

# Cyber-Physical System Design Space Exploration for Affordable Precision Agriculture

Pawan Kumar  
Arizona State University  
pkumar97@asu.edu

Hokeun Kim  
Arizona State University  
hokeun@asu.edu

**Abstract**—Precision agriculture promises higher yields and sustainability, but adoption is slowed by the high cost of cyber-physical systems (CPS) and the lack of systematic design methods. We present a cost-aware design space exploration (DSE) framework for multimodal drone-rover platforms to integrate budget, energy, sensing, payload, computation, and communication constraints. Using integer linear programming (ILP) with SAT-based verification, our approach trades off among cost, coverage, and payload while ensuring constraint compliance and a multitude of alternatives. We conduct case studies on smaller and larger-sized farms to show that our method consistently achieves full coverage within budget while maximizing payload efficiency, outperforming state-of-the-art CPS DSE approaches.

**Index Terms**—Design space exploration, Precision agriculture, Cyber-physical systems, Cost-aware optimization

## I. INTRODUCTION

Modern agriculture faces challenges to sustainably feed a growing global population amid workforce shortages, climate variability, and constrained natural resources [1]. Precision agriculture has emerged to tackle these issues with advanced sensing, automation, and data-driven management to optimize crop production. In particular, cyber-physical systems (CPS) integrate with sensors, microcontrollers, and real-time image processing, enabling continuous, non-destructive monitoring of crop growth and targeted interventions to boost yields while keeping hardware costs moderate [2], [3]. Such CPS-driven precision agriculture not only improves resource use efficiency (e.g., minimizing usage of water, fertilizer, and pesticide) but also helps offset labor constraints and environmental impacts, allowing farmers to achieve higher yields more sustainably [4].

Design space exploration (DSE) is critical for developing cost-effective CPS in precision agriculture, but it is challenging due to conflicting goals of minimizing cost while ensuring adequate coverage, payload, and runtime [4], [5]. Balancing these trade-offs requires systematic methods to explore hardware-software configurations under budget and operational constraints. Recent work emphasizes multimodal platforms that maximize field coverage and enable targeted interventions for sustainability and reduced chemical usage [6], [7]. This synergy between DSE and multimodal platforms provides a scalable pathway to practical, data-driven precision agriculture. Figure 1 shows an integrated multimodal platform hardware configuration, illustrating the trade-offs among size, coverage, and computing capacity to be evaluated.

### A. Related Work and State-of-the-Art Approaches

Recent studies have introduced cost-effective CPS prototypes and predictive modeling to non-destructively estimate

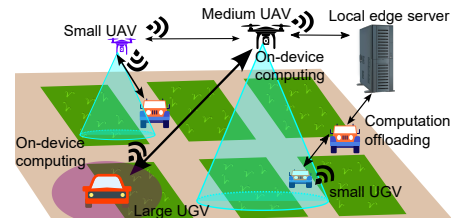


Fig. 1: Example illustration of DSE for cost-effective multimodal software-hardware platforms with unmanned ground vehicles (UGVs or rovers) and unmanned aerial vehicles (UAVs or drones) for precision agriculture.

plant growth, thereby promoting sustainability and improving crop yields [2], [8]–[10]. Doering *et al.* [11] propose a UAV-based platform with adaptive mission planning for precision agriculture. These methods integrate CPS in precision agriculture to facilitate data acquisition and adaptive decision-making in response to dynamic environmental conditions.

DSE is crucial in optimizing CPS configurations by balancing performance, cost, and computational efficiency. Mühleis *et al.* [12] integrate control performance directly into DSE. Fitzgerald *et al.* [13] develop a toolchain unifying discrete and continuous CPS models. Herget *et al.* [14] improve workload modeling for distributed CPS. Liao *et al.* [15] refine subspace pruning to cut computational overhead. Xiao *et al.* [16] propose ContrArc, which uses contract-based pruning to eliminate infeasible architectures before exploration. Yu *et al.* [17] explore symbiotic CPS designs using human-in-the-loop feedback to iteratively refine UAV architectures. Zheng *et al.* [18] develop a gradient-free framework for cross-domain UAV design exploration. There is, however, a critical gap that none of these methods target CPS DSE for precision agriculture. This motivates the need for a domain-aware CPS DSE framework tailored to the constraints and goals of agricultural automation.

### B. Research Problem and Contributions

A unique challenge in the agricultural CPS DSE problem is the co-parameterization of crops and applications, with non-linear relationships between crops and sensing/coverage requirements. Specifically, developing CPS for agriculture requires balancing competing objectives: minimizing cost while ensuring sufficient sensing, computation, and field coverage. Current state-of-the-art methods are limited to optimizing individual factors, leaving farmers without a systematic way to evaluate full platform designs within budget and constraints.

We address this gap with a cost-aware DSE framework that evaluates heterogeneous multimodal platform configurations across budget, energy, and sensing constraints. The underlying integer linear programming (ILP) formulation is non-trivial because many of these relationships are inherently non-linear,

requiring careful modeling and relaxation to obtain feasible solutions. To guarantee feasible outputs, we complement our approach with SAT-based verification to check the generated platforms and filter out invalid ones. Our framework produces alternative designs that achieve high coverage and sensing performance at the lowest possible cost, providing a principled method to navigate complex design trade-offs. We release the design, implementation, data sets, and evaluation results of our approach through an open-source repository.<sup>1</sup>

## II. PROPOSED APPROACH

This section outlines our proposed DSE method with explanations of user inputs and hardware factors to be explored, leading to feasible platforms for agriculture.

### A. User Inputs and Design Considerations

Users first provide key inputs that shape the design space: the total available budget, the size of the farm area, the target crop type, and the desired applications, as shown below.

