021-4069 at the AIAA ASCEND 2021, Las Vegas, Nevada & Virtual, November 15-17, 2021.
[†] 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.

$\mathcal{F}(-)$	=	Spacecraft sizing function
f	=	Objective function (subproblem)
$\boldsymbol{g}$	=	Inequality constraint
$\boldsymbol{h}$	=	Equality constraint
H_{vij}	=	Concurrency matrix
$\mathcal{J}$	=	Objective function
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
n	=	Dimension of variables
Q_{vijt}	=	Commodity transformation matrix
$\mathcal{T}$	=	Set of time steps
t_{mis}	=	Mission length
Δt_{ij}	=	Time of Flight (ToF)
u_{vijt}	=	Spacecraft flow variable
$\mathcal{V}$	=	Set of spacecraft
W_{ij}	=	Launch time window
$\boldsymbol{x}_{vijt}$	=	Commodity flow variable
$\boldsymbol{y}$	=	Shared variables
$\boldsymbol{z}$	=	Local variables
ζ	=	Propellant type
ϕ	=	Penalty function

Subscipt & Superscript

i	=	Node index (departure)
j	=	Node index (arrival)
k	=	Subproblem index

l	=	Subsystem index
q	=	Iteration count
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 incorporate 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. Even though this integration has been shown to exhibit substantial solution improvement compared to optimizing them individually [10], solving the MINLP 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 the piecewise linear (PWL) approximation of the nonlinear model to approximate the entire MINLP problem as a MILP problem that can be solved efficiently. However, this approach is an approximation model, and the resulting solution is not guaranteed to be feasible nor optimal in the original nonlinear problem. For a different yet related problem, satellite component selection and operation have been co-optimized as a mixed-integer programming problem [11, 12]; however, the existing techniques are based on the specific types of nonlinear relationships in the problem (e.g., power laws) and are not generally applicable to the integrated logistics 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 [13], 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. [14] 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, 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 [15] for MIQP; IPOPT [16] 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 [13, 17]. Specifically, the Augmented Lagrangian Coordination (ALC) approach [18] with the Analytical Target Cascading (ATC) structure [19, 20] 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 demonstration case study shows that, compared to the state-of-the-art method, the proposed new formulation converges substantially faster and the converged solution is also at least the same or better given the same computational time limit.

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 against the state-of-the-art 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.

$$\min \quad \mathcal{J} = \sum_{t \in \mathcal{T}} \sum_{(v,i,j) \in \mathcal{A}} \left(\boldsymbol{a}_{vijt}^T \boldsymbol{x}_{vijt} + a'^T_{vijt} m_{d_v} u_{vijt} \right) \tag{1}$$

subject to

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

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

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

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

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

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

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

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

Name	Description
Variables	
$\boldsymbol{x}_{vijt}$	Commodity flow variable, or the quantity of the commodity delivered from node i to j at time t by spacecraft v. $\boldsymbol{x}_{vijt} \geq 0$. Each component of this variable can contain either continuous variables (C_c) or discrete variables (C_d). This vector will be $\mathbb{R}^p$ 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 ($\in \mathbb{R}$). This variable is integer scalar.
$\boldsymbol{e}_v$	Spacecraft design variables and parameters. In this problem, it includes payload capacity m_p and propellant capacity m_f ($\in \mathbb{R}^2$).
m_{d_v}	Dry mass of spacecraft v ($\in \mathbb{R}$).
Parameters	
$\boldsymbol{a}_{vijt}$	Cost coefficient of commodity ($\in \mathbb{R}^p$).
a'_{vijt}	Cost coefficient of spacecraft ($\in \mathbb{R}$).
$\boldsymbol{d}_{it}$	Demands/supplies of different commodities and spacecraft at node i at time t ($\in \mathbb{R}^{p+1}$).
Q_{vijt}	Transformation matrix ($\in \mathbb{R}^{(p+1)\times(p+1)}$).
H_{vij}	Concurrency constraint matrix ($\in \mathbb{R}^{2\times p}$).
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.
Δt_{ij}	Time of Flight (ToF) from node i to j.
ζ_v	Propellant type for each spacecraft (predetermined).
Sets	
$\mathcal{A}(\mathcal{V}, \mathcal{N}, \mathcal{N}, \mathcal{T})$	Set of arcs realized by spacecraft.
C_c	Set of continuous commodity flow variables.
C_d	Set of discrete commodity flow variables.
$\mathcal{N}$	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 $\boldsymbol{a}_{vijt}$ and a_{vijt} so that the objective function corresponds to the sum of initial mass at low-earth orbit (IMLEO), a metric commonly used in the space logistics literature as can be seen in Ref. [3, 4, 6, 10].

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. It 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.

