

Received November 7, 2021, accepted November 16, 2021, date of publication November 18, 2021, date of current version November 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3129265

Multi-Objective Trajectory Planning for Slung-Load Quadrotor System

HALIT ERGEZER 

Mechatronics Engineering Department, Çankaya University, 06790 Ankara, Turkey


e-mail: halitergezer@cankaya.edu.tr

ABSTRACT In this article, multi-objective trajectory planning has been carried out for a quadrotor carrying a slung load. The goal is to obtain non-dominated solutions for path length, mission duration, and dissipated energy cost functions. These costs are optimized by imposing constraints on the slung-load quadrotor system's endpoints, borders, obstacles, and dynamical equations. The dynamic model of a slung-load quadrotor system is used in the Euler-Lagrange formulation. Although the differential flatness feature is mostly used in this system's trajectory planning, a fully dynamic model has been used in our study. A new multi-objective Genetic Algorithm has been developed to solve path planning, aiming to optimize trajectory length, mission time, and energy consumed during the mission. The solution process has a three-phase algorithm: Phase-1 is about randomly generating waypoints, Phase-2 is about constructing the initial non-dominated pool, and the final phase, Phase-3, is obtaining the solution. In addition to conventional genetic operators, simple genetic operators are proposed to improve the trajectories locally. Pareto Fronts have been obtained corresponding to exciting scenarios. The method has been tested, and results have been presented at the end. A comparison of the solutions obtained with MOGA operators and MOPSO over hypervolume values is also presented.

INDEX TERMS Multiobjective optimization, slung-load quadrotor system, trajectory planning.

NOMENCLATURE

$\mathcal{C} \in \mathbb{R}^3$: Configuration space.	\mathbf{R}	: The rotation matrix from Body Frame to Inertial Frame.
$\mathcal{C}_{obs} \in \mathbb{R}^3$: Configuration space for obstacles.	$\boldsymbol{\Omega}$: The angular velocity.
$\mathcal{C}_{free} \in \mathbb{R}^3$: Obstacle-free configuration space.	\mathbf{I}	: The inertia matrix.
E	: Consumed energy.	m_q	: The mass of the quadrotor.
f	: The lifting force.	l	: Length of Rope.
\mathbf{f}_m	: Objective function vector.	m_l	: The mass of the load.
\mathbf{x}_q	: The position vector of quadrotor's center of mass with respect to (wrt.) inertial frame.	ϕ	: The roll angle.
\mathbf{x}_l	: The position vector of load wrt. inertial frame.	ϕ_{min}	: Lower bound for the roll angle.
\mathbf{x}	: State vector.	ϕ_{max}	: Upper bound for the roll angle.
M_ϕ	: Roll moment acting on the quadrotor.	θ	: The pitch angle of the quadrotor.
M_θ	: Pitch moment acting on the quadrotor.	θ_{min}	: Lower bound for the pitch angle.
M_ψ	: Yaw moment acting on the quadrotor.	θ_{max}	: Upper bound for the pitch angle.
ω_j	: The angular velocity of rotor j .	\mathbf{p}^T	: The unit vector from the quadrotor to load in the inertial frame.
K_f	: Force-constant.	\mathbf{u}	: The vector that defines a Maneuver.
K_M	: Moment-constant.	$T_{mission}$: Scenario duration.

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Tang .

I. INTRODUCTION

The use of quadrotors is increasing day by day. One of the tasks quadrotors are employed recently is load transportation. One can equip a quadrotor with a gripper mechanism or

connect the load mass to a quadrotor with some wire (slung-load) to transport a load. The gripper mechanism adds to the quadrotor's total mass [2]; that is why the slung-load method is preferred. Besides, the gripper mechanism needs additional controllers and consumes energy while doing its job, which means it is more expensive to carry a load. Alternatively, it is preferred that the quadrotor takes the load with a tied rope or wire. In [1], this type of transportation task was carried out with unmanned helicopters; the helicopter-like systems' mathematical models and controller designs were examined for different wire connections.

It is possible to examine the literature on manipulating slung-load unmanned aerial vehicles, mainly in two categories. Studies in the first category are concerned with designing controllers for tracking desired payload trajectories [5], [7]–[9], and [17]. On the other hand, studies in the second category are trajectory planning studies to optimize a particular objective function [3], [6], [15], [18], and [19]. Our study belongs to the second category. For an extensive summary of load transportation studies using quadrotors, a recently published survey [3] can be investigated.

We think that in trajectory planning, usually, there is more than one criterion involved with the problem either directly or indirectly [29], [30]. There are studies to optimize objective functions such as minimum swing angles of load [6], [15], and [16], minimum mission time [24], and minimum energy [21], [22]. In [23], an objective function emphasizing the tradeoff between mission time and dissipated energy were investigated. Interestingly, there is no study aiming to optimize multiple objectives simultaneously to the best of our knowledge. It is one of our research contributions. The trajectory planning for a quadrotor slung-load system is investigated under numerous goals such as mission duration, path length, and dissipated energy.

Although there are many studies [5], [12]–[14] that advocate doing multi-objective path planning, most of them implement scalarization (weighted sum of multiple objectives are considered). For example, [5] aims to minimize the mission completion time, which is the sum of the time to set up a communication and find a target. They do not consider these two times separately; instead, they assign weights to prioritize. An online trajectory planning algorithm is proposed in [16] and [17] to minimize the load swing and target positioning error. Although there are multiple goals, the problem is not approached as multi-objective optimization; instead, the problems are handled separately.

Studies utilize the multi-objective optimization approach to solve trajectory planning problems to optimize path length, path safety, and path smoothness on the grid map without considering the vehicle dynamics constraints [26]. In [3], the firefly algorithm has been proposed to optimize objective functions [26]. Although a different application is discussed in [35] than the one we have covered in this study, a framework is proposed that makes good use of the concept of multi-objective optimization. Our study, conceptually similar to [35], aims to optimize more than one objective

simultaneously in the trajectory planning problem for quadrotor slung-load system makes the problem even more difficult.

Quadrotor slung-load system is an under-actuated system with eight degrees of freedom and four control inputs (i.e., it is challenging to obtain a trajectory plan even for only one of the objectives mentioned above). Furthermore, simultaneously optimizing more than one objective function makes the problem even more difficult.

Our aim here is to find out how to create a solution to optimize the length of trajectory, mission time, and energy consumed during the mission simultaneously. Intuitively, it is easier to understand that system will spend more energy on the fastest trajectory. However, it is helpful to make a few explanations regarding the relationship between “*mission time and path length*” and “*path length and energy spent*” objectives. The fastest trajectory may not be the shortest one because there are obstacles in the environment. It may be necessary to maneuver too much on the shortest trajectory. Since it may be required to accelerate and slow down during these maneuvers, we cannot say that the shortest trajectory is the least energy expenditure.

Even though these objective functions may not be completely conflicting, they are not entirely supportive of others. For example, as shown in Section 5, Figure 14, minimum length, minimum energy, and minimum time trajectories are different. In addition, a slight change in consumed energy immediately resulted in entirely different paths. Usually, there is no single solution that simultaneously optimizes each objective. Alternatively, it is possible to combine the individual objective functions into a single objective using suitable weights. Here the critical question is how one should select these weights. In this case, we will have a single optimum solution rather than a set of “non-dominated” solutions that can be examined for further tradeoffs (i.e., a set of non-dominated solutions is found corresponding to all possible weight selections). The collection of these non-dominated solutions is called the Pareto optimal set or Pareto-front [28]. In this set, solutions are non-dominated (i.e., none of the objective functions can be improved in value without degrading some of the other objective values) concerning each other. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in at least one objective(s) to achieve a certain amount of gain in the other(s). Pareto optimal solution sets are often preferred to single solutions because they can be practical when considering real-life problems.

A solution's superiority over other solutions is easily determined in the single-objective optimization problem by comparing their objective function values. In multi-objective optimization problems, the quality of a solution is determined by the importance of the optimization criteria. By investigating the Pareto-front, it is easy to observe how different objectives change as one moves on the Pareto-front. On the other hand, if one optimizes a weighted sum of objectives, only a point on the Pareto-front is found; one does not know what happens to the overall objectives if one of the weights is

slightly perturbed. There are many disadvantages to solving multi-objective problems with classical optimization methods [28]. Genetic Algorithms (GA) are frequently preferred in the solution of multi-objective optimization problems [27]. Our study used a GA-based algorithm by customizing the genetic operators for the problem at hand.

As stated in [36], where constrained path planning problem is examined, Evolutionary Algorithm (EA) is of great interest in solving such problems. Moreover, while algorithms such as SQP, IPM, which require numerical gradients, only guarantee convergence to the local optimum, heuristics are generally more likely to find the optimal result [33], [36]–[39]. In [33], it is underlined that PSO (Particle Swarm Optimization) requires less number of function evaluations among EAs. In [36], it is stated that direct application of MOPSO [45], [48] to the problem is lacking in finding better values for constraint compliance and objective functions. For this reason, a biased search towards the feasible region is proposed in [36]. Similar to the study we carried out in this article, modifications were made to the classical algorithm. In this context, in our study, not only a biased search application towards a feasible region but also an initial population has been generated in this way. In our study, we used only the main structure of GA as a base since it provides a more suitable structure than other EAs for describing the problem-specific operators. In [29] and [30], it is stated that it is vital to use problem-specific (mutation-like) operators to reach a global optimum when dealing with these challenging problems.

We believe that our approach yields many original contributions;

1. Trajectory planning as a multi-objective optimization problem for competing costs (energy, time, and length) was solved using a slung-load quadrotor system for the first time (to the best of our knowledge).
2. A new and successful Genetic Algorithm was developed for this problem (details of the algorithm are presented in Section 4). In trajectory planning problems, optimization variables are implicitly dependent, and the degree of dependence increases as one moves along the trajectory. The proposed approach almost completely removes this dependency between optimization variables.
3. Problem-dependent mutation-like operators were developed to avoid obstacles and to improve cost vectors locally.

The rest of the paper is organized as follows; Firstly, the problem definition is given in Section II. Secondly, a dynamic model of a slung-load quadrotor system is described in Section III. Thirdly, the details of the proposed trajectory planner are given in Section IV. Section V contains the resultant trajectories of the slung-load quadrotor system and Pareto Front. Performance comparison with classical GA and PSO algorithms and the analysis of gaps in Pareto Front are given in Section VI. Then, all sections are concluded in Section VII.

II. PROBLEM DEFINITION

Problem definition and associated assumptions are given below. Here we start with the assumptions first:

- The load is a point-mass,
- The cable is massless, and it is not stretchable,
- The cable is attached to the center of mass of the quadrotor,
- The cable is under tension throughout the mission.