1. **Budget:** Sets an upper limit on the total allowable monetary cost for hardware configurations, ensuring that our platforms are practical for farms with varying financial constraints.
2. **Farm area size:** Determines the minimum area coverage requirement that the platform must achieve. Larger farms may demand multiple vehicles or higher-capacity components to maintain efficiency.
3. **Target crop:** Determines the suitability of drones, rovers, or both for field operations. Indoor crops grown in controlled environments typically favor rovers, as drones are impractical in confined spaces. Densely planted crops like fiber and legumes increase the risk of ground vehicle damage, making drones a better choice. Similarly, early-stage paddy fields are often waterlogged, limiting rover use and favoring aerial monitoring. In contrast, crops such as cereals, trees, orchards, vines, vegetables, forage, and oilseeds benefit from hybrid drone-rover platforms, combining aerial surveys with ground-level intervention.
4. **Application:** Defines exact tasks that the platform must perform. TABLE I lists various agricultural applications that users can choose from when setting up our DSE framework. Each application has a recommended processing mode, either off-board (using a local edge server) or onboard (computing directly on the vehicle), and specifies whether it is better suited to drones, rovers, or both. Applications requiring control and actuation are handled onboard, while others are off-board.

Once these inputs are defined, our DSE framework systematically translates them into the design considerations that determine which combinations of chassis, motors, batteries, computing units, and sensors can meet the farm’s needs within the budget and platform constraints. It then evaluates each candidate using five key metrics:

| Application                        | Computation Mode | Platform |
|------------------------------------|------------------|----------|
| General crop monitoring            | Off-board        | Both     |
| Thermal imaging                    | Off-board        | Both     |
| Image stitching                    | Off-board        | Both     |
| Soil monitoring                    | Off-board        | Rover    |
| Yield estimation                   | Off-board        | Both     |
| Quality control                    | Off-board        | Rover    |
| Autonomous fruit/vegetable picking | Onboard          | Rover    |
| Mechanical weeding                 | Onboard          | Rover    |
| Soil pH sampling                   | Onboard          | Rover    |
| Climate mapping                    | Off-board        | Rover    |
| Fence/infrastructure inspection    | Off-board        | Drone    |
| Livestock monitoring               | Onboard          | Both     |
| Beehive inspection                 | Onboard          | Rover    |
| Frost & pest early-warning systems | Onboard          | Both     |
| Fertilizing                        | Onboard          | Rover    |

TABLE I: Overview of precision agriculture applications and their recommended processing modes and hardware platform compatibility with rover, drone, or both.

1. **Cost:** Sum of base vehicle, component (e.g., LiDAR, manipulators), and edge-server costs.
2. **Coverage:** Maximum operational coverage area (m<sup>2</sup>), calculated by estimating available battery energy after accounting for peak draws from motors, sensors, and onboard processors, then integrating over average platform speed.
3. **Payload capacity:** Mass (kg) available for additional components such as cameras, sensors, or manipulators, varying according to the target crop and application.
4. **Runtime:** Time spent (hours) before recharge, computed under the same maximum-component-load assumption.
5. **Crop–Application Parameterization:** Parameters tailored to specific crop types and application scenarios.

Our DSE framework encodes budget, area coverage, runtime thresholds, payload, communication-range requirements, and crop–application specifications into ILP. The goal of the objective function in Eq. 1 is to minimize a weighted combination of normalized cost, area coverage, and payload. By varying priority weights ( $\alpha, \beta, \gamma$ ), our ILP solver systematically generates all the possible drone-rover configurations that pass all constraints. Then, it screens them for feasibility using the crop–application platform compatibility and component power limits,

$$\min_x \sum_{i=1}^N \left( \alpha \tilde{C}_i x_i + \beta \tilde{A}_i x_i - \gamma \tilde{P}_i x_i \right) \quad (1)$$

$N$  denotes the total number of possible configurations, and  $i$  is the index of each configuration. For configuration  $i$ ,  $\tilde{C}_i$  is the normalized cost,  $\tilde{A}_i$  is the area coverage, and  $\tilde{P}_i$  is the per-platform payload. The variable  $x_i$  indicates the number of units selected for configuration  $i$ . Finally,  $\alpha, \beta$ , and  $\gamma$  are weight factors used to balance cost, area coverage, and payload capacity using the ROC weighting scheme described in III-A.

### B. Design Space Exploration

In this section, we formalize the framework requirements through a series of linear inequalities that encode budget limits, coverage and payload needs, runtime targets, component sizing rules, and communication-range expectations. These constraints approximate relationships that behave non-linearly in practice, such as the way weight influences energy consumption, how motor torque scales with usable payload,

<sup>1</sup><https://github.com/asu-kim/cps-dse-apa>

and how coverage depends on battery discharge and geometric range. Together, they define the feasible design space that our ILP solver searches to find valid drone-rover platforms.

We use equation (Eq. 2) to ensure the total cost does not exceed the user's budget,

$$(C_i + \text{extra\_cost}) \cdot x_i + \text{edge\_cost} \leq B \quad (2)$$

Here,  $C_i$  is the base monetary cost of configuration  $i$ , which includes the vehicle chassis, core hardware, and essential built-in components.  $B$  denotes the total user-defined budget. The term  $\text{extra\_cost}$  accounts for additional monetary cost for application-specific parts, while  $\text{edge\_cost}$  represents the monetary cost of the local edge server.

Eq. 3 ensures selected rovers and drones cover at least the required farm size,

$$A_i \cdot x_i \geq S \quad (3)$$

Here,  $S$  denotes the farm size specified by the user,  $x_i$  is the number of units selected for configuration  $i$ , and  $A_i$  represents the coverage area each unit of configuration  $i$  can achieve.