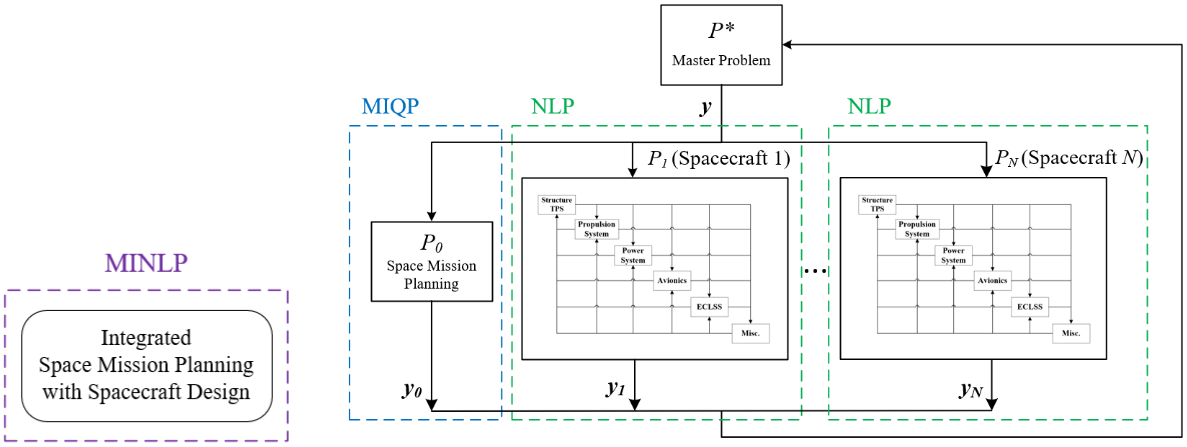
Along with Table 1, Eqs. (6), (7), and (8) show the definitions and domains of commodity flow variables, spacecraft flow variables, and spacecraft design variables, respectively.

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. In particular, the discrete variables represent the definition of the commodity flow and the number of spacecraft on the mission planning side of the problem. 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 $\boldsymbol{e}_v$ 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. Exploiting 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.



(a) All-in-one formulation **(b) Proposed decomposition-based formulation based on [18]**

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. [18].

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

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

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

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, $\boldsymbol{g}_k$ and $\boldsymbol{h}_k$, are only dependent on the auxiliary variables $\boldsymbol{y}_k$ and independent of the shared variables $\boldsymbol{y}$.

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

With the consistency constraints $\boldsymbol{c}_k$, which ensures that the auxiliary variables $\boldsymbol{y}_k$ are the same as the shared variables $\boldsymbol{y}$, 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:

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

The augmented Lagrangian penalty function for subproblem k, ϕ_k, is defined as follows.

$$\phi_k\left(\boldsymbol{c}_k\left(\boldsymbol{y}, \boldsymbol{y}_k\right)\right) = \boldsymbol{v}_k^T\left(\boldsymbol{y} - \boldsymbol{y}_k\right) + \left\|\boldsymbol{w}_k \circ \left(\boldsymbol{y} - \boldsymbol{y}_k\right)\right\|_2^2 \tag{12}$$

where $\boldsymbol{v_k}$ is the vector of Lagrange multiplier estimates, and $\boldsymbol{w_k}$ 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 $\boldsymbol{y}$. Note that even though the bi-level formulation is employed here, the ALC has the capability to handle multi-level hierarchical formulation as well [20].

(1) Master Problem

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

(2) Subproblem k

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

Adopting the above approach in 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 $1, \ldots, N$), 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 $\boldsymbol{y} = [\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_N^T]^T$, where $\boldsymbol{y}_v = [m_{p_v}, m_{f_v}, m_{d_v}]^T$, where v is the vehicle index such that $v = 1, \ldots, N$. Note that, in the considered problem, the subproblem index k and vehicle index v match (i.e., subproblem k handles the design problem of spacecraft v). For each spacecraft v, m_{p_v}, m_{f_v}, m_{d_v} respectively represent the payload capacity, propellant (fuel) capacity, and dry mass.

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.

$$
\begin{aligned}
\min_{\boldsymbol{x}_{vijt}, u_{vijt}, \boldsymbol{y}_0} \quad & \sum_{t \in \mathcal{T}} \sum_{(v,i,j) \in \mathcal{A}} \left(\boldsymbol{a}_{vijt}^T \boldsymbol{x}_{vijt} + a'^T_{vijt} m_{d_v} u_{vijt} \right) + \phi_0 \left(\boldsymbol{c}_0 \left(\boldsymbol{y}, \boldsymbol{y}_0 \right) \right) \\
\text{subject to} \quad & \text{Eqs. (2)–(4) and (6)–(8)} \\
\text{where} \quad & \boldsymbol{y} = [\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_N^T]^T \quad \text{and} \quad \boldsymbol{y}_v = [m_{p_v}, m_{f_v}, m_{d_v}]^T
\end{aligned}
\tag{15}
$$

In 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. Note that this subproblem does not optimize the spacecraft design against its own cost or mass. As the objective of the original problem is to minimize IMLEO, the purpose here is to find a feasible spacecraft design minimizing the IMLEO value. The spacecraft design 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.

