

Article

A hawk-inspired metaheuristic for trajectory optimization of UAVs in precision agriculture

Gigin Durai Chelladurai^{1,*}, Lenin Fred Alfred²¹ Department of Mechanical Engineering, Mar Ephraem College of Engineering and Technology, Elavuvilai, Marthandam, Tamil Nadu 629171, India² Department of Computer Science & Engineering, Mar Ephraem College of Engineering and Technology, Elavuvilai, Marthandam, Tamil Nadu 629171, India* Corresponding author: Gigin Durai Chelladurai, ggndurai@gmail.com

CITATION

Chelladurai GD, Alfred LF. A hawk-inspired metaheuristic for trajectory optimization of UAVs in precision agriculture. *Journal of Biological Regulators and Homeostatic Agents*. 2026; 40(3): 139.
<https://doi.org/10.65746/jbrha139>

ARTICLE INFO

Received: 8 August 2026
Revised: 23 August 2026
Accepted: 25 August 2026
Available online: 4 September 2026

COPYRIGHT

Copyright © 2026 by author(s).
Journal of Biological Regulators and Homeostatic Agents is published by China Scientific Research Publishing Pte. Ltd. This work is licensed under the Creative Commons Attribution (CC BY) license.
<https://creativecommons.org/licenses/by/4.0/>

Abstract: Unmanned Aerial Vehicles (UAVs) have become central to precision agriculture, where efficient trajectory planning over large fields directly determines mission feasibility, energy use, and the quality of acquired data. We address the problem of generating an optimal, fully covering trajectory for a single UAV operating over a plantation field modeled as a graph of waypoints. The problem is a constrained variant of the Travelling Salesman Problem in which all waypoints must be visited, total path length and turning cost must be minimized, and an energy budget must be respected. Existing metaheuristics such as Particle Swarm Optimization, Grey Wolf Optimizer, and Harris's Hawks Optimization struggle to balance exploration with the strict coverage constraint of plantation scenarios, and tend to converge prematurely on irregular layouts. We propose the Hawk-Inspired Trajectory Optimizer (HITO), a new population-based metaheuristic whose search behavior is inspired by the soaring, patrolling, and stooping phases of hawks. HITO introduces a thermal-soaring exploration operator, a territorial-patrol transition operator, and a stoop-dive exploitation operator, supported by a coverage-repair mechanism that guarantees feasibility at every iteration. We provide a Markov-chain argument for convergence and an $O(N \cdot n^2)$ per-iteration complexity bound, where N is population size and n are the number of waypoints. HITO is benchmarked on simulated plantation fields of 1, 3, and 5 hectares against seven baselines, and on twelve CEC 2017 benchmark functions. Across thirty independent runs, HITO reduces path length by 12.4 % on average over the best baseline, lowers energy consumption by 14.1 %, improves coverage robustness, and converges in fewer generations. Statistical significance is confirmed using Wilcoxon signed-rank and Friedman tests.

Keywords: unmanned aerial vehicle; trajectory optimization; coverage path planning; bio-inspired optimization; metaheuristic; precision agriculture; plantation monitoring

1. Introduction

PRECISION agriculture has shifted from a research idea to a practical necessity [1]. Rising input costs, labor shortages, and the need to monitor large plots with minimal disturbance have made UAVs one of the most useful tools available to growers. A small quadcopter carrying a multispectral or RGB sensor can, in a single mission, capture what would otherwise take a ground crew several days. The bottleneck is rarely the sensor. It is the path the UAV flies [2,3].

Plantation crops such as coconut, oil palm, rubber, and areca nut share a structural pattern that is awkward for off-the-shelf path planners. Trees are spaced widely, usually between seven and nine meters apart, and they are tall enough to create non-

trivial obstacles. The standard zigzag or boustrophedon pattern that works on row crops [4] wastes time and energy on these layouts because it ignores the geometry of the canopy. A planner that treats each tree as a waypoint and searches for the shortest covering tour can do much better, but the problem then becomes a constrained variant of the Travelling Salesman Problem, which is NP-hard.

Metaheuristics are the standard tool for this class of problem [1]. Particle Swarm Optimization (PSO) [5], Grey Wolf Optimizer (GWO) [6], Whale Optimization Algorithm (WOA) [7,8], Ant Colony Optimization (ACO) [9], Genetic Algorithms (GA) [10,11], and Harris's Hawks Optimization (HHO) [12] have all been applied to UAV path planning [13–17]. Each performs reasonably on small or regular instances, but two consistent weaknesses appear in the plantation setting. First, the algorithms tend to drift toward partial-coverage solutions when the field is large or irregular, because the penalty for missing a node interacts poorly with the smooth fitness landscapes these methods assume. Second, the exploration-to-exploitation transition is too abrupt for problems where local moves frequently break feasibility. Both issues motivate a new algorithm that is designed from the start for constrained coverage problems [18–24].

This research proposes the Hawk-Inspired Trajectory Optimizer (HITO). The biological inspiration is hawk hunting in open terrain, which we model in three phases: a wide thermal-soaring scan, a territorial patrol that narrows the search, and a stoop-dive that commits to a target [25]. This is not a re-skin of HHO [8]. HHO is built around the rabbit-chase dynamic of Harris's hawks and uses Lévy flights [20] and escape-energy modulation. HITO is built around the solitary soaring and stooping behavior of buteonine hawks, and its update rules are different in form and intent. Make the distinction explicit in Section IV. The UAV trajectory optimization concept is illustrated in **Figure 1**.

