More diverse reserve solutions for protecting biodiversity: an explicit difference criterion for producing a presentation set with exact optimisation methods

Adrien Brunel (adrien.brunel.pro@protonmail.com)
IRD, MARBEC (Univ. Montpellier, CNRS, IRD)
Jérémy Omer
Univ Rennes, INSA Rennes, CNRS, IRMAR, UMR 6625
Sophie Lanco-Betrand
IRD, MARBEC (Univ. Montpellier, CNRS, IRD)

Research Article

Keywords: optimal reserve site selection, presentation set, protected areas, conservation planning, integer linear programming, decision support tool.

Posted Date: May 18th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1644920/v1

More diverse reserve solutions for protecting biodiversity: an explicit difference criterion for producing a presentation set with exact optimisation methods

Adrien Brunel[1*], Jérémy Omer[2] and Sophie Lanco-Betrand[1]

[1*]IRD, MARBEC (Univ. Montpellier, CNRS, Ifremer, IRD), Avenue Jean Monnet, Sète, 34203, France.
[2]Univ Rennes, INSA Rennes, CNRS, IRMAR, UMR 6625, Rennes, 35000, France.

*Corresponding author(s). E-mail(s): adrien.brunel.pro@protonmail.com;
Contributing authors: jeremy.omer@insa-rennes.fr; sophie.bertrand@ird.fr;

Abstract

Protected areas are at the heart of current global policies against the erosion of biodiversity. For instance, to meet the pressing challenge of covering 30% of the surface of the sea under jurisdiction with a network of marine protected areas by 2030, reserve site selection models are increasingly mobilised. These models address the optimisation problem that seeks to cover biodiversity features at a minimum cost on human activities. A presentation set, *i.e.* a pool of alternative solutions, is often needed to increase the chances of satisfying unmodeled objectives that may be non-negligible in a conservation problem composed of various interests. The widely used Marxan repeats a metaheuristic algorithm based on random processes to produce diversity in the presentation set. Several works already demonstrated how exact optimisation methods outperform metaheuristics both in computation time and optimality. However, the generation of multiple solutions is still raised as a key feature of metaheuristics. In this work, we proposed two algorithms to generate a presentation set with exact optimisation methods using recursive procedures based on an explicit difference criterion. The resulting alternative solutions were generated by controlling the optimality gap but also the difference with the optimal solution. This work showed that the presentation set offered by metaheuristics can be outstripped by an explicit, transparent and replicable presentation set built by exact optimisation methods. Allowing to understand precisely why and how the result was arrived at, the framework presented here should contribute to a more equitable negotiation among stakeholders engaged in conservation planning processes.

Keywords: optimal reserve site selection; presentation set; protected areas; conservation planning; integer linear programming; decision support tool.

1 Introduction

Biodiversity and habitats are threatened worldwide [1] and one of the common conservation instrument used to combat this is to build comprehensive networks of nature reserves [2–4] known to bring benefits [5–7]. At sea for instance, current political objectives are to cover 30% of the waters under jurisdiction by 2030 with marine protected areas [8–10]. Similar concerns also exist on land [11, 12]. Given these objectives, there is a strong constraint to find the best compromises between the protection of biodiversity and the sustainability of human uses of these spaces. To do so, and to take into account the wide variety and heterogeneity of information required to analyse such problems, numerical optimisation methods are commonly implemented [13–16]. Such methods are often embedded in a software, *e.g.* Marxan or prioritizr. They are designed to systematically select reserve sites and are used as a decision support tools in real world cases[1] [17].

In such a decision-making framework, the views and objectives of stakeholders are usually competing, the spatial information documenting the region rarely exhaustive, and some issues at stake in the negotiations may not be easy to formulate in the context of a spatially-explicit numerical optimisation. The decision process, based *in fine* on negotiations, thus requires some latitude on the possible solutions to be considered. In addition, generating alternative solutions gives conservation practitioners the possibility of finding a solution that could be more satisfactory with respect to unmodeled objectives. For these reasons, the ability of decision support tools to produce a range of solutions instead of a single one, has been put forward frequently in conservation literature [18–21]. Recently, and more generally, [22] calls for more effort in the design of alternatives in decision aiding processes. That is why in conservation biology, reserve site selection tools such as Marxan or prioritizr also focus on generating near-optimal alternatives [23–25]. In particular, in a survey realised among Marxan users [26], "flexibility of generating multiple solutions was by far the most commonly noted strength of Marxan" over other optimization solvers.

More generally, the ability to produce alternative solutions were often presented as a key strength of metaheuristic algorithms over exact optimization approaches [18–21]. However, recent advances made exact solvers more attractive for conservation practitioners [27–30]. The "flexibility" claim thus seems to be the last argument remaining in favour of metaheuristic algorithms in the conservation biology literature. Technically speaking, metaheuristics rely on the repetition of random processes to implicitly create diversity in the generated solutions (see Appendix B-2.1 in [31]). On the contrary, exact solving methods usually produce the same solution if repeated and therefore cannot easily produce a range of different solutions. This major limitation

[1] More case study examples can be found at https://marxansolutions.org/community/ and https://prioritizr.net/articles/publication_record.html.

severely restricts exact solving methods ability to inform real-world conservation problems. Yet, and in the absence of explicit criteria, metaheuristic approaches do not provide any control over the alternatives generated nor they guarantee to have truly different solutions. However, the search for near-optimal alternative solutions with exact solvers was an issue addressed by the "modeling to generate alternatives" literature [32–35]. In particular, this point was already discussed by [36] who call on exact solving methods to seriously address the issue of generating alternatives close to optimality by relying on the existing literature of multi-objective optimisation. Few existing works in the conservation biology literature also had the concern of showing that alternative solutions can be generated with exact solving methods. [37] developed an algorithm to compute the exhaustive set of optimal solutions in the maximum coverage formulation with presence/absence data. In the same spirit, [38] described a branch and bound screening algorithm to show how suboptimal solutions, called the second-best solutions, can be derived with exact methods. As mentioned before, the reserve site selection tool prioritizr also provides additional functions[2] allowing users to build a "portfolio" of alternative solutions.

The main objective of this work was to show conservation practitioners how we could generate different near-optimal solutions of the reserve site selection problem thanks to exact solving methods. In particular, we proposed two recursive algorithms incorporating an explicit distance criterion to build a range of near-optimal solutions that were different from each other. Therefore, alternative solutions were selected based on a controlled optimality gap but also thanks to an explicit distance criterion. To choose a relevant difference criterion, we discussed and compared two metrics in our conservation biology context. Indeed, the natural distance metric led to generate alternative solutions that strictly included the optimal one. Observing that this was not a valuable alternative but only a degraded solution, the other distance metric that we proposed allowed to discriminate such case. Although similar ideas were discussed in a more operations research context [33, 34], we still think these ideas are not common in conservation biology. Practically, one of the contribution comes from the developments of these two recursive procedures that allow to guarantee the generation of alternatives different not only from the optimal solution but also from each other. These procedures occasioned the formulation of mixed integer linear programs solved using exact methods. Another important contribution is the comparative analysis of these two procedures in terms of solutions quality and difference.

The code related to this work is open, free and available[3]. We used Gurobi solver called through a code developed in Julia language [39, 40] using the JuMP optimisation library [41]. Gurobi is a commercial solver available under

[2] https://prioritizr.net/reference/portfolios.html
[3] https://github.com/AdrienBrunel/rssp_presentation_set.git

a free academic licence. We could have also solved the problem with the free and open-source integer programming solver Cbc [42]. The developed methods were numerically tested on the real example of Fernando de Noronha Brazilian archipelago in the tropical Atlantic composed of 3 conservation features and 756 planning units. We also assessed the generality of the approach by considering generated test cases. In brief, we proposed a methodology using exact solving methods to build a range of near-optimal solutions which are explicitly different from each other. This way, we hope to build alternative solutions that better fill unmodeled objectives in real-world conservation planning problems.

At this point, it seemed necessary to us to be more precise about terminology used in order to dissipate any ambiguity when one speaks about the generation of alternative solutions of an optimisation problem. Indeed, this concept had several names in the conservation biology literature. For example, [26] spoke of "flexibility" to designate this feature of Marxan. However, in the same context, this term referred to the ability of an optimisation model to be easily tuned [43, 44]. We thus preferred to avoid the term "flexibility". In prioritizr, the term "portfolio" was used in the name of functions generating alternative solutions. But again, we did not use this word in our work in order to avoid any confusion with the "modern portfolio theory" often referred in the risk-averse optimisation literature. Finally, in the "modelling to generate alternatives" literature, the term used was "presentation set" which made explicit the fact that these alternative solutions were intended to be presented to decision-makers and decided upon. That is why we used the word "presentation set" in this work to name the set of alternative solutions.

