Optimization Project I: Optimization for Biodiversity Conservation in Menorca

Rubén Asensi Valls, Pablo García Pérez, Manuel Martín Morante and Borja Rodrigo Garrido
School of Informatics (ETSINF)
Universitat Politècnica de València
Valencia, Spain

***Abstract*—In this project we address the design of a cost-effective conservation plan for the Biosphere Reserve of Menorca. The goal is to decide which cells of a spatial grid should be converted into suitable habitat and which ecological corridors should be built so that four target species (Atelerix, Martes, Eliomys and Oryctolagus) form viable and well-connected metapopulations under a limited budget. We started from an initial network-flow-based formulation, designed to ensure basic ecological continuity, but its direct implementation on the full island instance resulted in severe scalability and memory issues due to the heavy combination of flow variables and constraints. To overcome these limitations, we developed a three-stage modeling pipeline. First, we build a georeferenced graph and compute shortest paths between existing populations using a multi-source Dijkstra algorithm with great-circle distances and friction-based costs, pruning very expensive routes. Second, we use these precomputed paths as connectivity requirements and define a compact CP-SAT model that decides habitat activation, adaptation investments and corridor construction, while enforcing inter-species conflict constraints and minimum/maximum area shares per species. Finally, we visualize and audit the solution in geographic space and compute a detailed cost decomposition based on real distances. On the reference instance with a budget of $1000 k€, the optimal solution was proven in 130.97 seconds and invests $595.26 k€ (59.52%) in habitat adaptation and $404.89 k€ (40.48%) in 304 corridor segments totalling 241.81 km of ecological infrastructure.**

***Index Terms*—Biodiversity conservation, Mixed-integer programming, CP-SAT, Network optimization, Habitat connectivity, Dijkstra algorithm, Menorca.**

I. Iterative Design and Model Evolution

In this section, we describe the evolution of our solution through three modelling phases. We start from an early-stage network-flow formulation, designed to ensure basic ecological continuity, and progressively move towards a hybrid graph-based model that scales to the full Menorca instance.

A. Model 1: Initial Approach (Network-Flow Formulation)

Our first model was the initial implementable version. It implemented a simplified network-flow-based habitat expansion model (as described in the accompanying document $V.1$), primarily using a weighted sum objective to maximize total benefit.

- **Concept:** We coded the base mixed-integer model using a weighted sum objective (maximizing accumulated suitability and restoration gain), subject to a global budget, ecological conflict constraints, and minimal flow conservation laws ($f_{u,v,s} \in \mathbb{Z}_{\geq 0}$) used to prevent isolated restorations.
- **Critical Analysis:** On small subsets of the grid the model worked correctly, but when scaling to the full island the number of binary variables, integer flow variables ($f_{u,v,s}$), and constraints exploded. The solver suffered from memory exhaustion and very long solving times, failing to find an optimal solution within the computational budget. This issue was mainly due to the overhead of the flow structure required to enforce basic connectivity.
- **Transition:** These limitations motivated a second model focused on eliminating the computationally heavy flow variables and simplifying the objective, while still operating on the original grid.

B. Model 2: Intermediate Version (Path Implication)

The second model aimed to keep the overall structure but improve tractability by fundamentally changing how connectivity was enforced.

- **Improvements:** We leveraged the strong logical capabilities of the CP-SAT solver to replace the computationally heavy flow conservation laws ($f_{u,v,s}$) with a simpler Path Selection Strategy. Connectivity was henceforth enforced using Boolean implications ($x_{i,s} \leq z_e$). This change drastically reduced the number of variables, making the model solvable within time limits. We also kept the objective function as a weighted sum of species scores.
- **Limitations:** Although the model was computationally robust, it suffered from a lack of spatial realism. Connectivity was enforced using abstract grid distances rather than real geography. This deficiency produced unrealistic corridor geometries and biased the selection towards shorter, but potentially more expensive, topological paths.
- **Transition:** These persistent shortcomings led to the definitive model (Model 3), which required a major structural modification: the integration of a geodetic cost surface into the preprocessing phase to align corridor selection with the underlying real-world terrain.

C. Justification of the Definitive Model

The definitive model (Model 3) resolves the persistent shortcomings observed in Model 2 (lack of spatial realism due to abstract distances) by replacing the purely monolithic formulation with a robust hybrid strategy.

First, we perform a graph-based preprocessing using a multi-source Dijkstra algorithm on a georeferenced network with real great-circle distances (Haversine) and friction-based costs. This step identifies, for each candidate habitat cell and species, a small set of cost-efficient corridors needed to connect it to existing populations, and prunes all other edges.

Second, we build a compact CP-SAT model only on the surviving edges, using the precomputed paths as connectivity requirements. We also introduce explicit minimum and maximum area shares per species and a stress variable to capture predator–prey pressure.

This design significantly reduces the search space (as evidenced by the success in solving the previously intractable full-island instance), incorporates geographical realism into the corridor selection, and yields solutions that are ecologically meaningful and easier to interpret.

D. Preprocessing Algorithm

Before solving the optimization model, we perform a preprocessing phase that builds a georeferenced graph and computes shortest paths from existing populations. The algorithm proceeds as follows:

1) **Graph Construction:** We create an undirected graph $G = (I, E)$ where nodes correspond to grid cells. For each pair of neighboring cells (i, j), we compute the edge cost as:

$$C_{ij}^{\text{corr}} = \frac{\text{friction}_i + \text{friction}_j}{2} \times d_{ij} \times \texttt{SCALE\_COST},$$

where d_{ij} is the great-circle distance (computed using the Haversine formula) between the centroids of cells i and j, and $\texttt{SCALE\_COST} = 1000$ converts costs to integer units.