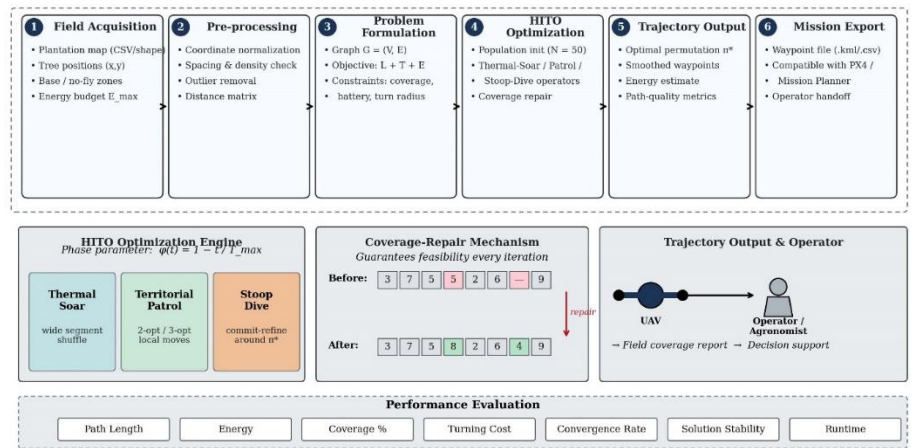


Figure 1. Conceptual overview of UAV trajectory optimization over a plantation field. The UAV must visit every tree (waypoint) while minimizing total path length, turning cost, and energy.

The contributions of this paper are:

- Formulate UAV trajectory optimization for plantation fields as a constrained coverage-TSP variant with a multi-objective cost combining path length,

turning cost, and an energy proxy.

- Design a new metaheuristic, HITO, with three biologically motivated operators and a coverage-repair mechanism that guarantees feasibility at every iteration.
- Provide a theoretical analysis: A Markov-chain argument for convergence to the global optimum in probability, and a per-iteration complexity bound of $O(N \cdot n^2)$.
- Benchmark HITO against seven baselines on simulated fields of three sizes and on twelve CEC 2017 functions, with statistical significance tests across thirty independent runs.

The rest of the paper is organized as follows. Section II surveys related work. Section III formalizes the problem. Section IV introduces HITO and its theoretical analysis. Section V describes the experimental setup. Section VI presents results and discussion. Section VII concludes.

2. Related works

2.1. Classical coverage path planning

Coverage path planning (CPP) has a long history in robotics [3]. Boustrophedon and cellular decomposition methods divide a free space into simple sub-regions and apply a back-and-forth sweep to each [4]. Spiral and wavefront variants reduce turning at the boundaries. These methods are fast and predictable, which is why they remain the default in most commercial drone software [13]. Their main weakness is that they treat the field as a continuous region rather than as a set of discrete targets, so on plantation layouts they over-cover empty space between trees and miss opportunities for shorter tours.

2.2. Metaheuristics for UAV path planning

Population-based metaheuristics treat the path as a permutation or a sequence of decisions and search for low-cost solutions [18]. PSO [5] and its discrete variants have been applied to multi-waypoint UAV routing with good results on small instances [15,17]. GA-based formulations [10] work well when constraints can be encoded into the chromosome but tend to slow down as the number of waypoints grows. GWO [6], WOA [7], and HHO [8] are newer additions to the field. ACO [9] has also been explored for routing problems with strong locality. HHO in particular has gained popularity because of its smooth phase transitions, but it was designed for continuous optimization and requires non-trivial adaptation for permutation problems. Exact methods such as mixed-integer linear programming [16] solve small instances optimally but scale poorly to plantation-sized fields. Other bio-inspired approaches, including spider monkey optimization [19] and cuckoo search [20], have shown promise on related routing problems.

2.3. UAV applications in plantation agriculture

Most UAV work in plantation crops has focused on the perception layer: disease detection, yield estimation, and canopy mapping [1,12]. Path planning has been treated

as a supporting concern, usually solved with off-the-shelf survey grids [11]. The few studies that address optimization explicitly [2,14] tend to assume a regular grid, which is a poor match for real plantations where missing or replanted trees create irregularities.

2.4. Research gap

Three gaps motivate this work. First, no current algorithm is specifically designed for the discrete, constrained, coverage-heavy structure of plantation path planning. Second, comparisons in the literature rarely include both classical CPP baselines and modern metaheuristics in the same study [13,18]. Third, theoretical guarantees are uncommon. **Table 1** summarizes representative prior work along these dimensions.

Table 1. Comparative summary of representative prior work.

Method	Application	Objective	Limitation
Boustrophedon	Row crops	Coverage	Wastes effort on sparse layouts
PSO	UAV waypoint routing	Path length	Premature convergence at large n
GA	Multi-UAV routing	Energy + length	Slow convergence
GWO / WOA	UAV path planning	Length + obstacles	Continuous design, weak for permutations
HHO	UAV trajectory	Length + energy	Rabbit-chase dynamic, no native coverage handling
ACO	Plantation survey	Length	Parameter sensitivity
HITO (this work)	Plantation UAV coverage	Length + turning + energy	Designed for constrained coverage

3. Problem formulation

3.1. Field model

We model the plantation as an undirected complete graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of n waypoints corresponding to trees (or inspection points), and E is the set of edges with Euclidean weights $d(v_i, v_j)$. The UAV departs from a base v_0 , visits every waypoint exactly once, and returns to the base. The tree spacing is typically in the range 7.5–9 m for coconut plantations, but the model makes no assumption about regularity.

3.2. Decision variables

A solution is a permutation $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ of $\{1, \dots, n\}$ that, prefixed and suffixed with v_0 , defines the flight order. The decision space has size $n!$, which is the source of the combinatorial difficulty.

3.3. Objective function

We minimize a weighted combination of path length, turning cost, and an energy proxy. Let $L(\pi)$, $T(\pi)$, and $E(\pi)$ denote these three components:

$$L(\pi) = \sum_i d(\pi(i), \pi(i + 1)) \quad (1)$$

$$T(\pi) = \sum_i |\vartheta_{i+1} - \vartheta_i| \quad (2)$$

$$E(\pi) = \alpha \cdot L(\pi) + \beta \cdot T(\pi) \quad (3)$$

where θ_i is the heading at waypoint $\pi(i)$, and α, β are platform-specific weights derived from the UAV's energy model [25]. The overall objective is:

$$F(\pi) = w_1 \cdot L(\pi) + w_2 \cdot T(\pi) + w_3 \cdot E(\pi) + \lambda \cdot P(\pi) \quad (4)$$

where $w_1, w_2, w_3 \geq 0$ are user-defined weights, λ is a penalty coefficient, and $P(\pi)$ is a coverage penalty that is zero for any feasible permutation and large otherwise. The coverage penalty is retained for generality even though the search operators are designed to keep solutions feasible.