Next is the optimization problem we are supposed to deal with:

$$\begin{aligned} \min_{\mathbf{u}} f_m(\mathbf{u}) &= (f_1(\mathbf{u}), f_2(\mathbf{u}), f_3(\mathbf{u}))^T \\ \text{subject to} & \\ & \text{State equations (Eq.5 – Eq.10)} \\ & \mathbf{LB} \leq \mathbf{u} \leq \mathbf{UB} \\ & \mathbf{x}_q = \mathbf{x}_{q\text{initial}} \text{ at } t = 0 \\ & \mathbf{x}_q = \mathbf{x}_{q\text{final}} \text{ at } t = T_{\text{mission}} \\ & \mathbf{x}_q, \mathbf{x}_l \in \mathcal{C}_{\text{free}} \quad \forall t \leq T_{\text{mission}} \\ & \phi_{\min} \leq \phi \leq \phi_{\max} \quad \forall t \leq T_{\text{mission}} \\ & \theta_{\min} \leq \theta \leq \theta_{\max} \quad \forall t \leq T_{\text{mission}} \end{aligned} \quad (1)$$

where $f_1(\mathbf{u})$, $f_2(\mathbf{u})$, $f_3(\mathbf{u})$ are the objective functions corresponding to the length of the path, total mission time, and energy consumed during the mission.

$$f_1(\mathbf{u}) = \sum_{i=1}^{N_L} \|\mathbf{x}_q(i) - \mathbf{x}_q(i-1)\| \quad (2)$$

where N_L is the number of points along a trajectory and can be calculated $N_L = 100 * T_{\text{mission}}$. The time step size of the simulation is 0.01 s.

The duration of the mission time can be determined at the end of the trajectory construction procedure (T_{mission}).

$$f_2(\mathbf{u}) = T_{\text{mission}} \quad (3)$$

The vector \mathbf{u} consists of the control inputs (i.e., control forces and moments acting on the quadrotor). The number of time instances these control inputs are supposed to be applied on the quadrotor. The vectors \mathbf{LB} and \mathbf{UB} are the lower and upper bounds of \mathbf{u} . $\mathcal{C}_{\text{free}} = \mathcal{C} \setminus \mathcal{C}_{\text{obs}} \in \mathbb{R}^3$ is obstacle-free configuration space (Figure 1). We also impose the constraints on roll (ϕ) and pitch (θ) angles of the quadrotor. The initial and final points of the trajectory are given as input to the problem. It should be noted the mission duration (i.e., T_{mission}) is a free variable.

III. DYNAMIC MODEL OF A SLUNG-LOAD QUADROTOR SYSTEM

In this section, the 3D dynamic model for the slung-load quadrotor system will be described. We start with the coordinate system definitions in the mathematical model, as shown in Figure 2. The mathematical model used in this study is borrowed from [9], and the detailed derivation of this model can be found there. This model is differentially flat, and the flat outputs are the position of the load and the yaw angle of

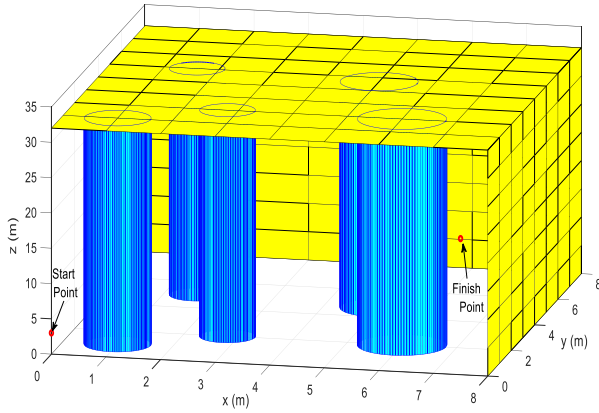


FIGURE 1. Sample scenario.

the quadrotor. Unlike [9], the full 3D dynamic model is preferred over the flat model since we imposed constraints on the quadrotor’s location and attitude. Additionally, different from previous studies [5], [6], and [18], we do trajectory planning for both the load and the quadrotor to avoid obstacles. The state vector is; $\mathbf{x}(t) = [x_L^T \mathbf{p}^T \mathbf{R} \dot{x}_L^T \dot{\mathbf{p}}^T \boldsymbol{\Omega}^T] \in \mathbb{R}^{18}$ where x_L^T is the position vector of load and \mathbf{p}^T is the unit vector from the quadrotor to load in the inertial frame. The input vector is $\mathbf{v} = [f \ M_\phi \ M_\theta \ M_\psi]^T$ where f is the lifting force acting on the system and M_ϕ, M_θ, M_ψ are roll, pitch, and yaw moments, respectively (see Figure 2).

$$\mathbf{v} = \begin{bmatrix} f \\ M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ K_M & -K_M & K_M & -K_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (4)$$

where $\omega_j, (j = 1, 2, 3, 4)$ is the angular velocity of rotor j , K_f aerodynamic force constant, K_M aerodynamic moment constant. The mathematical model equations are given below:

$$\frac{d}{dt}x_L = \dot{x}_L \quad (5)$$

$$\frac{d}{dt}\dot{x}_L = \frac{1}{m_q + m_l} (\mathbf{p} \cdot f \mathbf{R} \mathbf{e}_z - m_q l \|\dot{\mathbf{p}}\|) \mathbf{p} - g \mathbf{e}_z \quad (6)$$

$$\frac{d}{dt}\mathbf{p} = \dot{\mathbf{p}} \quad (7)$$

$$\frac{d}{dt}\dot{\mathbf{p}} = \frac{1}{m_q l} (\mathbf{p} \times (\mathbf{p} \times f \mathbf{R} \mathbf{e}_z) - \|\dot{\mathbf{p}}\| \mathbf{p}) \quad (8)$$

$$\frac{d}{dt}\mathbf{R} = \mathbf{R} \hat{\boldsymbol{\Omega}} \quad (9)$$

$$\frac{d}{dt}\boldsymbol{\Omega} = \mathbf{I}^{-1}(\mathbf{M} - \boldsymbol{\Omega} \times \mathbf{I} \boldsymbol{\Omega}) \quad (10)$$

where \mathbf{R} is the rotation matrix from Body Frame to Inertial Frame, $\boldsymbol{\Omega}$ is the angular velocity of the quadrotor. \mathbf{I} is the inertia matrix, and m_q and m_l are the mass of the quadrotor and the load, respectively. $\hat{\boldsymbol{\Omega}}$ is the skew-symmetric matrix described in [41]. The energy equation, a function of rotor

speeds and accelerations given in [20], has been used to calculate the quadrotor’s energy consumed during a mission.

$$f_3(\mathbf{u}) = E = \int_0^{t_{final}} \sum_{j=1}^4 \frac{I_r \dot{\omega}_j(t) + K_f \omega_j^2}{g_{r,j}(\dot{\omega}_j(t), \omega_j(t))} \omega_j(t) dt \quad (11)$$

IV. TRAJECTORY PLANNER

As stated earlier, the trajectory planning problem discussed in this study is expressed as a multi-objective optimization problem. A set of “non-dominated” [28] solution vectors are to be obtained. It is advisable to represent the Pareto front by a high number of non-dominated vectors. On the other hand, we expect them to be evenly distributed throughout the front. The nature of Genetic Algorithms (GA) is very suitable for solving multi-objective optimization problems [2], which is why a modified GA is developed and used in this study. As emphasized in [32], it is very effective to handle the problem using several stages in constraint trajectory planning problems.

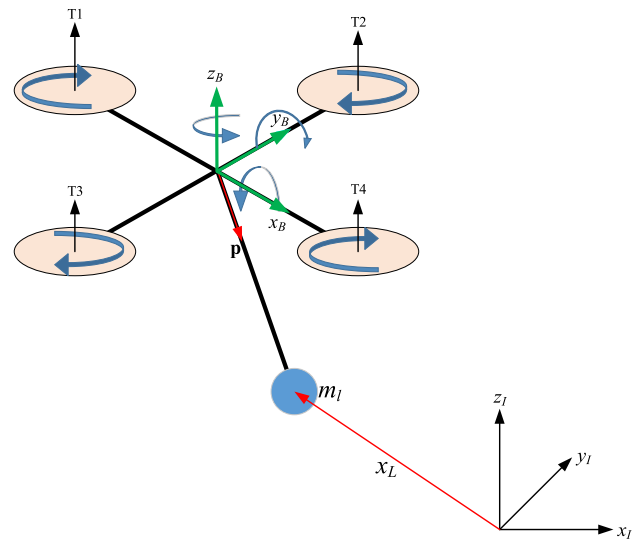


FIGURE 2. Slung-load quadrotor system coordinate frames.

The structure of a chromosome is shown in Figure 3. It consists of *slots*, each of which defines a maneuver. Each *slot* contains control inputs ($\mathbf{u} = [\mathbf{v} \text{ NoS}]$) and a variable ($\text{NoS}_i \ i = 1, 2, \dots, N$) that keeps information about how many time steps these control inputs are to be applied.

The maximum number of maneuvers is constant. Thus, while the chromosome length remains constant, the total mission duration is not persistent. We present some of the parameters we use in the scenarios below;

- The size of the Population is 150.
- The length of each Chromosome is at most 500 (i.e., 100 slots \times 5 variables).
- The step size in our simulations is 0.01 seconds.
- The maximum number of generations is 1000.

Reasonable initial estimates can produce better solutions with faster convergence if we have prior knowledge of

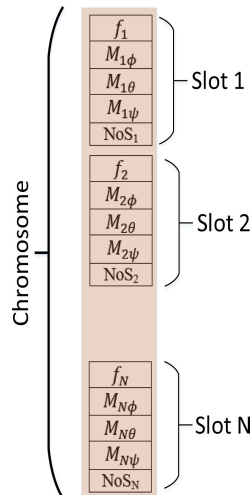


FIGURE 3. Chromosome structure.

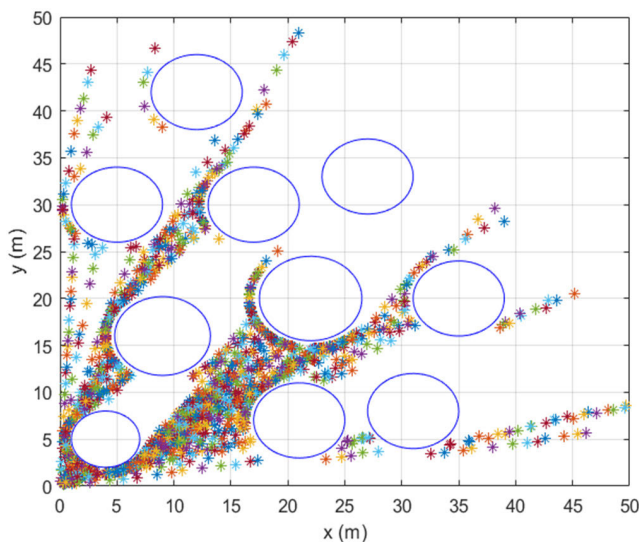


FIGURE 4. Points are found randomly between the starting and final points without obstacle violation.

the problem or can be produced at a low computational cost [46], [47]. The method of using good initial estimates is called seeding. In [46], five different algorithms were tested on 48 optimization problems using different seeding methods, and their performance was examined. It has been shown that the seeding process based on prior knowledge significantly reduces the computational cost. We did not randomly generate the initial population. Instead, it is obtained by using waypoints obtained by removing some randomly generated points out of the obstacle zone that does not violate the obstacles. In Phase 2, a structure was created between these waypoints so that the chromosome length gets longer and longer to reach the endpoint goal. In this way, when we come to Phase 3, not all individuals in the initial population have the same fitness value. In cases where the initial population is randomly generated, as noted in [28], nearly

all MOEAs evaluate all solutions of the first non-dominant front equally, assigning a similar or similar fitness to all individuals in the initial population. Neither of these solutions has a choice advantage. Naturally, it will take longer for the EAs to advance the search towards the Pareto optimal region. This search is difficult to achieve in the problem we consider, especially in complex scenarios, due to compelling dynamic constraints. Fortunately, the initial population generation technique in the proposed method greatly reduces the probability of not finding a solution in cases where the solution is very difficult to find.