2) **Pruning Strategy:** To reduce the search space, we eliminate routes whose cumulative cost exceeds a threshold. The cutoff value is computed as:

$$\text{cutoff} = \max(0.15 \times B \times \texttt{SCALE\_COST}, 6 \times \text{median}(C_{ij}^{\text{corr}})),$$

where B is the available budget in k€. This ensures that we only consider routes costing at most 15% of the budget, while maintaining a safety floor based on the median edge cost.

3) **Path Computation:** For each species s, we identify the set of source cells P_s where the species is initially present. Using NetworkX's multi-source Dijkstra algorithm with the computed edge weights and cutoff, we find the shortest path from any source in P_s to each reachable cell i. The resulting path is stored as $R_{i,s}$, a list of edges (ordered pairs) that must be activated if cell i becomes habitat for species s.

4) **Unreachable Cells:** If no path exists within the cutoff for a given cell-species pair, we set $R_{i,s} = \emptyset$ and force $x_{i,s} = 0$ during preprocessing, effectively removing these variables from the optimization model.

This preprocessing step dramatically reduces the number of corridor variables from potentially thousands (all neighbor pairs) to only those edges that lie on viable connection paths, making the subsequent CP-SAT model tractable.

II. Definitive Mathematical Model

This section presents the formal definition of our final model (Model 3), which combines a shortest-path preprocessing phase with a CP-SAT optimization model on a pruned conservation graph.

A. Sets and Indices

- I: set of grid cells, indexed by i.
- S: set of species, S = {Atelerix, Martes, Eliomys, Oryctolagus}, indexed by s.
- $E \subseteq I \times I$: set of undirected edges that survive the pruning phase, indexed by (i, j) with $i < j$.
- $P_s \subseteq I$: subset of cells where species s is initially present.
- $R_{i,s} \subseteq E$: set of edges lying on the precomputed shortest path that connects cell i to the initial population of species s (possibly empty or undefined if i is not reachable within the pruning budget).

B. Decision Variables

The definitive model uses the following binary decision variables:

- $x_{i,s} \in \{0, 1\}$: equals 1 if cell i is declared active habitat for species s, and 0 otherwise.
- $y_{i,s} \in \{0, 1\}$: equals 1 if we invest in adaptation actions in cell i for species s, and 0 otherwise.
- $z_{ij} \in \{0, 1\}$: equals 1 if a corridor is built along edge $(i, j) \in E$, and 0 otherwise.
- $\text{stress}_i \in \{0, 1\}$: equals 1 if cell i is in an ecologically stressed configuration (overlap of predator and prey), and 0 otherwise.

C. Constants and Parameters

The model uses the following parameters derived from the dataset and the preprocessing:

- A_i: area of cell i (in km^2, scaled by $\texttt{SCALE\_AREA} = 100$ for integer arithmetic).
- $Q_{i,s}$: habitat suitability of cell i for species s, ranging from 0.0 (Very Low) to 3.0 (High), computed from land cover types using species-specific rules.
- $C_{i,s}^{\text{adapt}}$: cost of adaptation actions in cell i for species s (scaled by $\texttt{SCALE\_COST} = 1000$).
- C_{ij}^{corr}: cost of building a corridor along edge (i, j), computed from the friction values of the adjacent cells and their great-circle distance (scaled by $\texttt{SCALE\_COST}$).
- W_s: ecological weight assigned to species s in the objective function. In our implementation, $W_{\text{Atelerix}} = 1.0$, $W_{\text{Martes}} = 1.0$, $W_{\text{Eliomys}} = 2.0$, and $W_{\text{Oryctolagus}} = 1.5$.
- $\alpha_s^{\min}, \alpha_s^{\max}$: minimum and maximum allowed percentage of the total active area that can be allocated to species s. We use $\alpha_s^{\min} \in \{5\%, 20\%, 5\%, 15\%\}$ and $\alpha_s^{\max} \in \{30\%, 60\%, 30\%, 50\%\}$ for Atelerix, Martes, Eliomys and Oryctolagus, respectively.

TABLE I: Key parameter values used in the implementation

Parameter	Value
`SCALE_COST`	1000
`SCALE_SCORE`	10
`SCALE_AREA`	100
`PENALTY_STRESS`	3500
`RESTORED_Q`	3.0
W_{Atelerix}	1.0
W_{Martes}	1.0
W_{Eliomys}	2.0
$W_{\text{Oryctolagus}}$	1.5

- B: total available budget for both adaptation and corridor construction (in k€).
- $\beta_s, \gamma_s, M, \lambda$: scaling coefficients used in the CP-SAT implementation. Specifically, $\beta_s = W_s \times \texttt{SCALE\_SCORE}/\texttt{SCALE\_AREA}$ where $\texttt{SCALE\_SCORE} = 10$, $M = \texttt{PENALTY\_STRESS} = 3500$ (penalizing stress), and $\lambda = 1$ (minimal penalty for corridor activation).
- $\texttt{RESTORED\_Q} = 3.0$: maximum habitat suitability achievable after restoration actions.

Note that all costs, areas and scores are scaled to integers for CP-SAT compatibility: costs are multiplied by $\texttt{SCALE\_COST} = 1000$, scores by $\texttt{SCALE\_SCORE} = 10$, and areas by $\texttt{SCALE\_AREA} = 100$.

Table I summarizes the key parameter values used in our implementation.