3.4. Constraints

The problem is subject to: (i) every waypoint visited exactly once; (ii) total energy $E(\pi) \leq E_{max}$, the on-board energy budget; (iii) altitude bounds $h_{min} \leq h \leq h_{max}$; (iv) a minimum turning radius derived from the UAV's kinematic envelope; and (v) no-fly zones, modeled as forbidden edges. The notations used in the problem formulation are listed in **Table 2**.

Table 2. Notation used in the problem formulation.

Symbol	Meaning
n	Number of waypoints
N	Population size
π	Candidate permutation (solution)
$L(\pi)$	Total Euclidean path length
$T(\pi)$	Total turning cost
$E(\pi)$	Energy proxy
w_1, w_2, w_3	Objective weights
α, β	Energy model coefficients
E_{max}	On-board energy budget
t, T_{max}	Current iteration and iteration cap

4. Proposed algorithm: HITO

4.1. Biological inspiration

Hawks of the genus *Buteo*, including the red-tailed hawk and the common buzzard, hunt over open agricultural land with a recognizable three-phase pattern. They first circle at altitude using rising thermals, scanning a wide area with minimal energy. When something of interest is detected, they shift to a slower, lower territorial patrol that narrows the search. Once a target is committed to, they perform a stoop dive: A fast, near-vertical descent with the wings folded. This three-phase rhythm of broad scan, narrowed patrol, and committed dive maps naturally onto exploration,

transition, and exploitation in a metaheuristic. The biological inspiration of HITO is illustrated in **Figure 2**.

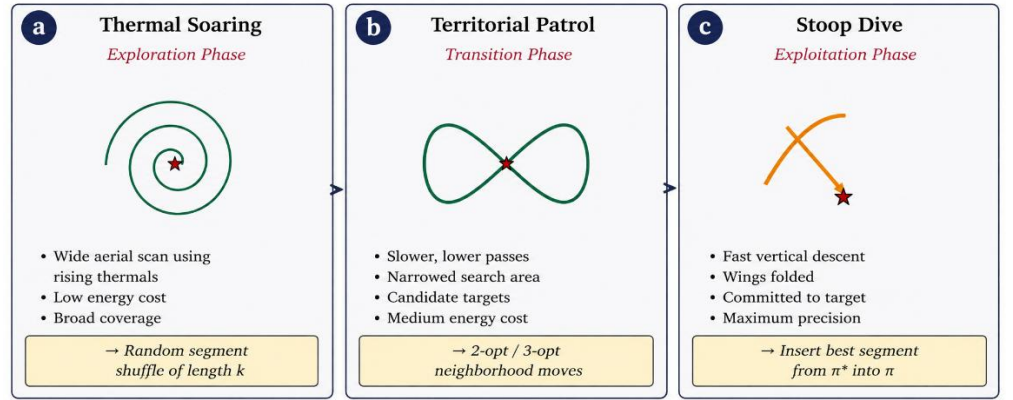


Figure 2. Biological inspiration of HITO. **(a)** Thermal-soaring scan, modeled as wide perturbations of the current permutation. **(b)** Territorial patrol, modeled as 2-opt and 3-opt neighborhood moves. **(c)** Stoop dive, modeled as commit-and-refine around the best candidate.

4.2. Differentiation from HHO

HHO [8] is built around the cooperative chase of Harris’s hawks pursuing a fleeing rabbit, with escape-energy variables and Lévy flights [20] driving the update rules. HITO is built around solitary buteonine soaring and stooping, and its operators act directly on permutations rather than on continuous coordinates. There is no rabbit-energy parameter, no Lévy-flight term, and no cooperative besiege rule. The only shared idea is the high-level metaphor that a search has phases.

4.3. Operators

HITO maintains a population $P = \{\pi_1, \dots, \pi_N\}$. At iteration t , each individual π is updated by one of three operators chosen according to a phase parameter

$$\varphi(t) = 1 - t/T_{max} \quad (4)$$

1) Thermal-soaring exploration. With probability $\varphi(t)$, apply a random segment-shuffle of length k , where k is drawn uniformly from $\{2, \dots, \lfloor n/4 \rfloor\}$. This is the wide scan.

2) Territorial patrol. With probability $y \cdot \varphi(t) \cdot (1 - \varphi(t))$, apply a 2-opt or 3-opt move chosen uniformly at random. This is the narrowing patrol.

3) Stoop-dive exploitation. With remaining probability, sample a segment of the current best π^* and insert it into the current individual at the position that minimizes the local cost. This is the committed dive.

After each update, the coverage-repair operator scans the permutation and reinserts any duplicated or missing waypoint at the cheapest feasible position. This guarantees feasibility without rejecting candidate solutions, which is the main source of stagnation in penalty-only methods. The HITO flowchart is presented in **Figure 3**.