A. PHASE-1 RANDOMLY GENERATED WAYPOINTS

To find non-dominated trajectories more quickly, we found waypoints between the starting and the final points without obstacle violation (Figure 4). These waypoints have been found by generating random points in polar coordinates. In particular, first, a radial line making an arbitrary angle to the radial line connecting the starting point $[0 \ 0 \ 3]^T$ and final point $[40 \ 40 \ 10]^T$ is generated. Next, points are created randomly along this line. If it exists, obstacles can be handled by moving these randomly generated points into the feasible region to have a pre-defined margin with the obstacles (Figure 4).

Although about 10 - 15 solutions are usually sufficient in real applications, the population size must be chosen much larger due to the nature of EA algorithms. Population size should be determined according to the complexity of the problem and the number of decision variables [44]. Population sizing is an iterative process; however, the graph based on the randomly generated population, given in [28 pp.403], can be used as a good starting point to determine the population size. The initial population in our study has been formed based on the data obtained at the former stages, not completely random. Therefore, the population size has been chosen as 150, aiming to find a nominal number of trajectories between 10 and 15.

B. PHASE-2 INITIAL NON-DOMINATED POOL CONSTRUCTION

One of the main difficulties in trajectory planning is that any change in a specific point significantly impacts the trajectory's remaining parts. In other words, optimization variables are implicitly dependent, and the degree of dependence increases as one moves along the trajectory [29].

The following idea can reduce this dependency; the trajectory is constructed stepwise. At each step, new waypoints are considered (Figure 5 and Figure 6). Equivalently, new slots are added to each Chromosome's end at each step, depending on the number of waypoints considered. Even though the solution obtained in previous steps (except the first step) is allowed to change within the algorithm's structure, it is observed that these changes are minimal. The operations outlined above can be repeated for all trajectory candidates, for all the waypoints on each trajectory are repeated, and

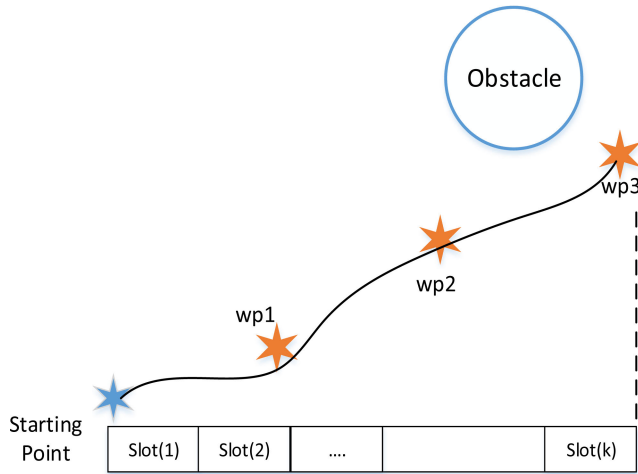


FIGURE 5. The first step of the algorithm with k slots.

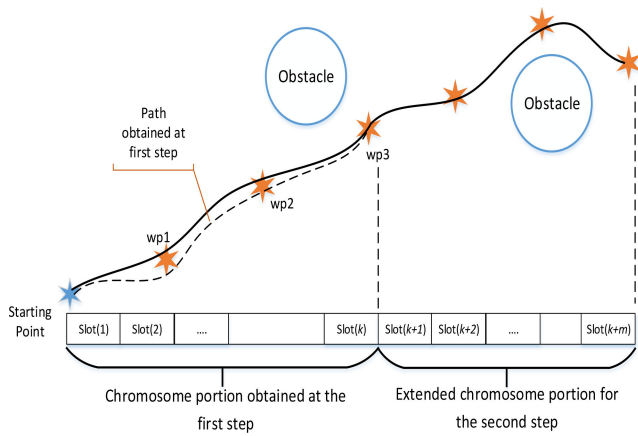


FIGURE 6. The i^{th} step of the algorithm with additional m slots.

finally, one obtains the initial non-dominated chromosome pool.

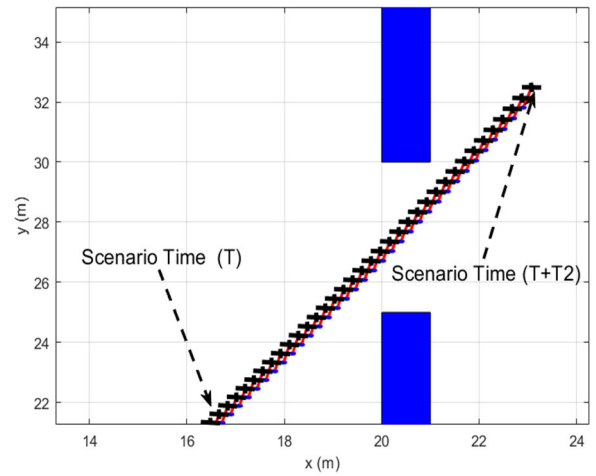
C. PHASE 3 MULTI-OBJECTIVE OPTIMIZATION

There are several multi-objective GA optimization codes in the literature. The most well-known of these are NSGA and NSGA-II [2]. NSGA-II is better than NSGA in terms of computational cost and elitism. NSGA-II uses a fast non-dominated sorting technique, an elitist-keeping technique, and a new niching genetic operator without parameters [2]. In this study, we have implemented a multi-objective optimization algorithm. This algorithm uses only the sorting structure of the NSGA-II to find a non-dominated solution. An important contribution of this study is that the chromosome length is variable, and the introduction of Phase-1 and Phase-2 will result in a very fast algorithm.

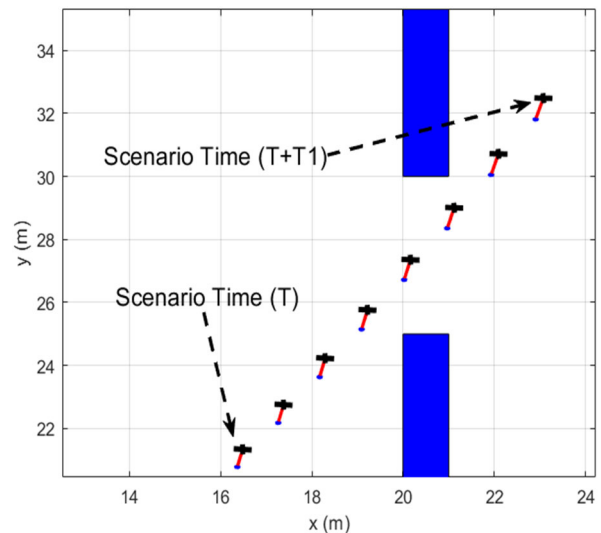
D. PROBLEM SPECIFIC OPERATORS

One should be careful in applying the crossover operation; the crossover points are selected between the slots.

Regularization Operation (RegOp) - Since the ranges of force and moment inputs are different (between 20 N



(a)



(b)

FIGURE 7. The mutated part of the trajectory, ComMut operator, has been applied, (a) trajectory before the operator, (b) trajectory after the operator ($T2 > T1$).

and 35 N for force, $\pm 0.5N.m$ for moments), it is not meant to change the force and moment creating values in the same range to mutate these inputs. For this reason, we perform the mutation operations for force and moments separately. The force and moment values have been changed per the maneuver to be made.

The number of Step Mutation (NoSMut) - As the NoS values are integers, a different mutation operator is also defined. By mutating these values, one can change the total mission duration. Again, this operator affects the dissipated energy by the quadrotor.

Combined Mutation (ComMut) - In this mutation operation, NoS is increased/decreased, and the objective functions are minimized by increasing or decreasing the force and moment values accordingly. Figure 7 presents this operator's effect; the slung-load quadrotor positions are plotted during that particular time interval. The operator decreases the NoS

Algorithm Trajectory Planner

1: MAIN
2: Inputs: $x_{qinitial}, x_{qfinal}$
 $C_{free} \in R^3, C_{obs} \in R^3$
3: Outputs: $u = [v \text{ NoS}]$
4: Call Phase-I
5: Call Phase-II
6: Call Phase-III
7: End MAIN

8: Phase-I #RGW
Input: $C_{obs} \in R^3$, and $NoRP$ # Number of Random Points.
9: NoRS # Number of Random point set.
Output: RPs #Random points between $x_{qinitial}, x_{qfinal}$
10: $\theta = \arctan\left(\frac{x_{qfinal}(2) - x_{qinitial}(2)}{x_{qfinal}(1) - x_{qinitial}(1)}\right)$
11: $r = \sqrt{(x_{qfinal}(2) - x_{qinitial}(2))^2 + (x_{qfinal}(1) - x_{qinitial}(1))^2}$
12: For n:1 to NoRS
For k:1 to NoRP
12: $\theta_{random} = N(\theta, 1), d_{random} = N\left(\frac{k*r}{NoRP}, 1\right)$
13: Call *Obstacle_Violation*($\theta_{random}, d_{random}, C_{obs}$)
14: If there is a violation **Then**
15: Move point to closest point in the feasible region.
16: End If
17: RPs(n, k) = ($\theta_{random}, d_{random}$)
18: End For
19: End For
20: End Phase-I

21: Phase-II #Initial non-dominated pool construction
22: Input: Random Points generated in Phase-I
Output: Initial non-dominated solutions.
23: Set *Chromosome_Length* to *minimumLength*
24: Set *Chromosome* to initial values.
25: Repeat For *Each Random Point Set* 1 to NoRS
26: For k:1 to NoRP-1
Call *FindNDPaths*($P(k), P(k+1), CL, \text{PreviousPopulation}$)
27: # non-dominated solutions between Points $P(k)$ and $P(k+1)$,
Points P are obtained in Cartesian coordinate from RPs.
28: Update *CL* # *Chromosome_Length* \leftarrow *Chromosome_Length* + *deltaLength*
29: End For
30: End Repeat
Def: FindNDPaths()
31: Inputs: $P(k), P(k+1), CL, \text{PreviousPopulation}$
Outputs: Population
32: Initialize Population # Initialize Population by preserving chromosome part for the previous path segment
For Each Member in Population Do
33: Call $[x(t) T_{mission}] \leftarrow \text{SimulatedDynamicModel}(v)$ #Call Simulink model for each input vector.
34: Call *CheckForConstraintViolation* #Discard paths if there is obstacle or border violation.
35: Call *CheckAngleConstraints*(v) #check for the angle of the p vector and penalize if the angle values exceed the defined limit.
36: Calculate $f_m(u)$
37: Call *NSGA-II_Sort* # Find non-dominated solutions using *NSGA-II* sort structure.
38: Call *Crossover*() # 64% of new population is generated
39: Call *RollMutation*() # 6% of new population is generated.
40: Call *PitchMutation*() # 6% of new population is generated.
41: Call *ForceMutation*() # 6% of new population is generated.
42: Call *ComMut*() # 12% of new population is generated.
43: Call *NoSMut*() # 6% of new population is generated.
44: End For
45: End FindNDPaths

