



**HAL**  
open science

# Producing a diverse set of near-optimal reserve solutions with exact optimisation

Adrien Brunel, Jérémy Omer, Sophie Lanco Bertrand

► **To cite this version:**

Adrien Brunel, Jérémy Omer, Sophie Lanco Bertrand. Producing a diverse set of near-optimal reserve solutions with exact optimisation. *Environmental Modeling & Assessment*, 2023, 28 (4), pp.619-634. 10.1007/s10666-022-09862-1 . hal-03519381v3

**HAL Id: hal-03519381**

**<https://hal.science/hal-03519381v3>**

Submitted on 6 Dec 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Producing a diverse set of near-optimal reserve solutions with exact optimisation

Adrien Brunel<sup>1\*</sup>, Jérémy Omer<sup>2</sup> and Sophie Lanco Bertrand<sup>1</sup>

<sup>1</sup>IRD, MARBEC (Univ. Montpellier, CNRS, Ifremer, IRD),  
Avenue Jean Monnet, Sète, 34203, France.

<sup>2</sup>Univ Rennes, INSA Rennes, CNRS, IRMAR, UMR 6625,  
Rennes, 35000, France.

\*Corresponding author(s). E-mail(s):

[adrien.brunel.pro@protonmail.com](mailto:adrien.brunel.pro@protonmail.com);

Contributing authors: [jeremy.omer@insa-rennes.fr](mailto:jeremy.omer@insa-rennes.fr);  
[sophie.bertrand@ird.fr](mailto:sophie.bertrand@ird.fr);

## Abstract

Reserves are at the heart of global policies to stop the erosion of biodiversity. Optimisation is increasingly used to identify reserve locations that preserve biodiversity at a minimum cost to human activities. Two classes of algorithms solve this reserve site selection problem: metaheuristic algorithms (such as simulated annealing, commonly implemented in Marxan) and exact optimisation (*i.e.* integer programming, commonly implemented in Prioritizr). Although exact approaches are now able to solve large-scale problems, metaheuristics are still widely used. One reason is that metaheuristic-based software provides a set of suboptimal reserve solutions instead of a single one. These alternative solutions are usually welcomed by stakeholders as they provide a better basis for negotiations among potentially conflictive objectives. Metaheuristic algorithms use random procedures to explore the space of suboptimal reserve solutions. Therefore, they may produce a large amount of similar, thus uninformative, alternative solutions, which usually calls for a heavy statistical post-processing. Effective methods for generating a diverse set of near-optimal solutions using exact optimisation are lacking. Here we present two new approaches for addressing this issue. Our algorithms explicitly control both the optimality gap and the dissimilarity between alternative reserve solutions. It allows the identification

2 *Environmental Modeling & Assessment*

of a parsimonious, yet meaningful set of reserve solutions. The algorithms presented here could potentially increase the uptake of exact optimisation by practitioners. These methods should contribute to less noisy and more efficient discussions in the design of conservation policies.

**Keywords:** optimal reserve site selection; presentation set; portfolio; protected areas; conservation planning; integer linear programming.

# 1 Introduction

Biodiversity and habitats are threatened worldwide [1]. Building comprehensive networks of nature reserves has become a popular conservation solution [2–4] and was shown to bring significant benefits [5–7]. At sea, for instance, current political objectives are to cover 30% of the marine spaces under jurisdiction by 2030 with marine protected areas [8–10]. Similar concerns also exist on land [11, 12]. Several methods (often embedded within a decision support tool software, *e.g.* Marxan or PrioritizR) select reserve sites given constraints and objectives.

A set of alternative reserve solutions is usually necessary to create effective nature reserves as they support better negotiations between different stakeholders. In a decision-making framework, constraints and objectives of reserve design may be difficult to formulate in the context of a spatially-explicit numerical optimisation. Any unmodeled phenomenon that may influence the decision can lead to divergences with the proposed solutions. For example, complex social mechanisms govern the final decision of wind farm locations [13, 14]. The decision process, based eventually on negotiations, thus requires some latitude on the possible solutions to be considered. Generating alternative solutions gives conservation practitioners the possibility of finding a solution that could be more satisfactory with respect to these 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 [15–19]. Consequently, reserve site selection tools do need options for generating different near-optimal alternatives.

