ARQII Publication
http://arqiipubl.com/ams
APPLICATIONS OF MODELLING AND SIMULATION
eISSN 2600-8084
VOL 10, 2026, 89-97

Development and Implementation of a Hybrid Non-dominated Sorting Genetic Algorithm (H-NSGA-II-III) for Greenhouse Pesticide Routing on Embedded Platforms

Abubaker Badi[1], Salinda Buyamin[1,*], Mohamad Shukri Zainal Abidin[1] and Mohd Saiful Azimi Mahmud[1]

[1]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

[*]Corresponding author: salinda@utm.my

Submitted 29 September 2025; Revised 23 December 2025; Accepted 11 January 2026; Available online 16 March 2026.

Abstract: Greenhouse pesticide application requires route optimization that balances operational efficiency with sapling-level infection severity. Classical formulations of the Capacitated Vehicle Routing Problem (CVRP) construct routes subject to vehicle capacity constraints. In agricultural applications, such as spraying robots with limited tank capacity, these formulations do not account for infection severity, which is critical for timely treatment of highly affected saplings. This limitation necessitates a multi-objective optimization (MOO) approach to capture competing routing and treatment priorities. However, achieving high-quality Pareto fronts with algorithms such as the Non-dominated Sorting Genetic Algorithm II (NSGA-II) and NSGA-III often requires long runtimes, limiting their practicality for real-time or embedded agricultural systems. To address these challenges, this study proposes H-NSGA-II-III, a hybrid algorithm integrating the diversity-preserving mechanism of NSGA-II with the reference-point selection strategy of NSGA-III. The approach extends the classical CVRP into a bi-objective formulation that minimizes total travel distance while incorporating sapling-level infection severity as a second objective. The algorithm was benchmarked through multiple independent runs on a standard computing platform (SCP) and a lightweight embedded platform (LEP). Results show that H-NSGA-II-III achieves 11-13% higher hypervolume than NSGA-II and 20-21% higher than NSGA-III, while reducing runtime by 9-32% on the SCP and 10-17% on the LEP. These findings indicate that H-NSGA-II-III provides an efficient and lightweight framework for infection-aware routing, feasible for embedded deployment in greenhouse robots.

Keywords: Capacitated vehicle routing problem; Greenhouse robotics; Hybrid NSGA-II-III; Multi-objective optimization; Sapling prioritization.

1. INTRODUCTION

The use of robotic systems for pesticide application in greenhouses has become increasingly important to address labor shortages, improve operational efficiency, and enhance pest- and disease-management in controlled environments [1], [2]. Unlike open-field agriculture, where larger equipment and frequent refills are feasible, greenhouse operations impose stricter constraints such as limited tank capacity, narrow aisles, and heterogeneous pest pressure [2], [3]. Beyond these operational constraints, timely treatment of severely infected saplings is critical, as pests can spread quickly to healthy areas, reducing the effectiveness of spraying and increasing overall pesticide use [4]. Together, these challenges extend the classical capacitated vehicle routing problem (CVRP), leading to a bi-objective formulation with two competing objectives: minimizing travel distance and prioritizing saplings with higher infection severity.

Multi-objective evolutionary algorithms (MOEAs) provide a natural framework for addressing trade-offs by generating Pareto-optimal sets of solutions that balance conflicting goals [5], [6]. Among these, NSGA-II introduced elitist sorting and crowding distance for diversity preservation, while NSGA-III extended this framework with reference-point based selection to improve solution spread in many-objective problems [7]. These algorithms have also been applied in agricultural robotics, though most studies emphasized operational objectives such as distance, turning angle, or pesticide dosage rather than sapling-level infection severity.

One of the earliest contributions in this domain is [3], which optimized greenhouse spraying routes as a bi-objective problem, minimizing travel distance and routing angle under tank capacity constraints. Routes were generated using a Probabilistic Roadmap (PRM), and NSGA-III solutions were benchmarked against NSGA-II, with results showing that NSGA-

III achieved superior solution quality but required slightly longer runtimes. Building on this, Zangina et *al*. [2] introduced an improved NSGA-III (INSGA-III) for variable-rate spraying under tank capacity constraints. The problem was formulated as a multi-objective VRP with three objectives: minimizing distance, reducing turning angle, and lowering total travel time. INSGA-III was validated against NSGA-III, demonstrating comparable solution quality with improved runtime efficiency. In parallel, Zangina et *al*. [8] investigated multi-robot fleet coordination by framing pesticide spraying as a vehicle routing problem. Their approach used a Model Predictive Control (MPC) method with Mixed-Integer Linear Programming (MILP), benchmarked against NSGA-III, Simulated Annealing (SA), Tabu Search Algorithm (TSA), and Multi-adaptive Particle Swarm Optimization (MaPSO). The results highlighted the superiority of MPC in terms of scalability and efficiency for autonomous fleet coordination. Later, Zhang et *al*. [6] addressed greenhouse robot navigation as a many-objective problem that explicitly considered both distance and turning angle. Their study compared the Grid-based Evolutionary Algorithm (GrEA), the Hypervolume Estimation Algorithm (HypE), the Knee Point-Driven Evolutionary Algorithm (KnEA), and NSGA-III, concluding that HypE offered the best solution quality, although with longer runtimes, and limited applicability to real-time deployment due to its offline design.