46: Phase-III # Multi-objective optimization
Def: MOOP($x_{qinitial}, x_{qfinal}, CL$)

47: Inputs: Initial non-dominated solutions from Phase-II (Population)
Outputs: Non-dominated Solutions and corresponding paths.
48: For Each Member in Population Do
49: Call $[x(t) T_{mission}] \leftarrow \text{SimulatedDynamicModel}(v)$ #Call Simulink model for each input vector.
50: Call *CheckForConstraintViolation* #Discard paths if there is obstacle or border violation.
51: Call *CheckAngleConstraints*(v) #check for the angle of the p vector and penalize if the angle values exceed the defined limit.
52: Calculate $f_m(u)$
53: Call *NSGA-II_Sort* # Find non-dominated solutions using *NSGA-II* sort structure.
54: Call *Crossover*() # 64% of new population is generated
55: Call *RollMutation*() # 6% of new population is generated.
56: Call *PitchMutation*() # 6% of new population is generated.
57: Call *ForceMutation*() # 6% of new population is generated.
58: Call *ComMut*() # 12% of new population is generated.
59: Call *NoSMut*() # 6% of new population is generated.
60: End For
61: End MOOP

and increases the applied force simultaneously. While the time spent in this part of the trajectory decreases, consumed energy increases, but the path length remains constant for this part.

Descriptions and probability rates for mutation operators are;

- Roll Moment mutation: A chromosome is selected randomly. One to eight slots are selected randomly on the Chromosome. For each selected slot, the roll moment value is changed by a randomly selected value between $+/- 10^{-4}$ N.m. (Probability rate is %6).
- Pitch Moment mutation: A chromosome is selected randomly. One to eight slots are chosen randomly on the Chromosome. For each chosen slot, the Pitch moment value is changed by a randomly selected value between $+/- 10^{-4}$ N.m. (Probability rate is %6).
- Force mutation: A chromosome is selected randomly. One to eight slots are chosen randomly on the Chromosome. A randomly selected value changes the force value between $+/- 0.5$ N for each selected slot. (Probability rate is %6).
- Step number mutation (*NoSMut*): A chromosome is selected randomly. One to eight slots are chosen randomly on the Chromosome. For each selected slot, the number of steps value is changed by ± 1 . (Probability rate is %6).
- Combined Mutation (*ComMut*): Described earlier. (Probability rate is %12).

The reason for changing the moments in these intervals is the dynamic constraints of the vehicle. The leaning angles of the vehicle must be within certain limits. Therefore, even if we give higher torque values, the vehicle will not fulfill these commands. We could have given it in lower steps, but it takes longer to find the result this time. For Force, choosing the

steps too small increases the time required to achieve results. We could have given larger force values, but in this case, when there is a velocity component in the horizontal, it increases the oscillation of the load by performing sudden accelerations, thus resulting in a long time to get the result.

It is not very important to choose 4% or 8% of the proportional values of the operators. Thanks to the robustness that the operators we define provide to GA, the effect of these choices on performance is almost equal. Increasing the ratio of these operators too much would be against the GA structure.

V. RESULTS

The simulations of case studies investigated in our approach were made using MATLAB/Simulink. The state equations (differential equations) and constraints have been taken care of implicitly by the Simulink solver. The simulations run on the personal computer, with processor configuration: 3 GHz, Memory (RAM): 32 GB. The overall algorithm takes 3612.145 s for the 1st Scenario and 3686.104 s for the 2nd Scenario. But the dynamic simulation of the quadrotor slung-load system’s model takes about 75% of the total time.

The method described in the previous sections has been tested and associated with Pareto Fronts for two exciting scenarios. The trajectories corresponding to some interesting points on Pareto Fronts have been given. The trajectories shown in these figures are essentially just one element of a trajectory set. Trajectories, which are in the same set, are similar in general shape. Since there are minor differences between trajectories, we prefer to present only one of them; giving the others would not make much sense.

The region for the first Scenario is shown in Figure 8. The borders of this region are defined as; $[0\ 0\ 0]^T \leq [x\ y\ z]^T \leq [50\ 50\ 32]^T$. The region boundary bounds the obstacles’ height; the vehicle cannot fly over the obstacles. We try to

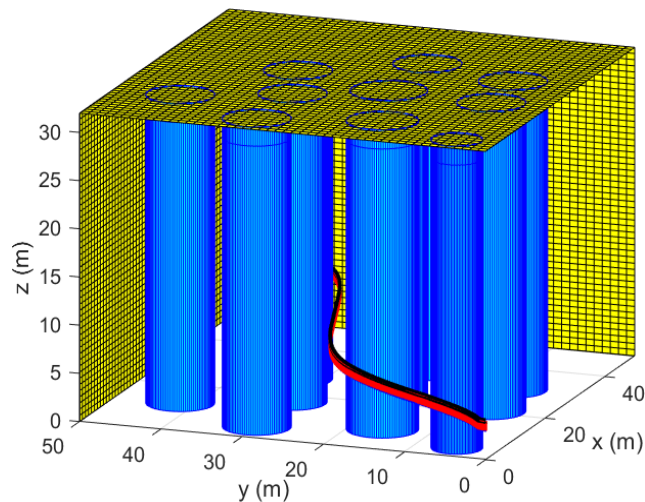


FIGURE 8. Resulting trajectory corresponding to the non-dominated solution in terms of path-length (3D view).

construct a relatively complicated region of interest to get more interesting Pareto Fronts at the end.

The quadrotor’s starting point is $[0\ 0\ 3]^T$, and the rope’s length is 1 meter. The final point of the quadrotor is $[40\ 40\ 10]^T$ in meters. In Figures 8 and 9, obstacles are shown in blue, borders of the region are shown in yellow, the quadrotor is in black, the load is in blue, and the rope is in red.

In Figure 9, one can observe the enlarged view of the trajectory of the slung-load quadrotor system.

One can observe in Figure 10 that the Pareto Front is a disjoint set of points in \mathbb{R}^3 consisting of two space curves. Please note that the objective function values in these figures are normalized to visualize the Pareto Front better.

In Figure 11, the projection of the Pareto Front is taken onto the Time vs. Energy plane. When the Path Length is removed from the multi-objective cost, the resultant Pareto Front in \mathbb{R}^2 becomes as shown in Figure 12.

In Figures 10 to 15, the Pareto Front in \mathbb{R}^3 associated with the multi-objective optimization problem in Scenario-1

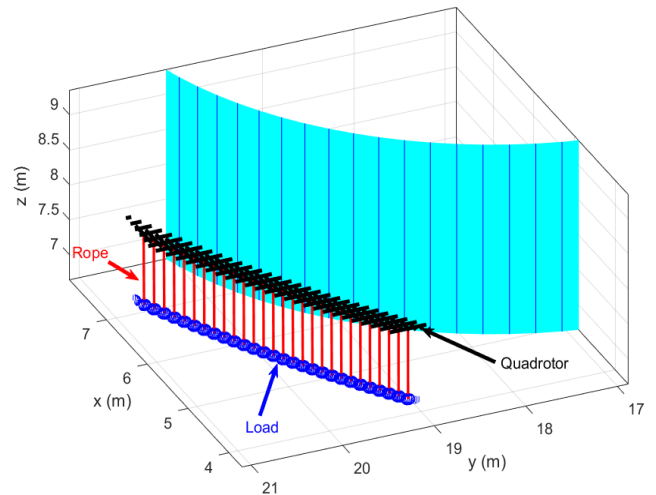


FIGURE 9. The resulting trajectory for the 1st scenario (zoomed view).

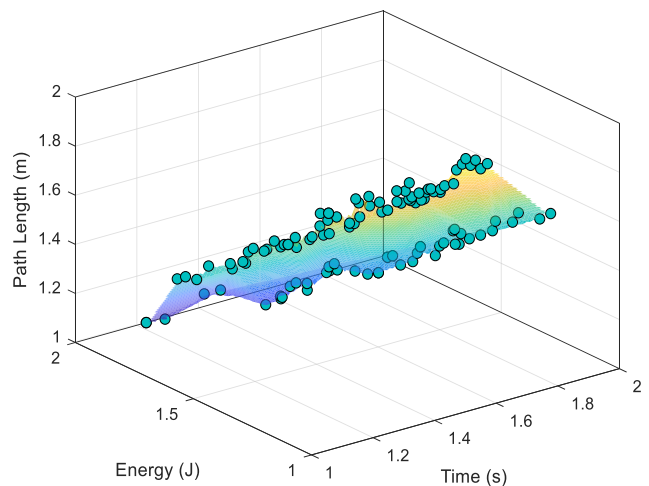


FIGURE 10. Pareto Front for scenario-1.

and its projections on different coordinate planes in \mathbb{R}^2 can be observed. This Pareto Front has been obtained according to path length, total mission time, and dissipated energy minimizations.

We have performed a similar analysis when the Pareto Front's projection is taken onto the Energy vs. Path Length plane (Figure 13). If the scenario duration is removed from the multi-objective cost, the resultant Pareto Front in \mathbb{R}^2 becomes as shown in Figure 14. It is interesting to observe the jump on the projected Pareto Front in this figure. The actual trajectories corresponding to the points (Pareto Point-1, Pareto Point-2 in Figure 14) at the beginning and the end of this jump are shown in Figures 15 and 16, respectively. It is important to note that the solution curves in Region-1 in Figure 14 are similar to those in Figure 15. Similarly, the solution curves in Region-2 in Figure 14 are similar to the curve in Figure 16.

The 3D plot of the Pareto Front points is shown in Figures 10, 11, and 13. Note that they are not Pareto Front surfaces (contrary to their representations in those figures). These figures have been obtained using the MATLAB function “*scatteredinterpolant*.”

Since we normalized the objective function values between 1 and 2, it does not give the maximum and minimum values. Still, the mean and standard deviation values are given in Table 1 for the first scenario.

We do not present the trajectory corresponding to Pareto Point-3 explicitly because it is very similar to the one corresponding to Pareto Point-2. The trajectory corresponding to Pareto Point-4 is illustrated in Figure 18 (its 3D version is shown in Figure 16).