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

B. Solution Algorithm and Iteration Scheme

This subsection introduces the iterative solution algorithm for the decomposition-based algorithm introduced in Section IV.A and Ref. [18]. The formulated decomposed optimization problems with ALC can be solved iteratively in two loops: the outer loop updates the augmented Lagrangian penalty parameters ($\boldsymbol{v}$, $\boldsymbol{w}$), while the inner loop solves the master problem and subproblems until the change in the subproblem objective function values is less than a defined tolerance. The outer loop terminates when two conditions are met simultaneously. First, the maximum consistency violation must be less than a specified tolerance. It ensures that the solution is feasible in each subproblem within the tolerance. Second, the maximum consistency violation change from the previous iteration must be smaller than the same tolerance. It allows for design space exploration and prevents premature convergences. The tolerance value of 10^{-3} is used in this study. The penalty parameters are updated in the outer loop based on the inner loop solution's consistency violation. Specifically, at q-th iteration, $\boldsymbol{v} = \left[\boldsymbol{v}_{\boldsymbol{0}}^T, \ldots, \boldsymbol{v}_{M-1}^T\right]^T$ is updated as follows:

$$\boldsymbol{v}^{q+1} = \boldsymbol{v}^q + 2\boldsymbol{w}^q \circ \boldsymbol{w}^q \circ \boldsymbol{c}^q \tag{17}$$

The r-th element w_r of the penalty weight vector $\boldsymbol{w} = \left[\boldsymbol{w}_{\boldsymbol{0}}^T, \ldots, \boldsymbol{w}_{M-1}^T\right]^T$ is updated based on the corresponding consistency constraint element c_r as follows:

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

where $\gamma_1 > 1$ and $0 < \gamma_2 < 1$. In this study, $\gamma_1 = 2$ and $\gamma_2 = 0.5$ are used. The initial penalty parameter values can take $\boldsymbol{v}^1 = \boldsymbol{0}$ and $\boldsymbol{w}^1 \approx \boldsymbol{1}$.

The updates for the inner loop are 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.

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

For our problem, we make an additional heuristics-based modification to the master problem to facilitate the convergence. 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 $\boldsymbol{y}$ into the regular shared variables $\boldsymbol{\alpha} = [m_{p_1}, m_{f_1}, \ldots, m_{p_N}, m_{f_N}]$ and the prioritized shared variables $\boldsymbol{\beta} = [m_{d_1}, \ldots, m_{d_N}]$ (i.e.,

$\boldsymbol{y} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$), and only $\boldsymbol{\alpha}$ is updated in the master problem.

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

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 β_0 is not used in the entire optimization architecture but only to facilitate the convergence of the whole optimization problem.

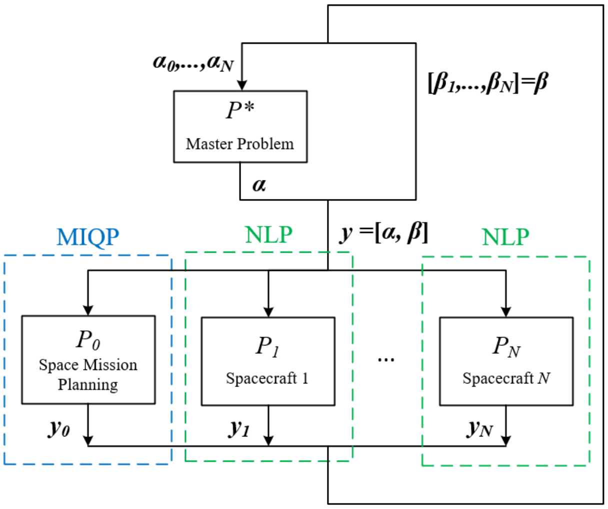


Fig. 2 Proposed decomposition-based optimization architecture with prioritized shared variables.

C. Automatic Initial Solution Generation via Piecewise Linear Approximation

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 approximate the nonlinear spacecraft model as a PWL model with chosen breakpoints so that an approximate initial solution to the original MINLP problem can be obtained using MILP [6]. Since MILP problems can be solved using a specialized solver, this approximate solution can be generated efficiently. 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. 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 in the spacecraft sizing constraint. To generate the PWL functions of the nonlinear constraints, we choose a series of equally-spaced "mesh" points over the feasible ranges of the independent spacecraft design variables and use them as breakpoints for the PWL function generation. Since the dry mass is an (implicit) function of the payload capacity and propellant capacity, we only use 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. The method used in this study to model PWL functions as MILP problems is presented in Ref. [21].