To determine how much area each selected configuration can cover, we use Eq.4 to find the operational coverage area for any drone or rover configuration in our DSE,

$$A_i = BATT_i \cdot T_i \cdot W_{\text{component},i} \cdot f_{\text{weight},i}(W_{\text{max},i}) \quad (4)$$

Here,  $BATT_i$  and  $T_i$  denote the number of batteries and tires in configuration  $i$ .  $W_{\text{component},i}$  is the weight calculated in Eq. 8  $W_{\text{max},i}$  is the maximum payload of configuration  $i$  calculated in Eq. 7, and  $f_{\text{weight},i}(W_{\text{max},i})$  captures the efficiency reduction due to weight, as defined in Eq. 5.

$$f_{\text{weight},i}(W_{\text{max},i}) = \frac{1}{1 + W_{\text{max},i}/100} \quad (5)$$

We use Eq. 6 to find the total force the motors generate, which determines how much weight the platforms can move,

$$W_{\text{carry}} = \tau \times \frac{nm}{TR \cdot g} \quad (6)$$

Here,  $W_{\text{carry}}$  is the effective motor weight (in kg), computed by dividing the total torque-based force  $\tau$  by the tire radius  $TR$ , and the gravitational acceleration  $g$ . The term  $nm$  denotes the number of motors in configuration  $i$ . For drone configurations, tire radius is omitted from the computation of  $W_{\text{motors}}$ .

We use Eq. 7 and Eq. 8 to find the maximum payload of configuration  $i$ .

$$W_{\text{max},i} = W_{\text{carry},i} - W_{\text{component},i} \quad (7)$$

$$W_{\text{component},i} = W_{\text{chassis},i} + W_{\text{motor},i} + W_{\text{battery},i} + W_{\text{tires},i} \quad (8)$$

Here,  $W_{\text{component},i}$  is the sum of physical weights.  $W_{\text{chassis},i}$ ,  $W_{\text{battery},i}$ ,  $W_{\text{motors},i}$ , and  $W_{\text{tires},i}$  represent the physical weights of the chassis, battery, motors, and tires, respectively of configuration  $i$ . For drone configurations, tire weight is omitted from the computation of  $W_{\text{component}}$ .

We use the following equation, Eq. 9, to ensure each selected drone or rover configuration includes at least one battery module to meet its energy requirements.

$$\sum_s BATT_{i,s} \geq 1, \quad \forall i \quad (9)$$

Here,  $BATT_{i,s}$  is a binary variable indicating whether battery option  $s$  is included in configuration  $i$ , and  $s$  represents the available battery choices for rover configurations.

Eq. 10 ensures each motor matches the chassis size, maintaining power-to-weight balance for reliable operation.

$$M_{\text{size},i} = CH_{\text{size},i}, \quad \forall i \quad (10)$$

Here,  $M_{\text{size},i}$  and  $CH_{\text{size},i}$  denote the motor size and chassis size, respectively, for configuration  $i$ .

We observe that drone configurations typically have a much shorter flight time than rovers due to higher power draw and limited battery capacity. To balance this gap and ensure that drones can perform meaningful tasks without frequent recharging, we add Eq. 11 below as a flight time condition.

$$f_d \geq F_{\text{min}} \quad (11)$$

Here,  $f_d$  denotes the estimated runtime (flight time in hours) of a specific drone configuration  $d$ , and  $F_{\text{min}}$  is the minimum required flight time per unit, set to 0.2 hours. This threshold reflects the flight time of commercial drones such as the Flyability Elios 3 [19] and XAG V40 [20], which typically operate for about 0.2 hours.

We ensure the drone-rover configurations maintain full communication coverage across the farm. Eq. 12 below calculates the number of cells<sup>2</sup> by dividing the farm area by the communication method's effective range.

$$l = \frac{\sqrt{S}}{R_{\text{comm}} \times 1000} \quad (12)$$

Here,  $S$  is the total farm size specified by the user in m<sup>2</sup> (square meter),  $R_{\text{comm}}$  is the range of a communication method (e.g., LoRa, cellular, Wi-Fi) in km, and  $l$  is the number of cells (minimum of 1 cell) required to cover the entire area. Eq. 13 then multiplies the number of cells from Eq. 12 by the cost per cell to compute the total communication cost.

$$C_{\text{comm}} = l \times C_{\text{cell}} \quad (13)$$

Here,  $C_{\text{cell}}$  denotes the cost of a single communication cell, and  $C_{\text{comm}}$  represents the total communication cost.

A suitable computing unit is selected based on the application's processing requirements as shown in (TABLE I). On-board processing requires real-time tasks such as image processing or autonomous navigation, so we assign a GPU or TPU. For off-board processing, the configuration may instead use a CPU, GPU, or TPU. Computation performance is evaluated using Geekbench 5.4.1 [21]–[23] multi-core scores, which are derived from workload execution time, the lower the execution time higher the score.

### C. SAT-based Verification

To ensure the correctness of the ILP even after modification of the ILP formula by the users, we verify that each configuration meets the total cost and total coverage constraints using a SAT solver, PySAT [24]. We encode feasibility as Boolean

<sup>2</sup>We use the term "cell" to refer to the area covered by each device of the communication method, analogous to cells in cellular networks.

variables, enabling efficient SAT solving. The solver checks conditions in Eq. 14.

$$\text{total\_cost} \leq B, \quad \text{total\_coverage} \geq S \quad (14)$$

where  $B$  is the budget and  $S$  the required farm size. Hardware components (motors, batteries, sensors, frames) contribute to these totals, keeping verification aligned with design. Similar SAT-based methods are used in many-core DSE [25], [26], validating our approach for CPS.