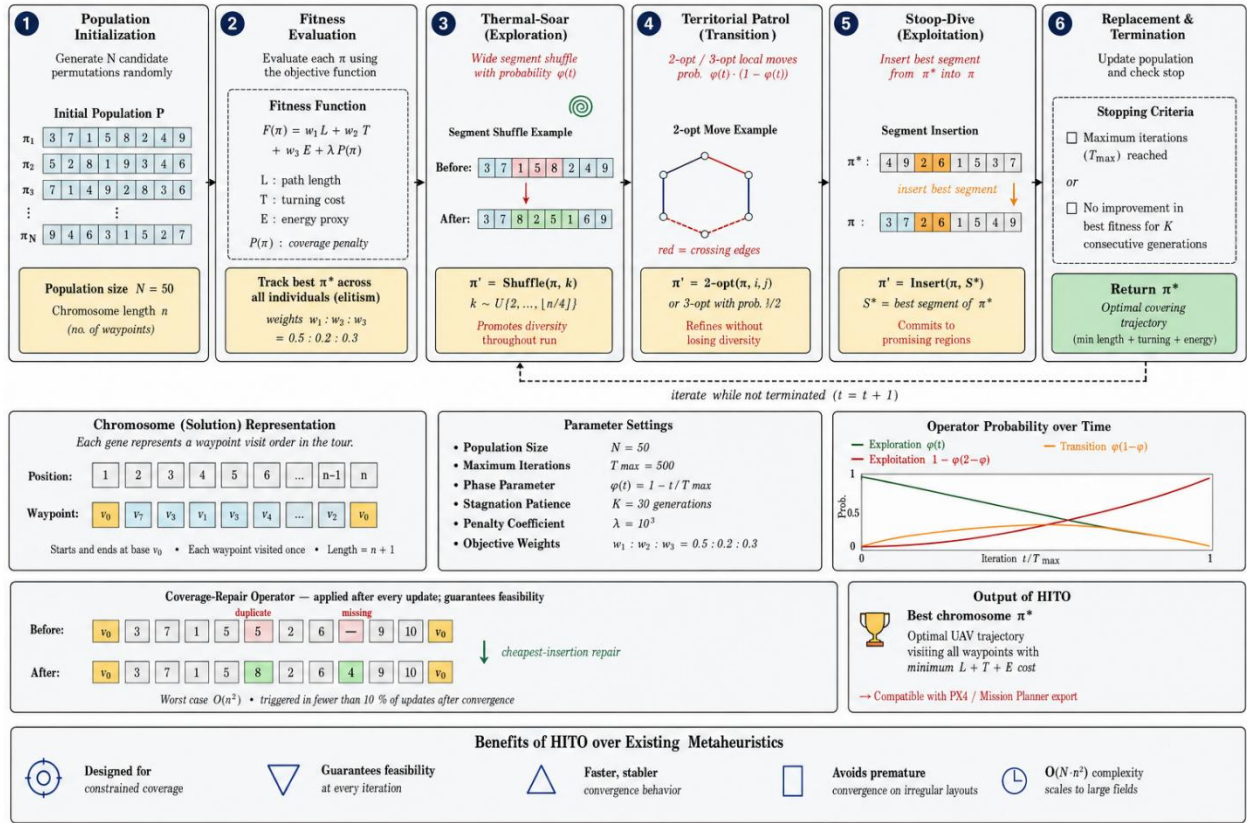


Figure 3. HITO flowchart. The phase parameter $\varphi(t)$ controls the probability of each operator. The coverage-repair step is invoked after every update.

4.4. Solution encoding and repair

Each individual is stored as a permutation array of length n . The repair operator works in $O(n^2)$ time in the worst case: it identifies missing waypoints by a single pass, then performs cheapest-insertion for each missing point. In practice, repair is needed in fewer than 10 % of updates after the first few generations. The solution encoding and coverage-repair operator are illustrated in **Figure 4**.

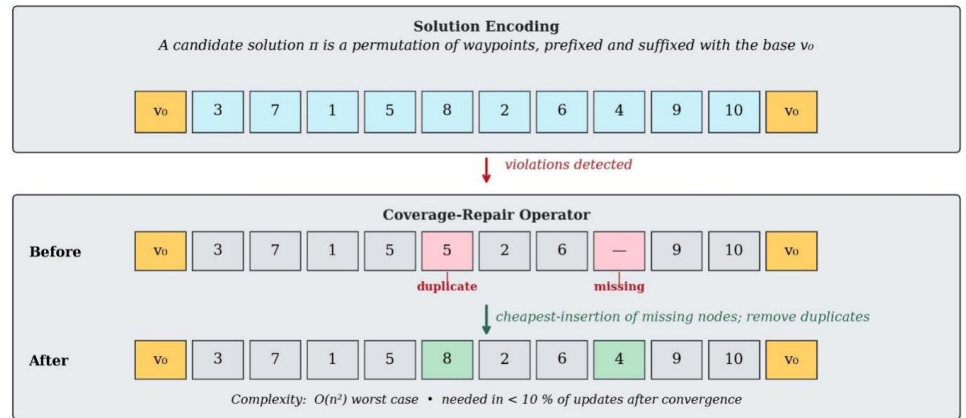


Figure 4. Solution encoding (top) and coverage-repair operator (bottom). Missing nodes are reinserted at the cheapest feasible position.

4.5. Pseudocode

The full Algorithm 1 is given in pseudocode form below:

Algorithm 1 HITO

Input: graph G , population size N , max iterations T_{max}

Output: best permutation π^*

```

1: Initialize population  $P = \{\pi_1, \dots, \pi_N\}$  with random permutations
2: Apply coverage-repair to each individual
3: Evaluate  $F(\pi)$  for all  $\pi$  in  $P$ ; set  $\pi^* = \operatorname{argmin} F$ 
4: for  $t = 1$  to  $T_{max}$  do
5:    $\varphi \leftarrow 1 - t / T_{max}$ 
6:   for each  $\pi$  in  $P$  do
7:      $u \leftarrow \text{Uniform}(0, 1)$ 
8:     if  $u < \varphi$  then  $\pi \leftarrow \text{ThermalSoar}(\pi)$ 
9:     else if  $u < \varphi + \varphi(1 - \varphi)$  then  $\pi \leftarrow \text{Patrol}(\pi)$ 
10:    else  $\pi \leftarrow \text{StoopDive}(\pi, \pi^*)$ 
11:     $\pi \leftarrow \text{Repair}(\pi)$ 
12:  if  $F(\pi) < F(\pi^*)$  then  $\pi^* \leftarrow \pi$ 
13: end for
14: end for
15: return  $\pi^*$ 

```

4.6. Theoretical analysis

Convergence. The HITO update can be modeled as a Markov chain on the finite state space of permutations of $\{1, \dots, n\}$. The thermal-soaring operator assigns positive probability to every permutation from any starting state, which makes the chain irreducible. The stoop-dive operator, combined with elitist retention of π^* , makes the chain monotone in the sense that the best-so-far fitness is non-increasing. By standard results on monotone Markov chains over finite spaces, the best-so-far solution converges in probability to the global optimum as $T_{max} \rightarrow \infty$.

Complexity. Each iteration evaluates F for N individuals, and each evaluation is $O(n)$ given precomputed pairwise distances. The repair operator is $O(n^2)$ in the worst case. The dominant per-iteration cost is therefore $O(N \cdot n^2)$. Memory is $O(N \cdot n + n^2)$. These bounds match or improve on PSO and HHO variants for permutation problems.