V. Case Study: Human Lunar Exploration Campaign

To demonstrate the effectiveness of the proposed approach, we perform several case study instances 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

1. Specifications of Human Lunar Exploration Campaign

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. We call this case study the default instance, and seven additional instances with different parameters are also studied. Their specifications are given in Table 4. The parameters not listed in Table 4 are the same as those in the default instance (Instance 1). In some instances, the supply and demand (i.e., mission payload) are increased, as shown in Table 3. Note that sizing models for only one kind of spacecraft, which is a single-stage lunar lander, are considered for simplicity. Hence, the designed lander is also used as other kinds of spacecraft, 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. For this reason, a so-called *aggressive* vehicle model with a lower structural mass estimation is used in some instances. In contrast, we refer to the original model as the conservative model. The sizing models are discussed in detail in the next section (V.A.2). For each case study instance, the computational time 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 [15] is used for the initial MILP problem and MIQP subproblem, and IPOPT [16] is chosen for the NLP subproblems.

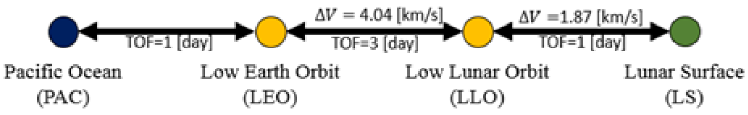


Fig. 3 Lunar campaign network model [6].

Table 2 Parameters used in the default instance (Instance 1) of the case study

Parameters	Assumed values
Spacecraft Propellant type	LH2/LOX
Propellant I_{sp}, s	420
Propellant density ρ_f, kg/m^3	360
Spacecraft miscellaneous mass fraction c_{misc} (see Eq. (21))	0.05
Spacecraft payload capacity range, kg	[500, 10000]
Spacecraft propellant capacity range, kg	[1000, 100000]
Type(s) 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	Date	Supply/Demand (increased instance)
Outbound to the Moon			
Crew	Earth	0, 365	4
Habitat, Equipment, and Propellant, kg	Earth	0, 365	∞
Crew	Moon	5, 370	-4
Habitat & Equipment, kg	Moon	5, 370	-2000 (-3000)
Inbound to the Earth			
Crew	Moon	8, 373	4
Returned mass, kg	Moon	8, 373	1000 (1500)
Crew	Earth	13, 378	-4
Returned mass, kg	Earth	13, 378	-1000 (-1500)

Table 4 Case study instances specifications

Instance index	Type(s) of SC desinged	Number of SC per type	SC sizing model	Mission payload
1 (default)	1	6	conservative	default
2	2	3	conservative	default
3	6	1	conservative	default
4	2	3	conservative	increased
5	1	6	aggressive	default
6	2	3	aggressive	default
7	6	1	aggressive	default
8	2	3	aggressive	increased

2. Spacecraft Sizing Model

The subsystem-level spacecraft model used as the spacecraft design constraint in Eq. (5) is developed by the least square curve fitting to the data from the lunar lander design database in Ref. [22, 23]. 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

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

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), ρ_f is the propellant density in 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.

The aforementioned aggressive sizing model has a lower structural mass estimation shown in Eq. (22). The only difference from Eq. (21) is the lowered coefficient of m_d, and the other subsystem mass relations remain the same. In the ranges of payload capacity and propellant capacity specified in Table 2, both models are non-convex.

$$m_{str} = {n_{stg}}^{-0.6705} \, (0.2694 \, m_d + 693.7 \, {m_p}^{0.04590}) \tag{22}$$

B. Optimization Results of the Proposed Decomposition-Based Formulation

This subsection introduces the optimization results of 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 for Instance 1 *. The results are shown in Table 5; note that the PWL-based initial solutions are not necessarily feasible, and so their IMLEOs are only reported for information purposes.

Table 5 Instance 1 optimization results by the proposed decomposition-based formulation

PWL increment, kg	PWL mesh points	Optimization time, s			IMLEO, kg	
		PLW-based initial solution	Decomposition-based iterations	Total	PWL-based initial solution	Final solution
10,000	13	3.461	20.02	23.48	741,115	724,776
5,000	36	4.769	13.47	18.24	700,684	694,224
2,500	120	4.754	11.98	16.73	677,035	676,862
1,250	425	28.09	13.83	41.92	677,343	677,204
625	1,595	465.5	13.18	478.7	677,315	677,072

Although the optimizer's computational time involves some randomness depending on the individual problems, some general trends 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 is 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. 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. 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 worth noting that the computational performance is not very sensitive to the choice of its value except for the extreme cases. In our case study and spacecraft sizing model, the 2,500 kg increment is chosen as the default increment for the remainder of the case study.

With the chosen default increment, the other seven instances are also tested. The results are summarized in Table 6.

*We have tested the integrated formulation (as a black box) with MINLP solvers including Outer Approximation [24], LP/NLP based Branch-and-Bound [25], and Extended Cutting Plane [26]; however, they failed to produce a feasible solution due to the non-convex sizing constraints.