2 Methods

2.1 General formulation of the reserve site selection problem

In a reserve site selection problem, the study area is discretized into a set of J planning units within which a set of I conservation features are distributed. The amount of conservation feature i in the planning unit j was denoted a_{ij}. Each planning unit has a cost c_j usually understood as the socio-economic cost associated with the closure of this unit. One then seeks to find the collection of planning units covering sufficient levels of each conservation features at a minimum cost. The covering of a conservation feature i is judged sufficient if it exceeds the user-defined level that we noted t_i. The decision is about whether to include the planning unit in the reserve. Thus, $x_j = 1$ if a planning unit j is selected in the reserve and $x_j = 0$ otherwise. Moreover, a compactness parameter β allows computing a more or less aggregated reserve, as it directly penalises the reserve perimeter within the objective function. The perimeter is computed as the total length of the boundaries between reserved and non-reserved planning units. To model this, the length of the shared boundary between planning unit j_1 and j_2 was denoted $b_{j_1j_2}$. Mathematically speaking, the general problem of reserve site selection is expressed as the following minimum set optimisation problem P_0:

$$P_0 : \begin{cases} \min\limits_x \ \sum\limits_{j\in J} c_j x_j + \beta \sum\limits_{j_1\in J} \sum\limits_{j_2\in J} b_{j_1j_2} x_{j_1}(1 - x_{j_2}) & \\ \text{s.t.} \ \sum\limits_{j\in J} a_{ij} x_j \geq t_i & \forall i \in I \\ \quad x_j \in \{0, 1\} & \forall j \in J \end{cases} \tag{1}$$

This combinatorial optimisation problem is a minimum set cover problem known to be NP-hard [45]. Also, it is a non-convex problem due to the binary nature of the decision variables. Yet, it can be expressed as an integer linear program (see Appendix A for the linearised model) and known solvers (like Gurobi or CbC) can solve it for realistic instances in a reasonable time. In the following, we denoted $x^\star$ and $z^\star$ respectively the optimal solution and the associated objective value of the original minimum set cover problem P_0. Let $\gamma = (z - z^\star)/z^\star$ be the algebraic relative variation of score z with respect to $z^\star$.

2.2 Measure of diversity between two reserve solutions

As a premise of our contribution, we needed a function characterising the difference between two solutions. The first natural idea was to use the distance defined by the absolute-value norm:

$$D(x, y) = \sum_{j\in J} |x_j - y_j| \tag{2}$$

When $x, y \in \{0,1\}^N$, the distance $D(x, y)$ was simply the total number of differences between x and y. It is a rigorous mathematical distance as function D verifies the symmetry, separation and triangular inequality properties. In order to avoid the absolute value in the distance definition, we considered the following linear expression, strictly equivalent[4] when $x, y \in \{0,1\}^N$:

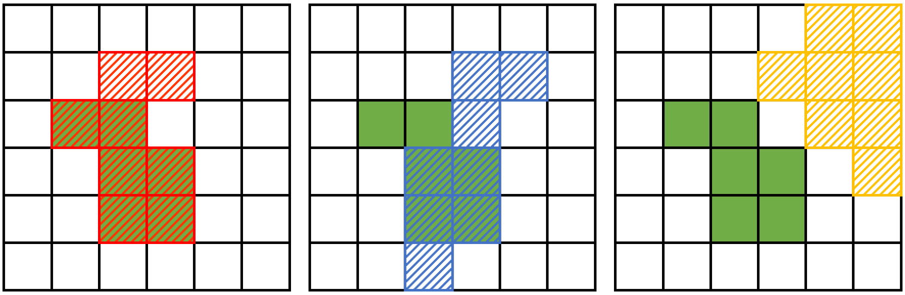
$$D(x, y) = \sum_{j \in J} x_j(1 - y_j) + y_j(1 - x_j) \tag{3}$$

However, we could easily build an alternative solution with a non-zero distance from the optimal solution $x^\star$ by choosing any solution that strictly includes $x^\star$. Such solution would logically meet the covering requirements, as it includes the optimal solution. Then, this solution would be at a distance from $x^\star$ equal to the number of planning units outside $x^\star$. Yet, we wanted to avoid this type of degenerated solutions as we thought they were not relevant alternatives but simply worse alternatives. Consequently, we finally selected the pseudo-distance $d(x, y)$ defined as the number of planning units selected in x and not in y:

$$d(x, y) = \sum_{j \in J} x_j(1 - y_j) \tag{4}$$

The term pseudo-distance is used as it can be understood as a distance, although it does not verify the rigorous mathematical definition of a distance. Observe that $D(x, y) = d(x, y) + d(y, x)$. In conclusion, the pseudo-distance d excluded alternative solutions we considered irrelevant, while the distance D would have tolerated these solutions. To illustrate our words, Figure 1 displays three examples where the distance D and pseudo-distance d were computed. The left and right panels show two specific examples: the green reserve is strictly included in the red reserve, while it has an empty intersection with the yellow reserve. The example shown in the middle panel is a more generic case, as the blue reserve has planning units both shared and not shared with the green reserve. In particular, we visually understand why the red solution is simply a worse alternative to the green solution and that we wanted to avoid this case. Then, the distance between red and green reserves is $D(\hat{x}, x) = 2$ while the pseudo-distance is $d(\hat{x}, x) = 0$. Consequently, according to the pseudo-distance d, the green and red reserve are the same. We wanted to use this feature to our advantage, as we did not want the red reserve to belong to the pool of alternative solutions. The pseudo-distance d could thus help to discriminate this type of reserve, while distance D could not. Finally, as illustrated with the blue and yellow reserve examples, the pseudo-distance still allows us to characterise differences between two reserves as much as distance D.

[4] Compare truth tables of these two expressions to be convinced.



(a) $d(\hat{x}, x) = 0$, $d(x, \hat{x}) = 2$, $D(\hat{x}, x) = 2$.

(b) $d(\hat{x}, x) = 2$, $d(x, \hat{x}) = 4$, $D(\hat{x}, x) = 6$.

(c) $d(\hat{x}, x) = 6$, $d(x, \hat{x}) = 8$, $D(\hat{x}, x) = 14$.

Fig. 1: Numerical examples of pseudo-distance d and distance D. The nominal reserve $\hat{x}$, depicted in green, includes 6 planning units. Other reserves, hatched in red, blue and yellow, include 8 planning units.

2.3 Imposing a pseudo-distance

2.3.1 Produce one alternative

Finding alternative solutions diverse enough can be done through adding a set of constraints to the initial optimisation problem P_0. Indeed, we proposed to explicitly constrain an alternative solution to differ, where diversity was measure with the pseudo-distance, from a given solution by at least δ planning units. We thus introduced the constraint $c_d(\hat{x}, \delta)$ which compelled the reserve solution x to have at least δ planning units selected in $\hat{x}$ that were not included in x:

$$c_d(\hat{x}, \delta) : d(\hat{x}, x) = \sum_{j \in J} \hat{x}_j (1 - x_j) \geq \delta \tag{5}$$

As discussed previously (cf. Section 2.2) and illustrated in Figure 2, the pseudo-distance d allowed forbidding the strict inclusion of the solution $\hat{x}$ within the considered alternatives thanks to $c_d(\hat{x}, \delta)$.

$$\begin{array}{cc|c} \hat{x} & x & d(\hat{x}, x) \\ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} & \begin{array}{c} +1 \\ +0 \\ +0 \\ +0 \end{array} \\ \hline & & \geq \delta \end{array}$$

Fig. 2: Illustration of constraint $c_d(\hat{x}, \delta)$. Example of pseudo-distance $d(\hat{x}, x)$ between solutions $\hat{x}$ and x.

2.3.2 Generate the presentation set

The most natural idea to produce a pool of diverse enough alternative solutions would be to repeat the procedure by excluding the alternative found the iteration before. This way, we would successively find the alternative solutions at a predefined pseudo-distance from the optimal solution $x^\star$. However, this approach did not ensure alternative solutions to be different from one another. To avoid the pitfall of producing similar alternative solutions although different from the optimal solution, we recursively constrained the alternative to differ not only from the optimal solution but also from every alternative solution found before. Practically, at each iteration $k \geq 0$, we successively added the constraint $c_d(x^{k-1}, \delta)$ to the initial optimisation problem P_0. If we denote $x^\star = x^0$ and $\{x^1, \ldots, x^{k-1}\}$ the set of alternative solutions derived before iteration $k \geq 0$, the integer linear program solved at iteration k is the following:

$$P_1^k : \begin{cases} \min\limits_x \sum\limits_{j \in J} c_j x_j + \beta \sum\limits_{j_1 \in J} \sum\limits_{j_2 \in J} b_{j_1 j_2} x_{j_1}(1 - x_{j_2}) & \\ \text{s.t. } \sum\limits_{j \in J} a_{ij} x_j \geq t_i & \forall i \in I \\ \quad d(x^l, x) \geq \delta & \forall l \in [\![0, k-1]\!] \\ \quad x_j \in \{0, 1\} & \forall j \in J \end{cases} \tag{6}$$

We detailed in Algorithm 1 the pseudocode of the recursive procedure described above. The procedure would have stopped whether the problem became infeasible or the maximum number of iteration n was reached. Infeasibility of the problem can be reached if the user requires alternative solutions with too many differences with one another.

Algorithm 1 Recursive search of n alternative solutions of problem P_0 with at least δ new planning units between successive iterated solutions.