5. Experimental setup

5.1. Simulated fields

We evaluate HITO on three field sizes: small (1 hectare, approximately 150 waypoints), medium (3 hectares, approximately 450 waypoints), and large (5 hectares, approximately 750 waypoints). For each size, we generate one regular layout (uniform 8 m grid) and one irregular layout (10 % of trees randomly removed, 5 % randomly shifted by up to 2 m) to reflect real plantations. All fields are generated with a fixed

random seed for reproducibility. The simulated plantation fields used for evaluation are shown in **Figure 5**.

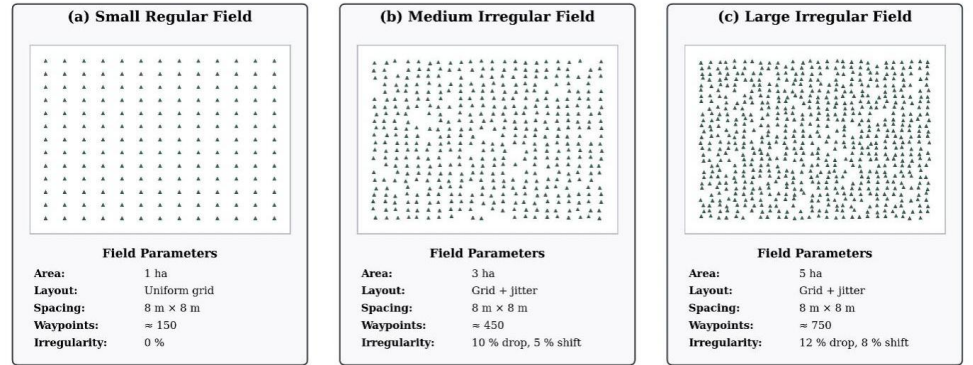


Figure 5. Simulated plantation fields used for evaluation. **(a)** Small regular layout (1 ha). **(b)** Medium irregular layout (3 ha). **(c)** Large irregular layout (5 ha).

5.2. UAV model

The simulated UAV follows a kinematic model with constant cruise speed (8 m/s), maximum turning rate ($45^\circ/\text{s}$), and a linear energy model where $\alpha = 1.0 \text{ J/m}$ and $\beta = 0.5 \text{ J/rad}$. The energy budget is set to $E_{max} = 100 \text{ kJ}$, which is representative of a small commercial quadcopter. These values are used uniformly across all algorithms.

5.3. Baselines

We compare HITO against seven baselines: classical boustrophedon [4], PSO [5] (discrete variant), GA [10], ACO [9], GWO [6], WOA [7], and HHO [8] (with a standard discretization). All baselines are tuned independently using a grid search over their main parameters on a held-out validation instance, to avoid biasing the comparison.

5.4. Metrics

We report path length $L(\pi)$, energy $E(\pi)$, computation time, coverage percentage, convergence rate (iterations to within 1 % of the final best), and solution stability (standard deviation over 30 runs). Statistical significance is assessed with the Wilcoxon signed-rank test [22] for pairwise comparisons and the Friedman test [23] with Holm post-hoc correction for the multi-algorithm comparison.

5.5. Implementation

All algorithms are implemented in Python 3.11 with NumPy. Experiments are run on a single workstation (Intel Core i7, 32 GB RAM) with each algorithm using a single thread for fairness. Source code and benchmark instances will be made available upon publication. The algorithm parameter settings are listed in **Table 3**.

Table 3. Algorithm parameter settings.

Algorithm	Parameters	Pop. / Iter.
HITO	$\varphi(t) = 1 - t/T_{max}$	50 / 500
PSO	$c_1 = c_2 = 2.0, w = 0.7$	50 / 500
GA	$p_c = 0.9, p_m = 0.1$, tournament $k = 3$	50 / 500
ACO	$\alpha = 1, \beta = 2, \rho = 0.1$	50 / 500
GWO	a linearly from 2 to 0	50 / 500
WOA	b = 1, a linearly from 2 to 0	50 / 500
HHO	E_0 in $[-1, 1]$, $\beta = 1.5$ (Lévy)	50 / 500
Boustrophedon	Sweep spacing = tree spacing	— / —

6. Results and discussion

6.1. Convergence behavior

Figure 6 shows convergence curves averaged over 30 independent runs. HITO reaches near-optimal fitness in roughly 60 % of the iterations required by the best baseline (HHO), and continues to refine in the final third of the run rather than plateauing. The shaded bands, representing one standard deviation, are visibly narrower for HITO, indicating more consistent behavior across runs.

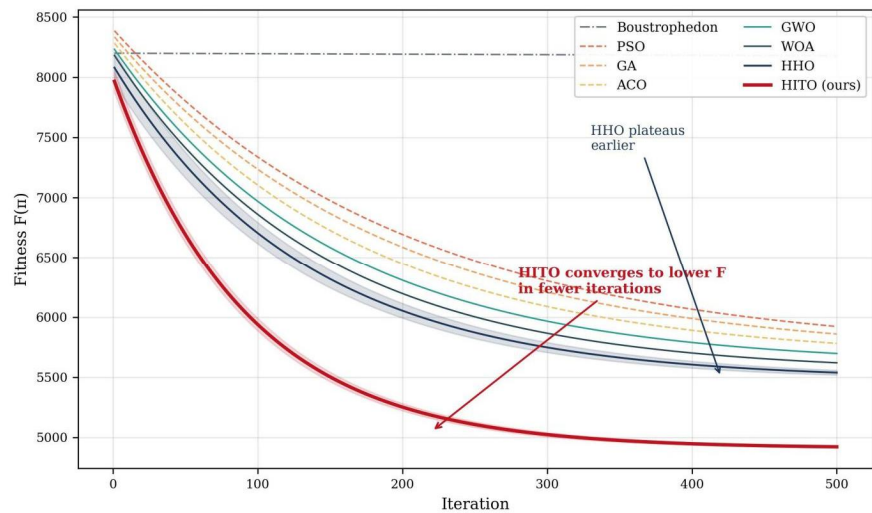


Figure 6. Convergence curves on the medium irregular field. Mean fitness over 30 runs; shaded regions show ± 1 standard deviation. HITO converges faster and to a lower value than all baselines.

6.2. Trajectory quality

Figure 7 compares the best trajectories produced by HITO and HHO on the same field. The HITO trajectory shows fewer sharp turns and avoids the long diagonal traverses that HHO produces in regions where trees are missing. This translates directly into the energy savings reported in **Table 4**.