In all instances, the proposed method can solve the problem with a reasonable amount of computational time and solution improvement from the initial guess. Note that the final IMLEO is not necessarily lower than that of the PWL-based initial solution because the original nonlinear problem might be infeasible with the initial guess spacecraft design (i.e., $y_v^0 = [m_{p_v}^0, m_{f_v}^0, \mathcal{F}(m_{p_v}^0, m_{f_v}^0)]^T$ for v-th vehicle).

Table 6 Decomposition-based optimization results with 2,500 kg increment for all instances

Instance	Optimization time, s			IMLEO, kg	
	PWL-based initial solution	Decomposition-based iterations	Total	PWL-based initial solution	Final solution
1 (default)	4.754	11.98	16.73	677,035	676,862
2	38.66	10.70	49.35	401,191	401,093
3	38.66	61.57	100.2	386,257	387,535
4	10.53	22.63	33.16	470,788	470,406
5	75.35	94.06	154.7	442,596	442,605
6	24.09	14.30	38.39	293,095	293,095
7	100.1	44.02	144.1	286,154	302,041
8	31.50	14.47	45.97	344,397	344,423

Overall, the proposed decomposition-based formulation can take the reasonable approximate solution by the PWL formulation and offer a better computational efficiency to achieve a high-quality solution.

C. Benchmark Formulation: Modified Embedded Optimization

While our formulation of the integrated mission planning and subsystem-level spacecraft design has not been directly tackled in the literature, we can extend a state-of-the-art approach for a similar problem straightforwardly as a benchmark to evaluate our newly proposed method. The identified 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. In this paper, the method is modified to solve the case study problem more efficiently.

In the original formulation of the embedded optimization, all spacecraft variables are separated from the problem and determined by a metaheuristics algorithm. However, in our problem, the spacecraft dry mass is constrained and can be uniquely determined by the payload and propellant capacity. Hence, we let the metaheuristics algorithm pick only the payload and propellant capacity of N types of vehicles, and calculate the corresponding spacecraft dry mass by the spacecraft sizing constraint. This process reduces the number of variables from $3N$ to $2N$ and turns the constrained problem into an unconstrained one. After obtaining the feasible vehicle design, these values are fed to the space mission planning problem, which is solved by the specialized MILP optimizer. Unlike the all-in-one formulation, the vehicle parameters are fixed within the mission planning part. Then, the corresponding objective function value (i.e., IMLEO) is returned to the metaheuristic optimizer for the evaluation for the next iteration. As a result, the metaheuristics only handles an unconstrained optimization problem with $2N$ variables (i.e., the payload capacity and propellant capacity for N spacecraft), where the evaluation of the constraints and the determination of the remaining variables are handled by the embedded MILP solver. When the problem is infeasible with the chosen spacecraft design, a predetermined large positive number is returned as the objective function value, following the death-penalty constraint handling [27]; this method is chosen because the large number of constraints in the problem makes other methods (e.g., repair-based methods) challenging or inapplicable. The problem to be optimized by the metaheuristic solver is expressed as Eq. (23).

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \text{IMLEO}(\boldsymbol{\alpha}, \mathcal{F}(\boldsymbol{\alpha})) \\ \text{where} \quad & \boldsymbol{\alpha} = [m_{p_1}, m_{f_1}, \ldots, m_{p_N}, m_{f_N}], \quad \boldsymbol{\alpha} \in \mathbb{R}^{2N} \end{aligned} \tag{23}$$

In accordance with the proposed decomposition-based method, the PWL MILP initial guess is given to the initial population. Since such an initial guess is not provided in the original method, this modification should lead to performance improvement. For the same reason as the proposed method, a reasonable increment for the PWL approximation is desired. The effect of the PWL increment value is examined (see Section V.D.1).

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) [28], the Genetic Algorithm (GA) [29], and the Particle Swarm Optimization (PSO) [30]. The population size is 10 for all, and other specific parameters to each algorithm are provided in Appendix B. The optimization is terminated when a predefined number of generations are populated; progress at different generation numbers are examined for each algorithm to explore the tradeoff between the computational time and accuracy. Due to the random nature of the metaheuristic optimizers, the optimization is run 10 times with the same setting.

D. Performance Comparison of the Proposed Formulation with Modified Embedded Optimization

In this section, the proposed formulation results are compared with the modified embedded optimization results, which depend on several factors, including the PWL increment (i.e., initial guess quality), algorithm, and problem to be solved. We first compare their performance with various increments and algorithms, then study the results of all case study instances. A set of sample results are shown in this section to discuss the findings; the complete dataset for every instance and algorithm is presented in Appendix B.

1. Performance Comparison with Various PWL Increments

First, we examine the results for the default instance with different increments. Table 7 shows how the three metaheuristics algorithms compare with the proposed method at 10 generations for various PWL increments. A generation number as low as 10 is used to match the time scale of the proposed method. Note that the optimization time in Table 7 does not include the initial guess generation time because it is the same as long as the increment is the same.