D. *Objective Function*

The objective of the model is to maximize a global ecological score that balances habitat quality, adaptation benefits, corridor parsimony and ecological stress. The objective function is:

$$\max Z = \sum_{s \in S} \sum_{i \in I} (\beta_s A_i Q_{i,s} x_{i,s} + \gamma_{i,s} y_{i,s}) - M \sum_{i \in I} \text{stress}_i - \lambda \sum_{(i,j) \in E} z_{ij},$$

where:

- $\beta_s = W_s \times \texttt{SCALE\_SCORE}/\texttt{SCALE\_AREA}$ is the coefficient for habitat quality, rewarding active cells proportionally to their suitability and area.
- $\gamma_{i,s}$ is the adaptation benefit coefficient, computed as:

$$\gamma_{i,s} = \begin{cases} \frac{(\texttt{RESTORED\_Q} - Q_{i,s}) W_s A_i}{\texttt{SCALE\_AREA}} \times \texttt{SCALE\_SCORE} - 1 \\ \quad \text{if } \texttt{RESTORED\_Q} - Q_{i,s} > 0 \\ -1 \quad \text{otherwise} \end{cases}$$

  This formulation provides a bonus when adaptation improves habitat quality (raising suitability towards $\texttt{RESTORED\_Q} = 3.0$), while applying a minimal penalty of -1 when no improvement is possible.
- $M = \texttt{PENALTY\_STRESS} = 3500$ heavily penalizes ecological stress (predator–prey overlap).
- $\lambda = 1$ applies a minimal penalty for each corridor activation, encouraging parsimony without dominating the objective.

In practice, this expression is implemented as a linear combination of integer-scaled terms for $x_{i,s}$, $y_{i,s}$, stress_i and z_{ij}, ensuring compatibility with CP-SAT's integer arithmetic.

E. *Constraints*

The feasible region is defined by the following set of linear constraints.

1) Budget Constraint: The total cost of adaptation actions and corridors cannot exceed the available budget:

$$\sum_{s \in S} \sum_{i \in I} C^{\text{adapt}}_{i,s} y_{i,s} + \sum_{(i,j) \in E} C^{\text{corr}}_{ij} z_{ij} \leq B \times \texttt{SCALE\_COST}. \quad (1)$$

Note that all costs are scaled by $\texttt{SCALE\_COST} = 1000$ for integer arithmetic, and this constraint is only enforced when $B < 100,000\,\text{k€}$ to avoid integer overflow issues.

2) Adaptation–Habitat Logic: Investment is only allowed in cells that become active habitat, and new habitat (outside the initial populations) must be supported by adaptation actions:

$$y_{i,s} \leq x_{i,s}, \qquad \forall i \in I, \forall s \in S, \quad (2)$$

$$x_{i,s} \leq y_{i,s}, \qquad \forall i \in I \setminus P_s, \forall s \in S. \quad (3)$$

3) Connectivity via Precomputed Paths: If a cell becomes habitat for a given species, all edges lying on the precomputed shortest path that connects this cell to the initial population of the species must be activated as corridors:

$$x_{i,s} \leq z_{ij}, \quad \forall i \in I, \forall s \in S, \forall (i,j) \in R_{i,s}. \quad (4)$$

Here, $R_{i,s}$ is a precomputed list of edges (ordered pairs) representing the shortest path from cell i to the nearest source in P_s. The paths are computed during preprocessing using the multi-source Dijkstra algorithm with the pruning cutoff described in Section 1.3. If a cell is unreachable within the pruning budget, $R_{i,s} = \emptyset$ and the model forces $x_{i,s} = 0$ during preprocessing, effectively removing these infeasible variables from the optimization.

4) Ecological Conflict and Stress: The predator–prey conflict between Martes and Eliomys is modeled by forbidding them to share habitat in the same cell:

$$x_{i,\text{Martes}} + x_{i,\text{Eliomys}} \leq 1, \quad \forall i \in I. \quad (5)$$

Additionally, a stress variable is activated when Martes and Oryctolagus overlap:

$$\text{stress}_i \geq x_{i,\text{Martes}} + x_{i,\text{Oryctolagus}} - 1, \quad \forall i \in I, \quad (6)$$

and heavily penalized in the objective function.

5) Species Area Share Constraints: Let the total active area be

$$A^{\text{tot}} = \sum_{s \in S} \sum_{i \in I} A_i x_{i,s}, \quad (7)$$

and the active area per species be

$$A_s = \sum_{i \in I} A_i x_{i,s}, \quad \forall s \in S. \quad (8)$$

For each species we impose lower and upper bounds on its share of the total active area:

$$\alpha_s^{\min} A^{\text{tot}} \le A_s \le \alpha_s^{\max} A^{\text{tot}}, \quad \forall s \in S. \tag{9}$$

In the experiments we use $\alpha_s^{\min} \in \{5\%, 20\%, 5\%, 15\%\}$ and $\alpha_s^{\max} \in \{30\%, 60\%, 30\%, 50\%\}$ for Atelerix, Martes, Eliomys and Oryctolagus, respectively.