In Figures 20 and 22, one can observe the global optimum solution corresponding to energy and time, respectively.

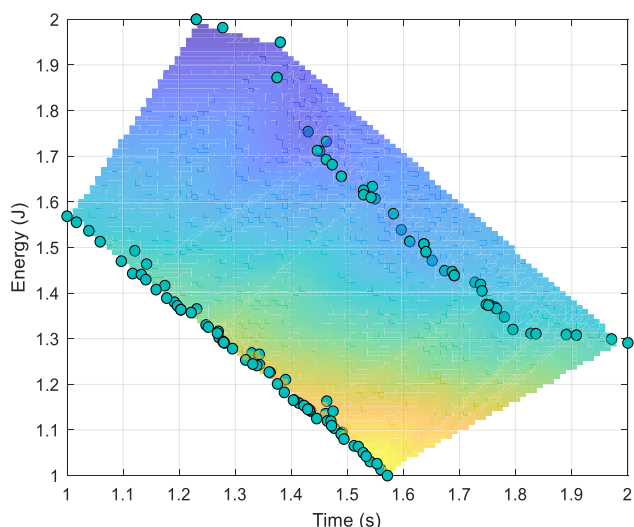


FIGURE 11. Projection of the Pareto front on time vs. energy.

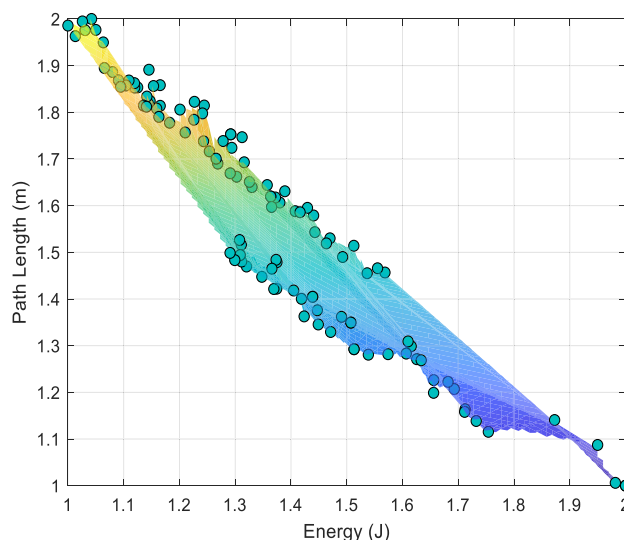


FIGURE 13. Projection of the Pareto front on energy vs. path length.

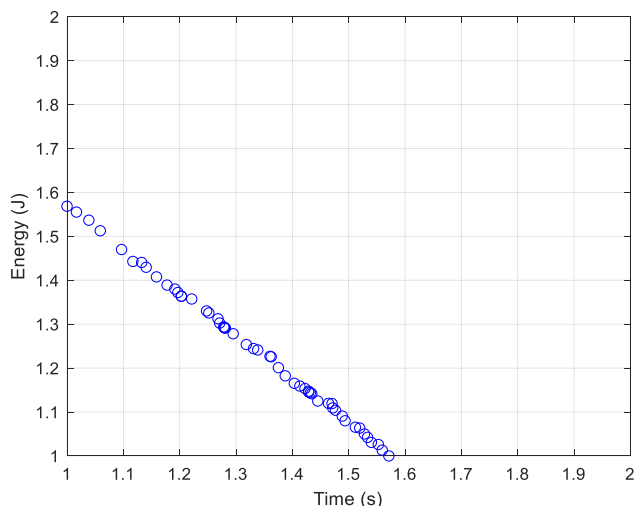


FIGURE 12. Pareto front when the path length is removed from the multi-objective cost.

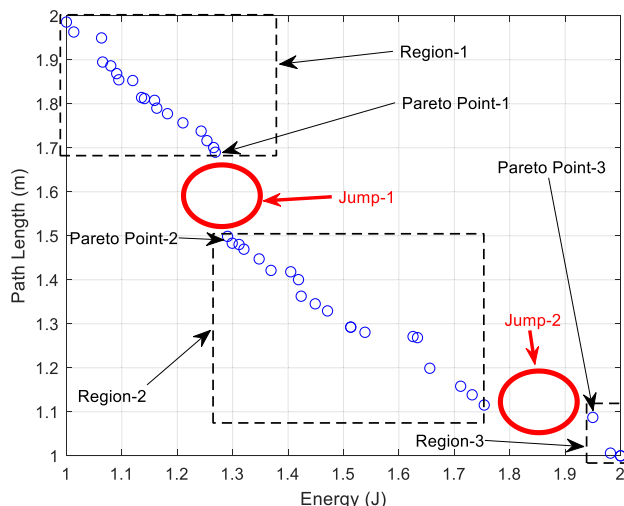


FIGURE 14. Pareto front when the scenario duration is removed from the multi-objective cost.

The 3D representations of the paths in Figures 20 and 22 are given in Figures 19 and 21.

Scenario-2 aims to construct obstacles so that jumps in the resultant Pareto Front and the corresponding differences in the solution trajectories will be more distinctive. For example, in the Pareto Front for the 2nd Scenario (Figure 23), the jump points are indicated as has been done earlier. In Figures 25, 26, and 27, one can observe the global-optimum trajectories corresponding energy (Pareto Point-1), path-length (Pareto Point-2), and scenario duration

TABLE 1. Statistical objective results (1st scenario).

	$f_1(\mathbf{u})$	$f_2(\mathbf{u})$	$f_3(\mathbf{u})$
Average	1.4195	1.3733	1.6066
Variance	0.1936	0.2427	0.2426

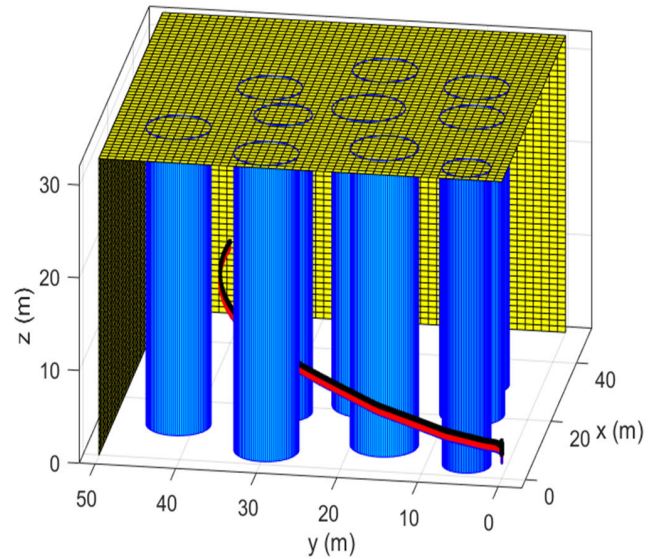


FIGURE 17. Resulting trajectory corresponding to the non-dominated solution in terms of path-length (3D view).

(Pareto Point-3), respectively. Additionally, the 3D representation of the solution corresponding to the minimum length is shown in Figure 24.

Since we normalized the objective function values between 1 and 2, it does not give the maximum and minimum values. Still, the mean and standard deviation values are given in Table 2 for the second scenario.

It can be seen that the quadrotor slows down near the obstacles (for example, see Figures 26 and 27). The effect of this slowdown on the time objective function is obvious. However, due to the possible accelerations and deceleration before and after this slowdown, the effect of the energy objective function may be different. We can compare this to the fuel consumption in traffic; while the short-term red light will slightly prolong our total driving time, the main effect

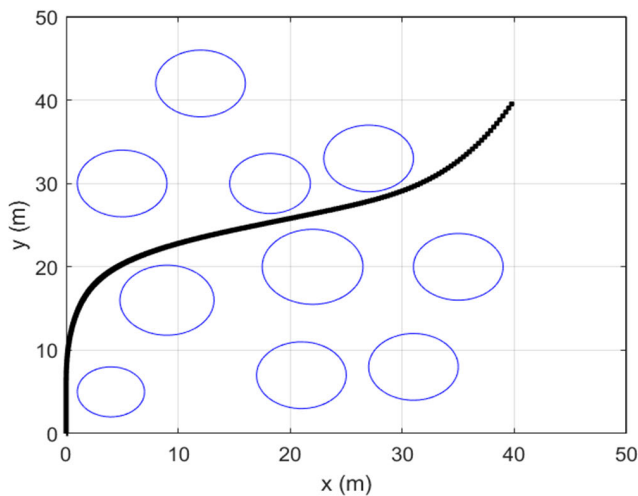


FIGURE 15. Resulting trajectory corresponding to the non-dominated solution in terms of path-length (3D view).

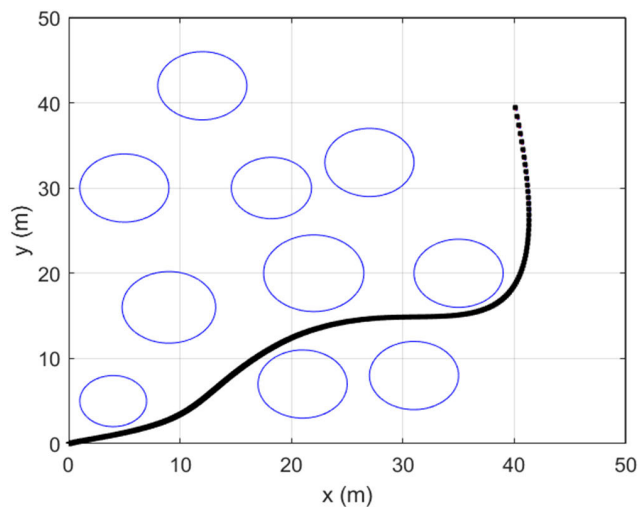


FIGURE 16. Resulting trajectory corresponding to the non-dominated solution in terms of path-length (3D view).

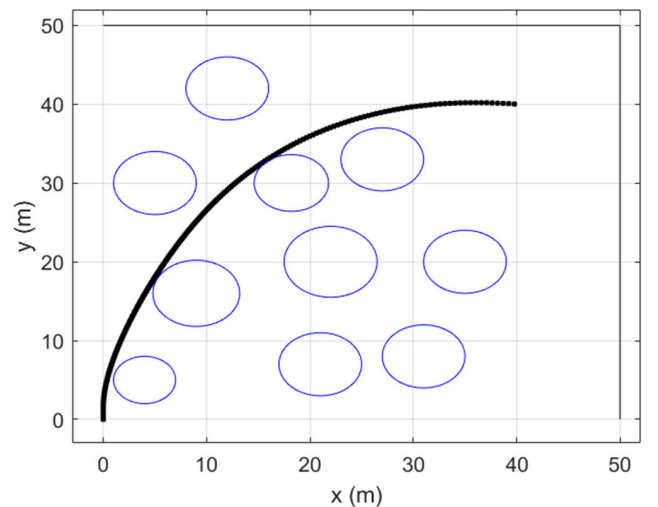


FIGURE 18. Resulting trajectory corresponding to the non-dominated solution in terms of path-length (top view).