### III. EVALUATION

To assess the effectiveness of our approach, we first evaluate the design aspects considered in our approach. Second, we compare our approach’s optimization method against state-of-the-art approaches.

- **Evaluation of design aspects (Section III-A)** – We conduct an ablation study of the ILP-based multi-objective optimizers: *Proposed Approach (PA)* ( $\text{cost+area+payload}$ ), *Area+Payload AP*, *Cost+Area CA*, and *Payload+Cost (PC)*.
- **Optimizer model evaluation (Section III-B)** – Comparison of the proposed ILP approach against optimization methods existing state-of-the-art CPS DSE approaches including: *simulated annealing*, *Bayesian optimization (BO)* [17], *random search*, *genetic algorithm*, *portfolio optimization*, *discrete search*, *Lengler* [18], and *PG-DSE with PSP and PEGA* [15] as summarized in TABLE III.

Our evaluation uses two farm settings as case studies:

**Case Study 1:** We assume a smaller-scale farm with a farm size of 4047 m<sup>2</sup> (approx. 1 acre), which aligns with farm size distribution data published by the USDA NASS in the 2022 Census of Agriculture [27] and by the EU in 2020 [28], with a budget of \$100K (USD), which aligns with entry-level costs for robotic platforms used in precision farming [29], [30]. We select the tree crop (Section II-A) for its high monitoring demands and economic value, as shown in reports from the US and Europe, with the value of \$2.9 billion USD in 2024 [31] for apple, and 1.17 million acres of farm land used for growing apples [32]. We use autonomous fruit and vegetable picking (TABLE I) as applications leveraging CPU, GPUs, or TPUs for efficient harvesting tasks on rovers.

**Case Study 2:** We assume a large-scale farm with a farm size of 40,469 m<sup>2</sup> (approx. 10 acres) [27], [28], with a budget of \$1M (USD), which aligns with the cost for robotic weeding machines [29], [30], [33]. We select vine (Section II-A) due to its high value and requirement for delicate handling, as shown in reports from the US and Europe, with the value of \$6.2 billion USD in 2024 [31] for grapes, and 7.9 million acres of farm land used for growing grapes [32]. We use general crop monitoring and yield estimation (TABLE I), which uses off-board processing with CPU, GPU, or TPU.

Across both evaluations, we visualize scatter plots and compute a unified weighted score for each optimizer.

#### A. Evaluation of Design Aspects

Figure 2 illustrates trade-offs across cost, payload, and area coverage for different optimization methods for Case Study 2.

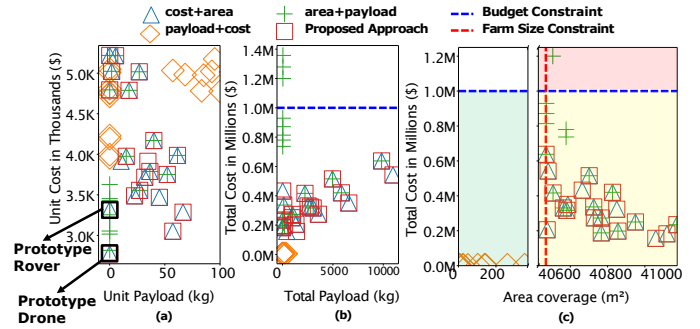


Fig. 2: Trade-offs among design aspects in Case Study 2 (\$1M, 10 acres): (a) unit cost vs. unit payload (b) total cost vs. total payload, (c) total cost vs. area coverage.

A unit refers to a single drone or rover instance, and a configuration may include multiple units to satisfy minimum farm area coverage. Fig. 2–(a) plots the unit cost vs. unit payload where, PA and CA yield balanced designs, whereas PC increases cost with increase in payload. AP produces a more scattered distribution. Fig. 2–(b) plots the total cost vs. total payload where, PC clusters near the origin, while PA and CA achieve high payloads within \$1M. Whereas, AP has lower payload and also exceeds the \$1M. Fig. 2–(c) plots the total cost vs. area coverage where, PA and CA maximize area coverage without exceeding \$1M (yellow shaded region). In contrast, AP attains minimum coverage but includes configurations that exceed \$1M (red shaded region), whereas PC stays under \$1M but fails coverage (green shaded region).

**Weighted Score Design:** We prioritize cost to achieve our primary goal, a cost-effective DSE. We assign area coverage the next highest priority so our framework can monitor the minimum required area for the applications. We consider payload as the least critical factor once the drone-rover carries the necessary sensors, as extra capacity adds little value. To reflect this, we now use the Rank Order Centroid (ROC) [34] method to determine the final weights, following the approach demonstrated in the works [35]–[37]. Using ROC, we rank the criteria in order of importance and derive their normalized weights as follows:

- Weight for cost: 0.611
- Weight for area coverage: 0.278
- Weight for payload: 0.111

| Design Aspects | Case Study 1 \$100K, 1 Acre | Case Study 2 \$1M, 10 Acres |
|----------------|-----------------------------|-----------------------------|
| AP             | 0.404                       | 0.377                       |
| PC             | 0.611                       | 0.611                       |
| CA             | 0.417                       | 0.389                       |
| PA             | 0.417                       | 0.389                       |

TABLE II: Weighted-score comparison for design aspects in our case.

Under these weights for Case Study 2, TABLE II shows that the PC achieves the highest score of 0.611, followed by PA and CA at 0.389. AP ranks lowest at 0.377.

A similar pattern appears in Case Study 1. PA and CA again achieve full area coverage within \$100K, while PC stays with the highest weighted score but lacks full area coverage, and AP exceeds \$100K despite minimum area coverage. Together, we confirm that the PA and CA deliver the best balance of cost and area coverage across different farm sizes.