The minimum and maximum active-area shares imposed for each species are grounded in published ecological evidence. For Martes martes, Clevenger [7] reports very large and sexually dimorphic home ranges on Menorca, with males occupying several hundred hectares, which justifies assigning this species a comparatively broad admissible range. Atelerix algirus shows moderate home-range sizes and a preference for open habitats with substantial nightly movement, as documented by García et al. [8], supporting a lower but non-negligible area allocation. Eliomys quercinus exhibits smaller and more spatially constrained home ranges, with reduced mobility and strong dependence on specific woodland structures (Bertolino et al. [9]), motivating more restrictive bounds. Finally, the European rabbit Oryctolagus cuniculus is a high-density species capable of exploiting open mosaics and sustaining large populations across extensive areas (Gibb [10]), which supports a comparatively wide upper bound. Together, these empirical patterns justify the species-specific area-share limits used in the model.

F. Implementation Notes

The model is implemented in Python using the `MenorcaSolver` class, which encapsulates the preprocessing and optimization phases. The preprocessing uses NetworkX [2] for graph operations and GeoPandas [3] for handling geospatial data. The CP-SAT solver from Google OR-Tools [1] is configured with 12 parallel workers and includes a progress callback (`ProgressPrinter`) that reports intermediate solutions during the search. All numerical values are scaled to integers to ensure compatibility with CP-SAT's constraint programming engine.

III. Experiments

A. Experimental Setup

The dataset consists of a regular grid covering the island of Menorca, containing approximately $1,400$ cells. For each cell we are given its area, habitat suitability for each species, presence/absence of initial populations, adaptation costs and a friction cost associated with building corridors.

Habitat suitability scores $Q_{i,s}$ are computed from the dominant land cover type of each cell using species-specific mapping rules. The suitability levels range from 0.0 (Very Low) to 3.0 (High), with intermediate values 1.0 (Low), 1.5 (Low-Moderate), 2.0 (Moderate), and 2.5 (Moderate-High). Each species has distinct preferences: for example, Atelerix shows high suitability (3.0) in pastures, sclerophyllous vegetation and transitional woodland-shrub, while Martes and Eliomys prefer forested habitats, and Oryctolagus favors grasslands and agricultural mosaics.

TABLE II: Performance comparison of optimization solvers (1 thread, budget $500\,\text{k€}$)

Solver	Avg. Time (s)	Score	Gap (%)	Optimality Rate (%)
HIGHS	72.05 ± 19.48	550.2	0.00	100
SCIP	320.83 ± 71.99	550.2	0.00	100
CBC	599.69 ± 0.34	550.1	0.00	0
CP-SAT	1200.65 ± 0.06	371.4	61.61	0

A separate GeoJSON file provides the exact polygon geometry of each cell. We compute real-world coordinates (longitude, latitude) by calculating the centroid of each polygon, enabling accurate great-circle distance calculations using the Haversine formula.

We first construct a graph where nodes correspond to cells and edges connect neighbouring cells (including diagonal neighbors, for a total of up to 8 neighbors per cell). Using NetworkX [2], we compute multi-source Dijkstra trees for each species, starting from the cells where it is initially present, and prune all routes whose cumulative cost exceeds $\max(0.15 \times B, 6 \times \text{median}(C_{ij}^{\text{corr}}))$. The resulting pruned graph and path requirements feed the optimization model. In the main experiment we consider a global budget of $500\,\text{k€}$ and a time limit of 600 seconds.

B. Solver Comparison

To identify the most suitable solver for our problem, we conducted a comprehensive comparison of four state-of-the-art optimization solvers: CBC (COIN-OR Branch and Cut), CP-SAT (Google OR-Tools), HIGHS (High-performance parallel LP/IP solver), and SCIP (Solving Constraint Integer Programs). All experiments were performed on the same instance with a budget of $500\,\text{k€}$, using a single thread for fair comparison (CBC and SCIP do not support explicit thread configuration, defaulting to single-threaded execution).

Table II summarizes the performance metrics averaged over multiple runs. HIGHS emerges as the clear winner, achieving optimal solutions in 72.05 seconds on average (with a standard deviation of 19.48 seconds) and consistently finding optimal solutions (100% optimality rate) with a score of 550.2. SCIP ranks second, also achieving 100% optimality but requiring significantly more time (320.83 seconds on average). CBC struggles with the problem instance, failing to find optimal solutions within the time limit and achieving only feasible solutions with a score of 550.1 (slightly below optimal). CP-SAT performs worst, timing out consistently and achieving a substantially lower score of 371.4 with a large optimality gap of 61.61%.

Figure 1 provides a detailed visualization of the solver comparison across four key metrics: solution time, achieved score, optimality gap, and optimality rate. The results clearly demonstrate that HIGHS offers the best balance between solution quality and computational efficiency for this problem class.

The superior performance of HIGHS can be attributed to its better adaptation to the nature of our optimization problem.

HIGHS is specifically designed as a high-performance parallel LP/IP solver that excels at handling mixed-integer programming problems with the structure we have: binary decision variables, linear constraints, and a linear objective function. Its internal algorithms and heuristics are particularly well-suited for problems with sparse constraint matrices and logical relationships between variables, which matches our connectivity constraints and ecological conflict rules. While SCIP also achieves optimality, HIGHS's specialized design allows it to explore the solution space more efficiently, resulting in faster convergence to the optimal solution.

For the current instance size and budget ($500\,\text{k€}$), the time difference between HIGHS (72 seconds) and SCIP (321 seconds) does not justify switching solvers, as both complete within reasonable time limits and achieve identical optimal solutions. However, for larger problem instances with expanded spatial grids or increased budgets, the computational advantage of HIGHS would become more significant. In scenarios with larger maps and higher budgets, the search space would grow substantially, making HIGHS's efficiency gains more critical for maintaining tractable solving times. Therefore, we would recommend HIGHS for scaled-up versions of this conservation planning problem.