Table 7 Instance 1 results comparison with different PWL increments at 10 generations

PWL increment, kg	Algorithm	Average opt. time, s	Best IMLEO, kg	Worst IMLEO, kg
10,000	ACO	66.93	707,882	724,776
	GA	72.81	684,755	724,776
	PSO	69.06	696,625	724,776
	Proposed	20.02	724,776	-
5,000	ACO	65.99	694,224	694,224
	GA	71.27	681,842	694,224
	PSO	67.98	694,224	694,224
	Proposed	13.47	694,224	-
2,500	ACO	72.27	705,550	infeasible
	GA	80.58	759,326	infeasible
	PSO	74.57	700,227	infeasible
	Proposed	11.98	676,862	-
1,250	ACO	65.93	677,303	677,303
	GA	70.16	677,303	677,303
	PSO	69.00	677,303	677,303
	Proposed	13.83	677,204	-
625	ACO	67.21	677,283	677,283
	GA	69.42	677,283	677,283
	PSO	68.33	677,283	677,283
	Proposed	13.18	677,072	-

There are several key findings. Most importantly, the proposed method converges substantially faster than any metaheuristics algorithms, even if the number of generations for metaheuristics is limited to 10. In most cases, the converged solution of the proposed method is better than even the best case (out of the 10 runs) for the modified embedded optimization. Note that it is not surprising that metaheuristics can occasionally perform better than the proposed method because it can stochastically escape from the local minima; however, it is worth stressing that, due to the stochasticity, the performance of the embedded optimization can vary significantly and possibly return infeasible solution in the worst case; this is true for the increment of 2,500 kg, where the initial PWL-based solution is not feasible and the embedded optimization is not able to lead it to a feasible one in the worst case of the 10 runs. In contrast, the performance of the proposed method is robust and repeatable due to its deterministic nature; it can find a nonlinear (local) solution regardless of the initial guess feasibility even though it might not always improve significantly from the initial guess.

2. Performance Comparison in Different Problem Instances

Next, we select PSO as our default algorithm and compare the results for all eight instances. The results by PSO for all instances are shown in Table 8. A similar trend is observed for all cases: the results at 10 generations suggest that there is a robust trend that the proposed method can achieve a better solution given a similar amount of time. In fact, note that in all cases, the computational time of the proposed method is substantially shorter than any other methods.

It is of interest to observe the performance of the metaheuristic methods with a higher number of generations. Thus, the 100-generation case is also shown in Table 8. In this case, we can see that there are (rare) cases where PSO can find a better solution than the proposed method in the best case (e.g., Instance 7); however, as mentioned previously, the stochastic nature of the metaheuristics algorithms makes their performance highly variable; for all cases tested, the proposed method reached the same or better solutions than those from the worst-case metaheuristics method even after 100 generations and with a substantially shorter computational time.

E. Advantages and Limitations of the Proposed Formulation

One advantage of the proposed method is its potential to integrate more complicated spacecraft design problems, such as models with more constraints or even MDO problems. Oftentimes, MDO problems require external and

Table 8 PSO embedded opt. results comparison: 2,500 kg PWL increment, all instances, 10 and 100 generations

Instance	Algorithm	Number of generations	Average opt. time, s	Best IMLEO, kg	Worst IMLEO, kg
1 (default)	PSO	10	74.57	700,227	infeasible
		100	765.3	680,234	710,650
	Proposed	-	11.98	676,862	-
2	PSO	10	84.70	436,363	infeasible
		100	881.6	402,976	412,622
	Proposed	-	10.70	401,093	-
3	PSO	10	153.0	423,873	infeasible
		100	1620	408,388	421,829
	Proposed	-	61.57	387,535	-
4	PSO	10	82.52	470,538	470,538
		100	842.3	470,538	470,538
	Proposed	-	22.63	470,406	-
5	PSO	10	75.36	451,269	470,869
		100	748.6	445,230	450,841
	Proposed	-	94.06	442,605	-
6	PSO	10	96.64	305,457	337,105
		100	949.7	298,414	306,893
	Proposed	-	14.30	293,095	-
7	PSO	10	172.3	302,791	358,768
		100	1772	298,361	310,755
	Proposed	-	44.02	302,041	-
8	PSO	10	94.21	355,767	395,748
		100	934.3	348,585	359,040
	Proposed	-	14.47	344,423	-

black-box analysis tools for specific disciplines where the analytical formulation or constraints might not be known by the user. As the ALC formulation is intended for such MDO problems, integration of disciplinary analysis tools should not be an issue. The method we employ to model PWL functions as MILP problems can handle black-box functions as well [21]. Furthermore, if a hierarchical MDO subproblem is considered, a multi-level hierarchical variation of the ALC formulation can be utilized [20]. When there are a large number of subproblems involved, parallel computation might further reduce the computational cost.