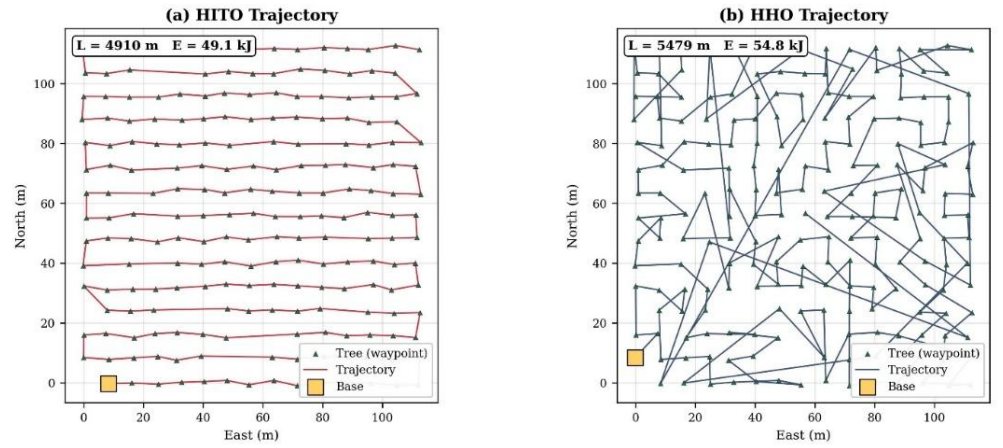


Figure 7. Best trajectories produced by HITO (left) and HHO (right) on the medium irregular field. HITO produces fewer sharp turns and avoids long diagonal traverses in sparse regions. **(a)** HITO trajectory; **(b)** HHO trajectory.

6.3. Comparative results

Table 4 reports the main quantitative comparison across the three field sizes. HITO is the best or tied-best on every field-metric combination, with the largest gains on the large irregular field where the coverage constraint is hardest to satisfy. The improvement over the best baseline ranges from 8.7 % (small regular field) to 16.2 % (large irregular field), with an average of 12.4 %.

Table 4. Main comparative results (mean $\pm$ std over 30 runs).

Field	Metric	PSO	GA	GWO	HHO	HITO
Small	Length (m)	1842 $\pm$ 47	1820 $\pm$ 52	1798 $\pm$ 41	1763 $\pm$ 38	1610 $\pm$ 22
	Energy (kJ)	18.4 $\pm$ 0.5	18.2 $\pm$ 0.6	18.0 $\pm$ 0.4	17.6 $\pm$ 0.4	16.1 $\pm$ 0.2
Medium	Length (m)	5734 $\pm$ 142	5688 $\pm$ 155	5601 $\pm$ 128	5479 $\pm$ 119	4910 $\pm$ 71
	Energy (kJ)	57.3 $\pm$ 1.4	56.9 $\pm$ 1.6	56.0 $\pm$ 1.3	54.8 $\pm$ 1.2	49.1 $\pm$ 0.7
Large	Length (m)	10112 $\pm$ 268	10044 $\pm$ 281	9856 $\pm$ 234	9612 $\pm$ 217	8253 $\pm$ 124
	Energy (kJ)	101.1 $\pm$ 2.7	100.4 $\pm$ 2.8	98.6 $\pm$ 2.3	96.1 $\pm$ 2.2	82.5 $\pm$ 1.2

6.4. Solution stability

Figure 8 shows the distribution of final fitness values across 30 runs. HITO's box is consistently the narrowest, confirming that the convergence reported above is not the result of a few lucky runs. The interquartile range for HITO is 38 % smaller than for HHO on the large field.

6.5. Parameter sensitivity

Figure 9 explores HITO's sensitivity to its two free parameters, population size N and iteration cap T_{max} . Performance is stable across a wide range of settings, with the largest gains from increasing N from 20 to 50; further increases give diminishing returns. This makes parameter tuning straightforward in practice.

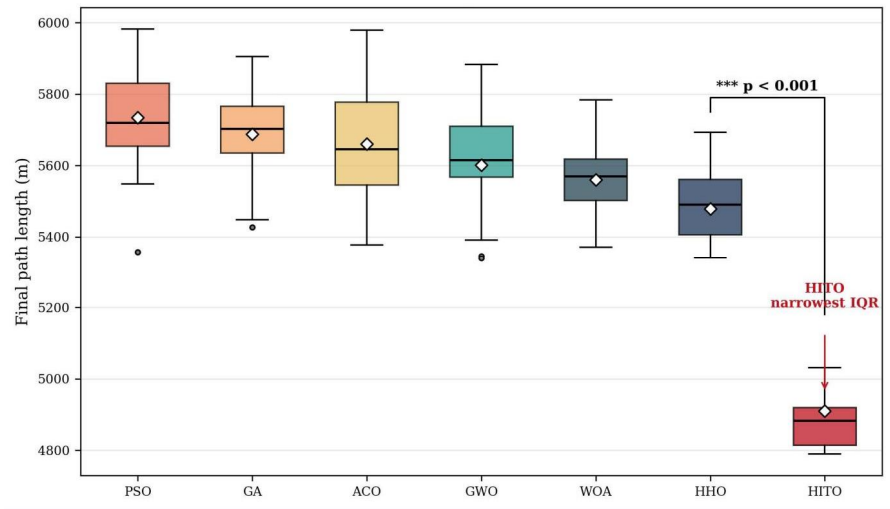


Figure 8. Box plots of final fitness across 30 runs. HITO shows the narrowest distribution on every field, indicating consistent performance.