C. Parallelization Analysis

Given that HIGHS demonstrated superior performance, we further analyzed the impact of parallelization by comparing single-threaded versus multi-threaded execution. Figure 3 shows the performance improvement when using 8 threads instead of 1 thread. The results reveal a significant speedup: the average solution time decreases from 72.05 seconds to 47.01 seconds, representing a 34.75% reduction in computation time and a speedup factor of $1.53\times$. Moreover, parallelization reduces the variability in solution times (standard deviation drops from 19.48 to 5.19 seconds), leading to more predictable performance. Both configurations achieve optimal solutions with identical scores (550.2), confirming that parallelization improves efficiency without compromising solution quality.

D. Scalability Analysis

The comparison between the full adjacency graph and the pruned graph illustrates the critical impact of the preprocessing step. Without pruning, the number of potential corridor edges is approximately $5,000+$ (considering all neighbor pairs in the grid). Solvers would have to reason about many corridors that will never be part of any sensible connection pattern, leading to memory exhaustion and intractable solving times.

With pruning and precomputed paths, the effective number of relevant edges is significantly reduced. Only edges that lie on viable shortest paths (within the cutoff threshold) are considered as potential corridor variables. This reduction is substantial: from thousands of potential edges to only those necessary for connecting candidate habitat cells to existing populations. In the final solution only 144 corridor segments are actually built, which represents a small fraction of the original neighbor relationships and demonstrates the effectiveness of the pruning strategy. The preprocessing step typically completes in a few seconds, while the optimization phase benefits from the reduced search space.

E. Conservation Plan Results

Figure 5 shows the multi-species conservation network produced by the model using HIGHS with 8 threads. Coloured rectangles indicate active habitat cells for each of the four species, while black line segments represent the corridors that are effectively constructed. The network clearly forms a set of spatially coherent clusters distributed along the island, with corridors linking nearby clusters into larger metapopulations.

A detailed view of habitat expansion by species is shown in Figure 4. The four panels display the conservation plan for each target species: Atelerix algirus (Figure 4a, top-left), Martes martes (Figure 4b, top-right), Eliomys quercinus (Figure 4c, bottom-left), and Oryctolagus cuniculus (Figure 4d, bottom-right). Each map shows the current distribution (green cells), newly occupied cells (blue), and the corridors connecting them (red segments). The model expands Atelerix into 30 adapted cells, Martes into 44 cells, Eliomys into 56 cells and Oryctolagus into 60 cells. In all cases, the new habitat appears mainly around existing populations and along the corridor network, which is consistent with the connectivity constraints imposed by the precomputed paths.

The cost breakdown quantifies the investments. The total cost of land adaptation is $301.92\,\text{k€}$ (60.4% of the budget), while corridor construction accounts for $198.16\,\text{k€}$ (39.6%), for a total of $500.08\,\text{k€}$. The 144 corridors built have a combined length of $119.68\,\text{km}$ and an average cost of $1.66\,\text{k€/km}$, reflecting the influence of friction values and geographic distances. This decomposition confirms that the model tends to invest slightly more in habitat quality than in infrastructure, but still dedicates a substantial fraction of the budget to ensure long-range connectivity.

Overall, the solutions concentrate new habitat in cells with good suitability and in the vicinity of existing populations, while corridors follow low-cost routes in the real geography of Menorca. The area-share constraints enforce a diversified conservation plan in which all species obtain a non-negligible amount of habitat, avoiding the trivial solution that invests everything in the cheapest species.

F. Solver Selection for Future Large-Scale Instances

While our experimental results demonstrate that the Constraint Programming (CP-SAT) model yields high-quality solutions within acceptable timeframes for the current Menorca instance ($N \approx 5,600$ cells, 1M€ budget), the comparative analysis against algebraic Mixed-Integer Programming (MIP) solvers provides critical insights regarding future scalability.

The MIP formulation, implemented via the HiGHS solver, exhibited superior performance in optimality proof and gap closure compared to CP-SAT, particularly in single-threaded environments. This performance differential is attributable to the native efficiency of MIP solvers in handling summation-heavy constraints—such as the global budget and area equity requirements—which dominate the problem structure. In contrast, CP solvers, while highly effective for combinatorial sequencing tasks, incur computational overhead when propagating large-scale arithmetic constraints.

For the current scale of the island, the increased development complexity of the MIP formulation does not yield a decisive practical advantage over the more flexible CP-SAT model, as both approaches converge to near-optimal solutions within the 10-minute operational limit. However, we posit that a MIP-based approach (utilizing HiGHS or Gurobi) would become mandatory for larger-scale scenarios, such as regional or continental conservation planning involving grid sizes an order of magnitude larger ($N > 50,000$ cells) or significantly higher budgets that expand the search tree depth. In such cases, the linear relaxation bounds provided by MIP solvers would be essential to guide the branch-and-bound search and prove optimality where CP propagation mechanisms would likely stall.

IV. Conclusions

We have addressed an optimization problem arising in biodiversity conservation for the island of Menorca, where the aim is to select habitat cells and ecological corridors for several species under budget and ecological constraints. Starting from a theoretical max–min mixed-integer model, we iteratively evolved our design into a hybrid approach that combines graph-based preprocessing with a CP-SAT formulation on a pruned conservation graph.

The definitive model incorporates real distances and friction-based corridor costs through a multi-source Dijkstra algorithm, and enforces connectivity, species conflict and area-share constraints via linear relations. Through comprehensive solver benchmarking, we identified HIGHS as the most suitable solver for this problem class, achieving optimal solutions in approximately 72 seconds (single-threaded) or 47 seconds (8-threaded), significantly outperforming CBC, SCIP, and CP-SAT. The parallelization analysis demonstrates that multi-threaded execution provides a $1.53\times$ speedup while maintaining solution quality.