Require: P_0, $x^\star$, n, δ
Ensure: x^1, ..., x^k

```
1: k ← 0; P ← P_0; x^0 ← x*                          ▷ initialisation
2: while P is feasible & k < n do     ▷ stop when infeasible or enough solutions
3:     k ← k + 1
4:     add c_d(x^{k-1}, δ) to P     ▷ impose pseudo-distance with past iterate
5:     solve P     ▷ get an optimal solution x^k or detect that P is infeasible
6: end while
```

2.4 Maximise the pseudo-distance

2.4.1 Produce one alternative

Another idea we developed was to seek for the most pseudo-distant solution at a user-defined extra cost relatively to the optimal score. Stated otherwise, we maximised the pseudo-distance from the optimal solution $x^\star$ under a fixed extra cost budget $\gamma z^\star$. The corresponding integer linear program is given by:

$$P_2 : \begin{cases} \max\limits_{x} \; d(x^\star, x) & \\ \text{s.t.} \; \sum\limits_{j\in J} c_j x_j + \beta \sum\limits_{j_1\in J}\sum\limits_{j_2\in J} b_{j_1 j_2} x_{j_1}(1-x_{j_2}) \leq (1+\gamma) z^\star & \\ \quad\;\; \sum\limits_{j\in J} a_{ij} x_j \geq t_i & \forall i \in I \\ \quad\;\; x_j \in \{0,1\} & \forall j \in J \end{cases} \tag{7}$$

2.4.2 Generate the presentation set

Once again, our interest was to produce a pool of alternative solutions. The most natural idea would have been to recursively produce the n most pseudo-distant alternative solutions from the optimal solution $x^\star$ at a fixed relative extra cost γ by successively excluding from the search space the solution found the iteration before. However, this method would have not ensured difference between alternatives and might produce a pool of similar alternative solutions, although at a maximum distance to the optimal solution $x^\star$. We thus preferred to recursively search for the solution maximising the minimum pseudo-distance from the past iterates. More precisely, let x^l be the solution found at iteration $l \in [\![0, k-1]\!]$. Let x be a candidate solution at iteration k and $\delta(x) = \min\{d(x^l, x), l \in [\![0, k-1]\!]\}$. We proposed to search for a solution x^k that maximised $\delta(x)$ among the solutions that did not exceed the degraded cost $(1+\gamma)z^\star$. Practically, the integer linear program P_2^k solved at iteration k is the following:

$$P_2^k : \begin{cases} \max\limits_{x,\delta} \; \delta & \\ \text{s.t.} \; \sum\limits_{j\in J} c_j x_j + \beta \sum\limits_{j_1\in J}\sum\limits_{j_2\in J} b_{j_1 j_2} x_{j_1}(1-x_{j_2}) \leq (1+\gamma) z^\star & \\ \quad\;\; \sum\limits_{j\in J} a_{ij} x_j \geq t_i & \forall i \in I \\ \quad\;\; \sum\limits_{j\in J} x_j^l (1-x_j) \geq \delta & \forall l \in [\![0, k-1]\!] \\ \quad\;\; x_j \in \{0,1\} & \forall j \in J \\ \quad\;\; \delta \in \mathbb{R}_+ & \end{cases} \tag{8}$$

The objective value δ would be the maximum of the minimum of the set of pseudo-distances with every past iterates. Variable δ was constrained to be smaller than every pseudo-distances, so was the minimum. As we wanted to maximise δ, it would logically have taken the value of one of the pseudo-distances, *i.e.* the smallest. Therefore, we maximised the minimum pseudo-distance with every past iterate. One should be careful as δ is a decision variable and no longer a parameter as in Section 2.3. In order to better understand what Algorithm 2 is doing, we gave more numerical details in Section 3.4. The corresponding procedure written in pseudocode is presented below:

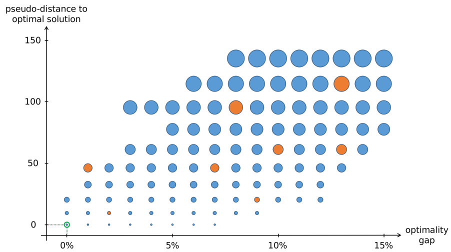
Algorithm 2 Recursive search of n alternative solutions maximising the minimum pseudo-distance from every past iterates at a relative extra cost γ.

Require: P_2, $x^\star$, $z^\star$, n, γ
Ensure: $x^1, \ldots, x^k$

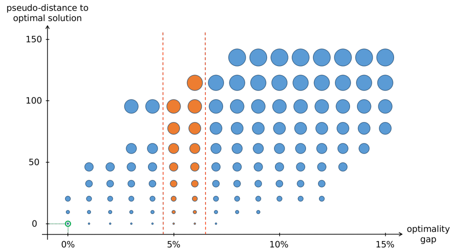
```
k ← 0; P ← P_2; x^0 ← x*                              ▷ initialisation
change the objective of P to δ
while P is feasible & k < n do    ▷ stop when infeasible or enough solutions
    k ← k + 1
    add c_d(x^{k−1}, δ) to P     ▷ impose pseudo-distance with past iterate
    solve P       ▷ get an optimal solution x^k or detect that P is infeasible
end while
```

2.5 Schematic illustration of methods used for generating a presentation set

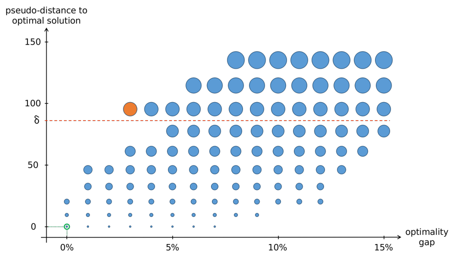
The objective of Figure 3 was to show schematically how the alternative solutions selected by different methods were positioned in a well-chosen plane, namely the optimality gap versus pseudo-distance to the optimum plane. We illustrated in Figure 3a that the repetition of a metaheuristic algorithm, *e.g.* simulated annealing, produces solutions that would be scattered in the considered plane. They have no guarantee to be both close to optimality and different from the optimal solution. The algorithm presented in the Appendix B or the *add_gap_portfolio* function of prioritizr have the principle of selecting the alternative solutions in a given optimality interval. This interval was materialised by the red dotted vertical lines in Figure 3b. As illustrated, the selected alternative solutions can have a pseudo-distance to the optimum more or less high without any guarantee. In Figure 3c, we showed how Algorithm 1 would select the leftmost solution from the solutions above the red dashed line. In other words, algorithm 1 would select the solution closest to the optimum at a fixed pseudo-distance. Similarly, for algorithm 2, the first alternative selected would be the solution with the largest pseudo-distance to a given optimality gap. In Figure 3d, this corresponds to the topmost solution among the solutions to the left of the red dashed line.



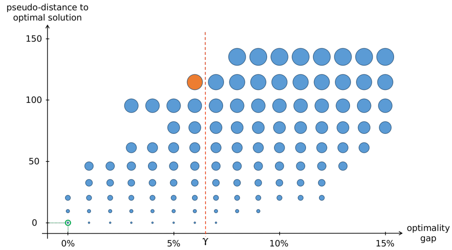
(a) Alternative solutions selected repeating a metaheuristic algorithm.



(b) Alternative solutions selected according to a predefined optimality gap, as in the *add_gap_portfolio* function of prioritizr.



(c) Alternative solution derived by algorithm 1 at the first iteration: the best solution at a predefined pseudo-distance δ is selected.



(d) Alternative solution derived by algorithm 2 at the first iteration: the most pseudo-distant solution involving at worse a budget γ is selected.

Fig. 3: Illustration of alternative solutions selected by metaheuristics, the *add_gap_portfolio* function of prioritizr, Algorithm 3 and 2. Each circle represent a reserve solution. The reserve solutions are represented by the optimality gap and pseudo-distance to the optimal solution $d(x^\star, x)$. The optimal solution $x^\star$ is circled in green at the bottom left of this plan. The bigger the circle, the greater the pseudo-distance to the optimal solution. Alternative solutions selected by an algorithm are depicted with orange circles.

3 Case study

3.1 Data preparation

The numerical assessment of the methods previously presented was mainly done on the real case of Fernando de Noronha. Other tests were also performed to get insights about the generality of the approach (*cf.* Appendix C).
Fernando de Noronha is a small oceanic archipelago in the western tropical Atlantic, made up of 21 islands, islets and rocks with a total surface area of 26 km^2, and constituting a genuine Brazilian natural and cultural heritage. Fernando de Noronha is a conservation showcase in Brazil, but it also faces many interests (oil industry, tourism intensification, fisheries) which results in an open laboratory for marine spatial planning. We used this region as a case study for the algorithms described in the previous section. The corresponding input data was processed in a study available in preprint[5]. We only summarize the main characteristics of the dataset below.

The geographical area was discretised according to a rectangular grid made of N=36×21=756 planning units with longitude and latitude respectively in [32.65°W, 32.30°W] and [3.95°S, 3.75°S] ranges. Planning units taking place in Fernando de Noronha land and harbour were a priori excluded from potential reserve site candidates. The scenario feeding the nominal optimisation problem P_0 through this work then considered three conservation features: fish biomass, continental shelf and shelf break habitats. Each feature was given a targeted protection level of 50%. The cost layer was made of the fishing pressure intensity. The compactness parameter considered was $\beta = 1$. We show in Figure 4 the details of the input data involved in the case study.

To get the cost and conservation features, the values of fish biomass was first estimated as the sum of fish echoes in nautical area back-scattering strength, i.e., s_A, over the water column. Interpolating between sample data allowed producing a continuous fish distribution within the sampling area. Outside this area, values were set to 0, although the actual fish distribution over there was unknown. Ocean depths were obtained from GEBCO online platform[6] and used as a surrogate of two suitable fish habitats, expressed as binary conservation features: the continental shelf and shelf break habitats, defined by specific ocean depth intervals. Finally, a segmentation model was applied to fisher GPS trajectories to derive one behavioural state for every measured location: fishing or travelling. This was used to build a quantitative proxy representing the intensity of the fishery activity.