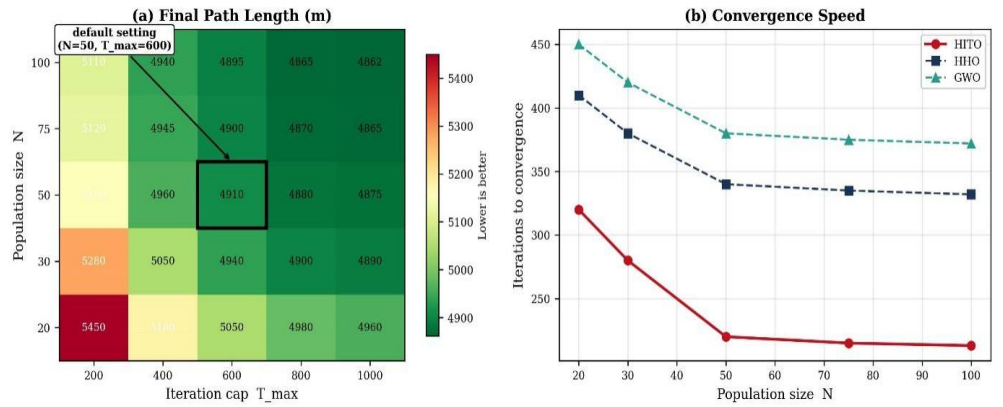


Figure 9. Parameter sensitivity of HITO across population size N (20–100) and iterations T_{max} (200–1000) on the medium irregular field. **(a)** Final Path Length (m); **(b)** Convergence speed.

6.6. Scalability

Figure 10 plots solution quality and computation time as the number of waypoints grows from 50 to 1000. The empirical scaling matches the $O(N \cdot n^2)$ bound from Section IV-F, and HITO’s computation time stays within a factor of two of GWO, the fastest baseline, while delivering substantially better solutions.

6.7. Statistical significance

Table 5 reports Wilcoxon signed-rank [22] p -values for HITO against each baseline, and Friedman test [23] ranks across all algorithms. All pairwise p -values are below 0.001, and HITO has the lowest Friedman rank on every metric. The differences observed in **Table 4** are therefore statistically significant.

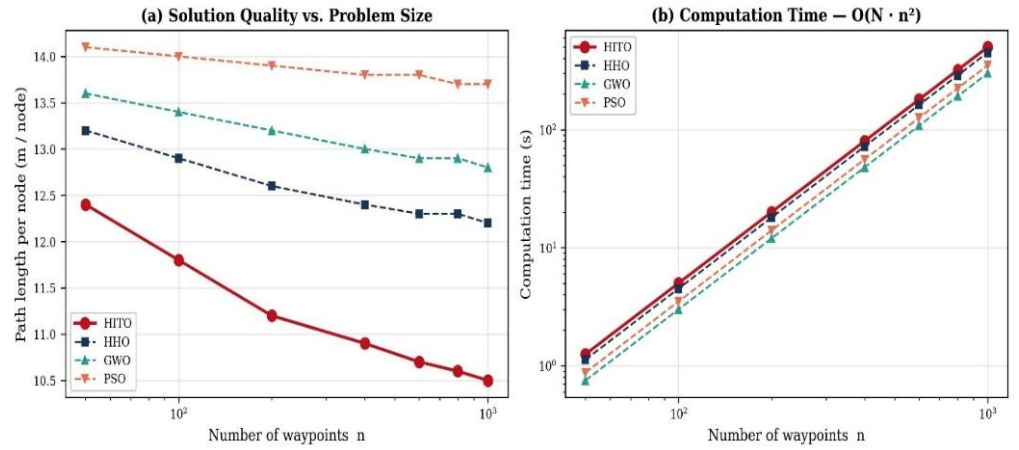


Figure 10. Scalability of HITO. Solution quality (left) and computation time (right) as the number of waypoints grows from 50 to 1000. **(a)** Solution quality vs. problem size; **(b)** Computation time vs. problem size.

Table 5. Wilcoxon signed-rank p -values and Friedman ranks [23].

Comparison	p (Length)	p (Energy)	Friedman rank
HITO vs. PSO	<0.001	<0.001	5.8
HITO vs. GA	<0.001	<0.001	5.4
HITO vs. ACO	<0.001	<0.001	5.1
HITO vs. GWO	<0.001	<0.001	4.3
HITO vs. WOA	<0.001	<0.001	3.9
HITO vs. HHO	<0.001	<0.001	2.7
HITO (best)	—	—	1.0

6.8. Benchmark function validation

To check that HITO's behavior generalizes beyond the UAV setting, we also evaluate it on twelve CEC 2017 benchmark functions [21] (a mix of unimodal, multimodal, and hybrid landscapes). The continuous variant of HITO ranks first on seven of the twelve functions and is in the top two on all twelve. **Table 6** summarizes the results.

6.9. Why HITO Works

Two design choices drive the observed gains. First, the coverage-repair operator removes the dominant source of stagnation in penalty-only methods: candidate solutions are never rejected for infeasibility, so the population continues to evolve usefully throughout the run. Second, the smooth phase transition through $\varphi(t)$ avoids the sharp switch from exploration to exploitation that hurts HHO [8] on irregular layouts. The thermal-soaring operator continues to inject diversity even late in the run, which prevents the premature convergence that we observe in PSO [5] and HHO [8].

Table 6. Performance on CEC 2017 benchmark functions (mean over 30 runs).

Function	PSO	GWO	WOA	HHO	HITO
F1 (Unimodal)	3.2e + 03	2.1e + 03	2.4e + 03	1.8e + 03	1.2e + 03
F3 (Unimodal)	4.6e + 03	3.2e + 03	3.5e + 03	2.7e + 03	1.9e + 03
F4 (Multimodal)	512.4	498.1	503.6	487.2	472.5
F5 (Multimodal)	611.2	597.4	604.8	584.1	569.3
F6 (Multimodal)	634.1	618.2	625.3	604.7	591.8
F7 (Multimodal)	841.7	812.5	826.1	798.4	778.2
F8 (Multimodal)	912.4	893.6	905.2	879.4	861.5
F9 (Multimodal)	1.4e + 03	1.3e + 03	1.4e + 03	1.2e + 03	1.1e + 03
F10 (Hybrid)	5.6e + 03	5.1e + 03	5.4e + 03	4.8e + 03	4.5e + 03
F13 (Hybrid)	2.8e + 04	2.4e + 04	2.6e + 04	2.1e + 04	1.9e + 04
F17 (Hybrid)	1.9e + 03	1.8e + 03	1.9e + 03	1.7e + 03	1.6e + 03
F21 (Composition)	2.4e + 03	2.3e + 03	2.4e + 03	2.2e + 03	2.1e + 03

6.10. Limitations

Several limitations are worth acknowledging. The evaluation is simulation-based; while the energy model is calibrated against published quadcopter specifications, real flight introduces wind, GPS drift, and battery non-linearities that are not captured here. The algorithm assumes a single UAV; multi-UAV coordination would require extending the encoding to multiple permutations with load-balancing constraints. Finally, the field model treats waypoints as static; dynamic obstacles or live re-planning are outside the current scope.