The ability to produce alternative solutions has often been presented as a key strength of metaheuristic algorithms over exact optimisation approaches [15–18]. In a survey realised among Marxan users [20], "generating multiple solutions was by far the most commonly noted strength of Marxan" over other reserve site selection algorithms. Yet, recent advances made exact optimisation methods more attractive for conservation practitioners because they provide the optimal solution even for large-scale instances in reasonable time [21–24]. The ability of generating multiple solutions thus seems to be the last argument remaining in favour of metaheuristic algorithms. Technically speaking, metaheuristics rely on random processes to create an implicit diversity within the set of solutions (see Appendix B-2.1 in [25]). On the contrary, exact solving methods produce, by construction, a single optimal solution and are not designed for producing a range of different solutions. This major limitation severely restricts the ability of exact solving methods to inform real-world conservation problems. However, in the absence of explicit criteria, metaheuristic approaches do not provide any control over the alternatives generated, nor do they guarantee to have truly different solutions. The search for near-optimal alternative solutions with exact solving methods began to be discussed in [26] and was explored in the general context of operational

research [27–30]. The algorithm developed in [31] computes the exhaustive set of optimal solutions of a reserve site selection problem. In the same line, a branch and bound screening algorithm showed how suboptimal solutions can be derived with exact methods [32]. The reserve site selection tool *PrioritizR* also provides additional functions allowing users to build a portfolio of alternative solutions<sup>1</sup>. Though, producing alternative solutions based on their distance to optimality with exact solving methods still does not guarantee to provide different solutions. This motivated us to explicitly introduce a dissimilarity measure in the search for alternative solutions.

In this work, we propose two recursive algorithms incorporating an explicit dissimilarity criterion to build a range of near-optimal solutions significantly different from each other with exact solving methods. Solutions are selected based on a controlled objective value degradation and using an explicit dissimilarity measure. We discuss and compare two metrics for the dissimilarity criterion. Our results show that generating alternative solutions according to the objective value interval can result in a low variability among solutions, as they are very similar to each other. These solutions which only differ by a few planning units are quite uninformative. They can hardly be considered as alternatives. Even worse, further from the optimal value, the variability among alternative solutions appears irrelevant because the procedure artificially increases the objective value by including pointless planning units. As such, it poorly answers the need for both good and different alternative solutions. The dissimilarity measure we incorporate allows us to overcome this limitation. The proposed algorithms explicitly seek to generate dissimilarity between reserve solutions and provide true alternatives. Similarly to [28, 29], the dissimilarity measure we define allows to avoid alternative reserves embedding the optimal one. 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 requires a statistical post-processing to identify a few distinct solutions. It often requires additional statistical analyses, *e.g.* the selection frequency of reserve sites or clustering analysis [20, 33, 34]. By contrast, our methods directly provide a presentation set composed of significantly distinct solutions. A few alternatives that are both good and different from each other can thus be sufficient.

At this point, it is necessary to precise the terminology chosen to dissipate any ambiguity about the generation of alternative solutions of an optimisation problem. Indeed, this concept is covered by different terminologies in conservation biology literature. In this work, we used "presentation set" to name the collection of alternative solutions of an optimisation problem. This term makes explicit the fact that these alternative solutions are intended to be presented to decision-makers and decided upon.

---

<sup>1</sup>More details can be found at <https://prioritizr.net/reference/portfolios.html>

## 2 Methods

In this section, we first present the general formulation of the reserve site selection problem. Then, we present the dissimilarity measure we used for quantifying the differences between two reserve solutions. Finally, we provide the mathematical formulation of new approaches incorporating our dissimilarity measure for generating presentation sets.

