biomimetics

MDPI

Article

Optimizing Tree Selection for Sustainable Logging in Amazonian Areas with Species Restrictions Using Bio-Inspired Algorithms

Sebastian Grajeda, Manuel Tupia * and Rony Cueva

Department of Engineering, Pontificia Universidad Católica del Perú, Av. Universitaria 1801, San Miguel, Lima 15086, Peru; s.grajeda@pucp.edu.pe (S.G.); cueva.r@pucp.edu.pe (R.C.)
* Correspondence: tupia.mf@pucp.edu.pe; Tel.: +51-962717353

Abstract

Forest management planning in the Peruvian Amazon must reconcile economic return, ecological sustainability, spatial coherence, and strict regulatory compliance within a single harvest decision. Biomimetic optimization—search that transposes natural mechanisms such as honeybee foraging, biological evolution, and physical annealing into computation—offers a promising route to such constrained combinatorial problems. This study proposes a set-theoretic discrete adaptation of the Artificial Bee Colony algorithm (Set-ABC), a swarm-intelligence metaheuristic inspired by the division of labor among employed, onlooker, and scout bees, for parcel-based tree selection; food sources are represented as sets of parcel identifiers and neighborhood search is driven by set-algebraic operators rather than continuous or binary encodings. The task is formalized as a regulated NP-hard combinatorial optimization with hard constraints anchored in Law No. 29763 and Supreme Decree No. 018-2015-MINAGRI, plus a penalized species-retention constraint, under an objective maximizing net economic return in monetary units (PEN). Set-ABC is benchmarked against three further nature-inspired paradigms—a Binary ABC (BABC), a Binary Genetic Algorithm (GA), and Simulated Annealing (SA)—on a semi-synthetic instance of approximately 90 parcels and 12 commercial species from a real operational inventory. After Grid Search calibration, the methods were evaluated over 20 paired runs. A Friedman test with Bonferroni-corrected Wilcoxon post hoc comparisons revealed significant differences across all pairs: GA attained the highest mean net return (1,059,364 PEN) and Set-ABC the lowest (983,873 PEN) but the second-most stable variance, establishing this biomimetic set-based formulation as a viable, stable contender for regulated parcel-based forest optimization.

Keywords: Artificial Bee Colony; genetic algorithm; combinatorial optimization; sustainable forest management; Amazonian tree selection

Check for updates

Academic Editor: Yongquan Zhou

Received: 8 August 2026
Revised: 9 September 2026
Accepted: 29 September 2026
Published: 1 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The Amazon basin concentrates one of the largest reserves of commercial timber and standing carbon on the planet, and its management under sustainable-yield regimes is simultaneously an economic activity and a conservation obligation [1,2]. In the Peruvian Amazon, harvest planning is not a purely economic exercise: it is bounded by a national regulatory framework that a valid management plan must satisfy. The Forest and Wildlife Law (Law No. 29763) mandates minimum primary-forest retention, canopy-cover thresholds, and seed-tree preservation ratios, while the national CITES registry restricts the

extraction of protected flora species [3–5]. A harvest proposal that maximizes revenue but breaches these thresholds is not merely suboptimal—it is legally inadmissible. Planning must therefore reconcile economic return, ecological sustainability, spatial coherence of the harvest block, and strict regulatory compliance within a single decision.

Despite this complexity, a substantial share of harvest planning in the region is still carried out through manual procedures or empirical criteria, which limits the exploration of alternative management scenarios and introduces calculation errors and delays [6,7]. The underlying difficulty is combinatorial: when the concession is discretized into spatial decision units—here, forest parcels aggregating multiple inventoried trees—the task of selecting which parcels to harvest becomes an NP-hard combinatorial optimization problem, structurally related to the Set Covering and Multidimensional Knapsack problems [8,9]. The number of candidate subsets grows exponentially with the number of parcels, and the feasible region is further fragmented by non-linear spatial requirements such as connectivity of the harvest block and tolerance limits on internal enclaves [9,10]. Because the input data (timber volumes, canopy estimates, species inventories) are themselves derived from sampling and carry uncertainty, exact methods stagnate computationally on realistic instances, motivating the use of approximate, metaheuristic search [7,11].

The metaheuristic literature applied to spatial forest planning has followed two main lines. Single-solution strategies—GRASP, Tabu Search, and Simulated Annealing—have proven efficient for coverage and infrastructure-allocation problems but are highly sensitive to neighborhood design and prone to local optima under overlapping constraints [9–11]. Population-based and bio-inspired methods, and in particular Genetic Algorithms (GA), have shown stronger global exploration for multi-objective tree selection and seed-tree preservation [12–14]. However, a specific gap persists: collective swarm-intelligence algorithms, and the Artificial Bee Colony (ABC) algorithm in particular, have not been adapted to the discrete, parcel-based tree-selection problem, nor systematically benchmarked against established population-based and single-solution metaheuristics on this class of regulated, spatially constrained forest problems. ABC is well established in continuous domains and in some binary reformulations [15–18], yet its cooperative balance between scout-bee exploration and employed/onlooker-bee exploitation remains untested in this setting.

Methodologically, this work is grounded in biomimetics: the transfer of problem-solving strategies observed in nature into computational search. The three metaheuristic families examined here are all biomimetic in origin—swarm intelligence, in which the Artificial Bee Colony (ABC) algorithm emulates the cooperative foraging of a honeybee colony through the division of labor among employed, onlooker, and scout bees; evolutionary computation, in which the Genetic Algorithm (GA) reproduces the dynamics of natural selection, crossover, and mutation; and physical-process analogs, in which Simulated Annealing (SA) mimics the controlled cooling of a solid to escape local optima. This biomimetic paradigm is a still-active line of research: recent studies continue to refine bee-inspired search through improved ABC variants with new crossover, lock, and feature-selection mechanisms [15,16], alongside other nature-inspired optimizers such as the Artificial Electric Field Algorithm [19].

This biomimetic machinery has proven particularly well suited to sustainable forest management, where the search space is large, nonlinear, and heavily constrained. Multi-objective genetic algorithms have been applied to seed-tree selection and to balancing timber yield against ecosystem services [12,13], and related optimization techniques have supported thinning and harvest decisions from multi-platform inventory data [8]. Within this biomimetic toolkit, however, the honeybee-foraging paradigm remains unexplored for discrete, parcel-based tree selection. The present study addresses that gap by contributing a biomimetic adaptation—Set-ABC—that reformulates the bees' foraging and perturbation

behavior through set-algebraic operators, and by situating it against three other biomimetic paradigms (BABC, GA, and SA) under a unified, regulation-aware evaluation.

Addressing this gap, the present work develops a set-theoretic discrete adaptation of the ABC algorithm (Set-ABC) in which food sources are represented as sets of parcel identifiers and neighborhood construction is governed by set-theoretic operators rather than continuous vector arithmetic or binary bit-masks [20,21]. Set-ABC and three reference metaheuristics—a Binary ABC (BABC), a binary GA, and Simulated Annealing (SA)—optimize a common objective expressed directly in net economic return (PEN), combining gross timber revenue and extraction cost while discounting statutory species-retention infractions through a penalty term, and assigning zero fitness to any solution that violates a hard constraint. The four methods are calibrated under an identical single-round Grid Search protocol and compared over 20 independent runs with matched random seeds, using a paired statistical pipeline (Friedman test with Bonferroni-corrected Wilcoxon post hoc comparisons). The main contributions are therefore: (i) a formal model of regulated parcel-based Amazonian tree selection with hard and penalized constraints anchored in Peruvian legislation; (ii) an operationally interpretable objective function expressed in monetary units that couples economic return with legal compliance; (iii) a novel set-theoretic discrete ABC adaptation driven by set-algebraic operators; and (iv) a statistically rigorous four-way benchmark of the proposed Set-ABC against BABC, GA, and SA on a semi-synthetic instance derived from a real operational concession, which identifies a well-tuned GA as the reference method to beat while establishing Set-ABC as a stable and feasible contender.

This paper is organized as follows. Section 2 reviews related work on optimization for spatial forest planning. Section 3 formalizes the problem, defining decision variables, hard and penalized constraints, and the objective function. Section 4 details the design of the four metaheuristics: the set-theoretic discrete ABC, the binary ABC, the binary GA, and simulated annealing. Section 5 describes the software architecture, the semi-synthetic dataset, the hyperparameter calibration protocol, and the statistical comparison pipeline. Section 6 reports empirical results, and Section 7 discusses their interpretation and limitations. Section 8 presents conclusions and future research directions.

2. Related Work

The application of optimization techniques in forest management has progressively expanded from primarily economic and timber-production objectives toward comprehensive frameworks that integrate ecological sustainability, regulatory requirements, and biodiversity conservation [1,6,8]. Forest harvest planning continues to face significant operational challenges because many planning activities are still carried out through manual procedures or empirical decision-making, limiting the evaluation of alternative management scenarios [6]. This traditional practice leads to calculation errors, significant delays, and an inability to process the massive combinatorial explosion of variables within reasonable execution times [7]. Spatial optimization problems in forestry are commonly formulated by discretizing the landscape into spatial decision units, such as management stands, production units, or grid cells, to represent large forest areas and enable the application of combinatorial optimization techniques [9,10,12]. Following this modeling paradigm, the present work adopts forest parcels as the spatial decision units, where each parcel aggregates multiple inventoried trees into a single optimization entity [8,12]. Under this representation, tree selection for harvesting or seed-tree preservation becomes a highly complex NP-hard combinatorial optimization problem, closely related to the Set Covering Problem and the Multidimensional Knapsack Problem [8,9].

In specialized literature, spatial forest planning has incorporated various ecological constraints to prevent inefficient decisions that cause ecosystem fragmentation or vulnerability during environmental audits [2,22]. Multiple authors emphasize that spatial cohesion and forest connectivity require harvest blocks to form connected subgraphs, avoiding isolated intervention zones [9,10]. Simultaneously, models must ensure project economic viability by exceeding fixed operational costs while strictly guaranteeing legal thresholds [6,23].

Although these constraints are commonly addressed in the literature from a general forest management perspective, their application in the Peruvian Amazon requires compliance with the national regulatory framework. Harvest planning must satisfy the provisions of the Peruvian Forest and Wildlife Law (Law No. 29763) and the national CITES registry for protected flora species, while also meeting sustainability requirements such as maintaining minimum primary forest retention, canopy cover, and seed-tree preservation ratios [3–5]. Additionally, models must manage enclave tolerance by limiting the number of unharvested parcels inside an active block to avoid disrupting access and transport logistics [6].

Because exact methods suffer from computational stagnation when scaled to real-world instances, research has increasingly turned to metaheuristic algorithms [7,11,19]. Single-solution strategies such as Greedy Randomized Adaptive Search Procedure (GRASP), Tabu Search (TS), and Simulated Annealing (SA) have demonstrated efficiency in discovering near-optimal solutions for forest coverage and infrastructure allocation [9,10]. However, these single-solution metaheuristics show high sensitivity to neighborhood structure design and frequently get trapped in local optima when handling overlapping constraints [7]. Conversely, population-based and bio-inspired metaheuristics, primarily Genetic Algorithms (GA), have shown greater global exploration capacity to optimize harvesting while maintaining landscape resilience and ecosystem services [12–14].

Despite these advances, a fundamental technical gap remains in spatial forest selection [7,8]. While GA implementation continues to be widely applied in multi-objective tree selection and seed-tree preservation [12,13], its reliance on rigid binary chromosome representations frequently generates infeasible solutions when handling non-linear spatial constraints, requiring heavy penalty functions that degrade search efficiency [12,13]. Furthermore, there is a distinct absence of studies applying collective swarm intelligence algorithms, specifically the Artificial Bee Colony (ABC) algorithm, to discrete parcel-based tree selection problems. Swarm-based bio-inspired techniques are widely recognized for their natural mechanism balancing global exploration by scout bees and local exploitation by employed and onlooker bees through cooperative foraging behavior [15,16]. Therefore, developing a discrete adaptation of the ABC algorithm based on set-theoretic operators rather than binary vectors constitutes a novel approach for solving parcel-based tree selection problems under the regulatory and ecological constraints that characterize sustainable forest management in the Peruvian Amazon [3,4,6,15].

3. Problem Formulation

This section formalizes the spatial tree selection problem for sustainable Amazonian harvesting. The objective is to identify an optimal subset of land parcels that maximizes economic yield and harvested area while satisfying non-linear ecological, regulatory, and spatial constraints.