[5] https://hal.archives-ouvertes.fr/hal-03445922
[6] GEneral Bathymetric Chart of the Ocean, https://download.gebco.net/

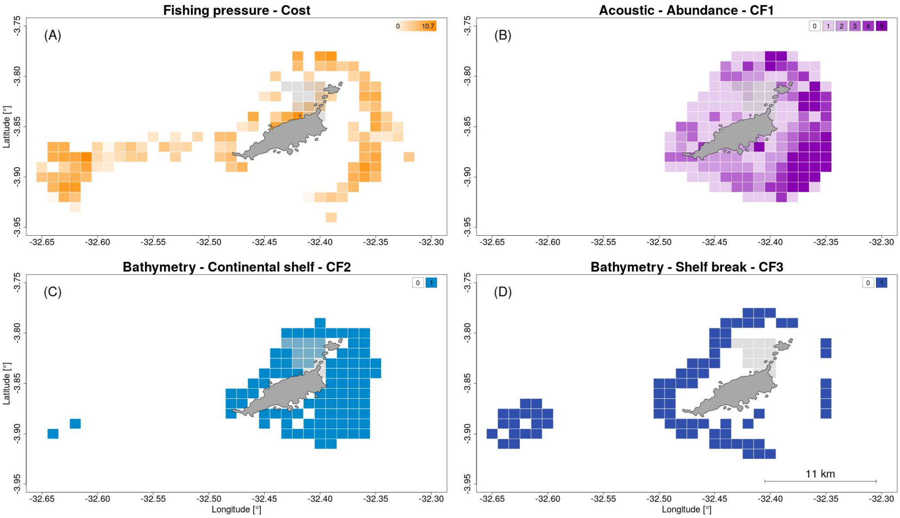


Fig. 4: Processed input data layers feeding the conservation problem. (A) Fishery-based cost layer is shown with a continuous orange colour gradient. (B) Fish Biomass conservation feature surrogate is depicted with a discrete purple colour gradient. (C) Continental shelf and (D) Shelf break habitat conservation feature surrogates are respectively illustrated in light and deep blue. Transparent grey pixels are the planning units a priori excluded from the solution.

3.2 Presentation set imposing an optimality gap

Exhaustive set of optimal solutions ($\gamma_1 = \gamma_2 = 0$)

We applied Algorithm 3 with $\gamma_1 = \gamma_2 = 0$ to derive the exhaustive set of optimal solutions. In this numerical application, it turned out we have 16 optimal solutions with an objective value $z^\star = 197.71$. Panel A of Figure 5 illustrates a map showing the selection frequency among optimal solutions, i.e., the percentage of time a planning unit was selected among the 16 optimal solutions. We observed a small variability as 84/93 planning units were selected at a 100% frequency. The nine planning units that changed were likely to be interchangeable (same cost and amount of each conservation features).

Alternative solutions in increasing order of optimality gap ($\gamma_1 = 0$, $\gamma_2 > 0$)

We computed the $n = 500$ following suboptimal solutions searched by increasing score order, thus from best to worst objective value, as it was a minimisation problem. The recursive procedure of Algorithm 3 was either stopped by γ_2 criterion or when the maximum amount of alternative solutions n was reached. To get n alternative solutions with this algorithm, we thus set γ_2 to a large value. The objective value of the last and worst solution returned by the algorithm was 198.98, which corresponded to $\gamma = 0.64\%$ relatively to the optimal score. So any value of γ_2 larger than 0.64 would have led to the same result. As above, Panel B of Figure 5 illustrates the selection frequency of these 500 alternative solutions[7]. Again, a small variability was observed, because many planning units had the same characteristics and can be interchangeable. The global visual impression was thus similar to the optimal solution exhaustive set.

***Alternative solutions within a targeted optimality gap* ($\gamma_1 > 0$, $\gamma_2 > 0$)**

We set $\gamma_1 > 0$ voluntarily to get suboptimal solutions where the relative optimality gap was at least γ_1. We chose γ_2 high enough to have $n = 100$ alternative solutions. Panel C and D in Figure 5 respectively show results for $\gamma_1 = 0.05$ and $\gamma_1 = 0.15$. Visually, we observed a greater variability, but when comparing to Figure 4, many planning units were selected while they did not increase the amount of conservation feature nor decrease the reserve perimeter. Consequently, these were planning units only useful to deteriorate the objective value and thus satisfy the optimality gap constraint. Although the variability appeared greater in Panel C and D compared to other panels, the core of the reserve was still globally similar to the optimal solutions.

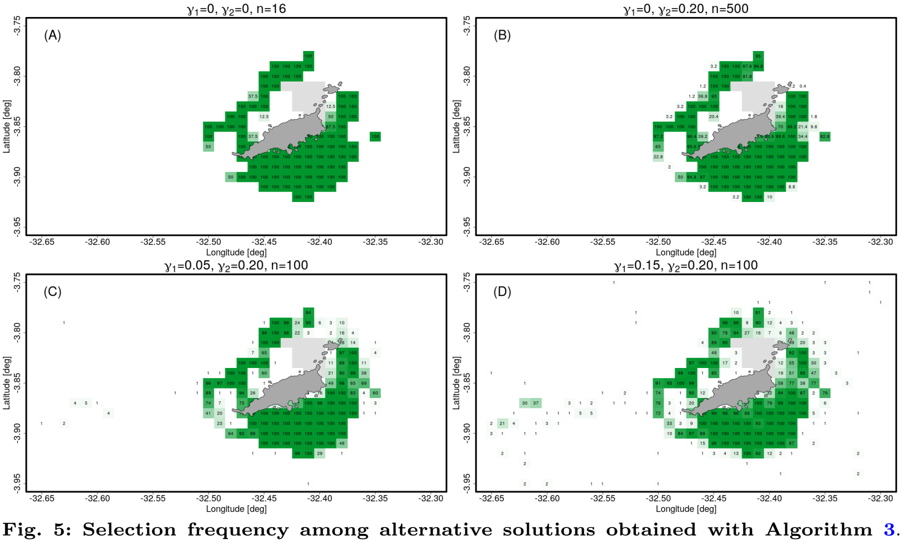


Fig. 5: Selection frequency among alternative solutions obtained with Algorithm 3. Selected planning units of alternative reserve solutions are represented with a green transparency gradient according to selection frequency expressed in percentage (black number inside planning unit).

3.3 Presentation set imposing a pseudo-distance

In this section, we applied Algorithm 1 to our case study. Therefore, we directly constrained successive alternative solutions to be pseudo-distant of at least δ

[7]among which we had 16 optimal solutions

planning units with every past iterates. Figure 6 shows four alternative examples of reserve solution found by the recursive procedure for $\delta = 20$. The optimal solution $x^\star$ is represented with planning units delimited by a thick black border. We first observed that at least 20 planning units selected in the optimal solution $x^\star$ were not found in the current solution. Those correspond to the white planning units with a thick black border. Indeed, Algorithm 1 found new planning units, because it forbade to have too many common pixels with the optimal solution. At first sight, the visual difference with the optimal solution appeared clearer than alternatives obtained with Algorithm 3. The alternative solutions proposed in Figure 6 seemed to visit different regions of the archipelago, although limited by the fact that positive conservation feature values are concentrated around the main island. In particular, in Panel B the southern region was privileged whereas north and east of Fernando de Noronha were preferred in Panel C. Panel A shows a solution similar to the optimal one, although two planning units were selected at the extreme west of the study area. Panel D displayed a solution cut into several pieces all around the main island.

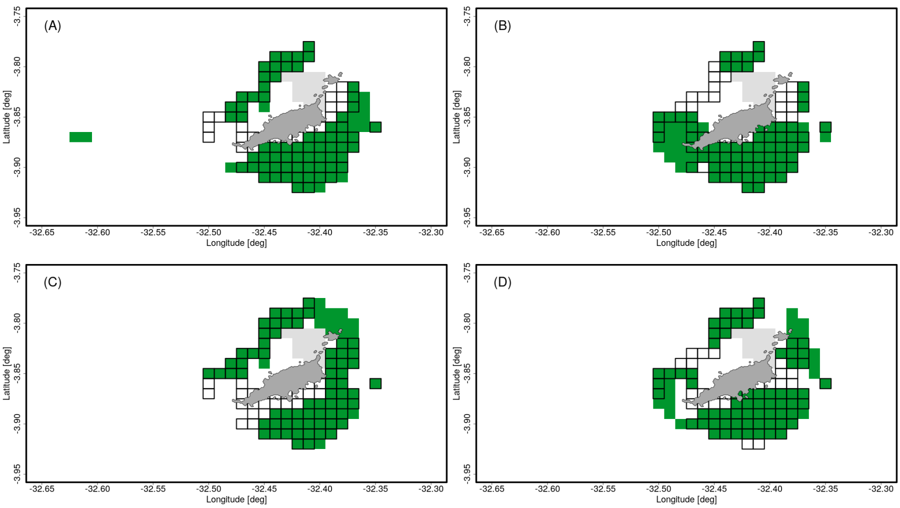


Fig. 6: Example of alternative solutions obtained with Algorithm 1 for $\delta = 20$. The alternative reserve solution is represented in green, while the optimal solution $x^\star$ is depicted with planning units delimited by a thick black border.

3.4 Presentation set maximising the minimum pseudo-distance between iterates

In this numerical application, we applied Algorithm 2 to find the $n = 4$ alternative solutions maximising the minimum pseudo-distance from past iterates. To do so, the integer linear program P_2^k was solved recursively. An example of derived reserve solutions is illustrated in Figure 7 for a relative extra cost budget $\gamma = 10\%$. Interestingly, a clear visual difference between the four alternative reserves appeared in Figure 7. The reserve in Panel A proposed a solution cut into 4 pieces, favouring the east of the archipelago. Reserve in Panel B showed a clear preference for the south of the island. Panel C was perhaps the most resembling to the optimal solution, although two planning units were found at the extreme west of the main island. Reserve in Panel D presented a reserve with two pieces, one in the north and one in the south.