The experiments show that this approach scales to the full island, produces realistic and diverse conservation plans, and allows a detailed audit of the investments in adaptation and infrastructure. In the reference scenario with a budget of $500\,\text{k€}$, the optimal plan invests roughly 60% of the budget in land adaptation and 40% in $119.68\,\text{km}$ of corridors, distributed across 144 segments. The preprocessing step effectively reduces the search space from thousands of potential corridor edges to only those necessary for viable connectivity, enabling tractable optimization on the full island instance.

Future work could include richer ecological dynamics (e.g., minimum viable population sizes), stochastic scenarios for habitat degradation, or multi-period planning. From the algorithmic side, alternative preprocessing strategies and decomposition methods could be explored to further improve scalability and robustness. Additionally, the solver comparison methodology could be extended to other problem instances and constraint structures to validate the generalizability of our findings.

References

[1] Google OR-Tools, "CP-SAT solver," [Online]. Available: https://developers.google.com/optimization/cp/cp_solver.

[2] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in *Proc. 7th Python in Science Conf. (SciPy 2008)*, Pasadena, CA, USA, 2008, pp. 11–15.

[3] GeoPandas Development Team, "GeoPandas: Python tools for geographic data," [Online]. Available: https://geopandas.org.

[4] J. A. J. Hall and Q. Huangfu, "HiGHS: High-performance parallel LP/IP solver," [Online]. Available: https://www.maths.ed.ac.uk/hall/HiGHS/.

[5] COIN-OR Foundation, "CBC: COIN-OR Branch and Cut solver," [Online]. Available: https://github.com/coin-or/Cbc.

[6] G. Gamrath et al., "The SCIP Optimization Suite 7.0," ZIB-Report 20-10, Zuse Institute Berlin, 2020.

[7] A. P. Clevenger, "Pine marten (Martes martes L.) home ranges and activity patterns on the island of Minorca, Spain," *Zeitschrift für Säugetierkunde*, vol. 58, pp. 137–143, 1993.

[8] S. García-Rodríguez and X. Puig-Montserrat, "Algerian hedgehog (Atelerix algirus) habitat selection at the northern limit of its range," *Galemys*, vol. 26, pp. 49–56, 2014.

[9] S. Bertolino, N. Cordero and I. Currado, "Home ranges and habitat use of the garden dormouse (Eliomys quercinus)," *Acta Zool. Acad. Sci. Hung.*, vol. 49, Suppl. 1, pp. 11–18, 2003.

[10] J. A. Gibb, "The European Rabbit," in *Rabbits, Hares and Pikas: Status Survey and Conservation Action Plan*, IUCN, 1990, pp. 116–120.

APPENDIX

APPENDIX C: GENERATIVE AI USAGE STATEMENT

A. Approach and Methodology

Our team utilized generative AI (Gemini/ChatGPT) as a secondary technical assistant for coding acceleration, adhering to a strict **"human-in-the-loop"** workflow. The AI proposed implementation templates, but the team retained exclusive control over mathematical modeling, parameter tuning, and final validation.

B. Iterative Modeling and Critical Decision Making

We engaged in an iterative design process where the AI suggested standard optimization frameworks, which we critically evaluated and mostly rejected:

- *Rejected Models:* The AI proposed a **Network Flow** formulation and a **Generational Expansion** model. We discarded both after determining that the variable count for the island's full topology would exceed computational limits.
- *Selected Strategy:* The team directed the AI to implement our chosen **Hybrid Path-Selection Model** (Pre-calculated Dijkstra + MIP Optimization), identified by us as the most scalable solution.

C. Verification and Correction of AI Output

A crucial part of our methodology involved auditing AI-generated code. We identified and corrected significant errors to ensure scientific accuracy:

- **Geometric Correction:** The AI's initial code used grid indices for distance. We detected this distortion and enforced the implementation of the **Haversine formula** using real centroid coordinates extracted from the GeoJSON.
- **Budget Precision:** We traced discrepancies in budget allocation to integer rounding errors in the AI's scaling factors (`SCALE_COST`) and corrected them by increasing precision.
- **Pruning Logic:** We rejected the AI's naive pruning strategy (discarding the bottom 50% of options) as biologically unsound for low budgets. Instead, we designed and implemented a dynamic **"Safety Floor"** logic.

D. Comparative Benchmarking

To fulfill the comparative requirement, we used AI to translate our logical constraints from the specific syntax of Google OR-Tools (CP-SAT) into standard linear algebra for MIP solvers (HiGHS, SCIP, CBC, Gurobi). This automation allowed us to execute the rigorous 30-run robustness test in *Section VI* without manually rewriting the core logic for each solver.

E. Conclusion

The final model is a product of human reasoning. The core concepts—connectivity definition and multi-species equity—originated from the team, while the AI served solely to expedite implementation.