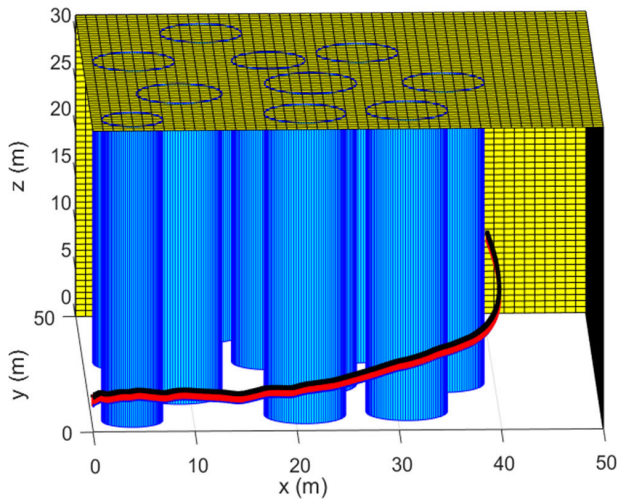


FIGURE 19. The resulting path corresponds to the non-dominated solution in terms of energy dissipation.

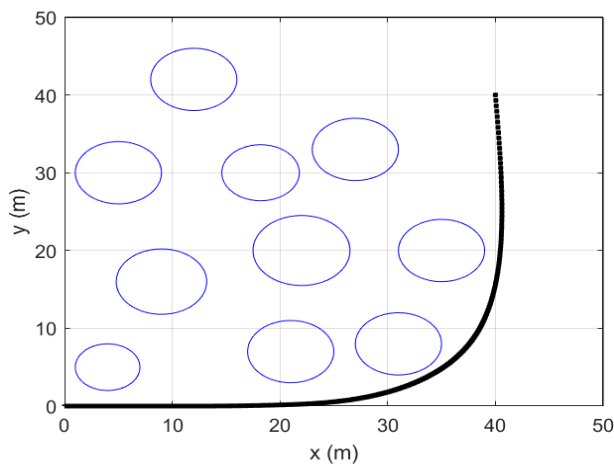


FIGURE 20. Resulting trajectory corresponding to the non-dominated solution in terms of energy dissipation (top view).

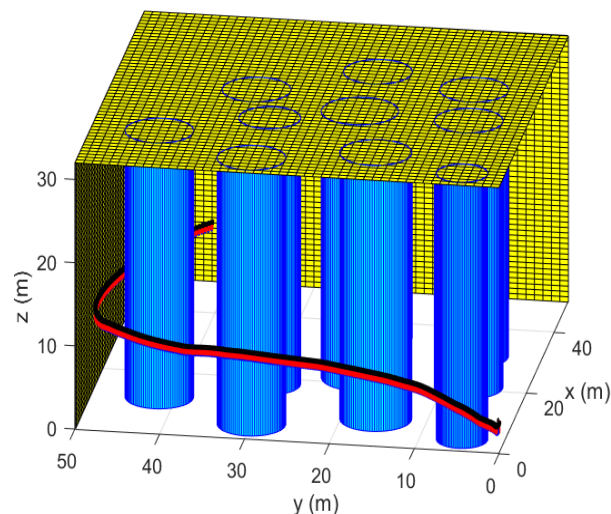


FIGURE 21. Resulting trajectory corresponding to the non-dominated solution in terms of scenario duration.

will be on fuel consumption due to stop-go or slow-speed movements. The positions of the obstacles with respect to

each other are also factors that directly affect the path-length objective function.

There is no study in the literature regarding the region of interest complexity. We have proposed a method to measure this complexity in our previous work [29], but this problem is still open to solving. From this point of view, we can say that obstacles are the most critical element in the trajectory planning problem. While determining the goal function, either knowledge about the complexity of the region in it should be produced or, as we suggest in our study, different goal functions should be looked at together. This study aimed to examine how different objective functions change when different regions are in question.

We think it is necessary to emphasize Scenario-1 and Scenario-2 covered in this study are pretty challenging.

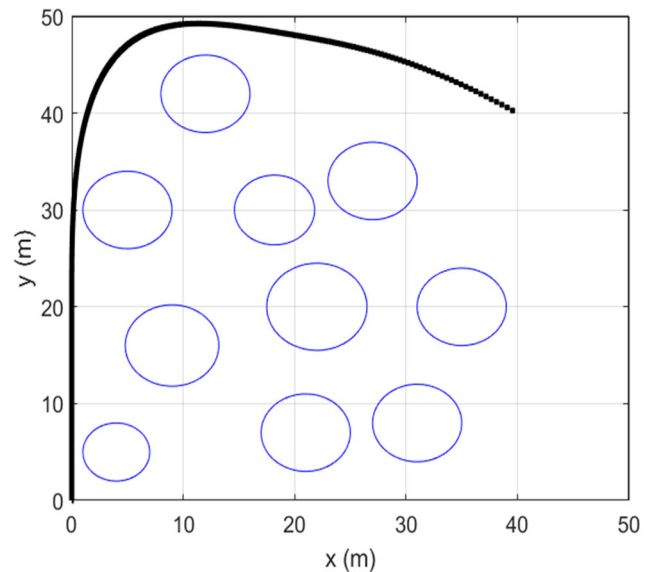


FIGURE 22. Resulting trajectory corresponding to the non-dominated solution in terms of scenario duration.

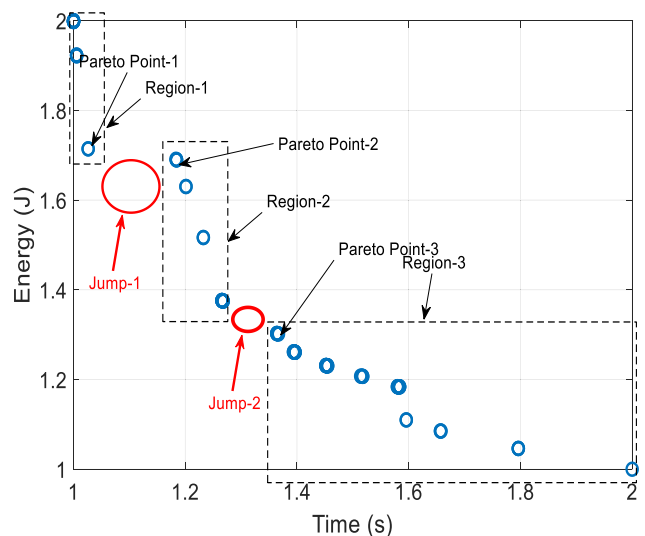


FIGURE 23. Pareto front for 2nd scenario, with the objective functions scenario duration and energy.

TABLE 2. Statistical objective results (2nd scenario).

	$f_1(\mathbf{u})$	$f_2(\mathbf{u})$	$f_3(\mathbf{u})$
Average	1.4232	1.3584	1.5945
Standard deviation	0.1969	0.2472	0.2809

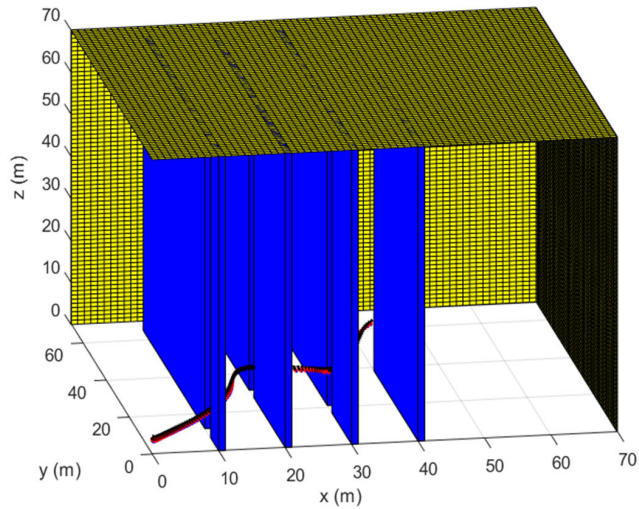


FIGURE 24. Resulting trajectory corresponding to the non-dominated solution in terms of scenario duration (3D view).

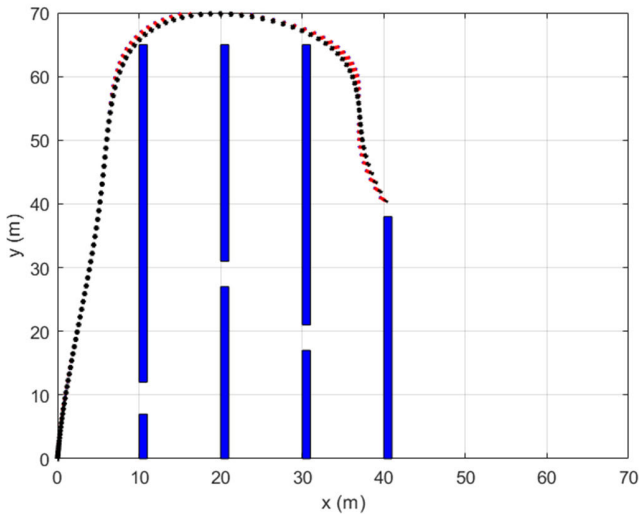


FIGURE 25. Top view of the resulting trajectory corresponding to the non-dominated solution in terms of energy.

Therefore, finding a Pareto Front with sufficient points and good spread with classical GA operators (i.e., MOGA) or MOPSO. This situation is investigated in a less challenging scenario (Scenario-3) using classical MOGA and MOPSO. We also have tried to solve multi-objective problems given in Scenario-1 and Scenario-2 by increasing the maximum number of generations to ten times that of our algorithm. They could only find trajectories given in Figure 28; they could not find the paths advancing through the obstacles as

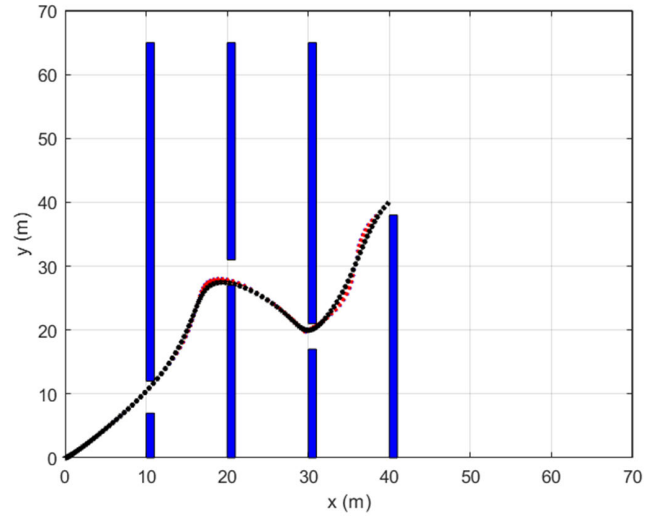


FIGURE 26. Resulting trajectory corresponding to the non-dominated solution in terms of path-length, top view of the trajectory shown in Figure 21.