Other applications of MOEAs in agriculture include greenhouse microclimate optimization [9], hybrid NSGA-II for spraying trajectories [10], and orchard spraying with splittable arc routing [11]. VRP has also been applied more broadly to agricultural logistics and product delivery [12], [13], but these studies focused on economic objectives rather than infection-aware greenhouse spraying.

Another persistent gap is computational feasibility on lightweight hardware. NSGA-II and NSGA-III often demand long runtimes and high generation counts to achieve competitive performance, limiting their practicality for real-time or embedded agricultural deployments. Prior studies on IoT- and edge-enabled agriculture [14]–[16] emphasize the importance of lightweight feasibility in agricultural robotics.

Collectively, these gaps highlight the need for an optimization approach that explicitly incorporates sapling-level infection severity into the route optimization process while remaining computationally feasible on embedded platforms. To meet this need, a hybrid evolutionary algorithm, H-NSGA-II-III, is proposed by combining the diversity-preserving mechanism of NSGA-II with the reference-point selection strategy of NSGA-III. The main contributions of this study are twofold: (a) the development of an infection-aware route optimization formulation that extends classical CVRP by incorporating sapling-level infection severity as a second objective alongside travel distance, and (b) the design of a computationally efficient hybrid evolutionary algorithm validated on both standard and embedded platforms.

The remainder of this paper is organized as follows. Section 2 presents the methodology, including baseline algorithms, the proposed H-NSGA-II-III, problem formulation, and performance indicators. Section 3 reports on the experimental setup and discusses the results. Finally, Section 4 concludes the paper and outlines directions for future work.

2. SAPLING-LEVEL MULTI-OBJECTIVE OPTIMIZATION

This section presents the methodology developed to optimize pesticide spraying routes with sapling-level infection prioritization. The proposed hybrid algorithm, H-NSGA-II-III, integrates NSGA-II, which uses non-dominated sorting and crowding distance for diversity preservation, with NSGA-III, which uses reference-point–based selection to guide convergence. Section 2.1 describes the baseline algorithms (NSGA-II and NSGA-III) that serve as comparative benchmarks; Section 2.2 details the structure and computational logic of H-NSGA-II-III; Section 2.3 formulates the bi-objective CVRP model with sapling-level infection severity; and Section 2.4 outlines the performance evaluation metrics used for Pareto front analysis.

2.1 Baseline Algorithms: NSGA-II and NSGA-III

The baseline multi-objective algorithms used in this study are NSGA-II and NSGA-III, both of which have been widely adopted for Pareto-based evolutionary optimization. NSGA-II is known for its efficient non-dominated sorting and the use of crowding distance to maintain solution diversity across the front. NSGA-III extends this framework by incorporating a predefined set of reference points in the objective space, guiding the population toward a more uniformly distributed Pareto front. Although this strategy is particularly effective in problems with more than two objectives, it remains relevant for the bi-objective sapling-level optimization problem considered here.

For fairness in comparison, all algorithms were executed under a fixed computational budget, denoted as E_{total}, defined in Equation (1) as the product of the population size, N and the number of generations, G [17]. By constraining NSGA-II, NSGA-III, and H-NSGA-II-III to the same evaluation budget, their performance can be assessed fairly in terms of both solution quality and computational cost. The procedural frameworks of NSGA-II and NSGA-III are summarized in Table 1 and provide the foundation for the hybrid H-NSGA-II-III algorithm.

$$E_{\text{total}} = N \times G \tag{1}$$

2.2 H-NSGA-II-III Algorithm

The H-NSGA-II-III algorithm integrates the key features of NSGA-II and NSGA-III in a two-stage hybrid framework designed to balance diversity and convergence for sapling-level optimization, as outlined in Table 2. The first stage emphasizes exploration, while the second stage emphasizes convergence.

Table 1. Procedural framework of NSGA-II and NSGA-III in sapling-level optimization.

Initialize a population P_0 of size N
for $g = 1$ to G do
Generate offspring population Q_g using order crossover and inversion mutation
Combine parent and offspring populations: $R_g = P_g \cup Q_g$
Perform non-dominated sorting on R_g to create fronts $F_1, F_2, \ldots \ldots$
NSGA-II: Select the next population $P_{(g+1)}$ using front ranking and crowding distance NSGA-III: Select the next population $P_{(g+1)}$ using reference point association and niching
end for
Return final Pareto front

Table 2. Two-stage optimization procedure of the proposed H-NSGA-II-III algorithm.