To summarize, Fig. 2 and TABLE II confirm that the PA and

| Approach          | Domain                | Platform            | Evaluation Methodology  | DSE Formulation   | Optimizers  | Cost Breakdown |
|-------------------|-----------------------|---------------------|---|---|---|----------------|
| DESTION 23 [17]   | General CPS           | UAVs & Robot Car    | Simulation-based performance evaluations include time & energy for robot car & four flight tasks for UAVs | Reward feedback & switching strategy                                    | Simulated Annealing, Bayesian Optimization (BO)     | Not available  |
| DESTION 22 [18]   | UAV architecture      | UAVs only           | 4 mission-based benchmarks (hover, straight, circle, oval)  | Graph-based design optimized by component selection or action sequences | Discrete, Lengler, Portfolio, FastGA, Random Search | Not available  |
| ASP-DAC 23 [15]   | ADAS                  | FPGA-Xilinx Artix-7 | ADAS system: 10 PSMs, 52 MCCs, 244 MCC alternatives, & Synthetic systems with design space sizes          | PSMs leverage component-level MCCs to model time-triggered behavior     | PG-DSE with PSP and PEGA                            | Not available  |
| Proposed approach | Precision agriculture | Rovers & UAVs       | Evaluation of design aspects & Optimizer model evaluation   | Minimize cost, maximize area coverage & payload                         | Integer Linear Programming                          | Available      |

TABLE III: Comparison of the proposed and state-of-the-art CPS DSE approaches.

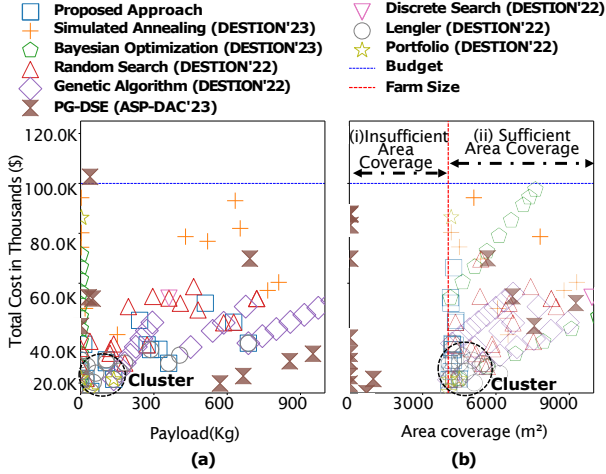


Fig. 3: Comparative evaluation of optimizers in Case Study 1 (\$100K, 1 acre): (a) total cost vs. payload, (b) total cost vs. area coverage.

CA achieve full area coverage while keeping costs within the \$1M budget in Case Study 2, which explains why they earn balanced weighted scores. Meanwhile, the PC scores higher due to its low cost and high payload, but it fails to fully meet the area coverage requirement. AP exceeds the budget even though it meets the area coverage target and ranks lowest in the weighted score.

### B. Optimizer Model Evaluation

Fig. 3–(a) plots the total cost vs. payload for Case Study 1, where the \$100K budget line appears as a blue dotted line. The PA spans a broad payload range while remaining below \$100K, indicating flexible scaling across configurations. **simulated annealing** concentrates around mid-to-high (approx. 400–800 kg) payload values within \$100K. **genetic algorithm** and **random search** show a gradual cost increase as payload increases. **discrete search** shows only one configuration at 400 kg payload with a cost of 60K. **BO** shows an increase in costs without huge changes in payload. **PG-DSE** covers a wide span, from light payload (approx. 50 kg) at mid cost (approx. \$60K) to heavy payload (approx. 900 kg) at low cost (approx. \$30K) with some exceeding \$100K. **Lengler** and **portfolio** position themselves mainly around low payload (approx. 150 kg) and cluster together, shown in a dotted circle.

Fig. 3–(b) plots the total cost vs. area coverage, where the farm size appears as a red dotted line, and the budget line appears as a blue dotted line. The PA sits close to the farm size line, sufficient area coverage (Fig. 3–(b)-(ii)) with varied

| State-of-the-Art Optimizers and Proposed Optimizer | Case Study 1: \$100K, 1 Acre | Case Study 2: \$1M, 10 Acres |
|--|------------------------------|------------------------------|
| Simulated Annealing (DESTION 23)                   | 0.325                        | 0.465                        |
| Bayesian Optimization (DESTION 23)                 | 0.498                        | N/A                          |
| Random Search (DESTION 22)                         | 0.496                        | 0.475                        |
| Genetic Algorithm (DESTION 22)                     | 0.503                        | 0.527                        |
| Discrete Search (DESTION 22)                       | 0.307                        | 0.600                        |
| Lengler (DESTION 22)                               | 0.634                        | 0.686                        |
| Portfolio (DESTION 22)                             | 0.634                        | 0.686                        |
| PG-DSE (ASP-DAC 23)                                | 0.619                        | 0.360                        |
| Proposed Approach                                  | 0.634                        | 0.686                        |

TABLE IV: Weighted-score comparison for optimizer evaluation in Case Study 1 and Case Study 2.

costs. **simulated annealing** extends beyond the minimum farm size, reaching sufficient area coverage (Fig. 3–(b)-(ii)) with costs upto approx. 95K. **BO** and **genetic algorithm** show a similar trend where area coverage increases with cost. **random search** achieves moderate area coverage of about 4100–8500 m<sup>2</sup> and stays under budget. **discrete search** shows only one configuration at approx. 10000 m<sup>2</sup>. **PG-DSE** explores a broader solution space using evolutionary operators, which can momentarily include configurations in insufficient area coverage (Fig. 3–(b)-(i)). **Lengler** and **portfolio** cluster near the farm size line and have similar cost, showing a similar trend between these optimizers.