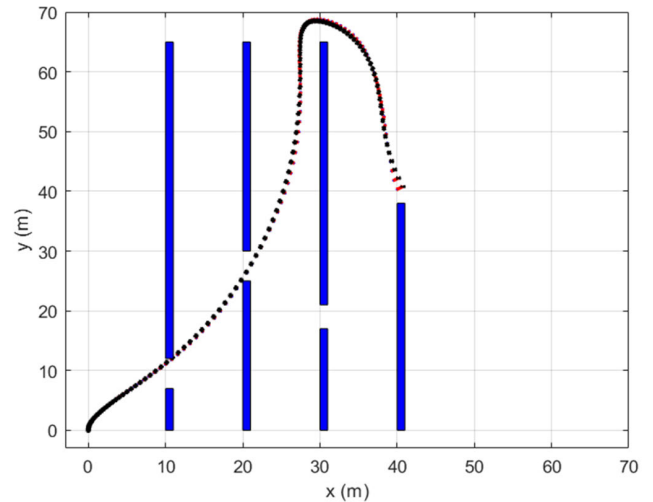


FIGURE 27. Resulting trajectory corresponding to the non-dominated solution in terms of time.

given in Figure 29. It cannot find the paths advancing through obstacles as given in Figure 26 and Figure 27. We tried to exemplify this situation through a less challenging scenario (Scenario-3) using MOGA and MOPSO.

In Figure 28, one can see from this figure; the slung-load quadrotor system reaches the final point by passing as close as possible to the obstacles thanks to the problem-specific operators. So, this solution dominates the solutions obtained using MOGA and MOPSO. In MOGA and MOPSO, it can find the trajectories that do not require too many maneuvers. Still, these two classical methods cannot solve complex trajectories, such as the route given in Figure 29.

We optimized the controller design for the Quadrotor slung-load system in the previous study [49]. Within the scope of this study, it is checked that the results are insensitive to the quadrotor controller's parameters as long as they are within 20% of the nominal parameters.

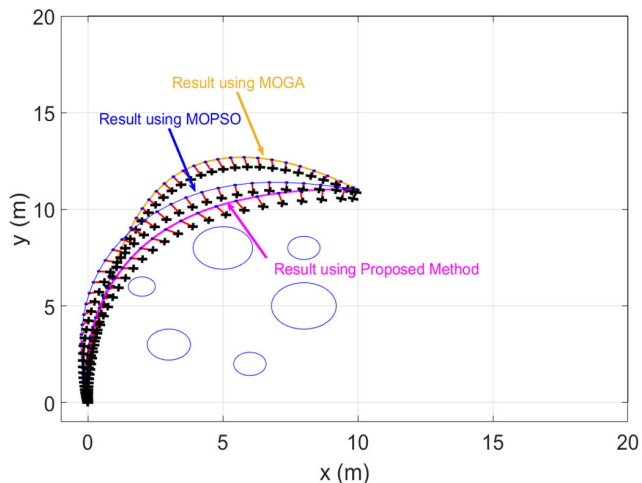


FIGURE 28. Resulting trajectories obtained using MOGA (yellow), MOPSO (blue), and the proposed method (pink).

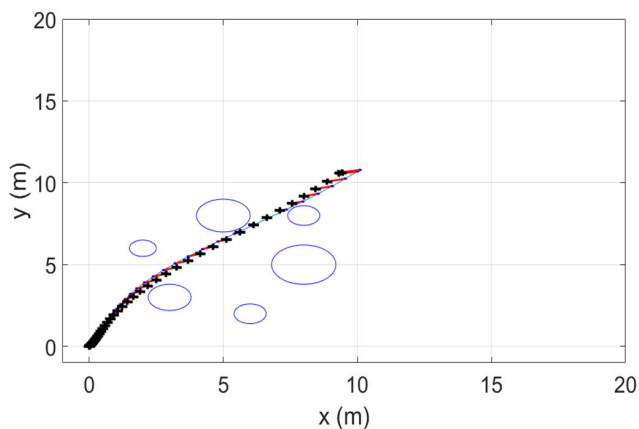


FIGURE 29. Resulting trajectory for the 3rd scenario obtained by the proposed method. MOGA and MOPSO cannot find a solution similar to this trajectory family.

Underlining the situation that the MOGA and MOPSO cannot find some trajectories that our proposed method can find, we compare the computation time required for an iteration of our method with the computation time required for an iteration of other algorithms over the 3rd scenario. Since our method performs extra calculations such as combined mutation, one iteration takes 1.6418 s, excluding model runs. For MOGA, this time is 1.4321 s; for MOPSO, it is 1.2123 seconds. Although the single iteration times are shorter, the number of iterations required to find similar trajectories is higher in MOGA and MOPSO. Hence, the total times are higher for these methods. Essentially, for this comparison to be more meaningful, all algorithms should have been able to find similar route families.

VI. PERFORMANCE EVALUATIONS

In single-objective optimization, it is reasonably easy to measure the quality of a particular solution; if the problem under consideration is a minimization problem, the smaller the objective function value, the better. However, evaluating

the quality of a Pareto set of solutions also requires examining several criteria, and 57 performance indicators are examined [42]. In our study, hypervolume [36], [43] is preferred to compare the proposed method with other methods. The hypervolume metric is sensitive to the scales of the objective functions; it is suggested to evaluate the [28] metric using normalized objective function values. Actually, in the Pareto Fronts, objective functions are evaluated as normalized values. Hypervolumes have been calculated using (Eq. 12) with normalized values [42].

$$HV(P, r) = L\left(\bigcup_{p \in P} [p; r]\right) \quad (12)$$

where P is the Pareto front approximation, r is the reference point dominated by all points in P , and L is the Lebesgue measure.

TABLE 3. Hypervolume results obtained from scenario-3 (reference point (2, 2, 2)).

	MOPSO	MOGA	Proposed method
Maximum	0.5091	0.5021	0.5329
Minimum	0.4723	0.4678	0.4905
Medium	0.4868	0.4789	0.5138

The gaps in the Pareto Fronts need special attention, as stated in [37]. We need to determine whether these gaps are due to the nature of the problem or an error in the algorithm. As mentioned earlier, there are two goals in the Multi-objective optimization problem: finding a well-converged and well-distributed set of Pareto-optimal solutions and choosing a single preferred solution from this set. Therefore, it must be determined that there is no solution (i.e., trajectory) associated with these gaps. A method for this issue is proposed in [37]. In the first stage of this method, it is investigated whether there is a gap or not. An Automated Gap finding procedure described in [37] has been used to find gap points ($G_j, j = 1, 2, \dots, M$). At the second stage, using the following two equations, we can find whether these gaps are formed due to the nature of the problem [37]. The method given in [40] has been used to find R near-PO point (i.e., RF_i)

$$d_{GE} = \frac{\frac{1}{M} \sum_{i=1}^M (\min_{j=1}^N (\|G_i - F_j\|_2))}{\frac{1}{N} \sum_{i=1}^N (\|F_i - F_j\|_2)} \quad (13)$$

The metric d_{GE} is the average normalized distance of gap-points G_i from their closest points on Pareto Optimal front F_j to the points on the Pareto Optimal. If this metric value is more significant than one (i.e., $d_{GE} > 1$), the obtained gap-point can be confirmed to exist.

$$d_{RG} = \frac{1}{R} \sum_{i=1}^R \frac{\min_{j=1}^K \|RF_i - G_j\|_2}{\min_{j=1}^N \|RF_i - F_j\|_2} \quad (14)$$

The metric d_{RG} is the average distance RF_i points to the gap-point compared to their distance from points on Pareto Optimal front. If $d_{RG} > 1$ there exists a natural gap in the

Pareto-set, and it means it is impossible to find any solution in the gap region. If $d_{RG} < 1$ the observed gap is not natural; it is due to the problems in the algorithm. The metric d_{RG} is calculated as 1.151 and 1.782 for the first and the second scenarios, respectively.

VII. CONCLUSION

In this study, multi-objective trajectory planning has been implemented for a slung-load quadrotor system. The path length, scenario duration, and the dissipated energy are taken as objectives (costs). In the literature, multi-objective optimization problems are frequently solved by transforming them into a single-objective optimization problem for a weighted sum of costs (scalarization). However, determining proper weights for scalarization is an arduous task. Also, the scalarization approach prevents the investigation of the problem strategically from the perspective of objectives. In our practice, we can easily choose the proper weights by observing, especially the points corresponding to jumps on the Pareto Front (whenever hops exist).

We think that Pareto Front must be obtained before converting from a multi-objective optimization problem to a single objective problem in the type we have discussed here, where more than one objective is essential, and the importance of these objectives may vary according to the mission to be performed. As we emphasized in the results section, there may be points where we can't solve the problem by giving different weights to the single objective function. Therefore, before looking for a solution to the problem, we need to see a solution, if any.

Using the Pareto front, one may determine a point with short mission duration, sufficiently small energy expenditure, and not too long path length. Then, we can find the associated trajectory together with planned input. Finally, the physical vehicle is run with these inputs (Actually, assuming the vehicle has an autopilot, the obtained trajectory will be used as a guidance command for the autopilot).

One could add safety to these objective functions (can be considered the load's swing angle). However, we added this as a constraint in our problem. In the future, trajectory planning can be performed by evaluating the security objective over the scenarios that can pass over the obstacles or as a constraint in the vertical direction. Besides, we plan to extend the method and the solution algorithm to cases where more than one, two, or three quadrotors carry the load. Our algorithm will work very well as long as the obstacle locations are known. However, we plan to extend the present algorithm to cases with uncertainties in obstacle locations.