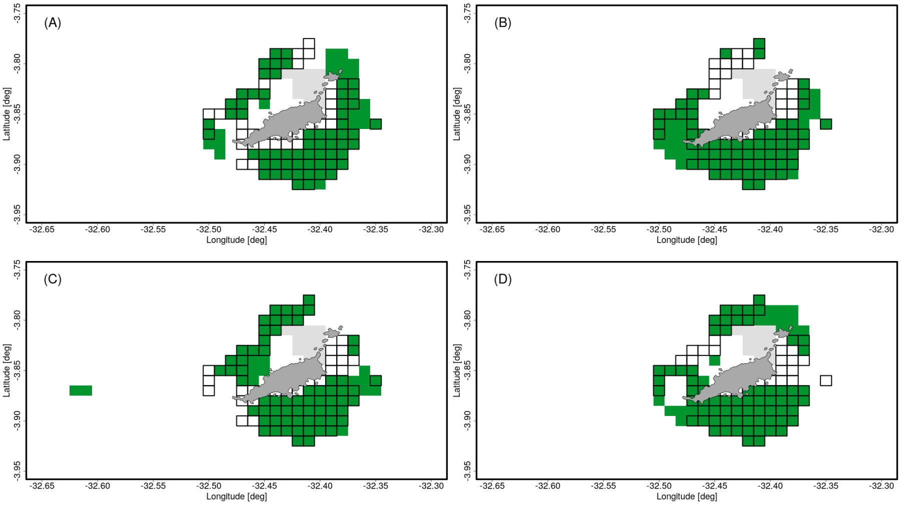


Fig. 7: Example of alternative solutions maximising the minimum pseudo-distance to past iterates at a fixed extra cost. We show four successive alternatives with an extra cost budget $\gamma = 10\%$. The alternative reserve solution is represented in green, whereas the optimal solution $x^\star$ is depicted with planning units delimited by a thick black border.

Find below more numerical details on what is realised in Algorithm 2:

- Reserve in Panel A is pseudo-distant of 27 planning units to the optimal solution $x^\star$. The first iteration simply solves P_2 which maximise the pseudo-distance to the optimal solution.
- The reserve in Panel B is pseudo-distant of 22 and 22 planning units respectively to the optimal solution and the reserve in Panel A. So the maximum minimum pseudo-distance between past iterates is 22.
- The reserve in Panel C is pseudo-distant of 22, 22 and 24 planning units respectively to the optimal solution, the reserve in Panel A and Panel B. So the maximum minimum pseudo-distance between past iterates is 22.
- The reserve in Panel D is pseudo-distant of 20, 20, 20 and 20 planning units respectively to the optimal solution, the reserve in Panel A, Panel B and Panel C. So the maximum minimum pseudo-distance between past iterates is 20.

3.5 Score versus pseudo-distance

We compared the mean relative scores and pseudo-distances of alternative solutions obtained with Algorithm 3, 1 and 2 for various values of parameters involved. More precisely, we performed a sensitivity analysis on γ_1 for Algorithm 3, δ for Algorithm 1 and γ for Algorithm 2. We observed a global increasing trend for all curves as the mean relative score was deteriorated with the pseudo-distance to the optimal solution. As expected, the highest curve was obtained with Algorithm 2 because it explicitly sought to maximise the minimum pseudo-distance to past iterates. Similarly, as pseudo-distance was not considered at all in Algorithm 3 but only the optimality gap, it was logical to observe the curve was the lowest and was not strictly increasing. Finally, the curve obtained with Algorithm 1 was in between the two others because it explicitly accounted for the pseudo-distance but did not seek to maximise it. We plotted mean relative scores and mean pseudo-distances computed on the number of alternatives, so these quantities are dependent on the number of iterations n.

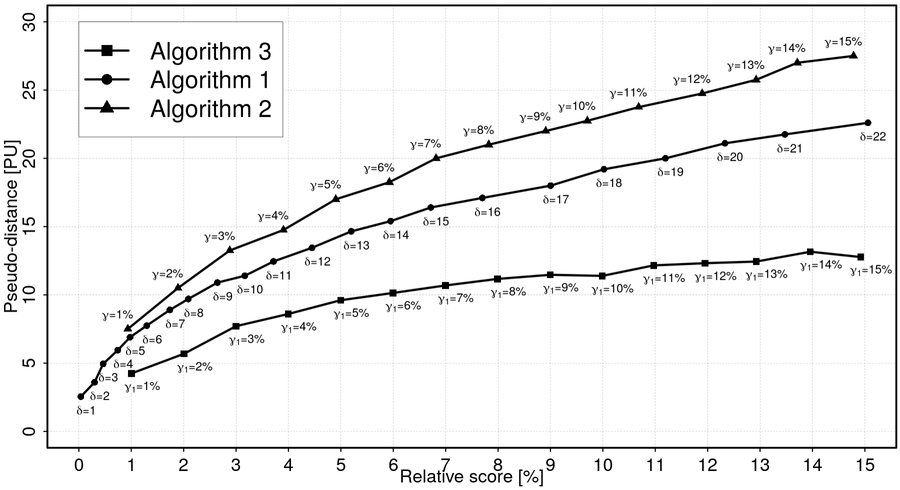


Fig. 8: Relative score γ versus pseudo-distance $d(x^\star, \cdot)$. Algorithm 3 for $\gamma_1 \in [\![1\%, 15\%]\!]$ and $n = 100$, Algorithm 1 for $\delta \in [\![1, 22]\!]$ and $n = 20$, Algorithm 2 for $\gamma \in [\![1\%, 15\%]\!]$ and $n = 4$. Relative scores and pseudo-distances are averaged on every alternative solution obtained with the considered algorithm.

4 Discussion

In this work, we demonstrated how we can produce a presentation set using exact solving methods in the reserve site selection optimisation problem. The presentation set is composed of alternative solutions that are not only different from the optimal solution but also different from each other thanks to the recursive modification of the optimisation problem. Indeed, we proposed two procedures to build a range of alternative reserve solutions according to an explicit difference criterion. We numerically illustrated the proposed methods on the real world example of Fernando de Noronha.

Our results showed that generating alternative solutions according to the optimality gap could result in a weak variability among solutions as they were very similar to each other. Indeed, these solutions which only differed by a few planning units turned out to be quite uninformative. Even worse, further from the optimal score, the variability among alternative solutions appeared irrelevant because the procedure artificially increased the objective value by including empty planning units. These planning units are pointless because they are empty relatively to the conservation features and do not contribute to decrease the reserve perimeter. Anyway, it poorly answered the conservation literature need for both *good* and *different* alternative solutions. This major limitation could also be found in the prioritizr *add_gap_portfolio* function or in the existing conservation literature [37, 38]. However, the diversity measure that we explicitly incorporate allowed to overcome this limitation. Indeed, the proposed algorithms explicitly sought to generate differences between reserve solutions which can be seen in the resulting alternatives. Similarly as in [33, 34], the diversity measure we chose allowed to avoid alternative reserves where the optimal solution is encompassed. Another pitfall, particularly striking in metaheuristic approaches, is the need to generate numerous alternative solutions in order to widely explore the solution space. This large amount of alternative solutions means a post-processing effort is needed to sort and identify a few distinct solutions. In particular, it often requires additional statistical analyses, *e.g.* the selection frequency of reserve sites or clustering analysis [26, 46]. On the contrary, our work directly provided a presentation set composed of very distinct solutions. This aimed to illustrate that a few alternatives that are both good and different from each other can be sufficient to feed a conservation planning process. In particular, no additional statistics were needed for our study case.

In summary, the strength of this work lies in the fact that only a few iterations are needed to generate a presentation set of truly different solutions. Moreover, the methods developed are highly customisable. For example, other metrics could be used in the our recursive procedures to judge the difference of solutions in the same spirit as in [35]. Indeed, this difference only depends on the definition of an arbitrary metric, this one can be adapted according to the application case. Then, another advantage of this type of approach is to be able to explicitly quantify the quality of the alternative solutions generated. Since the search for alternative solutions is carried out by exact resolution methods, this allows us to know the optimality gap which gives more control to the end user. Finally, the production of the presentation set is completely controlled by two parameters. Thus, it is the user who chooses exactly the trade-off between the difference of the alternative solutions and their optimality gap. The sensitivity analysis showing the difference of the alternative solutions relatively to their optimality gap for each algorithm is in particular a good illustration of this trade-off. Regarding the weaknesses, the proposed approaches are mostly limited by the computation time required. It can be significant on some instances and increases with the number of alternatives requested. However, the improvement of the computation time has not been the object of our attention and was not the focus of this work. In the current state of the algorithms, we can provide orders of magnitude of the computation time with a personal computer on generated instances (see Appendix) to generate 4 alternatives:

- 2-3 minutes for 500 planning units et 3 conservation features with Algorithm 2
- 10-20 seconds for 500 planning units et 3 conservation features with Algorithm 1
- 10-60 minutes for 1000 planning units et 5 conservation features with Algorithm 2
- 2-15 minutes for 1000 planning units et 5 conservation features with Algorithm 1

These computation times must be put into perspective. If we are not necessarily looking for a proof of optimality, they can be much lower. Indeed, our algorithms allow us to quickly provide interesting and feasible solutions if we decide to keep the current solution after a given maximum time. Finally, we recall that producing only 4 alternatives is a choice because they are really different alternatives that do not require additional statistical analysis.

In conclusion, although metaheuristics have been historically preferred to address the reserve site selection optimisation problem, several works demonstrated to what extent exact methods outperform metaheuristics [27, 29]. The last argument standing in favour of metaheuristics was the apparent impossibility to produce a presentation set with exact solving methods. However, our work suggested several methods showing how to build a relevant presentation set. Indeed, unlike what is commonly stated in the conservation literature [19, 26], our work showed that exact solvers used for the reserve site selection problem can also be advantageous to produce a range of alternative solutions. This ability is not inherent to metaheuristics. Besides, the inclusion of an explicit diversity criterion directly within a new optimisation model allowed to build a more controlled and transparent presentation set. Therefore, by seeking truly different solutions, we increase the chance to address objectives that are not necessarily modelled, such as socio-political or management objectives. In addition, we think the low number of alternatives needed with our methods may avoid unnecessary noise in the decision-making process. In other words, the proposed algorithms can potentially empower conservation practitioners by giving them more control over the alternatives produced and by removing the post-processing analysis usually needed. Therefore, we hope that these methods can at least shed a new light in conservation discussions and eventually bring more success in conservation decisions in practice.