Compute evaluation budget $E_{\text{total}} = N \times G$
Select Stage 1 budget fraction $\alpha,\ \beta$ subject to constraints $0.1 \leq \alpha \leq 0.9\ and\ 0.5 \leq \beta \leq 2.0$
Compute Stage 1 evaluations: $E_1 = \alpha N \times \beta G$
Run NSGA-II for $G_1 = \beta G$ generations, starting from an initialized population of size αN to evolve a diverse population P_{G_1}
Construct reference directions based on the size of P_{G_1}
Compute Stage 2 budget: $E_2 = E_{\text{total}} - E_1$
Compute number of generations for Stage 2: $G_2 = \left\lfloor \frac{E_2}{\lvert P_{G_1} \rvert} \right\rfloor$
Initialize NSGA-III using P_{G_1} as the population (population size $= \lvert P_{G_1} \rvert$) and reference directions
Run NSGA-III for G_2 generations using reference point-based selection
Return final Pareto front

In Stage 1, a portion of the total computational budget E_{total} is allocated to promote early exploration. The effective population size is scaled by a factor α, and the number of generations is scaled by a factor β, resulting in αN individuals evolved over βG generations. The corresponding number of evaluations is given by Equation (2).

$$E_1 = \alpha N \times \beta G \tag{2}$$

To prevent overcommitment of resources, a constraint-guided search ensures that E_1 does not exceed 60% of E_{total}. Accordingly, the scaling factors are bounded by $0.1 \leq \alpha \leq 0.9$ and $0.5 \leq \beta \leq 2.0$, values were selected empirically to balance exploration in Stage 1 with a sufficient remaining budget for Stage 2. A valid (α, β) configuration is selected prior to optimization and fixed for all runs, and the effective initial population of size αN is then processed using NSGA-II, yielding a set of non-dominated individuals P_{G_1}, which serve as the input population for the next stage.

In Stage 2, the population is initialized using the set of non-dominated solutions P_{G_1} obtained from Stage 1, such that the Stage 2 population size is $\lvert P_{G_1} \rvert$. The remaining evaluation budget is computed by Equation (3), and the number of generations is determined adaptively by Equation (4). This stage applies NSGA-III with reference directions to enhance convergence, guiding the search process along uniformly distributed points in the objective space. The design intent is to produce a diverse and well-balanced set of non-dominated solutions while operating strictly within the fixed evaluation budget.

Since the generation count in Equation (4) is an integer, the realized number of evaluations may be slightly below E_{total} by fewer than $\lvert P_{G_1} \rvert$ evaluations due to generation-level granularity.

$$E_2 = E_{\text{total}} - E_1 \tag{3}$$

$$G_2 = \left\lfloor \frac{E_2}{\lvert P_{G_1} \rvert} \right\rfloor \tag{4}$$

2.3 Problem Formulation

This study formulates sapling-level pesticide route optimization as a bi-objective problem that minimizes (i) the total travel distance of the spraying robot and (ii) a demand-based penalty that reflects infection severity weighted by treatment delay (route position). Let $V = v_0, v_1, \dots, v_n$ denote the set of vertices, where v_0 is the depot and each $v_i (i = 1, \dots, n)$ is a sapling. A symmetric distance matrix $D = [d_{ij}]$ gives the travel cost between any two vertices v_i and v_j. Each sapling $v_i \in V \setminus v_0$ has an associated demand s_i (pesticide dosage in liters), which is used as a proxy for sapling-level infection severity (higher s_i indicates more severe infection). The spraying robot has tank capacity C.

A route is represented as an ordered sequence that starts and ends at the depot, and a solution is a set of routes $R = \{r_1, \dots, r_m\}$ that collectively visit each sapling exactly once. In this study, routes are preconstructed in a preprocessing step by grouping spatially proximate saplings such that the cumulative demand in each route satisfies the capacity limit, i.e., $\sum_{v_i \in r_k} s_i \leq C, \forall\, r_k \in R$. The optimization stage then evaluates candidate visit orders within each route using two objectives. The first objective, defined in Equation (5), minimizes the total travel distance along a given route, while the second objective, defined in Equation (6), minimizes a position-weighted demand penalty that prioritizes early treatment of severely infected saplings.

$$\min f_1(r_k) = \sum_{(i,j) \in r_k} d_{ij} \tag{5}$$

$$\min f_2(r_k) = \sum_{v_i \in r_k} s_i \,.\, p_i \tag{6}$$

where p_i denotes the visit order index of sapling v_i within route r_k, with $p_i = 1$ corresponding to the first sapling visited after the depot and increasing sequentially along the route. Although s_i is measured in liters (L), the product $s_i \cdot p_i$ represents a position-weighted penalty rather than a physical quantity and is therefore reported in composite units of L × position.