The proposed method has several limitations as well, which can lead to future work. First, the objective function of the space mission planning problem must be linear or quadratic. Otherwise, the subproblem itself becomes a MINLP problem and negates the advantage of the proposed decomposition formulation. This limitation excludes general nonlinear objective functions. In a similar manner, the spacecraft design constraints can only have continuous variables although integer variables such as the number of engines and number of propellant tanks may be needed for high-fidelity spacecraft design. One approach to these cases is to revise the decomposition structure of the problem, which is left for future work. Finally, even though the proposed PWL solution can provide a near-optimal initial guess, the gradient-based NLP solver in the proposed method can still lead to a local optimum. Thus, embedded optimization may be preferred in certain contexts when the goal is to achieve a global optimum without any constraint on the available computational resources.

Overall, the case study demonstrates the high computational performance of the proposed method. The proposed formulation can achieve a high-quality solution robustly in a shorter computational time than the state-of-the-art modified embedded optimization method. It is also illustrated that the computational efficiency is not impaired by infeasible initial solutions.

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 (PWL) approximation of the nonlinear vehicle model, so that no user-defined initial guess is needed. In the case study, the computational efficiency of the proposed method is studied with several problem settings. The results demonstrate that, compared to the state-of-the-art method, the proposed formulation converges significantly faster, and the converged solution is also at least the same or better in the same computational time limit, even when started from the same PWL-based initial solution. 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. The parallelizable nature of the algorithm can potentially make the proposed method even more effective 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.

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. (24).

$$m_d = \sum m_{sub} = m_{str} + m_{prop} + m_{power} + m_{avi} + m_{ECLSS} + m_{misc} \tag{24}$$

where m_{sub} indicates the mass of each 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 includes landing legs and truss, TPS for the reentry to the earth, and a 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.

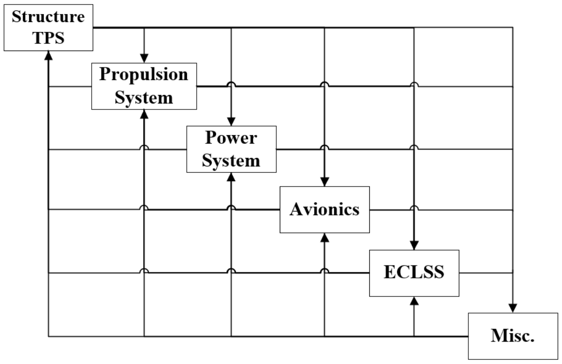


Fig. 4 Relationship of domains in a single-stage lunar lander.

For the defined subsystems, mass estimation relationships (MERs) are developed as functions of payload capacity, propellant capacity, propellant type ζ, 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 serve as the vehicle sizing constraints (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. [22, 23], 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 9 summarizes the independent variables, 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. (24) 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 9 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, ρ_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 10, 11, and 12 show the modified embedded optimization results by ACO, GA, and PSO, respectively. The initial guess increment is 2,500 kg for all cases. The algorithm implementations by pygmo [31], a scientific Python library, are used to generate the data. The parameters used for each algorithm are as follows.

For ACO, the kernel size, convergence speed parameter, oracle parameter, accuracy parameter, threshold parameter, and focus parameter are 10, 1, 10^9, 0, 7, 0, respectively. GA uses the exponential crossover [32] with 0.9 probability, polynomial mutation [33] with 0.02 probability, and tournament selection with size 2. For PSO, the inertia weight, social component, cognitive component, and maximum particle velocities are 0.7298, 1.05, 2.05, and 0.5, respectively.

Table 10 ACO embedded optimization results with 2,500 kg PWL increment

Instance	Number of generations	Optimization time, s	Best IMLEO, kg	Worst IMLEO, kg
1	10	72.27	705,550	infeasible
	50	360.9	705,550	830,350
	100	721.2	685,146	783,343
2	10	91.37	461,323	831,125
	50	452.4	420,662	631,469
	100	904.0	420,662	541,799
3	10	147.0	460,524	infeasible
	50	732.0	459,157	565,537
	100	1466	426,322	508,066
4	10	80.68	470,538	470,538
	50	403.6	470,538	470,538
	100	804.6	470,538	470,538
5	10	74.45	454,849	484,095
	50	371.3	453,073	466,563
	100	740.9	445,776	461,279
6	10	95.43	321,152	359,303
	50	476.7	299,690	336,976
	100	952.5	299,690	330,914
7	10	171.8	323,436	348,874
	50	858.8	315,515	326,136
	100	1716	313,764	323,436
8	10	94.14	375,746	425,749
	50	468.6	359,239	401,878
	100	934.7	351,292	394,226