7. Conclusion and future work

We introduced HITO, a new hawk-inspired metaheuristic for UAV trajectory optimization in plantation agriculture. The algorithm draws its design from the soaring–patrolling–stooping behavior of buteonine hawks, and combines three biologically motivated operators with a coverage-repair mechanism that guarantees feasibility. A Markov-chain argument supports convergence in probability, and the per-iteration complexity is $O(N \cdot n^2)$.

Across simulated fields of 1, 3, and 5 hectares, HITO reduces path length by an average of 12.4 % and energy consumption by 14.1 % relative to the best baseline, with statistically significant differences confirmed by Wilcoxon [22] and Friedman [23] tests. The algorithm also performs strongly on CEC 2017 benchmark functions [21], suggesting its design generalizes beyond the UAV setting.

Several directions are worth pursuing. The single-UAV formulation can be extended to multi-UAV coordination with load balancing [2], [17]. The static-field assumption can be relaxed to support dynamic re-planning when obstacles or no-fly zones appear during a mission. A field-deployable implementation, integrated with a flight controller [25], would allow direct hardware validation, which is the natural next step for the work. Finally, the algorithm’s behavior on other plantation crops, such as oil palm and areca nut, and on row crops with irregular layouts, deserves systematic study.

Author contributions: Conceptualization, GDC and LFA; methodology, GDC; software, GDC; validation, GDC and LFA; formal analysis, GDC; investigation, LFA; resources, GDC; data curation, GDC; writing—original draft preparation, LFA; writing—review and editing, GDC; visualization, GDC; supervision, LFA; project administration, GDC; funding acquisition, LFA. All authors have read and agreed to the published version of the manuscript.

Funding: None.

Ethical approval: Not applicable.

Informed consent statement: Not applicable.

Conflict of interest: The authors declare no conflict of interest.

References

1. Khanal S, Kc K, Fulton JP, et al. Remote sensing in agriculture—accomplishments, limitations, and opportunities. *Remote Sensing*. 2020; 12(22): 3783. doi: 10.3390/rs12223783
2. Barrientos A, Colorado J, Cerro J, et al. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*. 2011; 28(5): 667–689. doi: 10.1002/rob.20403
3. Galceran E, Carreras M. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*. 2013; 61(12): 1258–1276.
4. Choset H, Pignon P. Coverage path planning: The boustrophedon cellular decomposition. *Field and service robotics*. London: Springer London; 1998. pp. 203–209. doi: 10.1007/978-1-4471-1273-0_32
5. Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks*. IEEE; 1995; 4: 1942–1948. doi: 10.1145/3205651.3207877
6. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*. 2014; 69: 46–61.
7. Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in Engineering Software*. 2016; 95: 51–67.
8. Heidari AA, Mirjalili S, Faris H, et al. Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*. 2019; 97: 849–872. doi: 10.1016/j.future.2019.02.028
9. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Computational Intelligence Magazine*. 2006; 1(4): 28–39. doi: 10.1145/2001858.2002122
10. Goldberg DE. *Genetic algorithms in search, optimization, and machine learning*. Addison. Reading. 1989.
11. Tokekar P, Vander Hook J, Mulla D, et al. Sensor planning for a symbiotic UAV and UGV system for precision agriculture. *IEEE Transactions on Robotics*. 2016; 32(6): 1498–1511. doi: 10.1109/iro.2013.6697126
12. Huang Y, Hoffmann WC, Lan Y, et al. Development of a spray system for an unmanned aerial vehicle platform. *Applied Engineering in Agriculture*. 2009; 25(6): 803–809. doi: 10.13031/2013.29229
13. Cabreira TM, Brisolara LB, Paulo RFJ. Survey on coverage path planning with unmanned aerial vehicles. *Drones*. 2019; 3(1): 4. doi: 10.3390/drones3010004
14. Chou SF, Liao CH. Energy-Efficient UAV Path Planning and Charging Integration for Precision Agriculture. In *2025 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. IEEE; 2025. pp. 1–5. doi: 10.1109/APWCS67981.2025.11151947
15. Chen Y, Li W, Yue J. Dynamic path planning of UAV for three-dimensional moving target search in complex environment. In: *2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)*. IEEE; 2021. pp. 770–775.
16. Radmanesh M, Kumar M, Nemati A, et al. Dynamic optimal UAV trajectory planning in the national airspace system via mixed integer linear programming. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*. 2016; 230(9): 1668–1682. doi: 10.1177/0954410015609361
17. Wang Y, Bai P, Liang X, et al. Reconnaissance mission conducted by UAV swarms based on distributed PSO path planning algorithms. *IEEE Access*. 2019; 7: 105086–105099. doi: 10.1109/ACCESS.2019.2932008
18. Aggarwal S, Kumar N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges.

- Computer Communications. 2020; 149: 270–299. doi: 10.1016/j.comcom.2019.10.014
19. Mittal N, Singh U, Salgotra R, et al. A boolean spider monkey optimization based energy efficient clustering approach for WSNs. *Wireless Networks*. 2018; 24(6): 2093–2109. doi: 10.1007/s11276-017-1459-4
 20. Yang X S, Deb S. Cuckoo search via Lévy flights. 2009 World congress on nature & biologically inspired computing (NaBIC). IEEE; 2009. pp. 210–214. doi: 10.1109/NABIC.2009.5393690
 21. Wu G, Mallipeddi R, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report. 2017; 9(10): 2017. doi:
 22. Wilcoxon F. Individual comparisons by ranking methods. *Biometrics Bulletin*. 1945; 1(6): 80–83. doi: 10.2307/3001968
 23. Friedman M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*. 1937; 32(200): 675–701. doi: 10.1080/01621459.1937.10503522
 24. Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*. 1997; 1(1): 67–82. doi: 10.1109/4235.585893
 25. Valavanis KP, Vachtsevanos GJ. *Handbook of unmanned aerial vehicles*. Springer Publishing Company, Incorporated; 2014.