Appendix A Linearised model

Parameters and variables were defined in Section 2.1. Sets of planning units *a priori* excluded or included in the reserve are respectively noted $\mathcal{LO}$ and $\mathcal{LI}$. We can linearise the quadratic term of the objective function when decision variables are binary [47]. Considering this linearisation but also locked-in and locked-out planning units, we ended up with the full mathematical optimisation problem P_0^f of reserve site selection:

$$P_0^f : \begin{cases} \min\limits_{x,z} \sum\limits_{j \in J} c_j x_j + \beta(\sum\limits_{j_1 \in J} \sum\limits_{j_2 \in J} b_{j_1 j_2} z_{j_1 j_2} + \sum\limits_{j \in J} x_j b^*_{j,N+1}) & \\ \text{s.t. } \sum\limits_{j \in J} a_{ij} x_j \geq t_i & \forall i \in I \\ \quad z_{j_1 j_2} \leq x_{j_1} & \forall j_1 \in J, \forall j_2 \in J \\ \quad z_{j_1 j_2} \leq x_{j_2} & \forall j_1 \in J, \forall j_2 \in J \\ \quad z_{j_1 j_2} \geq x_{j_1} + x_{j_2} - 1 & \forall j_1 \in J, \forall j_2 \in J \\ \quad x_j = 0 & \forall j \in \mathcal{LO} \\ \quad x_j = 1 & \forall j \in \mathcal{LI} \\ \quad x_j \in \{0,1\} & \forall j \in J \\ \quad z_{j_1 j_2} \in \{0,1\} & \forall j_1 \in J, \forall j_2 \in J \end{cases}$$

We also accounted for the correction of the compactness parameter β undesirable edge effect[8], leading to the introduction of b^* where:

$$\forall j \in J = \{1, \cdots, N\},$$

$$b^*_{j,N+1} = \begin{cases} 1, & \text{if pixel } j \text{ shares a single side with the outer boundary} \\ 2, & \text{if pixel } j \text{ shares 2 sides with the outer boundary } (\textit{i.e.} \text{ located at a corner} \\ 0, & \text{otherwise} \end{cases}$$

[8]more details in https://hal.archives-ouvertes.fr/hal-03445922

Appendix B Imposing an optimality gap

We show here how we produced the presentation set composed of alternative solutions located at a predefined optimality gap. We developed our own algorithm but the function *add_gap_portfolio* of prioritizr allows to generate the same set of alternative solutions.

B.1 Optimality gap constraint

We present a set of constraints which compels a reserve score to fall within a predefined interval. Let $\gamma_1 \in \mathbb{R}^+$ and $\gamma_2 \in \mathbb{R}^+$, $\gamma_1 \leq \gamma_2$, be the boundaries of the relative optimality gap interval. The constraints $c_l(\gamma_1)$ and $c_u(\gamma_2)$ are imposing the objective to fall within a relaxed score interval $[(1+\gamma_1)z^\star, (1+\gamma_2)z^\star]$ relatively to the optimal score $z^\star$.

$$c_l(\gamma_1) : \sum_{j \in J} c_j x_j + \beta \sum_{j_1 \in J} \sum_{j_2 \in J} b_{j_1 j_2} x_{j_1}(1 - x_{j_2}) \geq (1+\gamma_1) z^\star$$

$$c_u(\gamma_2) : \sum_{j \in J} c_j x_j + \beta \sum_{j_1 \in J} \sum_{j_2 \in J} b_{j_1 j_2} x_{j_1}(1 - x_{j_2}) \leq (1+\gamma_2) z^\star$$

Note that if $\gamma_1 = \gamma_2 = 0$, we explore only the optimal solutions set while for $\gamma_1 > 0$, we explore strict suboptimal solutions with a maximum gap equal to γ_2.

B.2 Distance constraint

We present a constraint which compels the reserve to differ from another given solution by at least δ planning units. Indeed, the constraint $c_D(\hat{x}, \delta)$ makes the solution x to have at least δ different planning units with respect to $\hat{x}$ (cf. Figure 2). Importantly, for $\delta = 1$, we are simply forbidding x and $\hat{x}$ to be strictly equal.

$$c_D(\hat{x}, \delta) : D(\hat{x}, x) = \sum_{j \in J} \hat{x}_j (1 - x_j) + x_j (1 - \hat{x}_j) \geq \delta$$

B.3 Generate the presentation set

We detailed in Algorithm 3 a simple recursive procedure which explored suboptimal alternative solutions with an a priori defined relative optimality gap interval $[\gamma_1, \gamma_2]$. If $\gamma_1 = \gamma_2 = 0$, we provided the exhaustive set of optimal solutions. First, as we wanted to explore suboptimal alternatives, we forced the objective value to fall within the user-defined relaxation threshold. Practically, we imposed the optimisation problem to respect $c_l(\gamma_1)$ and $c_u(\gamma_2)$ at the beginning. Then, to derive a pool of alternative solutions, we excluded at iteration k the solution x^{k-1} derived at the previous iteration. For this, at each iteration k, we added the constraint $c_D(x^{k-1}, 1)$ which forbade the exact similarity with the solution x^{k-1} found at the previous iteration.

As we recursively excluded solutions from best to worst, the procedure would stop whether the objective function value exceeded the upper bound γ_2 or the maximum number of iteration n was reached. Importantly, if we set γ_2 high enough, the stopping criterion met would be the maximum number of iteration, which was what we generally wanted. For instance, if $\gamma_1 = 0$ and γ_2 were high enough, Algorithm 3 returned the n solutions with the smallest score. If n was chosen high enough, Algorithm 3 returned the exhaustive set of solutions with a relative optimality gap within $[\gamma_1, \gamma_2]$. Unlike metaheuristics where the optimality gap is unknown, we a priori established it thanks to this algorithm, thus offering users more control over the flexibility provided.

Practically, the integer linear program P_3^k solved at iteration k is the following:

$$P_3^k : \begin{cases} \min\limits_{x} \ \sum\limits_{j\in J} c_j x_j + \beta \sum\limits_{j_1\in J} \sum\limits_{j_2\in J} b_{j_1 j_2} x_{j_1}(1 - x_{j_2}) & \\ \text{s.t.} \ \sum\limits_{j\in J} c_j x_j + \beta \sum\limits_{j_1\in J} \sum\limits_{j_2\in J} b_{j_1 j_2} x_{j_1}(1 - x_{j_2}) \leq (1+\gamma_2) z^\star & \\ \qquad \sum\limits_{j\in J} c_j x_j + \beta \sum\limits_{j_1\in J} \sum\limits_{j_2\in J} b_{j_1 j_2} x_{j_1}(1 - x_{j_2}) \geq (1+\gamma_1) z^\star & \\ \qquad \sum\limits_{j\in J} a_{ij} x_j \geq t_i & \forall i \in I \\ \qquad \sum\limits_{j\in J} x_j(1 - x_j^l) + x_j^l(1 - x_j) \geq 1 & \forall l \in [\![0, k-1]\!] \\ \qquad x_j \in \{0,1\} & \forall j \in J \end{cases}$$

The corresponding algorithm can be found below:

Algorithm 3 Recursive search of n best alternative solutions with a relative optimality gap in $[\gamma_1, \gamma_2]$ relatively to score $z^\star$ of solution $x^\star$ of problem P_0.

Require: P_0, $x^\star$, $z^\star$, n, γ_1, γ_2
Ensure: x^1, ..., x^k

```
1: k ← 0; P ← P0; x^0 ← x*                                    ▷ initialisation
2: add c_l(γ1) and c_u(γ2) to P
3: while P is feasible & k < n do          ▷ stop when infeasible or enough solutions
4:     k ← k + 1
5:     add c_D(x^{k−1}, 1) to P                        ▷ exclude previous solution
6:     solve P      ▷ get an optimal solution x^k or detect that P is infeasible
7: end while
```

Appendix C The presentation set computed on generated data

For testing purposes, we developed a systematic way of building user-defined scenarios for reserve site selection optimisation problems. The idea is to provide the conservation literature tools to facilitate benchmarks of developed methods in conservation planning. Therefore, the main ambition is to generate realistic discrete spatial distributions of the considered conservation features.

C.1 Data generation

Technically speaking, we arbitrarily chose to compute the amount a_{ij} of a conservation feature $i \in I$ in a planning unit $j \in J$ by randomly drawing this value in a Gaussian distribution.

$$a_{ij} \sim \mathcal{N}(m_{ij}, \sigma_{ij}^2)$$

The mean value m_{ij} of the Gaussian distribution only depends on the distance d_{ij} to the closest (chosen or randomly drawn) N_{epi} epicentres associated to the conservation feature $i \in I$. To be more precise, the mean value m_{ij} depends on $d_{ij}^{\alpha_i}$, where α_i is a predefined parameter for each conservation feature $i \in I$ controlling the dispersion of the mean values distribution relatively to the epicentres.

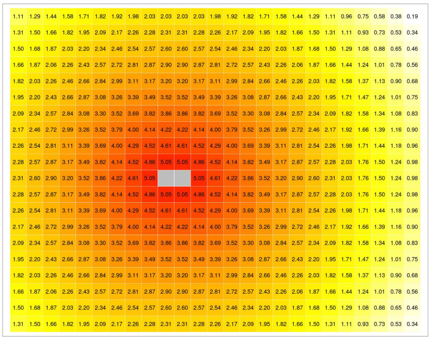
$$m_{ij} = \mu_i \left[1 - \left(\frac{d_{ij}}{d_{max}} \right)^{\alpha_i} \right]$$

The maximum mean value, *i.e.* the mean value at the epicentres, is a chosen parameter μ_i for each conservation feature $i \in I$. If no epicentres are provided, the mean value of the Gaussian distribution depends on the distance to the locked-out planning units supposed to represent a shoreline. The standard deviation σ_{ij} of the Gaussian distribution is such as $\sigma_{ij} = \sigma_i m_{ij}$ where σ_i is a chosen parameter for each conservation feature $i \in I$. The code used to generate data is available in open access[9]. The instance is characterised by the rectangular grid size N_x and N_y and the number of conservation features N_{cf}.