### 2.1 General formulation of a reserve site selection problem

In a reserve site selection problem, the study area is discretised into a set  $J$  of planning units within which a set  $I$  of conservation features are distributed. The amount of conservation feature  $i \in I$  within the planning unit  $j \in J$  is denoted  $a_{ij}$ . Each planning unit  $j \in J$  has a cost  $c_j$  representing the socio-economic cost of closing this unit. One then seeks to find the least cost collection of planning units covering a sufficient amount of each conservation features. The coverage of the conservation feature  $i$  is considered sufficient if it exceeds a user-defined target  $t_i$ . The decision is about whether to include the planning unit in the reserve. Consequently, we associate the decision variables  $x_j$  with each planning unit  $j$ :  $x_j = 1$  if a planning unit  $j$  belongs to the reserve and  $x_j = 0$  otherwise. The reserve compactness is the only spatial attribute incorporated in the model. Other spatial concerns, such as contiguity, minimum fragmentation, buffer zones, corridors, *etc.*, may govern the site selection but are not considered here. Since a small perimeter involves a compact reserve, the reserve perimeter is included in 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 units  $j_1$  and  $j_2$  is denoted  $b_{j_1 j_2}$ . A multiplier, noted  $\beta$ , is used within the objective function to reflect the importance of compactness relatively to the total cost of site selection. Mathematically speaking, the general problem of reserve site selection is expressed as the following combinatorial optimisation problem  $P_0$ :

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

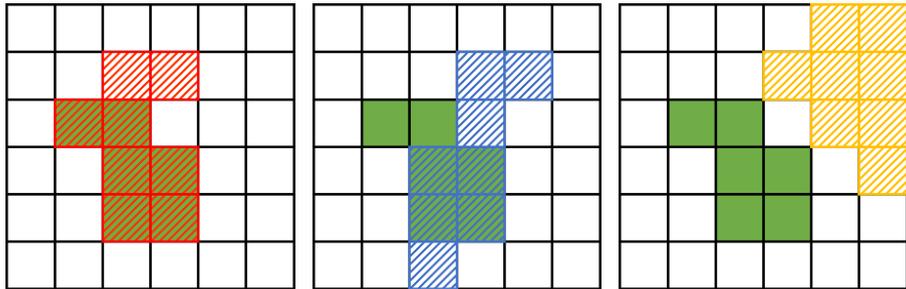
This problem is a generalization of the minimum set cover problem, which is known to be NP-hard [35]. It is a non-convex problem due to the binary nature of the decision variables. Yet, it can be expressed as an integer linear program (see Appendix A for the linearised model) and available solvers (*e.g.* Gurobi or Cbc) can solve realistic instances in a reasonable time. In the following, we respectively denote  $x^*$  and  $z^*$  the optimal solution and the associated objective value of  $P_0$ .

## 2.2 Measuring the dissimilarity between two reserve solutions

For providing a diverse presentation set, we first need a function characterising the dissimilarity between two solutions. Consequently, when  $x, y \in \{0, 1\}^N$ , we defined the following dissimilarity measure:

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

Dissimilarity measure  $d$  counts the number of planning units selected in  $x$  and not in  $y$ . This measure is rather a pseudo-distance than a distance, because it does not meet the separation property. Indeed,  $d(x, y) = 0$  does not imply that  $x = y$ . Instead,  $d(x, y) = 0$  whenever the reserve defined by  $x$  is included in that defined by  $y$ . This is actually an enjoyable property for the production of alternative solutions, because there is no real point in presenting an alternative solution that would strictly include an optimal solution  $x^*$ . As an illustration, Figure 1 displays three examples where the dissimilarity measure  $d$  is compared with the distance  $D(x, y) = d(x, y) + d(y, x)$ , which is equal to 0 only if  $x = y$ . For instance, on the leftmost example, we see that the red solution is simply a worse alternative to the green solution, so we wanted to avoid this case. This type of measure was proposed in the context of land-use planning through the 4 steps of the "Hop Skip and Jump" procedure [28].



(a) The green reserve is strictly included in the red reserve. We have  $d(x, y) = 0$  and  $D(x, y) = 2$ .

(b) The blue reserve has planning units both inside and outside the green reserve. We have  $d(x, y) = 2$  and  $D(x, y) = 6$ .

(c) The green reserve has an empty intersection with the yellow reserve. We have  $d(x, y) = 6$  and  $D(x, y) = 14$ .

**Fig. 1: Numerical examples of the dissimilarity measure  $d$  and distance  $D$ .** The reserve  $x$  depicted in green includes 6 planning units. Other reserves  $y$ , hatched in red, blue and yellow, include 8 planning units. According to the dissimilarity measure  $d$ , the green and red reserve are the same. The dissimilarity measure characterises differences between two reserves as much as distance  $D$  as illustrated with the blue and yellow reserve examples.