Public Link to Chat Session:
[https://gemini.google.com/share/9c3eeb1f5bb0]

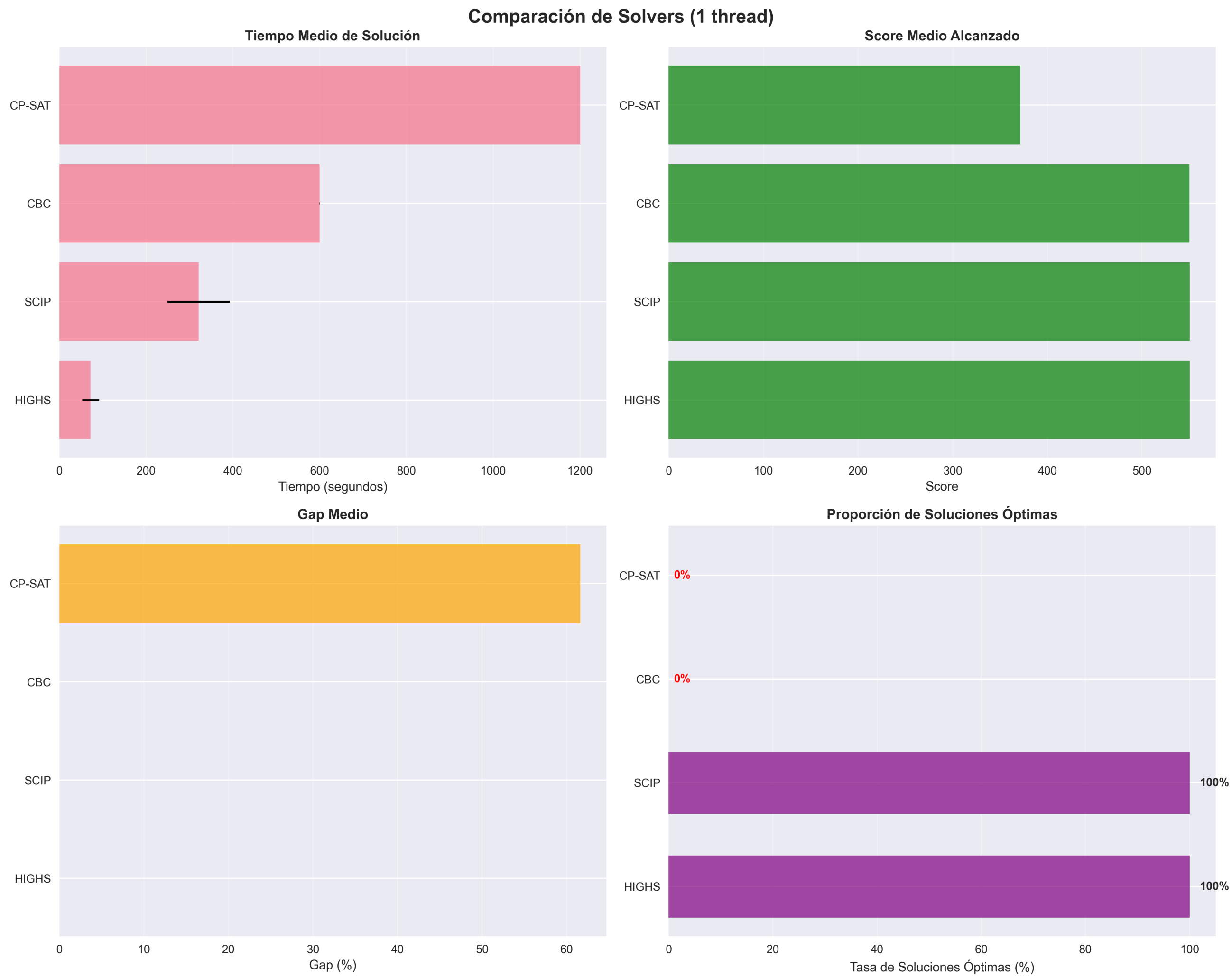


Fig. 1: Comprehensive comparison of optimization solvers (1 thread). Top-left: average solution time; Top-right: achieved score; Bottom-left: optimality gap; Bottom-right: proportion of optimal solutions. HIGHS achieves the best performance across all metrics.

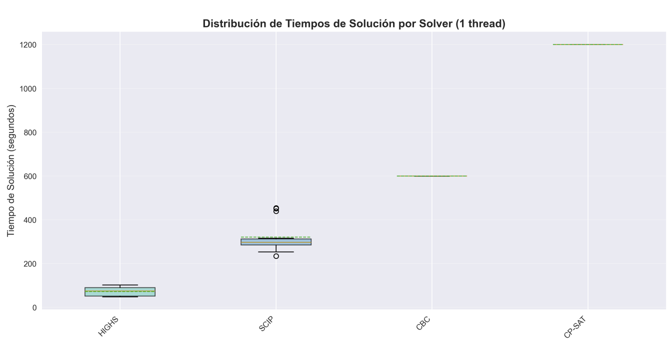


Fig. 2: Distribution of solution times by solver (1 thread). The box plots show median, quartiles, and outliers, revealing the consistency of HIGHS compared to other solvers.