3.1. Decision Variables, Parameters, and Sets

The underlying spatial structure of a forest concession is defined as a graph $G = (V, E)$, where $V = \{1, 2, 3, ..., N\}$ represents the set of total land parcels and E denotes the list of contact relationships between parcels. A candidate solution $X \subseteq V$ corresponds to the subset of selected parcels chosen for harvesting, containing $|X|$ selected parcels. The fundamental parameters and sets governing the model are defined as follows:

- Parcel Attributes: For each parcel $i \in V$, a_i denotes its land area, c_i represents the canopy cover area, c_i^{nat} represents the canopy cover area of primary/native trees, and $p_i \in \{0, 1\}$ indicates whether the parcel is designated as a protected area.
- Species Attributes: $S = \{1, 2, 3, ..., |S|\}$ represents the set of inventoried tree species. For each species $s \in S$, v_s is the commercial timber value per volume unit, c_s is the extraction cost per volume unit, p_s is the penalty per unit of volume exceeded, Q_s is the total initial tree count across the concession, W_s is the total concession wood volume of species s, and $r_s \in [0, 1]$ is the mandatory conservation ratio.
- Parcel-Species Relations: $q_{i,s}$ denotes the number of trees and $w_{i,s}$ denotes the timber volume of species s within parcel i.
- Concession Boundaries: S_{total} represents the total concession land area, C_{total} is the initial total canopy area, and C_{total}^{nat} is the initial total primary canopy area.
- Global Parameters: B_{proj} is the project budget that must cover extraction costs, R_{CC} is the minimum required canopy coverage ratio [24], R_{FC} is the minimum required primary forest conservation ratio, N_U is the maximum allowable unharvested internal parcels [24], G_{max} is the theoretical maximum revenue if all parcels were harvested, and $A_{max} = S_{total}$ is the maximum harvestable area.

3.2. Constraints and Solution Feasibility

A solution $X \subseteq V$ is subject to two classes of constraints: hard constraints (which render a solution completely infeasible with $F(X) = 0$) and penalized constraints (which progressively degrade solution fitness).

3.2.1. Hard Constraints (Feasibility Boundary)

If any of the following six conditions are violated, the fitness evaluation immediately assigns $F(X) = 0$:

1. Spatial Connectivity: The sub-graph induced by the selected parcels X must form a single connected component. Evaluated via Breadth-First Search (BFS) over the list of contact relationships between parcels E, X must satisfy $|X| \geq 3$ and $BFS(X) = |X|$.
2. Internal Enclave Limit: Complete spatial compactness around non-harvestable or protected units is often practically unachievable. Therefore, an operational tolerance threshold N_u is defined for the number of unharvested parcels embedded within the perimeter polygon generated by X, denoted as $H(X)$. This mechanism provides the spatial flexibility required for the metaheuristics to naturally bypass or contour around protected zones without triggering immediate disqualification, while preventing the formation of excessively hollow, fragmented, or structurally inefficient harvest blocks:

$$H(X) \leq N_u \tag{1}$$

3. Exclusion Zones Protection: National forestry regulations strictly forbid timber extraction within ecologically sensitive areas, riparian buffers, steep slopes, or officially declared conservation cores pursuant to Supreme Decree No. 018-2015-MINAGRI [24] and Law No. 29763 [3]. Consequently, no parcel designated as an exclusion/protected unit ($p_i = 1$) can be incorporated into X.

$$\sum_{i \in X} p_i = 0 \quad (2)$$

4. Economic Viability: A forest intervention must remain financially viable within allocated capital boundaries; thus, the total wood extraction $C(X)$ incurred by harvesting the selected parcels must not exceed the available project budget B_{proj}.

$$C(X) = \sum_{i \in X} \sum_{s \in S} (c_s \cdot w_{i,s}) < B_{proj} \quad (3)$$

5. Minimum Forest Canopy Coverage Constraint: Pursuant to the statutory definition of forest ecosystems established in Supreme Decree No. 018-2015-MINAGRI, post-harvest residual tree cover must retain at least the fraction R_{CC} = 0.25 of the initial concession canopy area C_{total}. This guarantees that canopy opening does not exceed the legal threshold that distinguishes managed productive forest from land-use degradation [24]:

$$\frac{C_{total} - \sum_{i \in X} c_i}{C_{total}} \geq R_{CC} \quad (4)$$

6. Primary Forest Conservation Constraint: Forest management in native Amazonian stands is bounded by statutory sustainability criteria governed by Law No. 29763 and the regulations approved under Supreme Decree No. 018-2015-MINAGRI [24], which mandate the preservation of natural forest structure, seed sources, and standing ecological biomass. To avoid permanent ecosystem depletion and severe administrative infractions evaluated under OSINFOR oversight guidelines [25], harvest operations must retain at least the legal fraction R_{FC} = 0.30 of the original native/primary forest canopy cover $C_{totalnat}$. Feasible solutions must strictly satisfy:

$$\frac{C^{nat}_{total} - \sum_{i \in X} c^{nat}_i}{C^{nat}_{total}} \geq R_{FC} \quad (5)$$

3.2.2. Penalty Constraints

Unlike hard constraints, penalty constraints enforce ecological conservation thresholds rooted in Peruvian forestry legislation (Law No. 29763) and regulatory guidelines established by OSINFOR [3,25]. Specifically, OSINFOR's oversight methodology penalizes infractions based on species threat status and ecological value. To reflect these standards, a Species Retention Constraint is considered: for each species $s \in S$, the proportion of remaining unharvested trees relative to their initial population Q_s must meet or exceed the species conservation ratio r_s:

$$\frac{Q_s - \sum_{i \in X} q_{i,s}}{Q_s} \geq r_s, \ \forall s \in S \quad (6)$$

Violating this threshold does not invalidate the candidate solution; rather, an indicator function identifies the breached boundary to compute the extent of non-compliance. The formal mathematical formulation of this penalty function, along with its weighting and integration into the overall objective function, is detailed in Section 3.3.

3.3. Objective Function and Design Justification

The objective function $F(X) \in [0, +\infty[$ evaluates solution quality based on economic profitability while strictly enforcing regulatory, spatial, and budgetary viability. The proposed objective function directly models the net economic return (in monetary currency, PEN) adjusted by linear economic penalties for species-retention infractions.

Formally, for a candidate harvest block $X \subseteq V$, the objective value is computed as:

$$F(X) = max(0, \frac{\lfloor 100 \cdot (G(X) - C(X) - P(X)) \rfloor}{100}) \quad (7)$$

where

- $G(X)$ represents the total gross timber revenue obtained from the harvested parcels:

$$G(X) = \sum_{i \in X} \sum_{s \in S} (v_s \cdot w_{i,s}) \quad (8)$$

- $C(X)$ is the total operational extraction cost across all harvested units:

$$C(X) = \sum_{i \in X} \sum_{s \in S} (c_s \cdot w_{i,s}) \quad (9)$$

- $P(X)$ is the total monetary penalty incurred when the harvested volume of restricted timber species breaches statutory retention quotas (detailed in Section 3.4).
- The scaling factor $\lfloor 100 \cdot (\cdot) \rfloor / 100$ applies a two-decimal floor truncation reflecting financial cent-level precision, and the non-negativity truncation $max(0, \cdot)$ prevents unviable negative returns.

The architectural design of $F(X)$ is justified through two principles:

1. Direct Operational Interpretability: Formulating $F(X)$ in net monetary units provides concession managers and forestry auditors with an immediate, tangible assessment of project profit margins, removing the ambiguity associated with synthetic dimensionless indices.
2. Strict Compliance Guarantee: Solutions breaching hard constraints (such as connectivity, enclave tolerance, exclusion zones, budgetary caps, canopy cover, or primary forest conservation) receive $F(X) = 0$, guaranteeing that only legally and operationally permissible harvesting schedules can achieve positive fitness.

3.4. Formulation of Penalties

Among the ecological restrictions, the species conservation threshold operates with a soft penalty mechanism that penalizes over-harvesting on an economic scale. Let $q_s(X)$ (Equation (10)) be the harvested tree count of species s, and L_s (Equation (11)) be the maximum allowable harvested limit.

$$q_s(X) = \sum_{i \in X} q_{i,s} \quad (10)$$

$$L_s = \lceil Q_s \cdot (1 - r_s) \rceil \quad (11)$$

If the tree count exceeds the permissible ceiling $q_s(X) > L_s$, the surplus harvested timber volume $\Delta W_s(X)$ beyond the legal volumetric quota $W_s \cdot (1 - r_s)$ is penalized:

$$\Delta W_s(X) = max(0, \sum_{i \in X} w_{i,s} - W_s \cdot (1 - r_s)) \quad (12)$$

The aggregate species violation penalty $P(X)$ is calculated as:

$$P(X) = \sum_{s \in S \mid q_s(X) > L_s} (\Delta W_s(X) \cdot p_s) \quad (13)$$

where p_s represents the unit penalty cost assessed per volume unit of timber extracted beyond the sustainable threshold for species s. This penalization rate is grounded in the

administrative enforcement framework and official fining methodology established in Presidential Resolution No. 021-2018-OSINFOR [25]. Under these supervisory guidelines, unauthorized or surplus commercial timber extraction is sanctioned through proportional economic fines scaled by the commercial value of the affected species and the volume unauthorizedly harvested. Integrating p_s directly into the objective function reflects the regulatory liability of exceeding species-level retention quotas, ensuring that the optimization algorithms heavily discount solutions that breach authorized silvicultural thresholds. If $q_s(X) \leq L_s$, no penalty is incurred for that species.

4. Algorithm Design

This section details the algorithmic framework designed for parcel-based forest optimization under Amazonian regulatory and spatial constraints. To thoroughly evaluate performance across distinct search paradigms and encoding schemes, four metaheuristic strategies are implemented: a novel Set-Theoretic Discrete Artificial Bee Colony (Set-ABC), a Binary Artificial Bee Colony (BABC), a traditional Binary Genetic Algorithm (GA), and a single-solution Simulated Annealing (SA) metaheuristic.

Because uniform random parcel selection yields disjoint subgraphs and violates exclusion restrictions with near certainty, all four approaches are initialized using a specialized graph-based constructive heuristic that guarantees spatial connectivity, exclusion zone compliance, and enclave feasibility. Subsequently, the algorithms explore the combinatorial landscape using either discrete set-algebraic operators (Set-ABC) or bitwise transformations over binary strings (BABC, GA, and SA) to evaluate the relative merits of set-based versus binary solution structures.

4.1. Feasible Solution Initialization

Generating feasible initial solutions uniformly at random over a spatial lattice is statistically improbable due to strict topological and ecological constraints: spatial connectivity and exclusion of protected areas. Selecting parcels independently would almost certainly produce disconnected subgraphs or violate protected barriers, leading to degenerate initial states where $F(X) = 0$.

To supply all evaluated metaheuristics with viable initial points, an iterative, graph-based constructive heuristic is implemented (Algorithm 1):

Algorithm 1: Pseudocode for the proposed random generation of solution

Input: n (length proposed of solution), G (All connected parcels)
Output: X
Visited ← InitializeVisitedMap(G,False)//Boolean vector with parcel index
parcel ← *Null*
while parcel = *Null* **or** is Protected(parcel) **then**
  parcel ← RandomParcelChoose(G)
  Visited[parcel] ← True
end while
Visited[parcel] ← True
Add(X, *parcel*)
for counter ← 0 **to** n **then**
  N← GetNeighborhoodOf(X, G)//all neighboring parcels of X without repetition
  for each neighbor **in** N
    if not is Protected(neighbor) **then**

Algorithm 1: *Cont.*

```
            Add(X, neighbor)
            break
        Visited[neighbor] ← True
    end for
```

4.2. Set-Theoretic Discrete Artificial Bee Colony (Set-ABC) Adaptation

Standard ABC implementations operate in continuous domains using floating-point vector arithmetic. To address the discrete combinatorial nature of parcel selection without relying on continuous rounding heuristics, this study develops a set-theoretic adaptation of ABC.

4.2.1. Food Source and Nectar Representation

- Nectar Unit: Represents the minimal atomic component of a solution, defined directly as a unique integer parcel identifier rather than a binary bit [17]. This formulation is hypothesized to enhance search performance and solution quality, as manipulating spatial subsets through direct set operators may guide neighborhood exploration more naturally than independent bit-flipping across fixed-length binary strings.
- Food Source (X_i): Represents a candidate harvest solution composed of a variable-length list of unique parcel identifiers (Equation (14)), its evaluated fitness value $F(X_i)$, and a visit counter $trial_i$.

$$X_i = \{p_1, p_2, \ldots, p_k\} \tag{14}$$

- Colony Population and Search Boundaries: The colony manages a population of SN food sources whose cardinality (number of selected parcels) is bounded by lower and upper operational search boundaries, L_{inf} and L_{sup}, respectively:

$$\left|L_{inf}\right| \leq |X_i| \leq \left|L_{sup}\right| \tag{15}$$

- Boundary Violation Replacement and Food Source Exhaustion: During the employed and onlooker bee phases, candidate neighbors V_i are generated through set perturbation. If the resulting parcel count of an accepted candidate breaches either operational threshold (Equation (15)), the solution is reset to the corresponding boundary benchmark to enforce spatial feasibility within the designated search window. In parallel, if a food source fails to improve its fitness over a predefined number of consecutive iterations $trial_i \geq Limit$, it is abandoned during the scout phase and replaced by a newly constructed feasible solution generated within $\left[\left|L_{inf}\right|, \left|L_{inf}\right|\right]$ via Algorithm 1, resetting $trial_i$ to 0.

4.2.2. Set-Theoretic Operators and Neighborhood Search

Neighboring food source construction V_i from an active food source X_i and a randomly chosen food source X_k is governed by discrete set algebra rather than continuous vector addition [20]. The discrete operators are defined as follows:

1. Set Addition ($\oplus$): Given parcel sets A and B, $C = A \oplus B$ generates the set union $A \cup B$ without duplicate parcel identifiers.
2. Set Subtraction ($\ominus$): Given parcel sets A and B, $C = A \ominus B$ generates the set difference A/B, removing all parcels present in B from A.