For example, as illustrated in Figure 1, where visit positions are indexed by $p = 1, 2, \dots$ along the route, the penalty depends jointly on demand magnitude and visit order. A sapling with relatively high demand visited early yields a moderate penalty (e.g., $s = 0.114$ L at $p = 1$ gives $0.114 \times 1 = 0.114$), while a sapling with low demand visited later also yields a moderate penalty (e.g., $s = 0.023$ L at $p = 3$ gives $0.023 \times 3 = 0.069$). The largest penalties arise when high demand is combined with late visit positions, explicitly prioritizing early treatment of severely infected saplings.

2.4 Performance Indicators for Pareto Front Evaluation

Evaluating the quality of the Pareto front generated by the proposed H-NSGA-II-III, in comparison with the baseline NSGA-II and NSGA-III algorithms, requires well-established performance indicators. These indicators quantify aspects such as solution dominance, convergence, distribution, and diversity in the objective space.

Performance indicators are classified in [18] into four categories, comprising cardinality, convergence, distribution, and combined convergence–distribution aspects. Based on this classification and given that the true Pareto front for this problem is unknown, a set of complementary metrics was selected to robustly evaluate algorithm performance in terms of dominance, convergence, diversity, and coverage (see Table 3). Specifically, the C-Metric [19] was chosen for cardinality, the Epsilon indicator (additive ε_+ and multiplicative $\varepsilon_\times$ variants) [20] for convergence, Spacing (SP) [21] for distribution, and Hypervolume (HV) [22] for both convergence and coverage of the objective space.

Table 3. Summary of performance indicators for Pareto front evaluation.

Metric	Range	Better When	Brief Explanation
C-Metric	[0,1]	Higher	Proportion of solutions in one front dominated by another; measures dominance
Additive Epsilon (ε_+)	$(-\infty, +\infty)$	Lower or Negative	Minimum additive shift needed to make one front dominate another
Multiplicative Epsilon ($\varepsilon_\times$)	$[1, \infty)$	Lower	Minimum scaling factor needed for domination
Spacing (SP)	$[0, \infty)$	Lower	Captures the variation in distance between neighbouring solutions on the Pareto front
Hypervolume (HV)	$[0, \infty)$	Higher	Measures the volume dominated by the Pareto front with respect to a reference point

		Saplings in route					
Route:	$depot$	$p = 1$	$p = 2$	$p = 3$	$p = 4$	… … … …	$depot$
Demand (s_i):	0	0.114 L	0.059 L	0.023 L	0.083 L	… … … …	0

Figure 1. Example route representation showing visit order and associated sapling demand values for a capacity-feasible route.

3. RESULTS AND DISCUSSION

The experiments in this study evaluated route optimization for a greenhouse spraying scenario with 30 infected saplings modeled as demand nodes and a central depot, forming a 31-node CVRP instance. NSGA-II, NSGA-III, and the proposed H-NSGA-II-III were implemented in Python using the Pymoo framework [23]. All algorithms were compared under an identical total evaluation budget and executed for 20 independent runs to account for stochastic variability. The baseline NSGA-II and NSGA-III algorithms employed fixed population sizes ($N = 600$), a fixed number of generations, and the same genetic operator types, including order crossover (*crossover=OrderCrossover(shift=False)*) and inversion mutation.

In contrast, the proposed H-NSGA-II-III redistributed the same total evaluation budget across two stages using a constraint-guided procedure (Section 2.2). For the proposed method, this procedure resulted in $\alpha = 0.3$ and $\beta = 2.0$, allocating approximately 60% of the total evaluation budget to Stage 1, with the remaining budget assigned to Stage 2. Regarding mutation settings, the baseline mutation probability of 0.9 was selected to provide a strong exploratory baseline and applied consistently across NSGA-II and NSGA-III. For the proposed H-NSGA-II-III, a stage-specific mutation strategy was adopted, with a moderate mutation probability of 0.6 in Stage 1 to enhance exploration and a reduced mutation probability of 0.1 in Stage 2 to emphasize convergence while retaining limited diversity under the same evaluation budget.

Performance was assessed using the indicators described in Section 2.4: C-Metric, additive-ε (ε_+), multiplicative-ε ($\varepsilon_\times$), spacing (SP), and hypervolume (HV). In addition, the number of non-dominated solutions was recorded to capture the richness of the obtained fronts. To evaluate both optimization quality and runtime feasibility, experiments were conducted in simulation on two platforms: a standard computing platform (SCP) represented by a laptop with Intel Core i5, 16 GB RAM, SSD, Windows 11, and a lightweight embedded platform (LEP) represented by a Raspberry Pi 5 with Broadcom BCM2712, Quad-core Cortex-A76, 8 GB RAM, Raspberry Pi OS 64-bit. To ensure fairness, the same precomputed distance matrix was loaded into memory on both devices. Execution times were measured using Python's *time.perf_counter()* by recording the elapsed time between the start and end of the optimization process.

Section 3.1 reports the comparative performance of H-NSGA-II-III against the baselines across both platforms, Section 3.2 presents the runtime benchmarking results, and Section 3.3 analyzes the effect of generation count on optimization quality and efficiency.