TABLE IV presents the weighted-score comparison for optimizer model evaluation, using the ROC weighting scheme described in III-A. In Case Study 1, the PA, **Lengler**, and **portfolio** top the ranking by striking the best balance among cost, area coverage, and payload. **PG-DSE** ranks next highest, driven by multiple low-cost configurations, though some solutions incur budget penalties or provide limited area coverage. **genetic algorithm** claims the next highest scores, leveraging strong area coverage with controlled spending. **BO**, **Random search** and **simulated annealing** follow, as they identify feasible configurations but exhibit a higher cost for similar area coverage or payload compared to **genetic algorithm**. **Discrete search** earns a slightly lower score due to limited designs.

In Case Study 2, the PA, **Lengler**, and **portfolio** achieve the highest overall score by effectively balancing payload, area coverage, and cost. **discrete search** follow at a distance, **genetic algorithm** with close behind, followed by **random search** and **simulated annealing**. **PG-DSE** scores the lowest due to budget penalties or limited area coverage. Also, **BO** from DESTION 23 [17] gives no results since we lack hyper-parameters (e.g., optimization calls, acquisition settings) that were unavailable, forcing untuned, general-purpose runs that cannot uncover viable configurations under tight constraints.

The trends shown in Fig. 3 align well with the weighted scores in TABLE IV. PA consistently shows strong trade-offs across cost, payload, and area coverage, explaining its top score in both weighted scores and plots. **genetic algorithm**, **BO** and **random search** follow a similar pattern by keeping costs balanced while reaching practical area coverage. **discrete search**, and **simulated annealing** show moderate performance with lower scores. Although **PG-DSE** scores a high weighted score, it has configurations that exceed the budget or provide limited area coverage. **Lengler** and **portfolio** perform similarly, with overlapping plots and identical scores.

### C. Verification of DSE Results

TABLE V summarizes the results of constraint verification by our SAT-based approach described in Section II-C for Case Study 1.

| Method Verified by SAT | Total Configs | Valid Configs | Invalid Configs |
|------------------------|---------------|---------------|-----------------|
| PA                     | 18            | 18            | 0               |
| AP                     | 20            | 17            | 3               |
| CA                     | 18            | 18            | 0               |
| PC                     | 13            | 0             | 13              |
| Simulated Annealing    | 12            | 12            | 0               |
| BO                     | 20            | 20            | 0               |
| PG-DSE                 | 20            | 4             | 16              |
| Random Search          | 20            | 20            | 0               |
| Genetic Algorithm      | 20            | 20            | 0               |
| Discrete               | 1             | 1             | 0               |
| Lengler                | 18            | 18            | 0               |
| Portfolio              | 4             | 4             | 0               |

TABLE V: SAT-based constraint verification across optimization methods, for Case Study 1.

The following approaches return only valid configurations without any invalid ones: PA, CA, **BO**, **discrete**, **Lengler**, **random search**, **genetic algorithm**, **portfolio**, and **simulated annealing**. In contrast, PC completely fails, returning all invalid results (13 out of 13), while AP and PG-DSE return 3 and 16 invalid configurations, respectively, out of 20 configurations. This highlights the strength of our ILP-based optimizer and the importance of verification in eliminating infeasible designs.

### D. Execution Time Analysis

Our approach can run on commercial off-the-shelf personal computers, such as laptops. We measure the execution time of our approach on a computer with an AMD Ryzen 7 processor (8 cores @ 3.6 GHz), 16GB RAM, Windows 11, and Python 3.10. In Case Study 1, PA requires 2.5 s to execute. PG-DSE completes in approximately 487 ms, while both **genetic algorithm** and **portfolio** finish in about 10 ms. Each of **Lengler**, **random search**, **simulated annealing**, and **discrete search** takes up to 5 ms to complete. In contrast, **BO** takes the longest time at 35.3 s. Also, our SAT solver completes verification almost instantly, in just 8.6 ms.

| Platform | Hardware Configurations                                 | Payload (kg) | Runtime (hours) | Cost (USD) |
|----------|---|--------------|-----------------|------------|
| Rover    | Plastic large body, large motor, medium battery, RPi 4B | 34.67        | 123.23          | 3449.56    |
| Drone    | metal body, large motor, large battery, RPi 4B          | 0.1          | 0.22            | 2345.34    |

TABLE VI: Prototype hardware specifications derived from Case Study 2 solver outputs, with the drone implemented using a carbon-fiber body instead of the solver-selected metal body.

### E. Hardware Prototype Implementation

To demonstrate the feasibility of our DSE approach for precision agriculture, we build and verify a prototype UAV-UGV platform derived directly from the solver outputs of the proposed method, as shown in Fig. 4 and TABLE VI.

Our prototype consists of a rover, a drone, and a local edge server. The rover uses a Raspberry Pi (RPi) 4B and a large plastic chassis with tires, providing sufficient runtime while supporting necessary sensors. The drone (Duckiedrone DD24-B [38]) follows the solver-selected configuration, except for the body. While the solver specifies a metal body, we use a carbon-fiber frame for the lightweight, while keeping all other hardware components unchanged. Also, a local edge server (\$2,000) provides off-board computation. Together, these elements confirm that ILP+SAT outputs are not abstract platforms but realizable, low-cost agricultural systems.

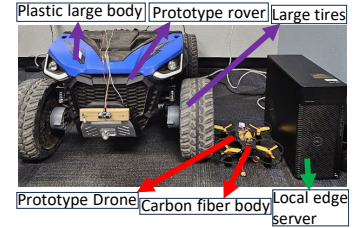


Fig. 4: Our prototype hardware constructed in line with the proposed approach's outputs.