4.2.3. New Food Source Construction Mechanism

In the standard continuous formulation of the Artificial Bee Colony (ABC) algorithm, a candidate food source V_i is generated by perturbing the current solution X_i with respect to a randomly selected food source X_k using continuous vector arithmetic (Equation (16)), where $\phi_{i,j} \in [-1, 1]$ is a uniform random number [18]:

$$V_{i,j} = X_{i,j} + \phi_{i,j} \cdot \left(X_{i,j} - X_{k,j} \right) \tag{16}$$

To adapt this mechanism to the discrete parcel-selection domain without relying on continuous rounding heuristics, this study reformulates the perturbation process using set-theoretic operations, similar to the process used in [21]. The logic for generating a new food source is defined as follows (Algorithm 2):

Algorithm 2: Pseudocode for the proposed food source construction

Input: i (food source index), SN
Output: V_i(new food source)
1: k ← RandomIndex(0, | SN |)
2: ϕ ← RandomRealNumber(−1,1)
3: X_i ← SN[i]
4: X_k ← SN[k]
5: **if** $i \neq k$ **and** $|\phi| > 0.01$ **then**
6: only_current ← $X_i \ominus X_k$
7: only_random ← $X_k \ominus X_i$
8: **if** $\phi > 0$**then**
9: n ← $\lfloor \phi \cdot |$only_random$| \rfloor$
10: V_i ← $X_i \oplus$ RandomNectarChoice(only_random, n)
11: **else**
12: n ← $\lfloor -\phi \cdot |$only_current$| \rfloor$
13: V_i ← $X_i \ominus$ RandomNectarChoice(only_current, n)
14: **end if**
15: **else**
16: V_i← X_i
17: **end if**

An important point to highlight is that in lines 5 and 16, having equal indexes or a very low ϕ, we double the original food source to force the $trial_i$ in the proposed food source evaluation (Algorithm 3) and increase the probability of replacement in the Scout phase (Algorithm 4).

4.2.4. Source Replacement and Bounded Evaluation

Source replacement follows a selection mechanism bounded by parcel search limits. If the newly generated candidate source V_i achieves a strictly higher fitness amplitude than X_i, it replaces the current solution. However, to maintain search feasibility within the designated spatial bounds, if the parcel count of V_i breaches either the upper limit or lower limit, the solution is reset to the boundary benchmark. If V_i fails to improve fitness, X_i is retained and its visit counter increments ($trial_i$) [16]. The replacement evaluation logic is defined as follows (Algorithm 3):

Algorithm 3: Pseudocode for the proposed food source evaluation

Input: X_i (current food source), V_i(new food source), L_{inf}(lower limit), L_{sup}(upper limit)
Output: V_i(new food source)

```
fitness_current ← F(X_i)(Equation (7))
fitness_new ← F(V_i)(Equation (7))
if fitness_new > fitness_current then
    if |V_i| > |L_sup| then
        V_i ← L_sup
    else
        if |V_i| < |L_inf| then
            V_i ← L_inf
    end if
    end if
else
    trial_i ← trial_i + 1
    V_i ← X_i
end if
```

4.2.5. Overall Set-Theoretic Discrete Artificial Bee Colony Algorithm

The overall execution flow of the adapted Set-Theoretic Discrete Artificial Bee Colony algorithm integrates the food source initialization, the three foraging phases (Employed, Onlooker, and Scout bees), and the global best solution memorization across iterations (Algorithm 4).

- Initialization Phase: Generates an initial colony of *Size* candidate solutions, where each food source is represented as a random set of unique parcel identifiers bounded by L_{inf}(lower limit) and L_{sup}(upper limit).
- Employed Bee Phase: Each employed bee explores the discrete neighborhood of its assigned food source by applying set-theoretic operations ($\oplus$ and $\ominus$) with a randomly selected peer, greedily retaining the candidate solution if its net economic return improves.
- Onlooker Bee Phase: Onlooker bees probabilistically choose promising food sources through roulette wheel selection based on relative fitness and conduct localized set perturbations to exploit high-yield foraging regions.
- Scout Bee Phase: When a food source fails to improve after a predetermined threshold (Limit), it is considered exhausted and abandoned, prompting a scout bee to generate a new random parcel set to prevent search stagnation.
- Memorization/Update Phase: Evaluates the entire updated population at the end of each iteration to record and preserve the global best harvest block (X_{best}, f_{best}) discovered so far.

4.3. Binary Artificial Bee Colony (BABC) Adaptation

Standard ABC implementations operate in continuous domains using floating-point vector arithmetic. To adapt the collective foraging mechanism to the binary combinatorial space of parcel selection without continuous rounding, this study implements a Binary Artificial Bee Colony (BABC) formulation based on bitwise Boolean operators [20].

Algorithm 4: Pseudocode for Set-Theoretic Discrete Artificial Bee Colony (Set-ABC)

Input: MaxIter, Limit, Size, L_{inf}(lower limit), L_{sup}(upper limit)
Output: X_{best} (best food source), f_{best} (best fitness)

SN ← InitializePopulation($Size, L_{inf}, L_{sup}$)
for it = 1 to MaxIter **do**
  for i = 1 to |SN| **do//**Employed phase
    X_i ← SN[i]
    V_i ← NewSourceConstruction(i, SN) (Algorithm 2)
    SN[i] ← FoodSourceEvaluation($X_i, V_i, L_{inf}, L_{sup}$) (Algorithm 3)
  end for
  for i = 1 to |SN| **do//**Onlooker phase
    k ← SelectionFoodSourceIndex("Roulette Wheel",SN)
    V_k ← NewSourceConstruction(k, SN) (Algorithm 2)
    SN[k] ← FoodSourceEvaluation($X_k, V_k, L_{inf}, L_{sup}$) (Algorithm 3)
  end for
  For each X_i **with** $trial_i \geq Limit$ **do//**Scout phase
    do
      n ← RandomPositiveNumber($\left|L_{inf}\right|, \left|L_{sup}\right|$)
    X_i ← RandomSolution(n, G) (Algorithm 1)
    while $F(X_i) = 0$ **or** SN contains X_i
    SN[i] ← X_i
    $trial_i$ ← 0
  end for
  (X_{best}, f_{best}) ← UpdateBest(SN, X_{best}, f_{best})
end for

4.3.1. Food Source and Nectar Representation Adaptation

- Nectar Representation: A binary bit where 1 indicates parcel selection for harvesting and 0 indicates exclusion [17].
- Food Source: A fixed-length binary vector (Equation (17)) of length |V| (total concession parcel count), its evaluated fitness value $F(X_i)$, $count(X_i)$ is the number of binary bits in 1 and a visit counter $trial_i$.

$$X_i = \left\{p_1, p_2, \ldots, p_{|V|}\right\} \tag{17}$$

- Colony Population and Search Boundaries: The colony manages a population of SN food sources whose cardinality (number of selected parcels) is bounded by lower and upper operational search boundaries, L_{inf} and L_{sup}, respectively:

$$count\left(L_{inf}\right) \leq count(X_i) \leq count\left(L_{sup}\right) \tag{18}$$

- Boundary Violation Replacement and Food Source Exhaustion: The candidate evaluation, boundary restriction mechanism, and scout abandonment logic in BABC follow the exact same formulation described for Set-ABC in Section 4.2.4 (Algorithm 2). The only distinction lies in the neighborhood generation operator, where the candidate solution is constructed using the bitwise Boolean formulation (Equations (17) and (18)) instead of the set-theoretic perturbation mechanism (Equations (13) and (14)).

4.3.2. Bitwise Operators and Neighborhood Search

Neighboring food source construction V_i from an active food source X_i and a randomly chosen food source X_k is governed by bitwise Boolean logic rather than continuous vector arithmetic [20]. The bitwise operators are defined as follows:

1. Bitwise Exclusive OR ($\oplus$): Given binary solutions A and B, $C = A \oplus B$ identifies coordinate-level discrepancies between both configurations, returning 1 for indices where the parcel selection states differ and 0 where they match.
2. Bitwise AND ($\bigotimes$): Given binary vectors A and B, $C = A \bigotimes B$ performs the standard conjunction operation across each bit position, yielding 1 if and only if the corresponding bits in both A and B are equal to 1, and 0 otherwise.

4.3.3. New Food Source Construction Mechanism Adaptation

In contrast to the set-theoretic operators deployed in Set-ABC, the Binary Artificial Bee Colony (BABC) generates candidate solutions over a binary domain using bitwise Boolean arithmetic. Given the current binary food source $X_i \in \{0,1\}^{|V|}$ and a distinct randomly selected neighbor $X_k \in \{0,1\}^{|V|}$, a candidate binary solution V_i is constructed via [18]:

$$V_i = X_i \oplus \left(B \bigotimes (X_i \oplus X_k) \right) \tag{19}$$

where:

- $\oplus$ *and* $\bigotimes$ denote the bitwise XOR and bitwise AND operators defined in Section 4.3.2, respectively.
- $B \in \{0,1\}^{|V|}$ is a stochastic binary vector of length units $|V|$ that replaces the continuous scaling factor ϕ. Each element B_j is sampled independently from a Bernoulli distribution with equal probability ($p = 0.5$):

$$B_j = \begin{cases} 1, & if\ U(0,1) > p \\ 0, & otherwise \end{cases} \tag{20}$$

Here, the term in $(X_i \oplus X_k)$ identifies the exact parcel selection discrepancies between the two solutions, while the conjunction with vector B selectively filters which discrepancies will perturb X_i. The final XOR operation applies these filtered changes, producing a new candidate binary configuration V_i whose active parcel count is recalculated prior to fitness evaluation (Algorithm 5):

Algorithm 5: Pseudocode for the food source construction BABC

Input: i (food source index), SN
Output: V_i(new food source)

```
k ← RandomIndex(0, |SN|)
X_i ← SN[i]
X_k ← SN[k]
if i ≠ k then
    B ← InitializeVector(|X_i|,0)
    for each B_j in B
        B_j ← RandomRealNumber(0,1) > 0.5
    end for
    V_i = X_i ⊕ (B ⊗ (X_i ⊕ X_k))
else
    V_i ← X_i
end if
```

4.3.4. Source Replacement and Bounded Evaluation Adapted

The source evaluation and selection mechanisms in BABC mirror the logic formulated for Set-ABC in Section 4.2.4 (Algorithm 2). The primary operational difference lies in how candidate solution cardinality is computed: because solutions in BABC are encoded as fixed-length binary vectors rather than variable-sized sets, the set-cardinality operator $|V_i|$ is replaced by $count(V_i)$, defined as the total number of active bits (1's) across the vector.

This active-bit count represents the exact number of selected harvesting parcels. If the newly generated binary solution V_i improves fitness over X_i. its active parcel count is checked against the search limits. If Equation (18) is false, the solution is reset to the corresponding boundary benchmark ($trial_i \leftarrow trial_i + 1$).

4.3.5. Overall Binary Artificial Bee Colony Algorithm

The overall execution flow of the Binary Artificial Bee Colony (BABC) algorithm follows the exact same structural framework as the Set-based ABC described in Section 4.2.5. It integrates the food source initialization, the three foraging phases (Employed, Onlooker, and Scout bees), and the global best solution memorization across iterations (Algorithm 6), differing only in the underlying representation—binary bit-vectors instead of parcel-identifier sets—and the corresponding discrete operators used for neighborhood construction.

Algorithm 6: Pseudocode for Binary Artificial Bee Colony (BABC)

Input: MaxIter, Limit, Size, L_{inf}(lower limit), L_{sup}(upper limit)
Output: X_{best} (best food source), f_{best} (best fitness)

SN ← InitializePopulation($Size, L_{inf}, L_{sup}$)
for it = 1 to MaxIter **do**
  for i = 1 to |SN| **do//**Employed phase
    $X_i \leftarrow$ SN[i]
    $V_i \leftarrow$ NewSourceConstruction(i, SN) (Algorithm 5)
    SN[i] ← FoodSourceEvaluation($X_i, V_i, L_{inf}, L_{sup}$) (Algorithm 3 using *count*)
  end for
  for i = 1 to |SN| **do//**Onlooker phase
    $k \leftarrow$ SelectionFoodSourceIndex("Roulette Wheel",SN)
    $V_k \leftarrow$ NewSourceConstruction(k, SN) (Algorithm 5)
    SN[k] ← FoodSourceEvaluation($X_k, V_k, L_{inf}, L_{sup}$) (Algorithm 3 using *count*)
  end for
  For each X_i **with** $trial_i \geq Limit$ **do//**Scout phase
    do
      n ← RandomPositiveNumber($count\left(L_{inf}\right), count\left(L_{sup}\right)$)
      $X_i \leftarrow$ RandomSolution(n,G) (Algorithm 1)
    while $F(X_i) = 0$ **or** SN contains X_i
    $trial_i \leftarrow 0$
  end for
  (X_{best}, f_{best}) ← UpdateBest(SN, X_{best}, f_{best})
end for

4.4. Genetic Algorithm (GA) Adaptation

To benchmark the performance of the set-theoretic ABC algorithm, a multi-point Genetic Algorithm (GA) is implemented using a traditional binary encoding scheme.