3.1 Comparative Performance on Standard and Lightweight Platforms

To select a practical configuration, multiple generation settings (40, 80, 120, 200, and 400) were tested on the LEP. The results indicated that 80 generations offered the best balance between solution quality and computational efficiency. Accordingly, the 80-generation configuration was adopted as the benchmark for comparing H-NSGA-II-III against NSGA-II and NSGA-III across both platforms.

The benchmarking results are summarized in Table 4. Overall, H-NSGA-II-III demonstrates consistent improvements over NSGA-III and competitive performance relative to NSGA-II across multiple indicators, particularly in terms of dominance, convergence, hypervolume, and solution richness. Dominance analysis shows that H-NSGA-II-III almost completely dominates NSGA-II (C ≈ 0.99 on SCP and ≈ 1.0 on LEP) and fully dominates NSGA-III (C = 1.0 on both platforms). The ε-indicators reinforce this result: additive ε_+ values are close to zero against NSGA-II (0.34 on SCP and -0.07 on LEP) and consistently negative against NSGA-III (-0.29 on SCP and -0.27 on LEP). Negative additive ε_+ values indicate that the Pareto front produced by H-NSGA-II-III dominates the reference front without requiring any objective translation, reflecting dominance in the sense of the ε-indicator definition. Complementarily, multiplicative $\varepsilon_\times$ values are ≈1.01 on SCP and ≈1.00 on LEP against NSGA-II, and exactly 1.0 against NSGA-III, indicating that the proposed algorithm achieves comparable or superior convergence without requiring objective scaling.

Solution diversity was further analyzed using spacing (SP) and hypervolume (HV). The spacing results indicate a moderate and competitive distribution of solutions for H-NSGA-II-III, with average values of 4.11 on the SCP and 3.83 on the LEP. NSGA-II exhibits more uniform spacing, while H-NSGA-II-III performs better than NSGA-III but remains slightly less uniform than NSGA-II. This behavior is consistent with [18], which notes that spacing may provide limited information when Pareto solutions form clustered structures. In contrast, the hypervolume results show that H-NSGA-II-III achieves higher overall performance on both platforms, indicating a more favorable trade-off between solution quality and coverage of the objective space. H-NSGA-II-III achieved 2907.7 on SCP and 2771.8 on LEP, compared with 2619.0/2454.5 for NSGA-II and 2416.7/2303.3 for NSGA-III, corresponding to approximately 11–13% and 20–21% improvements, respectively.

Finally, H-NSGA-II-III produced richer Pareto sets. On average, it generated ~59 solutions on SCP and ~58 on LEP, compared with ~32–33 for NSGA-II and ~14–16 for NSGA-III. This wider and more balanced set of trade-offs provides decision-makers with greater flexibility in balancing sapling infection severity against travel distance.

Table 4. Comparative performance of NSGA-II, NSGA-III, and H-NSGA-II-III on SCP and LEP (20 runs, 80 generations). Results are reported as mean ± SD.

Platform	Algorithm	C-metric C (H-NSGA-II-III, baseline)	Additive ε	Multiplicative ε	Spacing	Hypervolume (HV)	Solution Count
SCP	NSGA-II	≈ 1	0.34 ± 1.80	1.008 ± 0.037	2.28 ± 1.53	2619.0 ± 700.2	32.4 ± 7.9
SCP	NSGA-III	1.0	−0.29 ± 0.06	1.000	4.87 ± 2.62	2416.7 ± 642.6	14.1 ± 2.4
SCP	H-NSGA-II-III	–	–	–	4.11 ± 1.28	2907.7 ± 702.0	59.1 ± 10.2
LEP	NSGA-II	≈ 1	−0.07 ± 0.14	1.000 ± 0.002	1.84 ± 0.74	2454.5 ± 445.3	33.5 ± 5.8
LEP	NSGA-III	1.0	−0.27 ± 0.07	1.000	5.61 ± 3.40	2303.3 ± 419.8	15.6 ± 4.6
LEP	H-NSGA-II-III	–	–	–	3.83 ± 0.89	2771.8 ± 480.2	57.8 ± 10.8

3.2 Runtime performance on SCP and LEP

A central objective of this study was to assess whether H-NSGA-II-III can generate high-quality Pareto fronts without exceeding the computational limits of lightweight embedded hardware. Standard MOEAs such as NSGA-II and NSGA-III often demand high generation counts and long runtimes, limiting their practicality for real-time or embedded agricultural applications [7]. This limitation has also been observed in greenhouse-specific adaptations such as INSGA-III, where runtime reduction was necessary to improve feasibility [2]. The challenge is particularly critical in greenhouse robotics, where embedded platforms must compute solutions fast enough to respond to dynamic pest outbreaks and real-time sensor updates [14], [16].