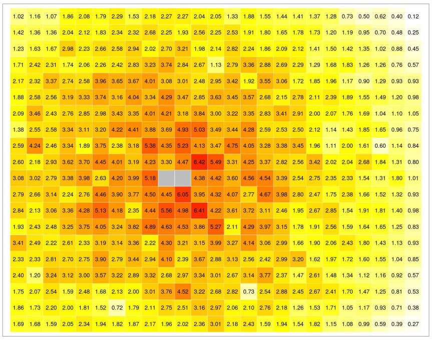
C.2 Scenarios

We build several scenarios to have some order of magnitudes for the computation time of the algorithms proposed in this work. We show the generated spatial distributions of two conservation features resulting from the data generation procedure in Figure C1. An example of a presentation set is given in Figure C2.

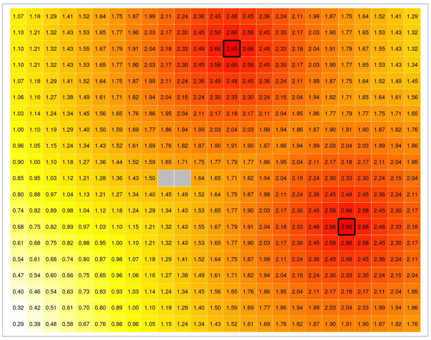
[9] https://github.com/AdrienBrunel/data_generation



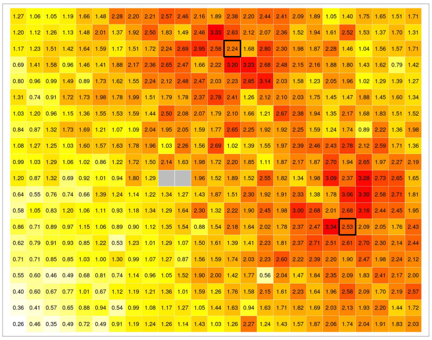
(a) Spatial distribution of the mean value m_{ij} when epicentres correspond to the locked-out planning units. The maximum mean value is 5.7.



(b) Random drawing from the Gaussian distribution, where the mean values are shown in Panel C1a. Dispersion coefficient is $\alpha_i = 0.75$.

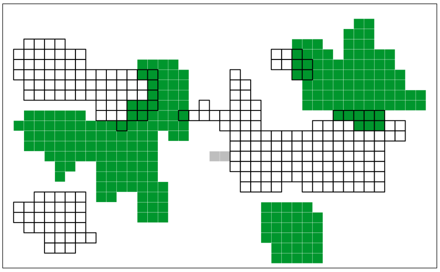


(c) Spatial distribution of the mean value m_{ij} where 2 epicentres are randomly drawn among planning units. The maximum mean value is 2.9.

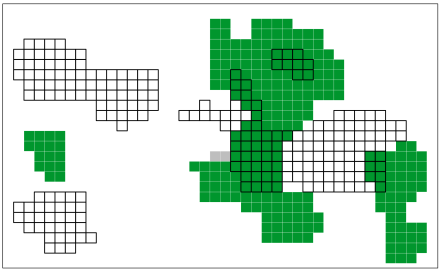


(d) Random drawing from the Gaussian distribution where the mean values are shown in Panel C1c. Dispersion coefficient is $\alpha_i = 0.78$.

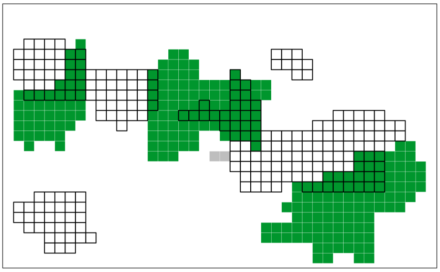
Fig. C1: Example of the generated spatial distribution for two different conservation features in a 25×20 rectangular grid. The amounts of considered conservation feature are shown with a yellow to red gradient. The corresponding numerical values are written in black inside the planning units. Locked-out planning units are represented in grey. We chose $\sigma_i = 0.20$.



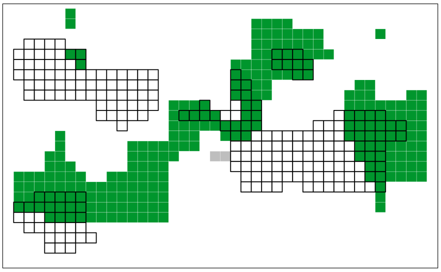
(a) Alternative reserve solution found at iteration 1 with Algorithm 2.



(b) Alternative reserve solution found at iteration 2 with Algorithm 2.



(c) Alternative reserve solution found at iteration 3 with Algorithm 2.



(d) Alternative reserve solution found at iteration 4 with Algorithm 2.

Fig. C2: Presentation set computed with Algorithm 2. The considered scenario was made of 40×25 planning units and 5 conservation features. We chose an extra cost budget of $\gamma = 0.10$. Relative targets for every conservation features were set to 25%. Green planning units represents the alternative reserve solution. Planning units with a black border indicates the initial optimal solution.

4 Author Declarations

4.1 Authors' contributions

A.B. and J.O. authors contributed to the study conception and design. S.L-B. was involved in material preparation, data collection, funding acquisition and project administration. A.B. performed the formal analyses and wrote the first draft of the manuscript. All authors read and approved the final manuscript.

4.2 Competing interests

The authors declare they have no competing interests.

4.3 Funding

A.B. is funded by Région Occitanie and Créocean. This work is a contribution to the *Planning in a Liquid World with Tropical Stakes* (PADDLE) Rise project (European Union's Horizon 2020 research and innovation program under Grant Agreement No. 734271) and to the *Tropical Atlantic Interdisciplinary laboratory on physical, biogeochemical, ecological and human dynamics International Laboratory* (LMI TAPIOCA). This project has received support from the TRIATLAS project (funding by the European Union's Horizon 2020 research and innovation program Grant agreement No. 817578).

4.4 Consent to Participate

Not applicable.

4.5 Ethics Declaration statement

Not applicable.

4.6 Consent for publication

Not applicable.

4.7 Availability of data and material/ Data availability

The datasets generated and analysed during the current study are available at the following GitHub repository : https://github.com/AdrienBrunel/data_generation.

References

[1] IPBES: Global assessment report on biodiversity and ecosystem services of the Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services. Technical report, Zenodo (May 2019). https://doi.org/10.5281/ZENODO.3831673. Version Number: 1. https://zenodo.org/record/3831673 Accessed 2022-04-08

[2] Ticco, P.C.: The use of marine protected areas to preserve and Enhance marine biological diversity: A case study approach. Coastal Management **23**(4), 309–314 (1995). https://doi.org/10.1080/08920759509362274. Accessed 2022-05-05

[3] Tundi Agardy, M.: Advances in marine conservation: the role of marine protected areas. Trends in Ecology & Evolution **9**(7), 267–270 (1994). https://doi.org/10.1016/0169-5347(94)90297-6. Accessed 2022-05-05

[4] Le Saout, S., Hoffmann, M., Shi, Y., Hughes, A., Bernard, C., Brooks, T.M., Bertzky, B., Butchart, S.H.M., Stuart, S.N., Badman, T., Rodrigues, A.S.L.: Protected Areas and Effective Biodiversity Conservation. Science **342**(6160), 803–805 (2013). https://doi.org/10.1126/science.1239268. Accessed 2022-05-05

[5] Claudet, J., Osenberg, C.W., Benedetti-Cecchi, L., Domenici, P., García-Charton, J.-A., Péerez-Ruzafa, A., Badalamenti, F., Bayle-Sempere, J., Brito, A., Bulleri, F., Culioli, J.-M., Dimech, M., Falcón, J.M., Guala, I., Milazzo, M., Sánchez-Meca, J., Somerfield, P.J., Stobart, B., Vandeperre, F., Valle, C., Planes, S.: Marine reserves: size and age do matter. Ecology Letters **11**(5), 481–489 (2008). https://doi.org/10.1111/j.1461-0248.2008.01166.x. Accessed 2021-03-10

[6] Stolton, S., Dudley, N.: Arguments for Protected Areas: Multiple Benefits for Conservation Use, (2010)

[7] Liu, P., Jiang, S., Zhao, L., Li, Y., Zhang, P., Zhang, L.: What are the benefits of strictly protected nature reserves? Rapid assessment of ecosystem service values in Wanglang Nature Reserve, China. Ecosystem Services **26**, 70–78 (2017). https://doi.org/10.1016/j.ecoser.2017.05.014. Accessed 2021-03-10

[8] IUCN: IUCN World Parks Congress 2014 Bulletin. International Institute for Sustainable Development **89**(16), 43 (2014)

[9] IUCN: IUCN Congress 2016 Bulletin **89**(16), 43 (2016)

[10] Commission, E.: EU Biodiversity Strategy for 2030, Bringing nature into our lives, COM(2020) 380 finals,Brussels, 20.05.2020. Technical report

(2020)

[11] Baillie, J., Zhang, Y.-P.: Space for nature. Science **361**(6407), 1051–1051 (2018). https://doi.org/10.1126/science.aau1397. Accessed 2022-05-06