4.4.1. Chromosome and Population Structure

- Gen Representation: A binary bit where 1 indicates parcel selection for harvesting and 0 indicates exclusion.
- Chromosome/Individual: A fixed-length binary vector (Equation (21)) of length |V| (total concession parcel count), paired with its calculated fitness $F(X_i)$.

$$X_i = \left\{ p_1, p_2, \ldots, p_{|V|} \right\} \tag{21}$$

- Population: A collection of individuals evolved over a fixed number of generations.

4.4.2. Genetic Operators

- Parent Selection: Employs roulette wheel selection proportional to individual fitness, complemented by a replacement rate to preserve top-performing solutions across generations.
- Multi-Point Crossover: Swaps multiple sub-segments of binary chromosomes between two parents using randomly chosen cut points. Segments are exchanged based on the crossover rate (CRate) [12,14].
- Single-Point Bit-Flip Mutation: Operates hierarchically at two levels to introduce local genetic diversity:
    - Probability Mutation (Pmut): Operates at the gene level. For each selected offspring, a target locus is flipped $(0 \rightarrow 1 \; or \; 1 \rightarrow 0)$ with probability Pmut, which corresponds to including or excluding an individual forest parcel from the harvest plan.
    - Mutation Rate (MRate): Governs the individual level, determining the proportion of offspring in the mating pool that is subjected to the mutation operator.
- Survivor Selection: Applies ranking-based replacement to construct the next generation, retaining high-fitness parents alongside newly mutated offspring [13].

4.4.3. Overall Binary Genetic Algorithm

The overall execution flow of the adapted Binary Genetic Algorithm integrates five sequential phases executed across successive generations (Algorithm 7):

- Initialization: Generates an initial population of binary chromosomes of size (Size), where each bit randomly represents parcel selection.
- Parent Selection: Chooses candidate parent chromosomes through roulette wheel selection biased toward higher fitness values, retaining an elitist fraction defined by Replacement Rate (RRate).
- Parent Crossover: Combines pairs of selected parents via multi-point crossover according to CRate, interchanging chromosome segments to produce preliminary offspring.
- Offspring Mutation: Modifies a fraction MRate of generated children by flipping individual parcel bits according to probability Pmut to maintain genetic diversity and prevent premature convergence.
- Survivor Selection: Evaluates the candidate offspring, sorts individuals using ranking selection, and constructs the next generation by merging the best-performing parents with the highest-fitness offspring.

Algorithm 7: Pseudocode for Genetic Algorithm (GA)

Input: MaxIter, Size, RRate (Replacement Rate), CRate (Crossover Rate), MRate (Mutation Rate), Pmut (Probability Mutation)
Output: X_{best} (best individual source), f_{best} (best fitness)
population ← InitializePopulation(*Size*)
for it = 1 to MaxIter **do**
  selectedParents ← ParentsSelection("Roulette Wheel", population, RRate)
  newChildren ← ParentsCrossover("Multi-Point Crossover", selectedParents, CRate)
  OffspringMutation("Single-Point Bit-Flip Mutation", newChildren, MRate, Pmut)
  population ← SurvivorSelection("Ranking", population, newChildren, RRate)
  $(X_{best},\ f_{best})$ ← UpdateBest(population, $X_{best},\ f_{best}$)
end for

4.5. Simulated Annealing (SA) Adaptation

Simulated Annealing is implemented as a single-solution trajectory metaheuristic to explore whether a local search mechanism guided by probabilistic uphill transitions can effectively navigate the constrained parcel-selection landscape [11].

4.5.1. Solution Representation and Initial State

A candidate solution X is represented as a dynamic list of parcel identifiers (Equation (22)):

$$X = \{p_1, p_2, \ldots, p_k\} \tag{22}$$

matching discrete set-based encoding utilized in Set-ABC. The search initiates from an initial state X_0 generated using Algorithm 8, guaranteeing that the trajectory starts from a regulation-compliant configuration with positive net profit $F(X) > 0$.

Algorithm 8: Pseudocode for Construct Feasible Solution

Input: G $(Graph\ of\ parcels)$
Output: X (Solution with $F(X) > 0$)
do
  $X \leftarrow$ RandomSolution($|G.V|, G$) (Algorithm 1)
while $F(X) = 0$
$(X_{best},\ f_{best}) \leftarrow (X,\ F(X))$
end for

4.5.2. Neighborhood Generation Strategy: Single-Element Variation

Candidate neighbors $X \in \mathcal{N}(X_0)$ are produced via the *Single-Element Variation* operator (Algorithm 9). At each step, a candidate copy is perturbed by either expanding or pruning the current harvest block, governed by a stochastic decision threshold ($p_{decision} \in [0,1]$):

- Parcel Addition ($random(0,1) < p_{decision}$): The operator queries the concession adjacency matrix to gather all unprotected candidate units bordering the active block. After shuffling this candidate neighborhood, adjacent parcels are sequentially tested by temporarily inserting them into the block and evaluating feasibility. The first neighbor yielding a feasible configuration (or upon reaching the search cap of tested candidates) is retained.

- Parcel Removal ($random(0,1) \geq p_{decision}$): The operator randomly shuffles the selected parcels in the current solution and tests the removal of individual parcels one by one. The first deletion that preserves block feasibility (ensuring connectivity is not broken and enclave limits are respected) is committed.

If no feasible single-element modification is found within the candidate exploration limit, the perturbed state is evaluated as generated, triggering the death-penalty assignment ($F(X) = 0$) if constraints are breached.

Algorithm 9: Pseudocode for Single-Element Variation Neighborhood Generation

Input: X_0 (Current solution), $p_{decision}$, max_eval, (maximum neighbor attemps, set to 1), G $(Graph\ of\ parcels)$
Output: X(Solution with $F(X) > 0$)

```
X ← clone(X_0)
p ← RandomRealNumber(0,1)
if p < p_decision then
    unprotected ← FilterUnprotectedParcels(G)
    visited ← InitializeVisitedMap(X_0, unprotected,True)
    N ← GetNeighborhoodOf(X, G, visited)//function used in Algorithm 1
    Shuffle(N)
    attempts ← 0
    for each neighbor in N
        Add(X, neighbor)
        if F(X) > 0 or attempts > max_eval then
            break
        end if
        Remove(X, neighbor)
        attempts ← attempts + 1
    end for
else
    selected_parcels ← extract(X)
    attempts ← 0
    for each parcel in selected_parcels
        Add(X, parcel)
        if F(X) > 0 or attempts > max_eval then
        Remove(X, parcel)
        attempts ← attempts + 1
    end for
end if
```

4.5.3. Acceptance Criterion, Cooling Schedule, and Algorithmic Workflow

Because the objective function calculates net economic profit in monetary units (PEN), raw fitness deviations between states can vary by multiple orders of magnitude. Evaluating unscaled differences would instantly saturate the exponential acceptance term. Consequently, relative transitions are evaluated through a normalized objective difference δ_{norm} scaled against the current state fitness (Equation (23)):

$$\delta_{norm} = \frac{F(X_0) - F(X)}{F(X_0)} \tag{23}$$

For a maximization framework:

- If ($\delta_{norm} < 0$): the candidate neighbor yields higher net profit and is unconditionally accepted ($X_0 \leftarrow X$).
- If ($\delta_{norm} \geq 0$): the deteriorating transition is accepted probabilistically according to the Metropolis criterion (Equation (24)):

$$P(accept) = e^{\frac{-\delta_{norm}}{T}} \tag{24}$$

where a random value $x \in [0,1]$ is sampled; if $x < e^{\frac{-\delta_{norm}}{T}}$, the worsening move is committed to allow escaping local optima.

System temperature decays geometrically according to $T_{k+1} = \alpha T_k$ where $\alpha \in]0,1[$ denotes the cooling coefficient applied after running an equilibrium chain of iterations ITER at each temperature stage. The entire optimization sequence is detailed in Algorithm 10.

Algorithm 10: Pseudocode for Simulated Anneling (SA)

Input: T_0 (Initial Temperature), T_{min} (Stopping Temperature), α (Cooling rate), ITER (iterations per temperature level), $p_{decision}$ (Decision Probability), max_eval (Maximum number of attempts)
Output: X_{best} (best food source), f_{best} (best fitness)

$X_0 \leftarrow$ ConstructFeasibleSolution(G) (Algorithm 8)
$T \leftarrow T_0$
$(X_{best}, f_{best}) \leftarrow (X_0, F(X_0))$
while $T > T_{min}$ **do**
  for iter = 1 **to** ITER **do**
    $X \leftarrow$ GenerateSingleChangeNeighbor(X_0, $p_{decision}$, max_eval, G) (Algorithm 9)
    if $F(X) = 0$ **then**
      $X \leftarrow$ ConstructFeasibleSolution(G) (Algorithm 8)
    end if
    $\delta_{norm} = \frac{F(X_0)-F(X)}{F(X_0)}$
    if $\delta_{norm} > 0$ **then**
      $X_0 \leftarrow X$
      $(X_{best}, f_{best}) \leftarrow$ UpdateBest(X_0, X_{best}, f_{best})
    else if randomRealNumber(0,1) $< e^{\frac{-\delta_{norm}}{T}}$ **then**
      $X_0 \leftarrow X$
    end if
  end for
  $T \leftarrow \alpha T$
end while

5. Implementation and Experimental Setup

This section outlines the software stack, execution environment, synthetic dataset specifications, hyperparameter tuning protocol, and statistical comparison pipeline used to evaluate both metaheuristics.

5.1. Software Architecture and Execution Environment

The optimization platform was developed in Java 21 using the Spring Boot framework (v4.0.3) for service orchestration and experiment automation. Automated testing and unit validation were implemented using the Spring Boot Starter Test module, incorporating JUnit 5 and Mockito frameworks. Experiments were executed on an 8-core x86_64 architecture

running 64-bit Windows 10 Pro, powered by an Intel Core i5-10310U CPU @ 1.70 GHz (2.21 GHz boost) with 16.0 GB of installed RAM.

5.2. Dataset and Instance Generation

Algorithm performance was benchmarked on a semi-synthetic dataset derived from the real operational forest inventory of the ecological plantation "EL PAISA" located in Nauta, Loreto, Peru. The concession landscape discretizes a total surface area (S_{total}) of approximately 99 hectares into $|V| \approx 90$ contiguous land parcels. The instance incorporates 12 commercial Amazonian timber species identified across the operational inventory [12].

To construct a realistic yet controlled benchmark under the updated regulatory and economic formulation, data parameters were integrated and synthesized from multiple official and experimental sources:

- Spatial Parcel Topography and Floristic Inventory**:** Baseline tree species occurrences, diameter classes, and spatial distributions were drawn from the primary forest census at "EL PAISA". To adapt these counts to parcel-level management units, tree densities, species counts per parcel, and canopy coverage fractions were generated using bounded uniform sampling calibrated against the operational field density ranges.
- Statutory and Conservation Baselines: Minimum forest canopy cover ($R_{CC} = 0.25$), native primary forest conservation ratios ($R_{FC} = 0.3$), and protected exclusion zones were anchored in national forestry thresholds mandated by Supreme Decree No. 018-2015-MINAGRI [24]. Species-specific conservation ratios reflect official categorization lists and CITES-Peru flora classifications [4].
- Economic Valuation, Extraction Costs, and Penalties: Commercial wood sale values per cubic meter were established using regional timber market references from SERFOR [5]. Species-level extraction costs and infraction penalty coefficients were derived from the administrative valuation methodology formulated in Presidential Resolution No. 021-2018-OSINFOR (*Metodología N.° 001-2018-OSINFOR*) [25].
- Experimental Parameters: Operational limits such as the available project execution budget, timber volume conversions, and the internal enclave tolerance ceiling were configured experimentally to provide a balanced, non-trivial combinatorial search space. Furthermore, candidate parcel-search boundaries governing solution initialization and replacement bounds in the Artificial Bee Colony variants (Set-ABC and BABC)—spanning exploration windows from 25% to 80% of the total dataset parcels $|V|$ are fully documented and accessible within the public project repository.
- The global problem parameters are summarized as follows:
- Available Project Budget (B_{proj}): 900,000 PEN.
- Minimum Forest Canopy Coverage Ratio (R_{CC}): 25% (0.25).
- Primary Forest Conservation Ratio (R_{FC}): 30% (0.30).
- Maximum Internal Enclave Tolerance (N_u): 5 unharvested internal parcels.

5.3. Hyperparameter Calibration

To guarantee a rigorous and computationally fair comparison, algorithm hyperparameters were calibrated using a systematic single-round Grid Search (GS) over the semi-synthetic forest instance. Unlike multi-round or asymmetric heuristics, all population-based metaheuristics (Discrete Set-based ABC, Binary ABC, and Binary GA) operated under a common computational budget fixed at 1000 iterations (or generations) and a population size of 300 individuals (or food sources) [14,26].