## IV. CONCLUSION

In summary, we propose a cost-aware DSE framework for multimodal UAV-UGV platforms in precision agriculture. By formulating the problem as an ILP with SAT verification, we address the challenges of balancing cost, coverage, and payload under strict resource constraints, while outperforming existing methods in terms of feasibility, cost, and the number of alternative configurations. Limitations, scope, and future work of this paper are further discussed below.

**Limitations and Scope:** To optimize the ILP, our framework targets static, deterministic conditions. These choices streamline the ILP but limit its ability to reflect soil conditions, weather patterns, and long-term degradation. Communication and sensing are modeled at a high level, which can miss interference effects or routing complexity in large farms. Linear approximations of non-linear behaviors such as battery discharge, weight-efficiency decay, and torque-payload scaling further narrow realistic behavior.

**Future Work:** We plan to extend our framework to other CPS domains with non-linear trade-offs, such as warehouse monitoring and disaster response. These settings introduce challenges like coverage vs. communication, mobility vs. energy, and safety vs. latency. We continue modeling these within an ILP and verifying feasibility with SAT. Although each domain requires its own constraint set and weighting, the same systematic exploration process remains, making this work a foundation for broader CPS optimization.

## ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation (NSF) under grants #2449200 (NSF I/UCRC for IDEAS) and POSE-#2449200 (An Open-Source Ecosystem to Coordinate Integration of Cyber-Physical Systems).