REFERENCES

- [1] M. Bisgaard, "Modeling, estimation, and control of helicopter slung load system," Ph.D. dissertation, Dept. Electron. Syst. Sect. Automat. Control, Aalborg Univ., Aalborg, Denmark, 2007.
- [2] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach," *Soft Comput.*, vol. 21, no. 4, pp. 949–964, Feb. 2017.
- [3] D. Zamoski, G. Starr, J. Wood, and R. Lumia, "Rapid swing-free transport of nonlinear payloads using dynamic programming," *J. Dyn. Syst., Meas., Control*, vol. 130, no. 4, Jul. 2008, doi: [10.1115/1.2936384](https://doi.org/10.1115/1.2936384).
- [4] S. Hayat, E. Yanmaz, T. X. Brown, and C. Bettstetter, "Multi-objective UAV path planning for search and rescue," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017, pp. 5569–5574, doi: [10.1109/ICRA.2017.7989656](https://doi.org/10.1109/ICRA.2017.7989656).
- [5] K. Sreenath, N. Michael, and V. Kumar, "Trajectory generation and control of a quadrotor with a cable-suspended load—A differentially-flat hybrid system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4888–4895.
- [6] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2691–2697.
- [7] K. Klausen, T. I. Fossen, and T. A. Johansen, "Nonlinear control of a multirotor UAV with suspended load," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2015, pp. 176–184.
- [8] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load," in *Proc. 52nd IEEE Conf. Decis. Control*, Dec. 2013, pp. 2269–2274.
- [9] T. Lee, "Geometric control of quadrotor UAVs transporting a cable-suspended rigid body," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1, pp. 255–264, Jan. 2018.
- [10] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.
- [11] S. Geetha, G. M. Chitra, and V. Jayalakshmi, "Multi objective mobile robot path planning based on hybrid algorithm," in *Proc. 3rd Int. Conf. Electron. Comput. Technol.*, Apr. 2011, pp. 251–255, doi: [10.1109/ICECTECH.2011.5942092](https://doi.org/10.1109/ICECTECH.2011.5942092).
- [12] W. Hao and S. Qin, "Multi-objective path planning for space exploration robot based on chaos immune particle swarm optimization algorithm," in *Artificial Intelligence and Computational Intelligence (Lecture Notes in Computer Science)*, vol. 7003, H. Deng, D. Miao, J. Leis, and F. Wang, Eds. Berlin, Germany: Springer, 2011, pp. 42–52, doi: [10.1007/978-3-642-23887-1_6](https://doi.org/10.1007/978-3-642-23887-1_6).
- [13] E. Masehian and D. Sedighzadeh, "Multi-objective PSO- and NPSO-based algorithms for robot path planning," *Adv. Electr. Comput. Eng.*, vol. 10, no. 4, pp. 69–76, 2010, doi: [10.4316/AECE.2010.04011](https://doi.org/10.4316/AECE.2010.04011).
- [14] B. Saicharan, R. Tiwari, and N. Roberts, "Multi objective optimization based path planning in robotics using nature inspired algorithms: A survey," in *Proc. IEEE 1st Int. Conf. Power Electron., Intell. Control Energy Syst. (ICPEICES)*, New Delhi, India, Jul. 2016, pp. 1–6, doi: [10.1109/ICPEICES.2016.7853442](https://doi.org/10.1109/ICPEICES.2016.7853442).
- [15] B. Xian, S. Wang, and S. Yang, "An online trajectory planning approach for a quadrotor UAV with a slung payload," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6669–6678, Aug. 2020, doi: [10.1109/TIE.2019.2938493](https://doi.org/10.1109/TIE.2019.2938493).
- [16] D. Hashemi and H. Heidari, "Trajectory planning of quadrotor UAV with maximum payload and minimum oscillation of suspended load using optimal control," *J. Intell. Robot. Syst.*, vol. 100, nos. 3–4, pp. 1369–1381, Dec. 2020, doi: [10.1007/s10846-020-01166-4](https://doi.org/10.1007/s10846-020-01166-4).
- [17] M. E. Guerrero, D. A. Mercado, R. Lozano, and C. D. Garcia, "Passivity based control for a quadrotor UAV transporting a cable-suspended payload with minimum swing," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2015, pp. 6718–6723, doi: [10.1109/CDC.2015.7403277](https://doi.org/10.1109/CDC.2015.7403277).
- [18] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2216–2222, doi: [10.1109/ICRA.2015.7139492](https://doi.org/10.1109/ICRA.2015.7139492).
- [19] I. Palunko, A. Faust, P. Cruz, L. Tapia, and R. Fierro, "A reinforcement learning approach towards autonomous suspended load manipulation using aerial robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4896–4901.
- [20] F. Yacef, N. Rizoug, O. Bouhali, and M. Hamerlain, "Optimization of energy consumption for quadrotor UAV," in *Proc. Int. Micro Air Vehicle Conf. Flight Competition (IMAV)*, Toulouse, France, Sep. 2017, pp. 215–222.
- [21] D. C. Gandolfo, L. R. Salinas, A. Brandao, and J. M. Toibero, "Stable path-following control for a quadrotor helicopter considering energy consumption," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1423–1430, Jul. 2017.

- [22] F. Morbidi, R. Cano, and D. Lara, "Minimum-energy path generation for a quadrotor UAV," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 2–8.
- [23] H. Lu, K. Chen, X. B. Zhai, B. Chen, and Y. Zhao, "Tradeoff between duration and energy optimization for speed control of quadrotor unmanned aerial vehicle," in *Proc. IEEE Symp. Product Compliance Eng. Asia (ISPCE-CN)*, Dec. 2018, pp. 1–5.
- [24] S. Spedicato and G. Notarstefano, "Minimum-time trajectory generation for quadrotors in constrained environments," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1335–1344, Jul. 2018, doi: [10.1109/TCST.2017.2709268](https://doi.org/10.1109/TCST.2017.2709268).
- [25] J.-H. Wei and J.-S. Liu, "Generating minimax-curvature and shorter n3-spline path using multi-objective variable-length genetic algorithm," in *Proc. Int. Conf. Netw., Sens. Control (ICNSC)*, Apr. 2010, pp. 319–324.
- [26] F. Ahmed and K. Deb, "Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms," *Soft Comput.*, vol. 17, no. 7, pp. 1283–1299, 2013.
- [27] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Rel. Eng. Syst. Saf.*, vol. 91, no. 9, pp. 992–1007, Sep. 2006.
- [28] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.
- [29] H. Ergezer and K. Leblebicioglu, "Path planning for UAVs for maximum information collection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 1, pp. 502–520, Jan. 2013.
- [30] H. Ergezer and K. Leblebicioglu, "3D path planning for UAVs for maximum information collection," *J. Intell. Robot. Syst.*, vol. 73, pp. 737–762, 2014.
- [31] J. Zeng, P. Kotaru, M. W. Mueller, and K. Sreenath, "Differential flatness based path planning with direct collocation on hybrid modes for a quadrotor with a cable-suspended payload," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3074–3081, Apr. 2020, doi: [10.1109/LRA.2020.2972845](https://doi.org/10.1109/LRA.2020.2972845).
- [32] R. Chai, A. Tsourdos, A. Savvaris, Y. Xia, and S. Chai, "Real-time reentry trajectory planning of hypersonic vehicles: A two-step strategy incorporating fuzzy multiobjective transcription and deep neural network," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6904–6915, Aug. 2020, doi: [10.1109/TIE.2019.2939934](https://doi.org/10.1109/TIE.2019.2939934).
- [33] R. Chai, A. Savvaris, A. Tsourdos, Y. Xia, and S. Chai, "Solving multi-objective constrained trajectory optimization problem by an extended evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1630–1643, Apr. 2020, doi: [10.1109/TCYB.2018.2881190](https://doi.org/10.1109/TCYB.2018.2881190).
- [34] R. Chai, A. Savvaris, A. Tsourdos, and S. Chai, "Multi-objective trajectory optimization of space manoeuvre vehicle using adaptive differential evolution and modified game theory," *Acta Astronautica*, vol. 136, pp. 273–280, Jul. 2017.
- [35] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. L. P. Chen, "Multiobjective optimal parking maneuver planning of autonomous wheeled vehicles," *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10809–10821, Dec. 2020, doi: [10.1109/TIE.2019.2962482](https://doi.org/10.1109/TIE.2019.2962482).
- [36] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, and Y. Xia, "Solving constrained trajectory planning problems using biased particle swarm optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 3, pp. 1685–1701, Jun. 2021, doi: [10.1109/TAES.2021.3050645](https://doi.org/10.1109/TAES.2021.3050645).
- [37] P. V. Pellicer, M. I. Escudero, S. F. Alzueta, and K. Deb, "Gap finding and validation in evolutionary multi- and many-objective optimization," in *Proc. Genetic Evol. Comput. Conf.*, Jun. 2020.
- [38] A. Rahimi, K. D. Kumar, and H. Alighanbari, "Particle swarm optimization applied to spacecraft reentry trajectory," *J. Guid., Control, Dyn.*, vol. 36, no. 1, pp. 307–310, 2012.
- [39] D. Spiller, C. Circi, and F. Curti, "Particle swarm with domain partition and control assignment for time-optimal maneuvers," *J. Guid., Control, Dyn.*, vol. 41, no. 4, pp. 968–977, Apr. 2018.
- [40] K. Deb, J. Sundar, N. Uday, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *Int. J. Comput. Intell. Res.*, vol. 2, no. 6, pp. 273–286, 2006.
- [41] T. Lee, "Computational geometric mechanics and control of rigid bodies," Ph.D. dissertation, Dept. Aerosp. Eng., Univ. Michigan, Ann Arbor, MI, USA, 2008.
- [42] C. Audet, J. Bignon, D. Cartier, S. L. Digabel, and L. Salomon, "Performance indicators in multiobjective optimization," *Eur. J. Oper. Res.*, vol. 292, no. 2, pp. 397–422, 2021.
- [43] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "Reference point specification in hypervolume calculation for fair comparison and efficient search," in *Proc. Genetic Evol. Comput. Conf.*, Berlin, Germany, Jul. 2017, pp. 585–592.
- [44] G. Harik, E. Cantu-Pas, D. E. Goldberg, and B. L. Miller, "The gambler's ruin problem, genetic algorithms, and the sizing of populations," *Evol. Comput. J.*, vol. 7, no. 3, pp. 231–254, 1999.
- [45] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Unified multi-objective optimization scheme for aeroassisted vehicle trajectory planning," *J. Guid., Control Dyn.*, vol. 41, no. 7, pp. 1521–1530, 2018.
- [46] T. Friedrich and M. Wagner, "Seeding the initial population of multi-objective evolutionary algorithms: A computational study," *Appl. Soft Comput.*, vol. 33, pp. 223–230, Aug. 2015.
- [47] A. G. Hernandez-Diaz, C. A. C. Coello, F. Perez, R. Caballero, J. Molina, and L. V. Santana-Quintero, "Seeding the initial population of a multi-objective evolutionary algorithm using gradient-based information," in *Proc. IEEE Congr. Evol. Comput., IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 1617–1624.
- [48] R. Chai, A. Savvaris, and A. Tsourdos, "Violation learning differential evolution-based hp-adaptive pseudospectral method for trajectory optimization of space maneuver vehicle," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 4, pp. 2031–2044, Aug. 2017.
- [49] A. Ates, O. Cosan, B. Degirmenci, H. Ergezer, and M. Ozgehan, "Controller design for quadrotor-slung load system with swing angle constraints using particle swarm optimization," in *Proc. Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Aug. 2021, pp. 1–6.



HALIT ERGEZER received the bachelor's degree in electrical and electronics engineering from Gazi University, in 2000, the master's degree in computer engineering from Baškent University, in 2003, and the Ph.D. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 2013.

From 2000 to 2006, he was a Research Assistant in computer engineering from Baškent University.

From 2006 to 2018, he worked as a Senior Design Engineer in various projects in different defense industry companies. He has been an Assistant Professor with the Department of Mechatronics Engineering, Çankaya University, since 2018. His research interests include robot motion planning with their applications and coordinated guidance of multiple UAVs. He also specializes in systems engineering, radar systems, and electronic warfare systems.

Dr. Ergezer was a recipient of the Thesis of the Year Award from Middle East Technical University, in 2013.

• • •