For Simulated Annealing (SA), the geometric cooling schedule $T_k = T_0 \alpha^k$ was analytically calibrated to match this computational budget. The initial temperature was set to $T_0 = -0.25/\ln(0.9) \approx 2.3728$ and the minimum (stopping) temperature to

$T_{min} = -0.001/\ln(0.0001) \approx 1.0857 \times 10^{-4}$. Under a cooling rate $\alpha = 0.99$, the outer temperature loop yields:

$$K = \frac{\ln(T_{min}/T_0)}{\ln(\alpha)} = 995 \approx 1000 \text{ steps} \quad (25)$$

To align SA with the 300 evaluations per iteration of the population-based methods, the internal search budget was factored into combinations satisfying Iterations per temperature level (ITER) times Neighborhood candidates (max_eval), ensuring an identical evaluation budget across all four algorithms.

The structural baseline parameters held constant across the calibration are detailed in Table 1.

Table 1. Baseline parameters across evaluated metaheuristics.

SA		BABC		SetABC		GA	
Name	**Value**	**Name**	**Value**	**Name**	**Value**	**Name**	**Value**
T_0	2.3728	Iteration	1000	Iteration	1000	Iteration	1000
T_{min}	1.0857×10^{-4}	Population	300	Population	300	Population	300
α	0.99	Constructor	Algorithm 1	Constructor	Equation (7)	Parent selection	Roulette wheel
-	-	Evaluation	Equation (3)	Evaluation	Equation (3)	Type crossover	Multi-point crossover
-	-	-	-	-	-	Type mutation	Single-point bit-flip
-	-	-	-	-	-	Survivor selection	Ranking

The discrete parameter grid evaluated candidate operators governing exploration-exploitation trade-offs, neighborhood perturbation windows, and selection pressures. The complete grid of tested candidate values is summarized in Table 2.

Table 2. Candidate hyperparameter search space tested in the Grid Search.

Algorithm	Hyperparameter	Value
SA	Iterations per temperature time Neighbors	(15,20); (20,15); (10,30)
	Decision constant	0.33, 0.40, 0.50, 0.60, 0.66
BABC	Abandonment limit	20, 30, 40
	Search boundaries	[30%, 70%], [25%, 75%], [50%, 80%] of $\|V\|$
	Food source selection	Roulette Wheel, Tournament (33% of $\|SN\|$), Tournament (25% of $\|SN\|$)
SetABC	Abandonment limit	20, 30, 40
	Search boundaries	[30%, 70%], [25%, 75%], [50%, 80%] of $\|V\|$
	Food source selection	Roulette Wheel, Tournament (33% of $\|SN\|$), Tournament (25% of $\|SN\|$)
GA	Crossover Rate	0.1, 0.3, 0.5
	Mutation Rate	0.1, 0.3, 0.5
	Probability Mutation	0.5, 0.7, 0.9
	Replacement Rate	0.5, 0.6, 0.7

Note: The GA parameters (Crossover Rate = 0.1, Mutation Rate = 0.1, Probability Mutation = 0.5, Replacement Rate = 0.6) were used by Gomide [14].

The optimal configuration for each metaheuristic was determined through a single-round Grid Search evaluated across four representative test seeds (42, 640, 1000, and 2000). The hyperparameter selection followed a modal-frequency criterion: the configuration most frequently identified as optimal across the four seeds was selected as the final setting. In the event of a tie in selection frequency between candidate configurations, the tie was

resolved in favor of the parameter set that achieved the strictly higher net economic return (fitness) in Table 3.

Table 3. Grid Search results.

Algorithm	Seed	Best Parameters	Best Fitness
BABC	42	Limit 30, [50%, 80%], Roulette	1,035,331.34
	640	Limit: 40, [50%, 80%], Roulette	1,042,712.98
	1000	Limit: 30, [30%, 70%], Roulette	1,044,223.15
	2000	Limit: 40, [50%, 80%], Roulette	1,030,204.39
Set-ABC	42	Limit: 30, [30%, 70%], Roulette	985,833.42
	640	Limit: 30, [25%, 75%], Roulette	986,147.00
	1000	Limit: 30, [50%, 80%], Roulette	983,385.56
	2000	Limit: 30, [25%, 75%], Roulette	986,147.00
GA	42	CRate = 0.5, MRate = 0.5, Pmut = 0.9, RRate = 0.7	1,061,032.19
	640	CRate = 0.5, MRate = 0.5, Pmut = 0.9, RRate = 0.7	1,061,032.19
	1000	CRate = 0.3, MRate = 0.5, Pmut = 0.9, RRate = 0.7	1,061,032.19
	2000	CRate = 0.5, MRate = 0.5, Pmut = 0.9, RRate = 0.6	1,061,032.19
SA	42	(20 iter./T × 15 neighbor), Decision = 0.50	1,039,281.63
	640	(15 iter./T × 20 neighbor), Decision = 0.50	1,037,601.93
	1000	(20 iter./T × 15 neighbor), Decision = 0.60	1,041,478.61
	2000	(20 iter./T × 15 neighbor), Decision = 0.60	1,036,141.17

The final selected hyperparameter configuration (Table 4):

- Genetic Algorithm (GA): The parameter configuration CRate = 0.5, MRate = 0.5, Pmut = 0.9, RRate = 0.7 emerged as the modal setting across test seeds (selected in 50% of the trials, seeds 42 and 640). This parameter consistently reached the maximum global net return of 1,061,032.19 PEN prior to generation 300, demonstrating rapid convergence and high search stability.
- Set-based ABC: The configuration characterized by an abandonment limit of Limit = 30, parcel search boundaries bounded between [25%, 75%] of $|V|$, and roulette-wheel selection was the most frequently selected setup across seeds (seeds 640 and 2000). Furthermore, this setting attained the highest overall net economic return recorded for the set-theoretic approach (986,147.00 PEN).
- Binary ABC (BABC): The parameter set combining Limit = 40, search boundaries bounded between [50%, 80%] of $|V|$, and roulette-wheel selection was selected in half of the tuning runs (seeds 640 and 2000). However, when compared with the combinations of Limit = 30 (seeds 1000 and 42), the [50%, 80%] threshold consolidated a presence in 3 of the 4 seeds evaluated.
- Simulated Annealing (SA): The cooling and neighborhood exploration scheme of 20 inner iterations per temperature with 15 candidate neighbors per iteration proved dominant, appearing in three of the four tuning seeds (42, 1000, and 2000). To resolve the tie between decision constants 0.50 and 0.60, the parameter 0.60 was chosen as it achieved the strictly superior net return on seed 1000 (1,041,478.61 PEN versus 1,039,281.63 PEN).

Table 4. Final configuration.

Algorithm	Best Parameters	Best Fitness
SA	Iterations per temperature	20
	Neighborhood sample size	15
	Decision constant (delta)	0.60
	Initial temperature (T0)	$-0.25/\ln(0.9)$ (~2.373)
	Minimum temperature (Tmin)	$-0.001/\ln(0.0001)$ (~1.086×10^{-4})
	Cooling rate (alpha)	0.99
BABC	Iterations	1000
	Population (Food sources)	300
	Abandonment limit (Limit)	30
	Search boundaries	[50%, 80%] of $\|V\|$
	Selection operator	Roulette Wheel
Set-ABC	Iterations	1000
	Population (Food sources)	300
	Abandonment limit (Limit)	30
	Search boundaries	[25%, 75%] of $\|V\|$
	Selection operator	Roulette Wheel
GA	Generations (Iterations)	1000
	Population size	300
	Parent selection	Roulette Wheel
	Crossover operator	Multi-Point Crossover
	Crossover rate (CRate)	0.50
	Type Mutation	Single-point bit-flip
	Mutation rate (MRate)	0.5
	Mutation probability (Pmut)	0.90
	Survivor selection	Ranking
	Replacement rate (RRate)	0.70

5.4. Comparative Protocol and Statistical Pipeline

To systematically evaluate algorithmic performance, stability, and convergence behavior under identical stochastic baselines, all four metaheuristics (Set-based ABC, Binary ABC, Binary GA, and Simulated Annealing) were executed over 20 independent runs across a shared set of 20 random numerical seeds. Because each algorithm was evaluated on the exact same sequence of pseudo-random seeds, the resulting fitness distributions constitute dependent (paired) samples across treatments [27].

Statistical validation followed an inferential decision tree structured to verify distributional assumptions before executing omnibus hypothesis testing and post hoc pairwise procedures:

1. Normality Assessment: The Shapiro–Wilk test was independently applied to each 20-run sample distribution at a significance threshold of ($\alpha = 0.05$) to assess parametric normality across all four algorithms.
2. Variance Homogeneity Test: Levene's test was conducted simultaneously across all four groups ($\alpha = 0.05$) to verify the assumption of homoscedasticity.
3. Omnibus Hypothesis Testing:
    a. If all four sample groups satisfied normality ($p > 0.05$) and the joint variance was homoscedastic ($p > 0.05$), a repeated-measures one-way Analysis of Variance (RM-ANOVA) was scheduled to test the global null hypothesis of equal algorithm means ($H_0 : \mu_{SetABC} = \mu_{GA} = \mu_{BABC} = \mu_{SA}$).
    b. If parametric normality was violated by any algorithm or if heteroscedasticity was present, the non-parametric Friedman test for paired samples was sched-

uled as the primary omnibus rank test (H_0 : $Median_{SetABC} = Median_{GA} = Median_{BABC} = Median_{SA}$).

4. Post hoc Pairwise Comparisons: If the global omnibus test rejected $H_0 (p < 0.05)$, post hoc pairwise comparisons were conducted using the two-sided Wilcoxon signed-rank test for matched pairs across all six algorithm combinations (k = 6). To strictly preserve the family-wise error rate (FWER) and prevent Type I error inflation, the Bonferroni correction was applied, yielding an adjusted significance threshold of (Equation (26)):

$$\alpha' = \frac{\alpha}{k} = \frac{0.05}{6} \approx 0.0083 \quad (26)$$

Differences satisfying ($p < \alpha'$) were deemed statistically significant.

6. Results

This section presents the empirical performance comparison among the proposed Discrete Set-based Artificial Bee Colony (Set-ABC) algorithm, Binary Artificial Bee Colony (BABC), Binary Genetic Algorithm (GA), and Simulated Annealing (SA) when solving the parcel-based tree selection problem for sustainable Amazonian harvesting. To ensure a fair and rigorous comparative evaluation, all four metaheuristics were evaluated under an equivalent computational budget across 20 independent runs executed with a shared set of 20 pseudo-random seeds. The resulting best net economic return values (Equation (7)) obtained per run are reported in Table 5, providing the empirical baseline for the descriptive and inferential statistical pipeline.

Table 5. Best solution fitness values obtained over 20 independent algorithm runs.

Seed	SA	BABC	Set-ABC	GA
42	1,018,659.80	1,041,572.84	982,066.69	1,060,753.26
640	1,026,093.26	1,028,228.12	980,975.58	1,060,753.26
201	1,014,990.02	1,028,228.12	986,147.00	1,058,878.00
123	1,042,909.59	1,035,408.84	983,185.12	1,061,032.19
456	1,005,002.79	1,028,228.12	985,913.70	1,060,753.26
789	992,965.63	1,031,876.71	984,525.38	1,061,032.19
987	1,007,408.92	1,028,228.12	986,428.93	1,061,032.19
654	1,002,808.42	1,021,303.94	981,959.64	1,061,032.19
321	1,008,331.75	1,029,164.99	988,543.93	1,055,438.33
111	1,025,215.46	1,021,083.18	986,487.05	1,061,032.19
28	1,025,856.74	1,028,228.12	982,703.57	1,058,878.00
839	995,280.45	1,036,061.36	982,740.46	1,058,878.00
450	1,006,942.30	1,029,164.99	987,386.03	1,061,032.19
401	1,015,757.64	1,028,228.12	984,090.45	1,058,878.00
908	1,019,937.10	1,028,228.12	982,906.89	1,053,897.36
222	1,025,709.29	1,018,878.82	978,877.75	1,061,032.19
653	996,565.62	1,035,042.34	981,621.19	1,058,878.00
104	989,305.75	1,028,228.12	980,225.82	1,054,154.77
110	1,008,364.52	1,022,664.81	989,016.90	1,058,878.00
777	1,032,351.71	1,029,164.99	981,650.48	1,061,032.19

To determine the appropriate inferential statistical pipeline (parametric Repeated-Measures ANOVA vs. non-parametric Friedman test) as outlined in Section 5.4, descriptive statistics were compiled, and formal distributional assumption tests were executed across

all four treatment groups. Table 6 synthesizes the central tendency metrics, dispersion properties, parametric normality evaluations, variance homogeneity assessments, and the omnibus hypothesis test result.

Table 6. Descriptive statistics and inferential test results across 20 independent runs.