Figure 2 presents runtime performance across 20 independent runs, with error bars denoting standard deviations. On the LEP, H-NSGA-II-III consistently achieved the fastest completion time (8.42 ± 0.27 s), outperforming NSGA-II (9.32 ± 0.26 s) and NSGA-III (10.19 ± 0.09 s). All algorithms completed within 11 s on the LEP, confirming that multi-objective optimization can be executed in real time even under constrained processing resources. For comparison, runtimes on the SCP were roughly half those of the LEP, yet the same ranking held—H-NSGA-II-III remained the fastest, averaging 3.63 ± 0.13 s compared with 3.99 ± 0.14 s for NSGA-II and 5.31 ± 0.23 s for NSGA-III.

These results demonstrate the practicality of H-NSGA-II-III for greenhouse robotics. By combining superior Pareto quality (Section 3.1) with runtimes that remain feasible on resource-constrained hardware, H-NSGA-II-III addresses one of the major barriers to translating evolutionary optimization from simulation into real-world agricultural deployment. The next subsection (3.3) explores how generation count influences this balance between optimization quality and efficiency, with particular attention to maintaining feasibility on the LEP.

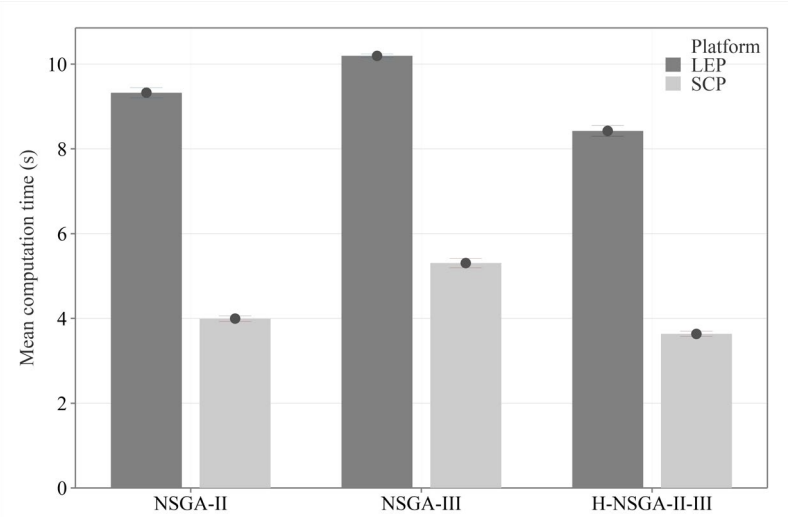

Figure 2. Mean runtime (20 runs) of NSGA-II, NSGA-III, and H-NSGA-II-III on SCP and LEP.

3.3 Per-Generation Efficiency of H-NSGA-II-III (40–400 Generations on LEP)

To evaluate the trade-off between runtime and solution quality, H-NSGA-II-III was executed on the LEP under five generational configurations: 40, 80, 120, 200, and 400. Each setting was repeated over 20 independent runs, with the 400-generation configuration serving as the reference baseline. Table 5 summarizes solution counts, computation times, hypervolume, spacing, and additive ε-indicators relative to the 400-generation front. For clarity, Figure 3 presents representative Pareto fronts for 40, 80, and 400 generations, as these illustrate the most distinct trade-offs.

The results show that the number of non-dominated solutions increased steadily with generation count, from 33 at 40 generations to 110 at 400. However, this expansion of the Pareto set came at the expense of substantially higher runtime. Mean execution times rose nearly linearly with generation count, ranging from 4.54 s at 40 generations to 42.53 s at 400. Importantly, improvements in Pareto quality diminished beyond 80 generations. The 80-generation configuration achieved an HV of 2771.8, comparable to the 2686.9 and 2875.1 observed at 120 and 200 generations, while 400 generations yielded only a modest further gain at 3160.6. Spacing results showed a similar trend, decreasing from 5.14 at 40 generations to 3.83 at 80, then stabilizing around 3.3–3.5 from 120 to 400 generations.

The additive ε-indicator relative to the 400 baseline provides further evidence of convergence. At 40 generations, ε_+ was high (10.97), indicating poor approximation of the reference front. At 80 generations, ε_+ dropped substantially to 5.78, while further increases in generation count produced smaller improvements (4.40 at 120 and 2.86 at 200). These results confirm that although longer runs continue to refine convergence, the additional runtime required becomes disproportionate. From these results, the 80-generation configuration offers the most practical balance between solution quality and computational efficiency. It provides sufficiently accurate Pareto approximations for embedded deployment on the LEP while avoiding the heavy runtime penalties observed at higher generation counts.

Table 5. Performance of H-NSGA-II-III on LEP under varying generation counts (mean ± SD over 20 runs). The additive ε_+ indicator is reported relative to the 400-generation baseline.