Table 11 GA embedded optimization results with 2,500 kg PWL increment

Instance	Number of generations	Optimization time, s	Best IMLEO, kg	Worst IMLEO, kg
1	10	80.58	759,326	infeasible
	50	413.1	712,515	infeasible
	100	814.8	712,515	infeasible
2	10	98.80	437,974	infeasible
	50	511.9	437,275	infeasible
	100	1010	414,034	infeasible
3	10	158.7	427,682	infeasible
	50	840.4	419,610	infeasible
	100	1686	398,507	infeasible
4	10	85.71	470,538	470,538
	50	434.4	470,538	470,538
	100	872.8	470,538	470,538
5	10	84.28	455,676	484,630
	50	406.7	455,676	479,622
	100	806.1	455,676	476,852
6	10	106.2	306,077	359,839
	50	512.0	297,363	329,513
	100	1020	297,183	316,152
7	10	178.5	304,792	345,599
	50	866.8	296,631	323,739
	100	1708	291,836	305,488
8	10	111.7	358,082	513,027
	50	518.8	353,113	388,664
	100	998.2	349,502	371,646

Table 12   PSO embedded optimization results with 2,500 kg PWL increment

Instance	Number of generations	Optimization time, s	Best IMLEO, kg	Worst IMLEO, kg
1	10	74.57	700,227	infeasible
	50	381.7	680,234	731,419
	100	765.3	680,234	710,650
2	10	84.70	436,363	infeasible
	50	440.0	403,448	439,154
	100	881.6	402,976	412,622
3	10	153.0	423,873	infeasible
	50	794.1	421,689	436,608
	100	1620	408,388	421,829
4	10	82.52	470,538	470,538
	50	419.9	470,538	470,538
	100	842.3	470,538	470,538
5	10	75.36	451,269	470,869
	50	373.9	445,230	458,624
	100	748.6	445,230	450,841
6	10	96.64	305,457	337,105
	50	476.7	301,583	314,667
	100	949.7	298,414	306,893
7	10	172.3	302,791	358,768
	50	881.8	298,361	313,493
	100	1772	298,361	310,755
8	10	94.21	355,767	395,748
	50	466.1	350,294	363,609
	100	934.3	348,585	359,040

Acknowledgments

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

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] Norheim, J., and de Weck, O., "Co-optimizing Spacecraft Component Selection, Design, and Operation with MINLP," *2021 IEEE Aerospace Conference*, IEEE, 2021. https://doi.org/10.1109/aero50100.2021.9438148.

[12] Norheim, J., "Satellite Component Selection with Mixed Integer Nonlinear Programming," *2020 IEEE Aerospace Conference*, IEEE, 2020. https://doi.org/10.1109/aero47225.2020.9172505.

[13] Sobieszczanski-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.

[14] 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.

[15] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," , 2021. URL https://www.gurobi.com.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] Vielma, J. P., Ahmed, S., and Nemhauser, G., "Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions," *Operations Research*, Vol. 58, No. 2, 2010, pp. 303–315. https://doi.org/10.1287/opre.1090.0721.

[22] 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.

[23] 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.

[24] Duran, M. A., and Grossmann, I. E., "An outer-approximation algorithm for a class of mixed-integer nonlinear programs," *Mathematical Programming*, Vol. 36, No. 3, 1986, pp. 307–339. https://doi.org/10.1007/bf02592064.

[25] Quesada, I., and Grossmann, I., "An LP/NLP based branch and bound algorithm for convex MINLP optimization problems," *Computers & Chemical Engineering*, Vol. 16, No. 10, 1992, pp. 937–947. https://doi.org/https://doi.org/10.1016/0098-1354(92)80028-8.

[26] Westerlund, T., and Pettersson, F., "An extended cutting plane method for solving convex MINLP problems," *Computers & Chemical Engineering*, Vol. 19, 1995, pp. 131–136. https://doi.org/https://doi.org/10.1016/0098-1354(95)87027-X.

[27] Morales, A. K., and Quezada, C. V., "C.V.: A universal eclectic genetic algorithm for constrained optimization," *In: Proceedings 6th European Congress on Intelligent Techniques and Soft Computing*, Verlag, 1998, pp. 518–522.

[28] 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.

[29] Holland, J. H., "Genetic Algorithms," *Scientific American*, Vol. 267, No. 1, 1992, pp. 66–73. URL http://www.jstor.org/stable/24939139.

[30] 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.

[31] Biscani, F., and Izzo, D., "A parallel global multiobjective framework for optimization: pagmo," *Journal of Open Source Software*, Vol. 5, No. 53, 2020, p. 2338. https://doi.org/10.21105/joss.02338, URL https://doi.org/10.21105/joss.02338.

[32] Storn, R., and Price, K., "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, Vol. 11, No. 4, 1997, pp. 341–359. https://doi.org/10.1023/a:1008202821328.

[33] Deb, K., and Goyal, M., "A Combined Genetic Adaptive Search (Geneas) for Engineering Design," *Computer Science and Informatics*, Vol. 26, No. 4, 1996, pp. 30–45.