Statistic/Test	SA	BABC	SetABC	GA
Mean Fitness	1,013,022.84	1,028,860.64	983,872.63	1,059,363.79
Median Fitness	1,011,677.27	1,028,228.12	983,046.01	1,060,753.26
Standard Deviation Fitness	14,205.60	5442.21	2823.15	2315.86
Max Fitness	1,042,909.59	1,041,572.84	989,016.90	1,061,032.19
Min Fitness	989,305.75	1,018,878.82	978,877.75	1,053,897.36
IQR Fitness	20,884.72	1614.80	4335.13	2154.19
Mean Run Time	77.32 s	179.51 s	203.35 s	246.22 s
Shapiro–Wilk Normality (p-value)	W = 0.9742 p = 0.8392 (Normal)	W = 0.9115 p = 0.0682 (Normal)	W = 0.9628 p = 0.6013 (Normal)	W = 0.7210, $p = 7.1135 \times 10^{-6}$ (Non-Normal)
Levene's Variance Test (p-value)	Stat = 21.0248, $p = 5.1164 \times 10^{-10}$ (Heteroscedastic)			
RM-ANOVA	-			
Friedman test (χ^2, p-value)	$\chi^2 = 56.16$ | $p = 3.8833 \times 10^{-12}$ | reject H_0			

- Assessment of Normality: The Shapiro–Wilk test indicated that the fitness distributions for SA (p = 0.8392), BABC (p = 0.0682), and the proposed Set-based ABC (p = 0.6013) did not significantly depart from parametric normality at the α = 0.05 threshold. In contrast, GA exhibited a severe violation of the normality assumption (W = 0.7210, $p = 0.71135 \times 10^{-5}$), caused by its steep concentration and acute rightward kurtosis around the upper fitness boundary of 1,061,032.19 PEN.
- Assessment of Variance Homogeneity: Levene's test confirmed pronounced heteroscedasticity across the four metaheuristics (Stat = 21.0248, $p = 5.1164 \times 10^{-10} < 0.05$). Simulated Annealing displayed the highest dispersion (SD = 14,205.60 PEN), reflecting the trajectory sensitivity inherent to single-solution annealing schedules, whereas the population-based methods achieved much tighter distributions, with GA reaching the lowest standard deviation (SD = 2315.86 PEN).
- Inferential Hypothesis Testing: Because the parametric assumptions of normality and homoscedasticity were both violated, the parametric RM-ANOVA was ruled out. Instead, the non-parametric Friedman test for paired samples across the 20 matched random seeds was conducted. The test yielded Chi-square = 56.16 with $p = 3.8833 \times 10^{-12} < 0.05$, decisively rejecting the null hypothesis of equal median algorithmic performance.

Following the rejection of the omnibus null hypothesis by the Friedman test, pairwise post hoc comparisons were conducted using the two-sided Wilcoxon signed-rank test for paired samples across the 20 matched random seeds. To control the family-wise error rate across all six pairwise combinations (k = 6), the Bonferroni adjustment established a corrected significance threshold of 0.0083 (Equation (26)). As summarized in Table 7, all pairwise comparisons yielded p-values strictly below α', confirming statistically significant performance differences among all four metaheuristics. Specifically, GA significantly

outperformed all candidate algorithms ($p < 10^{-4}$), while Binary ABC attained a statistically significant advantage over both Simulated Annealing ($p = 4.8256 \times 10^{-4}$) and the proposed Set-based ABC ($p = 1.9073 \times 10^{-6}$).

Table 7. Post hoc paired Wilcoxon signed-rank test results with Bonferroni correction.

Comparison Pair	Wilcoxon Statistic (W)	*p*-Value	Statistical Decision
SA vs. BABC	18.0	4.8256×10^{-4}	Statistically Significant
SA vs. Set-ABC	0.0	1.9073×10^{-6}	Statistically Significant
SA vs. GA	0.0	1.9073×10^{-6}	Statistically Significant
BABC vs. Set-ABC	0.0	1.9073×10^{-6}	Statistically Significant
BABC vs. GA	0.0	8.7824×10^{-5}	Statistically Significant
Set-ABC vs. GA	0.0	1.9073×10^{-6}	Statistically Significant

To visually examine the structural performance hierarchy, distributional spread, and convergence stability across all four metaheuristics, Figure 1 illustrates the comparative boxplots paired with overlaid jittered individual runs across the 20 independent trials. The visualization confirms a clear, separated progression in median net economic return: the proposed Set-based ABC is centered at 983,046.01 PEN, followed by Simulated Annealing at 1,011,677.27 PEN, Binary ABC at 1,028,228.12 PEN, and GA at 1,060,753.26 PEN. Beyond median performance, the plot highlights distinct dispersion behaviors: Simulated Annealing exhibits the widest interquartile range (IQR = 20,884.72$ PEN) due to trajectory-dependent thermal exploration, whereas Binary GA displays extreme distributional concentration (IQR = 2154.19 PEN) around its top-performing state. Furthermore, while Binary ABC achieves higher median returns than Set-based ABC, the proposed discrete set formulation maintains a remarkably stable and compact variance profile (SD = 2823.15 PEN), demonstrating consistent convergence across independent stochastic evaluations.

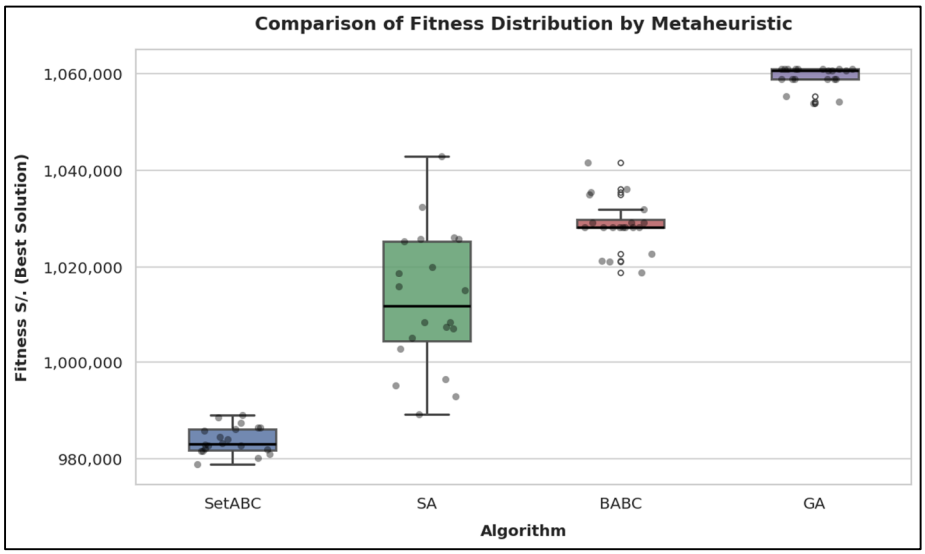


Figure 1. Boxplot comparing fitness distributions between the four metaheuristics across 20 independent runs.

To evaluate search efficiency, local optima avoidance, and convergence trajectories under an equivalent computational budget, the evolution of both local-best and cumulative global-best solutions was monitored across 1000 iterations for each method. The individual convergence profiles exhibit distinct exploratory behaviors as can be seen in Figure 2:

- Genetic Algorithm (GA): Characterized by an aggressive early ascent during the initial 50 generations, maintaining steady genetic refinement until achieving its definitive fitness plateau near generation 290. Beyond this stage, the iteration-best solution closely tracks the cumulative best-so-far trajectory, indicating that the elitist replacement and survivor ranking mechanisms drove the population into a stable, high-yield consensus attractor without premature stagnation.

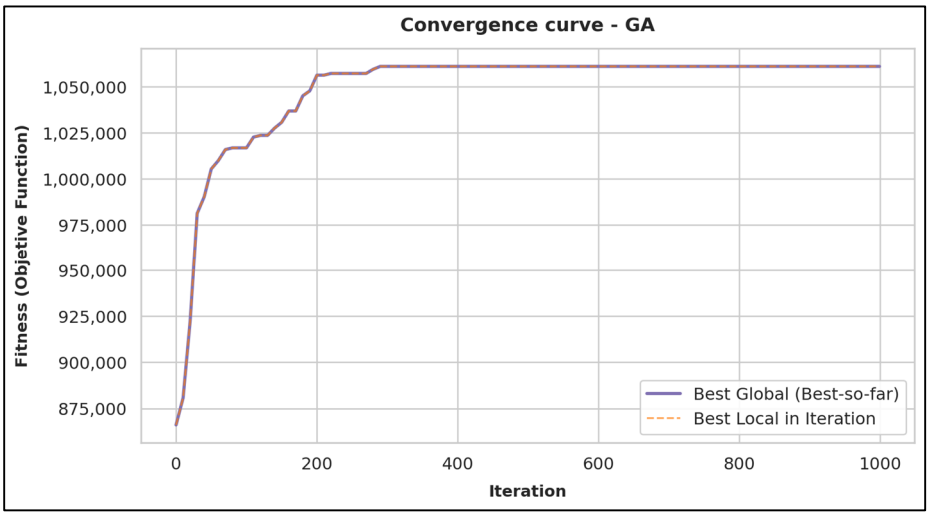


Figure 2. Convergence Curve GA best parameters showing iterations for each multiple of 10.

- Binary Artificial Bee Colony (BABC): Demonstrates rapid initial exploitation across the first 40 iterations, followed by stepped improvements resulting from scout-bee perturbations. The local iteration-best trajectory displays controlled oscillations below the cumulative curve, confirming that employed and onlooker bees maintained active neighborhood sampling while retaining the global best-so-far solution.
- Set-based Artificial Bee Colony (Set-ABC): Converges rapidly toward its primary plateau around iteration 100, with a final localized adjustment near iteration 470. Throughout the search, moderate local fitness fluctuations reflect continuous exploratory shuffling via set union $\oplus$ and difference $\ominus$ operators; however, these discrete perturbations stabilize within a lower-fitness basin compared to the binary representations.
- Simulated Annealing (SA): Exhibits severe variance and abrupt oscillations in the local iteration-best curve during the first 600 outer cooling cycles. This behavior is a direct consequence of high-temperature Metropolis–Hastings acceptance transitions that facilitate escape from local optima by accepting deteriorating moves. As temperature decays below critical thresholds, the trajectory gradually stabilizes, consolidating its final harvest block near iteration 720.

To directly compare learning velocity, escapability from deceptive local optima, and the final solution quality across methods, Figures 3–6 synthesizes the cumulative best-so-far fitness trajectories over the 1000 iterations for all four evaluated metaheuristics respectively:

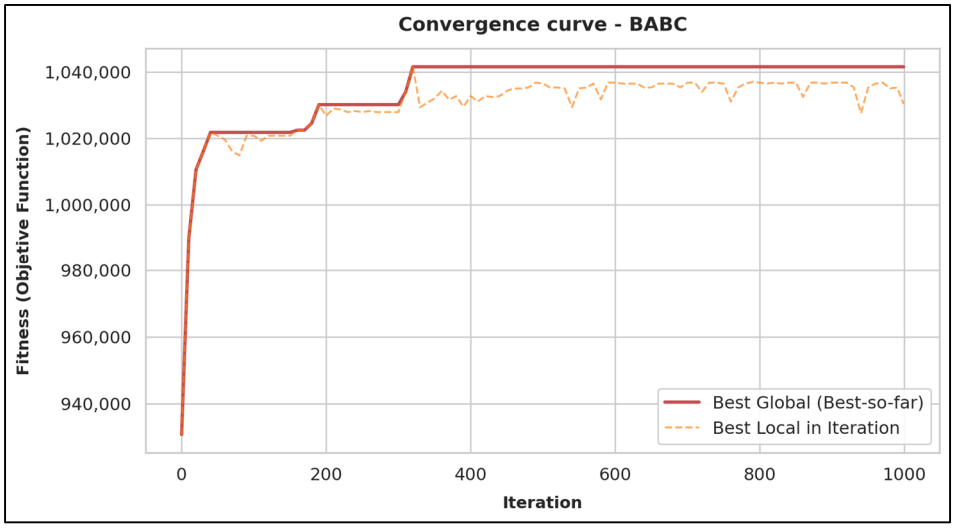


Figure 3. Convergence Curve BABC best parameters showing iterations for each multiple of 10.

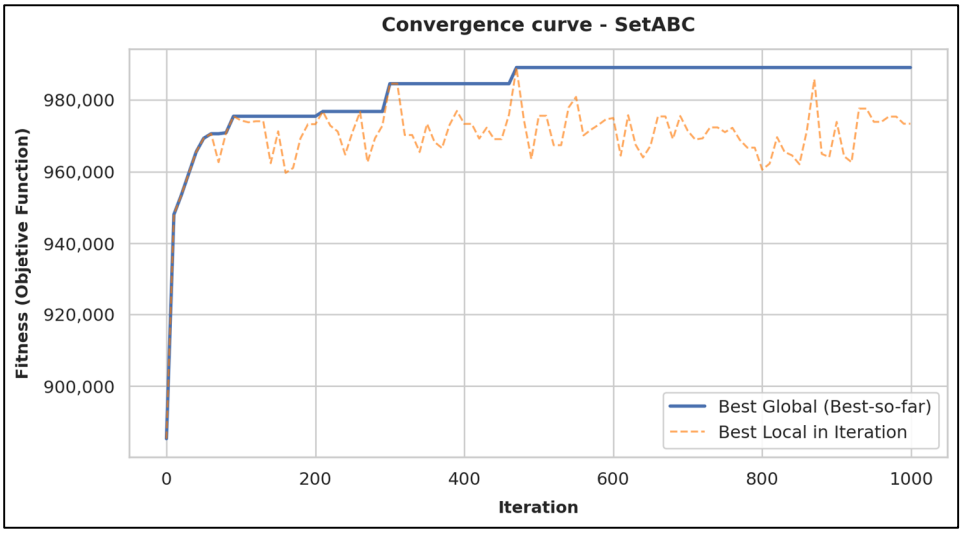


Figure 4. Convergence Curve SetABC best parameters showing iterations for each multiple of 10.