Generation count	Sol. Count	Comp. Time (s)	HV	SP	ε_+ vs 400
40	32.8 ± 6.5	4.54 ± 0.34	2847.6 ± 602.6	5.14 ± 2.33	10.97 ± 5.27
80	57.8 ± 10.8	8.42 ± 0.27	2771.8 ± 480.2	3.83 ± 0.89	5.78 ± 3.89
120	79.5 ± 14.2	12.73 ± 0.27	2686.9 ± 463.1	3.46 ± 0.83	4.40 ± 3.97
200	96.6 ± 9.7	20.81 ± 0.40	2875.1 ± 465.2	3.28 ± 0.43	2.86 ± 2.56
400	109.8 ± 11.5	42.53 ± 0.87	3160.6 ± 442.9	3.41 ± 0.49	– (baseline)

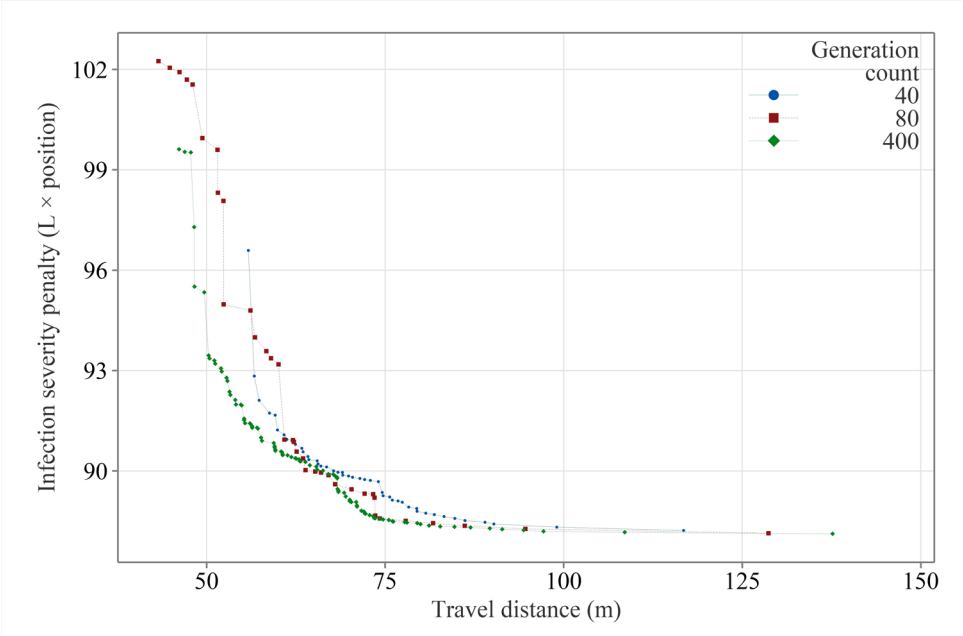

Figure 3. Pareto fronts of H-NSGA-II-III on LEP under 40, 80, and 400 generations, showing the trade-off between travel distance and infection severity penalty.

4. CONCLUSION

This study evaluated the proposed H-NSGA-II-III algorithm for capacity-constrained pesticide spraying routes in greenhouse environments. The findings confirm that H-NSGA-II-III consistently outperforms baseline algorithms, NSGA-II and NSGA-III, by achieving superior Pareto quality while remaining computationally efficient across both standard computing (SCP) and lightweight embedded (LEP) platforms. On average, H-NSGA-II-III delivered 11–13% higher hypervolume than NSGA-II and ~20% higher than NSGA-III, while producing ~70–80% more non-dominated solutions than NSGA-II and nearly three times more than NSGA-III. Runtime performance further highlighted its efficiency: compared with NSGA-II and NSGA-III, H-NSGA-II-III reduced execution time by ~9% and ~32% on the SCP, and by ~10% and ~17% on the LEP.

These advantages become even clearer when contrasted with the baseline algorithms at higher generation counts. NSGA-II and NSGA-III can approximate competitive fronts only when running for very long durations, but such improvements come at the cost of five- to ten-fold longer runtimes, undermining their feasibility for embedded deployment. By contrast, H-NSGA-II-III reached high-quality solutions with only 80 generations, capturing the essential Pareto front without exhausting computational resources. The per-generation analysis reinforced this balance: while 400 generations yielded ~90% more solutions, the hypervolume gains were modest (~14%) and came at a fivefold increase in runtime, whereas 40 generations were insufficient, with ~11% lower hypervolume and reduced solution diversity.

In summary, H-NSGA-II-III provides a practical balance between optimization quality and runtime efficiency, making it a promising approach for infection-aware greenhouse spraying under real-time constraints. Future work will focus on enhancing the algorithm's robustness and integrating it into mobile spraying robots for seamless deployment in greenhouse operations.

ACKNOWLEDGMENT AND FUNDING

The authors would like to acknowledge the financial support provided by Universiti Teknologi Malaysia and the Ministry of Higher Education Malaysia through research grants (Vote No. R.J130000.7723.4J626 and Q.J130000.3823.31J58). The first author also gratefully recognizes the support received from the International Doctoral Fellowship (IDF).