[12] Dinerstein, E., Vynne, C., Sala, E., Joshi, A.R., Fernando, S., Lovejoy, T.E., Mayorga, J., Olson, D., Asner, G.P., Baillie, J.E.M., Burgess, N.D., Burkart, K., Noss, R.F., Zhang, Y.P., Baccini, A., Birch, T., Hahn, N., Joppa, L.N., Wikramanayake, E.: A Global Deal For Nature: Guiding principles, milestones, and targets. Science Advances **5**(4), 2869 (2019). https://doi.org/10.1126/sciadv.aaw2869. Accessed 2022-05-06

[13] Margules, C., Pressey, R.L.: Systematic conservation planning. Nature **405**, 243–253 (2000)

[14] Moilanen, A., Wilson, K.A., Possingham, H.P. (eds.): Spatial Conservation Prioritization: Quantitative Methods and Computational Tools. Oxford biology. Oxford University Press, Oxford ; New York (2009). OCLC: ocn277205285

[15] Ando, A., Camm, J., Polasky, S., Solow, A.: Species Distributions, Land Values, and Efficient Conservation. Science **279**(5359), 2126–2128 (1998)

[16] Stewart, R.R., Possingham, H.P.: Efficiency, costs and trade-offs in marine reserve system design. Environmental Modeling & Assessment **10**(3), 203–213 (2005). https://doi.org/10.1007/s10666-005-9001-y. Accessed 2022-05-05

[17] Hanson, J.O., Rhodes, J.R., Butchart, S.H.M., Buchanan, G.M., Rondinini, C., Ficetola, G.F., Fuller, R.A.: Global conservation of species' niches. Nature **580**(7802), 232–234 (2020). https://doi.org/10.1038/s41586-020-2138-7. Accessed 2022-05-02

[18] Pressey, R.L., Humphries, C.J., Margules, C.R., Vane-Wright, R.I., Williams, P.: Beyond Opportunism: Key Principles for Systematic Reserve Selection **8**(4), 5 (1993)

[19] Possingham, H., Ball, I., Andelman, S.: Mathematical Methods for Identifying Representative Reserve Networks. In: Quantitative Methods for Conservation Biology, pp. 291–306. Springer, New York (2000). https://doi.org/10.1007/0-387-22648-6_17. http://link.springer.com/10.1007/0-387-22648-6_17 Accessed 2021-05-04

[20] Possingham, H.P., Wilson, K.A., Andelman, S., Vynne, C.H.: Protected areas: Goals, limitations, and design. In: Principles of Conservation Biology, 3rd edn., pp. 507–549 (2006)

[21] Sarkar, S.: Complementarity and the selection of nature reserves: algorithms and the origins of conservation planning, 1980–1995. Archive for History of Exact Sciences **66**(4), 397–426 (2012). https://doi.org/10.1007/s00407-012-0097-6. Accessed 2021-05-04

[22] Ferretti, V., Pluchinotta, I., Tsoukiàs, A.: Studying the generation of alternatives in public policy making processes. European Journal of Operational Research **273**(1), 353–363 (2019). https://doi.org/10.1016/j.ejor.2018.07.054. Accessed 2022-03-18

[23] Ball, I., Possingham, H.: Marine Reserve Design using Spatially Explicit Annealing, 70 (2000)

[24] Ball, I.R., Possingham, H.P., Watts, M.E.: Marxan and relatives: software for spatial conservation prioritisation. In: Spatial Conservation Prioritisation: Quantitative Methods and Computational Tools vol. 14, pp. 185–196 (2009)

[25] Hanson, J.O., Schuster, R., Morrel, N., Strimas-Mackey, M., Watts, M.E., Arcese, P., Bennett, J., Possingham, H.P.: prioritizr: Systematic Conservation Prioritization in R. (2020). https://CRAN.R-project.org/package=prioritizr

[26] Ardron, J.A., Possingham, H.P., Klein, C.J.: Marxan Good Practices Handbook, (2010)

[27] Schuster, R., Hanson, J.O., Strimas-Mackey, M., Bennett, J.R.: Exact integer linear programming solvers outperform simulated annealing for solving conservation planning problems. PeerJ **8**, 9258 (2020). https://doi.org/10.7717/peerj.9258. Accessed 2021-05-04

[28] Hanson, J.O., Schuster, R., Strimas-Mackey, M., Bennett, J.R.: Optimality in prioritizing conservation projects. Methods in Ecology and Evolution **10**(10), 1655–1663 (2019). https://doi.org/10.1111/2041-210X.13264. Accessed 2021-05-04

[29] Beyer, H.L., Dujardin, Y., Watts, M.E., Possingham, H.P.: Solving conservation planning problems with integer linear programming. Ecological Modelling **328**, 14–22 (2016). https://doi.org/10.1016/j.ecolmodel.2016.02.005. Accessed 2021-05-04

[30] Rodrigues, A.S.L., Gaston, K.J.: Optimisation in reserve selection procedures—why not? Biological Conservation **107**(1), 123–129 (2002). https://doi.org/10.1016/S0006-3207(02)00042-3. Accessed 2021-09-21

[31] Serra-Sogas, N., Kockel, A., Game, E.T., Possingham, H., McGowan, J.: Marxan User Manual: For Marxan version 2.43 and above. The Nature

Conservancy (TNC), Arlington, Virginia, United States and Pacific Marine Analysis and Research Association (PacMARA), Victoria, British Columbia, Canada (2020)

[32] Chang, S.-Y., Brill, E.D., Hopkins, L.D.: Efficient random generation of feasible alternatives: a land use example. Journal of Regional Science **22**(3), 303–314 (1982). https://doi.org/10.1111/j.1467-9787.1982.tb00754.x. Accessed 2022-04-08

[33] Brill, E.D., Chang, S.-Y., Hopkins, L.D.: Modeling to Generate Alternatives: The HSJ Approach and an Illustration Using a Problem in Land Use Planning. Management Science **28**(3), 221–235 (1982). https://doi.org/10.1287/mnsc.28.3.221. Accessed 2022-04-08

[34] Chang, S.-Y., Brill, E.D., Hopkins, L.D.: Use of mathematical models to generate alternative solutions to water resources planning problems. Water Resources Research **18**(1), 58–64 (1982). https://doi.org/10.1029/WR018i001p00058. Accessed 2022-04-08

[35] Makowski, D., Hendrix, E.M.T., van Ittersum, M.K., Rossing, W.A.H.: A framework to study nearly optimal solutions of linear programming models developed for agricultural land use exploration. Ecological Modelling **131**(1), 65–77 (2000). https://doi.org/10.1016/S0304-3800(00)00249-0. Accessed 2022-04-08

[36] Fischer, D.T., Church, R.L.: The SITES reserve selection system: A critical review. Environmental Modeling & Assessment **10**(3), 215–228 (2005). https://doi.org/10.1007/s10666-005-9005-7. Accessed 2022-04-08

[37] Arthur, J.L., Hachey, M., Sahr, K.: Finding all optimal solutions to the reserve site selection problem: formulation and computational analysis. Environmental and Ecological Statistics, 13 (1997)

[38] Önal, H.: First-best, second-best, and heuristic solutions in conservation reserve site selection. Biological Conservation **115**(1), 55–62 (2004). https://doi.org/10.1016/S0006-3207(03)00093-4. Accessed 2021-03-23

[39] Bezanson, J., Karpinski, S., Shah, V.B., Edelman, A.: Julia: A Fast Dynamic Language for Technical Computing. arXiv:1209.5145 [cs] (2012). arXiv: 1209.5145. Accessed 2021-03-16

[40] Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A Fresh Approach to Numerical Computing. arXiv:1411.1607 [cs] (2015). arXiv: 1411.1607. Accessed 2021-03-02

[41] Dunning, I., Huchette, J., Lubin, M.: JuMP: A Modeling Language for Mathematical Optimization. SIAM Review **59**(2), 295–320 (2017). https:

//doi.org/10.1137/15M1020575. arXiv: 1508.01982. Accessed 2021-03-16

[42] Forrest, J., Ralphs, T., Vigerske, S., LouHafer, Kristjansson, B., Jpfasano, EdwinStraver, Lubin, M., Santos, H.G., Rlougee, Saltzman, M.: Coin-Or/Cbc: Version 2.9.9. Zenodo (2018). https://doi.org/10.5281/ZENODO.1317566. https://zenodo.org/record/1317566 Accessed 2021-05-04

[43] Rodrigues, A.S., Orestes Cerdeira, J., Gaston, K.J.: Flexibility, efficiency, and accountability: adapting reserve selection algorithms to more complex conservation problems. Ecography **23**(5), 565–574 (2000). https://doi.org/10.1111/j.1600-0587.2000.tb00175.x. Accessed 2021-10-27

[44] Cocks, K.D., Baird, I.A.: Using mathematical programming to address the multiple reserve selection problem: An example from the Eyre Peninsula, South Australia. Biological Conservation **49**(2), 113–130 (1989). https://doi.org/10.1016/0006-3207(89)90083-9. Accessed 2021-05-04

[45] Garey, M.R., Johnson, D.S.: Computers and Intractability: a Guide to the Theory of NP-completeness, 27. print edn. A series of books in the mathematical sciences. Freeman, New York [u.a] (1979)

[46] Harris, L.R., Watts, M.E., Nel, R., Schoeman, D.S., Possingham, H.P.: Using multivariate statistics to explore trade-offs among spatial planning scenarios. Journal of Applied Ecology **51**(6), 1504–1514 (2014). https://doi.org/10.1111/1365-2664.12345. Accessed 2022-05-04

[47] Billionnet, A.: Optimisation Discrète, de la Modélisation à la Résolution Par des Logiciels de Programmation Mathématique, Dunod edn. (2007). https://hal.archives-ouvertes.fr/hal-01125300