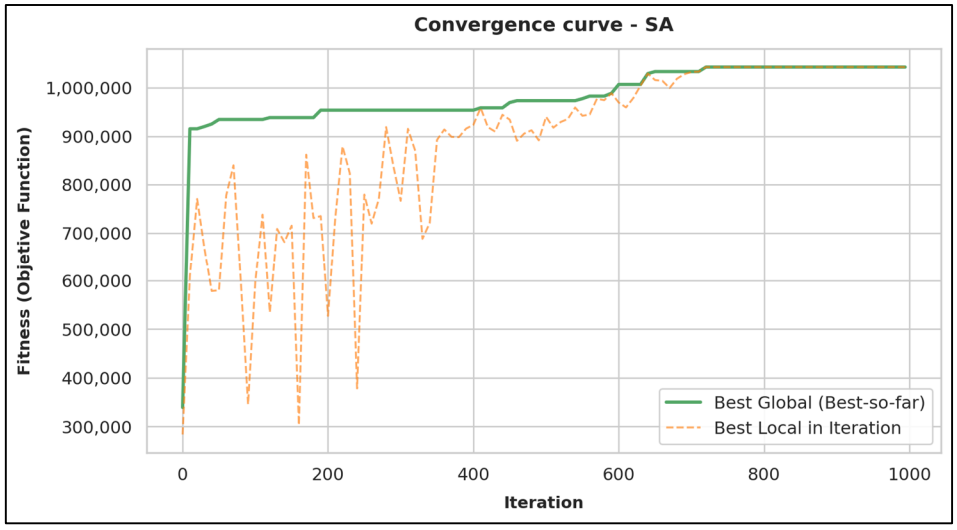


Figure 5. Convergence Curve SA best parameters showing iterations for each multiple of 10.

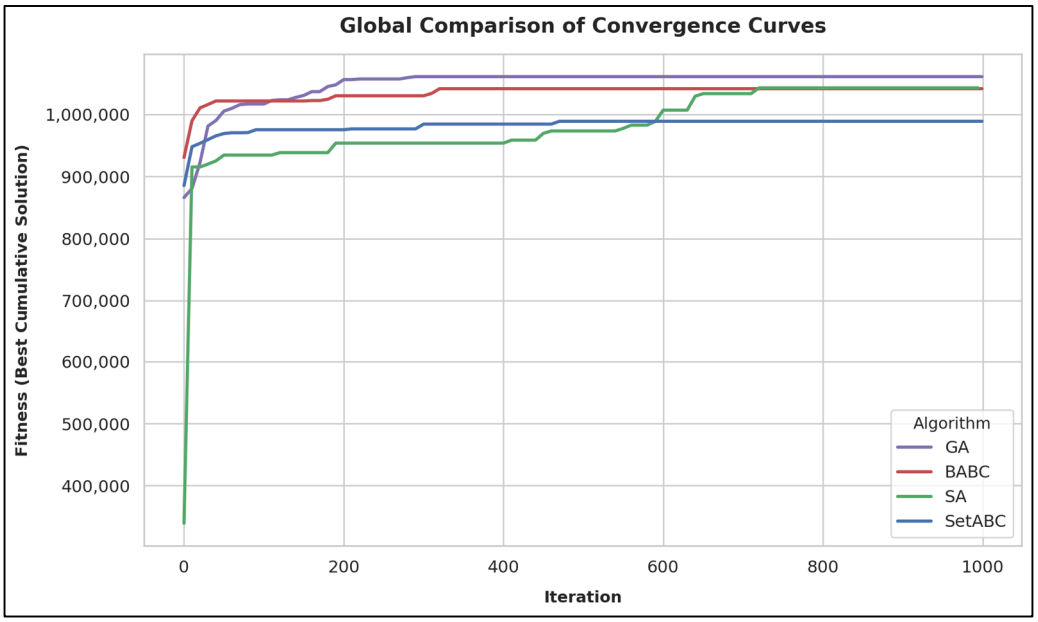


Figure 6. General comparisons.

To address operational feasibility, constraint compliance, and practical forest management implications, the definitive harvest blocks identified by both the median and peak runs of each metaheuristic were analyzed across their spatial, economic, and biological dimensions. Table 8 details the parcel selection properties, harvested land and canopy distributions, economic returns, budget utilization, and legal retention ratios for the median solutions across the 90-parcel concession. Correspondingly, Table 9 provides the complete operational and ecological breakdown for the best overall solution discovered by each metaheuristic, establishing a direct baseline to evaluate management trade-offs under representative and peak algorithmic performance.

Table 8. Operational, ecological, and economic breakdown of the median solution found by each metaheuristic.

Operational/Ecological Metric	Simulated Annealing (SA)	Genetic Algorithm (GA)	Set-Based ABC (Set-ABC)	Binary ABC (BABC)	Threshold/Legal Boundary
Representative Random Seed	201	42	123	201	—
Selected Parcels (|X|/90)	60 (66.67%)	61 (67.78%)	62 (68.89%)	61 (67.78%)	Max 90 parcels
Harvested Land Area (ha/%)	65.06 ha (65.88%)	66.05 ha (66.88%)	67.82 ha (68.67%)	66.40 ha (67.24%)	Total: 98.75 ha
Harvested Canopy Area (ha/%)	47.17 ha (65.05%)	48.34 ha (66.65%)	49.63 ha (68.44%)	48.83 ha (67.34%)	Initial: 72.52 ha
Remaining Total Canopy Ratio	34.95% (Feasible)	33.35% (Feasible)	31.56% (Feasible)	32.66% (Feasible)	Mandatory RCC >= 25.00%
Harvested Primary Canopy Area (ha/%)	39.24 ha (65.65%)	40.95 ha (68.51%)	41.70 ha (69.76%)	39.81 ha (66.61%)	Initial: 59.77 ha
Remaining Primary Forest Ratio	34.35% (Feasible)	31.49% (Feasible)	30.24% (Feasible)	33.39% (Feasible)	Mandatory RFC >= 30.00%
Gross Timber Revenue (PEN)	2,047,656.29	2,055,509.93	2,023,832.06	2,033,654.86	Max: 3,099,712.93 PEN

Table 8. *Cont.*

Operational/Ecological Metric	Simulated Annealing (SA)	Genetic Algorithm (GA)	Set-Based ABC (Set-ABC)	Binary ABC (BABC)	Threshold/Legal Boundary
Total Extraction Cost (PEN)	896,764.50	899,382.80	897,095.93	894,985.49	Max: 900,000.00 PEN
Budget Utilization (%)	99.64%	99.93%	99.68%	99.44%	Compliant (Phard = 0)
Economic Excess Penalties (PEN)	135,901.26	95,373.87	143,551.01	110,441.25	Soft ecological penalty
Net Economic Return (F(X) in PEN)	1,014,990.02	1,060,753.26	983,185.12	1,028,228.12	Objective function
Restricted Species Limit Violations	Palo Sangre (+42), Aguano Pashaco (+59)	Palo Sangre (+31), Aguano Pashaco (+49)	Palo Sangre (+52), Aguano Pashaco (+45)	Palo Sangre (+37), Aguano Pashaco (+49)	17 preserved, 2 penalized
Hard Ecological Constraints Status	Feasible (Phard = 0)	Feasible (Phard = 0)	Feasible (Phard = 0)	Feasible (Phard = 0)	100% legal compliance
Convergence Iteration/Budget	744/1000	345/1000	958/1000	89/1000	1000 iterations

Table 9. Operational, ecological, and economic breakdown of the best solution found by each metaheuristic.

Operational/Ecological Metric	Simulated Annealing (SA)	Genetic Algorithm (GA)	Set-Based ABC (Set-ABC)	Binary ABC (BABC)	Threshold/Legal Boundary
Representative Random Seed	123	789	110	42	—
Selected Parcels (|X|/90)	61 (67.78%)	61 (67.78%)	61 (67.78%)	61 (67.78%)	Max 90 parcels
Harvested Land Area (ha/%)	67.27 ha (68.12%)	65.95 ha (66.78%)	66.76 ha (67.60%)	66.01 ha (66.84%)	Total: 98.75 ha
Harvested Canopy Area (ha/%)	49.49 ha (68.24%)	48.29 ha (66.59%)	48.67 ha (67.11%)	48.44 ha (66.79%)	Initial: 72.52 ha
Remaining Total Canopy Ratio	31.76% (Feasible)	33.41% (Feasible)	32.89% (Feasible)	33.21% (Feasible)	Mandatory RCC >= 25.00%
Harvested Primary Canopy Area (ha/%)	40.47 ha (67.70%)	40.05 ha (67.01%)	41.39 ha (69.26%)	40.20 ha (67.26%)	Initial: 59.77 ha
Remaining Primary Forest Ratio	32.30% (Feasible)	32.99% (Feasible)	30.74% (Feasible)	32.74% (Feasible)	Mandatory RFC >= 30.00%
Gross Timber Revenue (PEN)	2,067,155.66	2,065,103.95	1,998,573.64	2,064,844.60	Max: 3,099,712.93 PEN
Total Extraction Cost (PEN)	899,522.85	899,929.91	897,075.09	898,488.01	Max: 900,000.00 PEN
Budget Utilization (%)	99.95%	99.99%	99.68%	99.83%	Compliant (Phard = 0)
Economic Excess Penalties (PEN)	124,723.22	104,141.84	112,481.64	124,783.75	Soft ecological penalty
Net Economic Return (F(X) in PEN)	1,042,909.59	1,061,032.19	989,016.90	1,041,572.84	Objective function
Restricted Species Limit Violations	Palo Sangre (+41) Aguano Pashaco (+51)	Palo Sangre (+33) Aguano Pashaco (+49)	Palo Sangre (+32) Aguano Pashaco (+59)	Palo Sangre (+38) Aguano Pashaco (+57)	17 preserved, 2 penalized
Hard Ecological Constraints Status	Feasible (Phard = 0)	Feasible (Phard = 0)	Feasible (Phard = 0)	Feasible (Phard = 0)	100% legal compliance
Convergence Iteration/Budget	719/1000	290/1000	461/1000	312/1000	1000 iterations

7. Discussion

The four-algorithm comparison yields a clear and statistically robust performance hierarchy. The Friedman test rejected the null hypothesis of equal median performance ($\chi^2 = 56.16$, $p = 3.88 \times 10^{-12}$), and all six Bonferroni-corrected Wilcoxon signed-rank comparisons were significant at the adjusted threshold $\alpha' = 0.0083$. Ranked by mean net economic return, the ordering is GA (1,059,364 PEN) > BABC (1,028,861 PEN) > SA (1,013,023 PEN) > Set-ABC (983,873 PEN): GA significantly outperformed every competitor, BABC was significantly superior to both SA and the proposed Set-ABC, and SA in turn significantly exceeded Set-ABC. Two findings stand out immediately. First, a well-tuned Binary GA is the method to beat on this regulated parcel-selection instance. Second, and less comfortably for the encoding hypothesis of this work, within the ABC family the binary variant outperformed the proposed set-theoretic adaptation under a matched computational budget.

Central tendency, however, tells only part of the story; dispersion adds a complementary reading. GA was simultaneously the most accurate and the most stable method (SD = 2315.86 PEN), while the proposed Set-ABC, despite the lowest mean, achieved the second-lowest standard deviation (SD = 2823.15 PEN), converging tightly and reproducibly to a consistent basin. Simulated Annealing sat at the opposite extreme (SD = 14,205.60 PEN), the expected signature of single-solution trajectory sensitivity—yet its best individual run (1,042,909.59 PEN) exceeded BABC's best (1,041,572.84 PEN), showing that high-temperature exploration occasionally reaches excellent solutions it cannot reliably reproduce. The practical implication is that the results describe a quality–stability–cost frontier rather than a single dominant choice: GA for reliable peak quality, Set-ABC for stable and novel discrete search, and SA for cheap but volatile exploration.

Interpreted in operational terms, the gaps are narrower than the ranking suggests. All four methods converged to feasible, regulation-compliant harvest blocks of comparable spatial footprint—61 of the 90 parcels (~67.8%), roughly 66–67 ha harvested, with total-canopy and primary-forest retention comfortably above the statutory thresholds (R_CC = 25%, R_FC = 30%). The ~7% mean gap separating GA from Set-ABC therefore does not correspond to a categorically different management plan, but to finer optimization of the harvested species and volume mix. Concretely, GA paired high gross revenue with the lowest ecological penalties of the panel (104,142 PEN), whereas Set-ABC's deficit is concentrated on the gross-revenue side (1,998,574 PEN versus ~2,065,000 PEN for the other three methods), indicating that it settles on a less valuable—but still fully legal—species composition. Every method delivers an auditable, compliant harvest schedule; they differ mainly in how much legal value they extract from a spatially similar block.