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

REFERENCES

[1] Z. A. Yahaya, S. Buyamin, H. Chiroma, M. S. Azimi Mahmud, F. Hassan and A. A. H Badi, Paths planning for agricultural robots: Recent development, taxonomy, challenges, and opportunities for future research, *Proceedings of IEEE 20th Student Conference on Research and Development (SCOReD)*, Bangi, Malaysia, 2022, 7-12.

[2] U. Zangina, S. Buyamin, M. S. Z. Abidin and M. S. A. Mahmud, Agricultural rout planning with variable rate pesticide application in a greenhouse environment, *Alexandria Engineering Journal*, 60, 2021, 3007-3020.

[3] M. S. A. Mahmud, M. S. Z. Abidin, Z. Mohamed, M. K. I. A. Rahman and M. Iida, Multi-objective path planner for an agricultural mobile robot in a virtual greenhouse environment, *Computers and Electronics in Agriculture*, 157, 2019, 488-499.

[4] K. K. Kethineni, S. P. Mohanty, E. Kougianos, S. Bhowmick and L. Rachakonda, SprayCraft: Graph-based route optimization for variable rate precision spraying, *arXiv.org,* 2024, 2412.12176.

[5] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6, 2002, 182-197.

[6] X. Zhang, Y. Guo, J. Yang, D. Li, Y. Wang and R. Zhao, Many-objective evolutionary algorithm based agricultural mobile robot route planning, *Computers and Electronics in Agriculture*, 200, 2022, 107274.

[7] K. Deb and H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints, *IEEE Transactions on Evolutionary Computation*, 18, 2014, 577-601.

[8] U. Zangina, S. Buyamin, M. N. Aman, M. S. Zainal Abidin and M. S. A. Mahmud, Autonomous mobility of a fleet of vehicles for precision pesticide application, *Computers and Electronics in Agriculture*, 186, 2021, 106217.

[9] L. Wang, Greenhouse microclimate control optimization based on improved NSGA-II algorithm, *Proceedings of Chinese Control and Decision Conference (CCDC)*, Hefei, China, 2020, 2365-2370.

[10] C. Xu, W. Liu, Q. Zhang, Y. Wan and P. Liu, Multi-objective trajectory planning for spraying robot based on hybrid polynomial interpolation and HMONSGA-II, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 239, 2025, 2014-2029.

[11] Q. Wan, R. García-Flores, S. A. Bowly, P. Kilby and A. T. Ernst, The agricultural spraying vehicle routing problem with splittable edge demands, *arXiv.org*, 2023, 2308.15108.

[12] W. M. Fernando, A. Thibbotuwawa, H. N. Perera, P. Nielsen and D. K. Kilic, An integrated vehicle routing model to optimize agricultural products distribution in retail chains, *Cleaner Logistics and Supply Chain*, 10, 2024, 100137.

[13] D. Wu and C. Wu, Research on the time-dependent split delivery green vehicle routing problem for fresh agricultural products with multiple time windows, *Agriculture*, 12, 2022, 793.

[14] A. Mana, A. Allouhi, A. Hamrani, A. Jamil, K. O. A. Barrahmoune and D. Daffa, Survey review on artificial intelligence and embedded systems for agriculture safety: a proposed IoT agro-meteorology system for local farmers in Morocco, *Smart Embedded Systems and Applications*, 2022, 211-242.

[15] X. Liu, Z. Zhao and A. Rezaeipanah, Intelligent and automatic irrigation system based on internet of things using fuzzy control technology, *Scientific Reports*, 15, 2025, 14577.
[16] D. F. Yépez-Ponce, W. Montalvo, X. A. Guamán-Gavilanes and M. D. Echeverría-Cadena, Route optimization for UGVs: A systematic analysis of applications, algorithms and challenges, *Applied Sciences*, 15, 2025, 6477.
[17] Y. Ma, B. Li, W. Huang and Q. Fan, An improved NSGA-II Based on multi-task optimization for multi-UAV maritime search and rescue under severe weather, *Journal of Marine Science and Engineering*, 11, 2023, 781.
[18] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel and L. Salomon, Performance indicators in multiobjective optimization, *European Journal of Operational Research*, 292, 2021, 397-422.
[19] E. Zitzler and L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *IEEE transactions on Evolutionary Computation*, 3, 1999, 257-271.
[20] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca and V. G. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Transactions on evolutionary computation*, 7, 2003, 117-132.
[21] J. R. Schott, *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, Master Thesis, Massachusetts Institute of Technology (MIT), Cambridge, USA, 1995.
[22] E. Zitzler, K. Deb and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation*, 8, 2000, 173-195.
[23] J. Blank and K. Deb, Pymoo: Multi-objective optimization in Python, *IEEE Access*, 8, 2020, 89497-89509.

.