## REFERENCES

- [1] J. McFadden, E. Njuki, and T. Griffin, "Precision agriculture in the digital era: Recent adoption on U.S. farms," United States Department of Agriculture, USDA Miscellaneous 333550, 2023, dOI: 10.22004/ag.econ.333550. [Online]. Available: <https://ideas.repec.org/p/ags/usdami/333550.html>
- [2] P. Kumar and H. Kim, "Cost-effective cyber-physical system prototype for precision agriculture with a focus on crop growth," in *2024 International Workshop on Rapid System Prototyping (RSP)*. IEEE, October 2024, pp. 35–41.
- [3] P. Castillejo *et al.*, "Aggregate farming in the cloud: the AFarCloud ECSEL project," *Microprocessors and microsystems*, vol. 78, p. 103218, August 2020.
- [4] A. Pretto *et al.*, "Building an aerial-ground robotics system for precision farming: An adaptable solution," *IEEE Robotics & Automation Magazine*, vol. 28, no. 3, pp. 29–49, August 2021.
- [5] D. Madroñal, F. Palumbo, A. Capotondi, and A. Marongiu, "Unmanned vehicles in smart farming: A survey and a glance at future horizons," in *Proceedings of the 2021 Drone Systems Engineering and Rapid Simulation and Performance Evaluation: Methods and Tools Proceedings*, ser. DroneSE and RAPIDO '21. Association for Computing Machinery, January 2021, pp. 1–8.
- [6] P. Kumar, Y. Dube, and H. Kim, "Cost-effective rover for farms," in *Proceedings of the ACM/IEEE 16th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2025)*, ser. ICCPS '25, May 2025, pp. 1–2.
- [7] United States Government Accountability Office, "Precision agriculture: Benefits and challenges for technology adoption and use," 2024, accessed: 2025-06-26. [Online]. Available: <https://www.gao.gov/products/gao-24-105962>
- [8] S. Bouarourou, C. Kanzouai, A. Zannou, E. H. Nfaoui, and A. Boulaalam, "Crop yield prediction in IoT: A hybrid feature selection approach using machine learning models," in *2024 3rd International Conference on Embedded Systems and Artificial Intelligence (ESAI)*. IEEE, December 2024, pp. 1–5.
- [9] H. Qiu, Z. Zhou, Y. Peng, J. Liang, J. Shang, X. Xi, and R. Zeng, "Multi-terminal cooperative AIoT smart agriculture system based on growth prediction model," in *2024 IEEE International Conference on Smart Internet of Things (SmartIoT)*. IEEE, November 2024, pp. 407–414.
- [10] M. Shilpa *et al.*, "Enhancing crop yield and growth prediction using IoT-based smart irrigation systems and machine learning algorithms," in *2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON)*. IEEE, August 2024, pp. 1–5.
- [11] D. Doering, A. Benenmann, R. Lerm, E. P. de Freitas, I. Muller, J. M. Winter, and C. E. Pereira, "Design and optimization of a heterogeneous platform for multiple UAV use in precision agriculture applications," vol. 47, no. 3, January 2014, pp. 12272–12277, 19th IFAC World Congress.
- [12] N. Mühleis, M. Glaß, L. Zhang, and J. Teich, "A co-simulation approach for control performance analysis during design space exploration of cyber-physical systems," *ACM SIGBED Review*, vol. 8, no. 2, pp. 23–26, June 2011.
- [13] J. Fitzgerald, C. Gamble, P. G. Larsen, K. Pierce, and J. Woodcock, "Cyber-physical systems design: Formal foundations, methods and integrated tool chains," in *2015 IEEE/ACM 3rd FME Workshop on Formal Methods in Software Engineering*, May 2015, pp. 40–46.
- [14] M. Herget, F. S. Saadatmand, M. Bor, I. G. Alonso, T. Stefanov, B. Akesson, and A. D. Pimentel, "Design space exploration for distributed cyber-physical systems: State-of-the-art, challenges, and directions," in *2022 25th Euromicro Conference on Digital System Design (DSD)*. IEEE, August 2022, pp. 632–640.
- [15] Y. Liao, T. Adegbija, and R. Lysecky, "Efficient system-level design space exploration for high-level synthesis using Pareto-optimal subspace pruning," in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, January 2023, pp. 567–572.
- [16] Y. Xiao, C. Oh, M. Lora, and P. Nuzzo, "Efficient exploration of cyber-physical system architectures using contracts and subgraph isomorphism," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, March 2024, pp. 1–6.
- [17] S.-J. Yu, I. Incer, V. Prabhhu, A. Chatteraj, E. Vin, D. Fremont, A. Mehta, A. Sangiovanni-Vincentelli, S. Sastry, and S. A. Seshia, "Symbiotic CPS design-space exploration through iterated optimization," in *Proceedings of Cyber-Physical Systems and Internet of Things Week 2023*, May 2023, pp. 92–99.
- [18] H. Zheng, J. Betz, A. Ramamurthy, H. Jin, and R. Mangharam, "Combinatorial and parametric gradient-free optimization for cyber-physical system design," in *2022 IEEE Workshop on Design Automation for CPS and IoT (DESTION)*. IEEE, May 2022, pp. 34–41.
- [19] Flyability, "Flyability Elios 3," 2024, accessed: 2025-02-17. [Online]. Available: <https://www.flyability.com/hubfs/Knowledge%20Base%20Files/Documents/Manuals/E3%20and%20equipments/Elios%20Base%20RAD%20User%20Manual%20v1.0.pdf>
- [20] XAG, "XAG V40 HSE-UAV," 2024, accessed: 2025-02-17. [Online]. Available: <https://www.xa.com/en/v40/v40specs>
- [21] Primate Labs, "Geekbench 5 (benchmark)," 2021, accessed: 2025-9-12. [Online]. Available: <https://www.geekbench.com/legacy/>
- [22] —, "Geekbench 5.4.1 (benchmark)," 2021, accessed: 2025-9-12. [Online]. Available: <https://www.primatelabs.com/release/geekbench5/>
- [23] S. Chetoui and S. Reda, "Workload- and user-aware battery lifetime management for mobile SoCs," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2021, pp. 1679–1684.
- [24] A. Ignatiev, A. Morgado, and J. Marques-Silva, "PySAT: A Python toolkit for prototyping with SAT oracles," in *International Conference on Theory and Applications of Satisfiability Testing*, June 2018, pp. 428–437.
- [25] V. Rithhammer, T. Schwarzer, S. Wildermann, J. Teich, and M. Glaß, "Architecture decomposition in system synthesis of heterogeneous many-core systems," in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1–6.
- [26] T. Schwarzer, A. Weichslgartner, M. Glaß, S. Wildermann, P. Brand, and J. Teich, "Symmetry-eliminating design space exploration for hybrid application mapping on many-core architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 2, pp. 297–310, 2017.
- [27] United States Department of Agriculture, "2022 census of agriculture data," 2022, accessed: 2025-06-26. [Online]. Available: [https://www.nass.usda.gov/Publications/Highlights/2024/Census22\\_HL\\_FarmsFarmland.pdf](https://www.nass.usda.gov/Publications/Highlights/2024/Census22_HL_FarmsFarmland.pdf)
- [28] Eurostat, "Farms and farmland in the EU," 2022, accessed: 2025-09-11. [Online]. Available: [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Farms\\_and\\_farmland\\_in\\_the\\_European\\_Union\\_-\\_statistics](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Farms_and_farmland_in_the_European_Union_-_statistics)
- [29] AllyNav, "Agriculture robot price - professional farm automation solutions," 2025, accessed: 2025-12-18. [Online]. Available: <https://www.allynnav.com/precision-agriculture-solutions/agricultural-robots/agriculture-robot-price>
- [30] G. Gil *et al.*, "Why the low adoption of robotics in the farms? challenges for the establishment of commercial agricultural robots," *Smart Agricultural Technology*, vol. 3, p. 100069, February 2023.
- [31] United States Department of Agriculture, "National agricultural statistics service," 2022, accessed: 2025-06-26. [Online]. Available: [https://www.nass.usda.gov/Statistics\\_by\\_State/Washington/Publications/Current\\_News\\_Release/2025/FRUIT.pdf](https://www.nass.usda.gov/Statistics_by_State/Washington/Publications/Current_News_Release/2025/FRUIT.pdf)
- [32] Eurostat, "Apple orchards in the EU," 2019, accessed: 2025-09-11. [Online]. Available: [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Agricultural\\_production\\_-\\_orchards](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Agricultural_production_-_orchards)
- [33] S. Bloch, "Robotic weeders are racing to replace glyphosate and dicamba," 2019, accessed: 2025-12-18. [Online]. Available: <https://thecounter.org/robot-weeders-glyphosate-dicamba-herbicide-replacement/>
- [34] F. H. Barron, "Selecting a best multiattribute alternative with partial information about attribute weights," *Acta psychologica*, vol. 80, no. 1-3, pp. 91–103, 1992.
- [35] T. Varshney, A. V. Waghmare, V. P. Singh, M. Ramu, N. Patnana, V. P. Meena, A. T. Azar, and I. A. Hameed, "Investigation of rank order centroid method for optimal generation control," *Scientific Reports*, vol. 14, pp. 1–21, December 2024.
- [36] E. Roszkowska, "Rank ordering criteria weighting methods – a comparative overview," *Optimum. Studia ekonomiczne*, pp. 14–33, 2013.
- [37] B. Suhendarwan, "Rank order centroid as a method to weight dimensional variables for developing a composite index on policing activities in mining and quarrying areas," *Journal Research of Social Science, Economics, and Management*, vol. 3, pp. 1608–1616, March 2024.
- [38] Duckietown, "Duckiedrone (DD24-B)," 2025, accessed: 2025-09-07. [Online]. Available: <https://get.duckietown.com/products/autonomous-raspberrypi-quadcopter-duckiedrone-dd24>