Convergence behavior under the shared 1000-iteration budget was equally distinctive. GA ascended aggressively and reached its plateau by roughly generation 290; BABC exploited early and refined through scout-driven steps (convergence near iteration 312); Set-ABC reached its primary plateau by about iteration 100 with a late adjustment near iteration 470 but stabilized within a lower-fitness basin; and SA oscillated through approximately 600 cooling cycles before consolidating near iteration 720. Computationally, the ordering inverts the quality ranking: SA was the cheapest method (43–77 s), GA the most expensive (142–246 s), with the two ABC variants in between. GA thus purchases its accuracy with the highest objective-function-call count and the longest runtime—a trade-off that matters where evaluation is costly or where near-real-time replanning is required. This asymmetry in maturity is central to interpreting the notion of "superiority." GA relies on a well-understood operator set—roulette-wheel selection, multi-point crossover, and bit-flip mutation—supported by an extensive tuning literature, and a single-round Grid Search over four seeds converged it to a stable, aggressive configuration. Set-ABC, by contrast, is

a novel adaptation whose dynamics depend on parameters—the abandonment limit, the parcel-selection boundaries, and the discrete perturbation magnitude drawn from φ—for which no consolidated tuning heuristics exist in this domain, and it was calibrated over the same grid without encoding-specific priors. Its consistent convergence into a lower but stable basin is compatible with sub-calibration rather than an intrinsic performance ceiling: the set-algebraic operators reliably locate a compact family of feasible solutions, but the current boundary and abandonment settings appear to under-explore the higher-value region that BABC and GA reach. That the binary ABC outperformed the set-theoretic one under identical budgets points specifically to the encoding–operator interaction as the place where dedicated tuning is most likely to yield gains.

Two limitations bound the generality of these conclusions. First, the evaluation rests on a single semi-synthetic instance of approximately 90 parcels derived from one operational distribution (the "EL PAISA" ecological plantation, Nauta, Loreto); the relative ordering could shift on larger or structurally different concessions, where the broader exploratory profile of swarm intelligence might become advantageous against GA's tendency to concentrate around a single attractor. Second, all runs were executed on modest commodity hardware (an Intel Core i5 mobile CPU with 16 GB RAM), and iteration caps were set partly by runtime tractability rather than by convergence alone; a larger computational budget could reshape the trajectories of both methods, and of SA and Set-ABC in particular, given their later and more erratic discovery iterations. These observations are consistent with the broader forest-optimization literature, in which a well-tuned GA remains a strong baseline for spatial harvest problems [12–14], while swarm-based methods are typically reported as competitive rather than dominant until problem-specific adaptations mature [15–17]. Read in this light, the present study contributes not a claim that set-theoretic ABC surpasses established methods, but a statistically rigorous four-way benchmark establishing that GA is the reference to beat, that the proposed Set-ABC is a stable, feasible, and genuinely novel contender whose principal current weakness is gross-value extraction under an unspecialized calibration, and that the binary-versus-set encoding contrast is itself an informative result motivating encoding-aware tuning as the most promising next step.

8. Conclusions

This study addressed sustainable tree selection in the Peruvian Amazon, formulated as a regulated NP-hard combinatorial optimization over forest parcels, and delivered four contributions. First, a formal model of the problem was established, separating hard constraints—spatial connectivity, enclave tolerance, exclusion zones, budget, canopy cover, and primary-forest conservation—from a penalized species-retention constraint, all anchored in Law No. 29763, the national CITES registry, and Supreme Decree No. 018-2015-MINAGRI. Second, an objective function expressed directly in net economic return (PEN) was proposed, combining gross timber revenue and extraction cost while discounting statutory infractions through a species-level penalty, so that only legally and operationally admissible harvest blocks attain positive fitness. Third, a novel set-theoretic discrete adaptation of the Artificial Bee Colony algorithm (Set-ABC) was developed, representing food sources as parcel-identifier sets and driving neighborhood search through set-algebraic operators rather than continuous or binary encodings. Fourth, Set-ABC was benchmarked against a Binary ABC, a Binary Genetic Algorithm, and Simulated Annealing under a common computational budget and a paired statistical pipeline (Friedman test with Bonferroni-corrected Wilcoxon post hoc comparisons).

The four-way comparison produced a statistically robust performance hierarchy. All pairwise differences were significant at the adjusted threshold, with mean net return ordered GA (1,059,364 PEN) > BABC (1,028,861 PEN) > SA (1,013,023 PEN) > Set-ABC

(983,873 PEN). A well-tuned Binary GA is thus the reference method to beat on this regulated instance, achieving both the highest accuracy and the lowest dispersion. Within the ABC family, the binary variant outperformed the proposed set-theoretic adaptation under matched budgets—an informative encoding-versus-set contrast rather than a dismissal of the discrete formulation. Indeed, the operational analysis showed that all four methods converged to feasible, regulation-compliant harvest blocks of comparable spatial footprint (61 of 90 parcels, retention above statutory thresholds); Set-ABC's deficit was concentrated in gross-value extraction, not in feasibility or stability, and it achieved the second-lowest variance of the panel, converging reproducibly to a compact solution basin. Its consistent settling into a lower but stable region is compatible with sub-calibration under an unspecialized parameter grid rather than an intrinsic performance ceiling.

Several avenues follow directly from these findings. The most immediate is encoding-aware calibration of the ABC family: because Set-ABC was tuned over the same grid as the binary methods without operator-specific priors, dedicated tuning of its abandonment limit, parcel-selection boundaries, and discrete perturbation magnitude is the change most likely to close the observed gap. A natural extension is a Multi-Objective ABC (MOABC) that returns a Pareto front over net return, harvested area, and conservation compliance instead of collapsing them into a single scalar, yielding decision-support output better matched to real management trade-offs. Replacing random initialization with a GRASP-style constructive procedure could seed every method with high-quality feasible blocks and further mitigate early-search infeasibility, while hybrid memetic schemes coupling population-based exploration with local search would broaden the competitive landscape. Finally, extending the evaluation beyond the current single 90-parcel semi-synthetic instance to larger and structurally diverse concessions—ideally validated against field inventory data—would test the scalability of all four methods and clarify whether the broader exploratory profile of swarm intelligence becomes advantageous at scale.

Author Contributions: Conceptualization, S.G., M.T. and R.C.; methodology, S.G.; software, S.G.; validation S.G.; formal analysis, S.G.; investigation, S.G., M.T. and R.C.; resources, S.G.; writing—original draft preparation, S.G.; writing—review and editing, S.G., M.T. and R.C.; supervision, M.T. and R.C.; project administration, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study, including parcel records, adjacency matrices, generation scripts, and algorithm implementations, are openly available in a GitHub repository at https://github.com/SebastianGrajeda28/Optimizing-tree-selection-using-bio-inspired-algorithms (accessed on 28 September 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABC	Artificial Bee Colony
BFS	Breadth-First Search
GA	Genetic Algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
TS	Tabu Search
SA	Simulated Annealing

References

1. FAO. *Evaluación de los Recursos Forestales Mundiales 2020: Informe Principal*; FAO: Rome, Italy, 2020.
2. FAO; INEI; MINAM. *Cuenta de Bosques del Perú*; Instituto Nacional de Estadística e Informática: Lima, Peru, 2021.
3. Congreso de la República del Perú. *Ley Forestal y de Fauna Silvestre, Ley N° 29763*; Diario Oficial El Peruano: Lima, Peru, 2011.
4. Ministerio del Ambiente. *Listado de Especies de Flora Silvestre CITES–Perú*; Dirección General de Diversidad Biológica, Viceministerio de Desarrollo Estratégico de los Recursos Naturales: Lima, Peru, 2018.
5. SERFOR. *Anuario Forestal y de Fauna Silvestre 2022*; Servicio Nacional Forestal y de Fauna Silvestre: Lima, Peru, 2023.
6. Aguiar, M.O.; da Silva, G.F.; Mauri, G.R.; de Mendonça, A.R.; da Silva, E.F.; Figueiredo, E.O.; Pereira Martins Silva, J.; Alves da Silva, V.; Freitas Silva, R.; Lessa Lavagnoli, G. Integrated Planning of Forest Exploration Infrastructures in an Amazonian Sustainable Forest Management Area. *For. Ecol. Manag.* **2023**, *549*, 121265. [CrossRef]
7. Naderializadeh, N.; Crowe, K.A.; Rouhafza, M. Solving the Integrated Forest-Harvest Scheduling Model Using Metaheuristic Algorithms. *Oper. Res.* **2022**, *22*, 2437–2463. [CrossRef]
8. Lee, Y.; Sim, W.; Lee, S.; Lee, J. Development and Evaluation of a Thinning Tree Selection System Using Optimization Techniques Based on Multi-Platform LiDAR. *Forests* **2025**, *16*, 1776. [CrossRef]
9. Ciarleglio, M.; Barnes, J.W.; Sarkar, S. ConsNet—A Tabu Search Approach to the Spatially Coherent Conservation Area Network Design Problem. *J. Heuristics* **2010**, *16*, 537–557. [CrossRef]
10. Mesquita, M.; Marques, S.; Marques, M.; Marto, M.; Constantino, M.; Borges, J.G. An Optimization Approach to Design Forest Road Networks and Plan Timber Transportation. *Oper. Res.* **2022**, *22*, 2973–3001. [CrossRef]
11. Dong, L.; Bettinger, P.; Liu, Z.; Qin, H. A Comparison of a Neighborhood Search Technique for Forest Spatial Harvest Scheduling Problems: A Case Study of the Simulated Annealing Algorithm. *For. Ecol. Manag.* **2015**, *356*, 124–135. [CrossRef]
12. Nanos, N.; García-del-Rey, E.; Gil, L. Optimal Selection of Seed-Trees Using the Multi-Objective NSGA-II Algorithm and a Seed Dispersal Model. *Forests* **2024**, *15*, 499. [CrossRef]
13. Castro, I.; Salas-González, R.; Fidalgo, B.; Farinha, J.T.; Mendes, M. Optimising Forest Management Using Multi-Objective Genetic Algorithms. *Sustainability* **2024**, *16*, 10655. [CrossRef]
14. Gomide, L.R.; Arce, J.E.; Silva, A.C.L. Uso do Algoritmo Genético no Planejamento Florestal Considerando Seus Operadores de Seleção. *Cerne* **2009**, *15*, 460–467.
15. Haber, Z.; Uguz, H.; Hakli, H. An Improved Artificial Bee Colony Algorithm with a Probabilistic Crossover and Lock Mechanism. *Biomimetics* **2026**, *11*, 187. [CrossRef] [PubMed]
16. Lin, Y.; Wang, J.; Li, X.; Zhang, Y.; Huang, S. An Improved Artificial Bee Colony for Feature Selection in QSAR. *Algorithms* **2021**, *14*, 120. [CrossRef]
17. He, Y.; Xie, H.; Wong, T.L.; Wang, X. A Novel Binary Artificial Bee Colony Algorithm for the Set-Union Knapsack Problem. *Future Gener. Comput. Syst.* **2018**, *83*, 311–325.
18. Karaboga, D.; Basturk, B. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
19. Hussien, A.G.; Pop, A.; Kumar, S.; Hashim, F.A.; Hu, G. A Novel Artificial Electric Field Algorithm for Solving Global Optimization and Real-World Engineering Problems. *Biomimetics* **2024**, *9*, 186. [CrossRef] [PubMed]
20. Hançer, E. An ABC Algorithm Inspired by Boolean Operators for Knapsack and Lot Sizing Problems. *Acad. Platf. J. Eng. Sci.* **2018**, *6*, 142–152. [CrossRef]
21. Beheshti, Z. BMNABC: Binary Multi-Neighborhood Artificial Bee Colony for High-Dimensional Discrete Optimization Problems. *Cybern. Syst.* **2018**, *49*, 452–474. [CrossRef]
22. Binoti, D.H.B.; Binoti, M.L.M.d.S.; Leite, H.G.; Gleriani, J.M.; de Oliveira, M.L.R.; Campos, J.C.C.; Monte, M.A. Regulation of Even-Aged Forests with Inclusion of Environmental Constraints. *Cerne* **2014**, *20*, 139–147. [CrossRef]
23. Aguiar, M.O.; da Silva, G.F.; Mauri, G.R.; da Silva, E.F.; de Mendonça, A.R.; Martins Silva, J.P.; Freitas Silva, R.; Silva Santos, J.; Lavagnoli, G.L.; Figueiredo, E.O. Metaheuristics Applied for Storage Yards Allocation in an Amazonian Sustainable Forest Management Area. *J. Environ. Manag.* **2020**, *274*, 110926. [CrossRef] [PubMed]
24. MINAGRI. *Decreto Supremo N.° 018-2015-MINAGRI: Reglamento para la Gestión Forestal*; Diario Oficial El Peruano: Lima, Peru, 2015.
25. OSINFOR. *Resolución de Presidencia N.° 021-2018-OSINFOR: Metodología del Cálculo del Monto de las Multas a Imponer por Infracción a la Legislación Forestal y de Fauna Silvestre*; Organismo de Supervisión de los Recursos Forestales y de Fauna Silvestre: Lima, Peru, 2018.

26. Momma, M.; Bennett, K.P. A Pattern Search Method for Model Selection of Support Vector Regression. In Proceedings of the Second SIAM International Conference on Data Mining, Arlington, VA, USA, 11–13 April 2002; pp. 261–274.
27. Gutiérrez Pulido, H.; de la Vara Salazar, R. *Análisis y Diseño de Experimentos*, 3rd ed.; McGraw-Hill/Interamericana: Mexico City, Mexico, 2012.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.