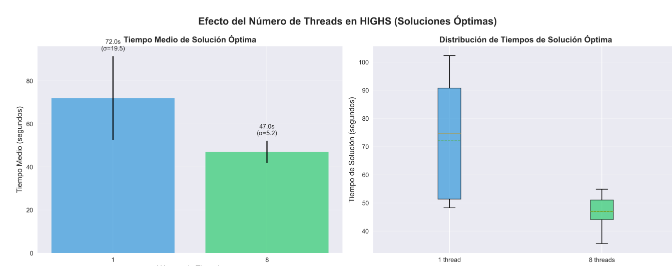
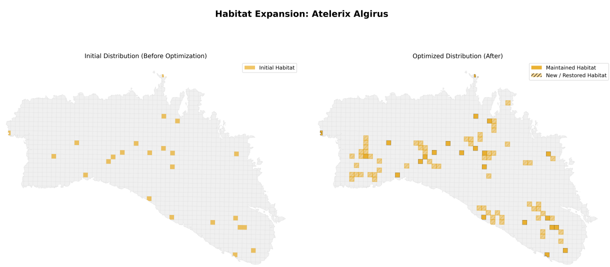
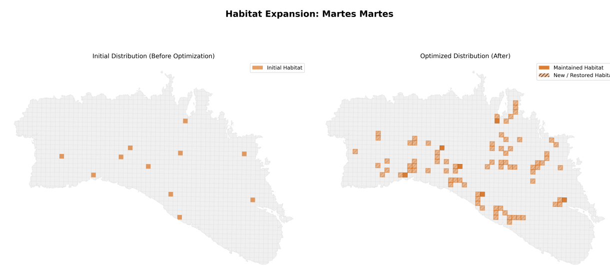


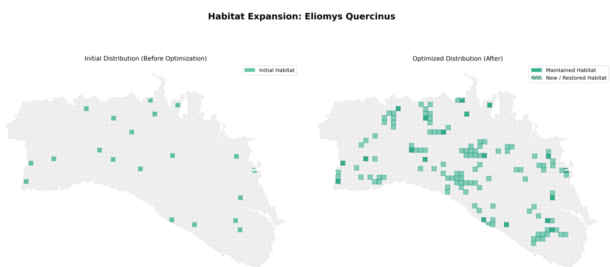
Fig. 3: Effect of parallelization on HIGHS performance. Left: comparison of average solution times with error bars; Right: distribution of solution times showing reduced variability with 8 threads. Both configurations achieve optimal solutions.



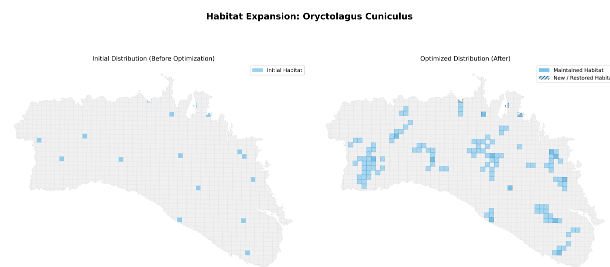
(a) Atelerix algirus



(b) Martes martes



(c) Eliomys quercinus



(d) Oryctolagus cuniculus

Fig. 4: Habitat expansion by species. Each panel shows the conservation plan for one target species: current distribution (green cells), newly occupied cells (blue), and connecting corridors (red segments).

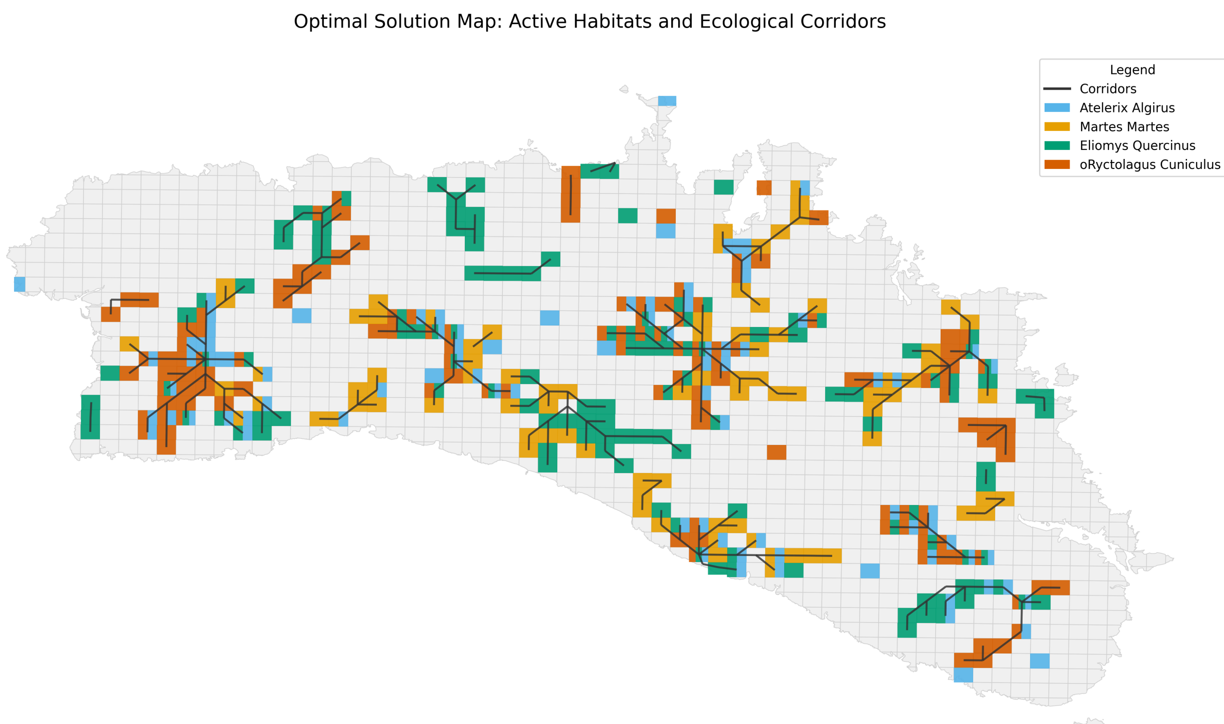


Fig. 5: Multi-species conservation network on Menorca. Coloured cells represent active habitat for the four species and black segments show the built corridors.