## 2.3 Computing a presentation set

In this section, we describe our two methods that compute a presentation set using the dissimilarity measure  $d$ . In the following, the approach proposed by the `add_gap_portfolio` function<sup>2</sup> of `PrioritizR` is referred to as the Algorithm `AddGapPortfolio`. This method produces alternative solutions by enforcing a given optimality gap. This algorithm is described in Appendix B.

### 2.3.1 Imposing dissimilarity between alternative solutions

For some predefined dissimilarity threshold  $\delta$  and  $k$  feasible solutions  $x^0, \dots, x^{k-1}$  of  $P_0$ , we may impose that a new alternative solution  $x$  differs sufficiently from  $x^l$  for every  $l \in \llbracket 0, k-1 \rrbracket$  by constraining  $d(x^l, x)$  to be at least equal to  $\delta$ . More formally, this can be achieved by adding the following constraints to the initial optimisation problem  $P_0$ :

$$c_d(x^l, \delta) : d(x^l, x) = \sum_{j \in J} x_j^l (1 - x_j) \geq \delta, \quad \forall l \in \llbracket 0, k-1 \rrbracket.$$

The integer linear program solved at iteration  $k \geq 1$  is  $P_1^k$  such as:

$$P_1^k : \begin{cases} \min_x & \sum_{j \in J} c_j x_j + \beta \sum_{j_1 \in J} \sum_{j_2 \in J} b_{j_1 j_2} x_{j_1} (1 - x_{j_2}) \\ \text{s.t.} & \sum_{j \in J} a_{ij} x_j \geq t_i & \forall i \in I \\ & d(x^l, x) \geq \delta & \forall l \in \llbracket 0, k-1 \rrbracket \\ & x_j \in \{0, 1\} & \forall j \in J \end{cases}$$

Algorithm `MinDegradation` details the pseudocode of the recursive procedure we implemented to produce the presentation set. The procedure stops if the problem becomes infeasible or the maximum number of iterations  $n$  is reached. Infeasibility is reached when no alternative satisfying the dissimilarity constraints can be found. If the user wants a larger presentation set, they may choose a smaller threshold  $\delta$ .

---

<sup>2</sup>Find more details at [https://prioritizr.net/reference/add\\_gap\\_portfolio.html](https://prioritizr.net/reference/add_gap_portfolio.html).

---

**Algorithm MinDegradation** Recursive search of  $n$  alternative solutions of  $P_0$  with at least  $\delta$  dissimilarities to the past solutions.

---

**Require:**  $P_0, x^*, n, \delta$

**Ensure:**  $x^1, \dots, x^k$

- 1:  $k \leftarrow 0; P \leftarrow P_0; x^0 \leftarrow x^*$  ▷ initialisation
  - 2: **while** ( $P$  is feasible &  $k < n$ ) **do** ▷ stop when infeasible or enough solutions
  - 3:      $k \leftarrow k + 1$
  - 4:     add  $c_d(x^{k-1}, \delta)$  to  $P$  ▷ impose dissimilarities to the past solutions
  - 5:     solve  $P$  ▷ get an optimal solution  $x^k$  or detect that  $P$  is infeasible
  - 6: **end while**
- 

### 2.3.2 Maximising dissimilarity between alternative solutions

Another option we investigated was to iteratively seek for the most different solution at a user-defined extra cost relatively to the optimal value. More formally, assume that  $k - 1$  alternative solutions  $x^1, \dots, x^{k-1}$  have been previously computed. At iteration  $k \geq 1$ , we seek for an alternative solution  $x^k$  that maximises  $\Delta(x) = \min \{d(x^l, x) : l \in \llbracket 0, k - 1 \rrbracket\}$  among the solutions that do not exceed the cost  $(1 + \gamma)z^*$ . Given that  $\Delta(x)$  is not a linear function of  $x$ , we needed to linearize its expression. The classical method to do this uses that the minimum value among a finite set is the maximum lower bound of the set, *i.e.*,  $\max_x \Delta(x) = \max_{\Delta} \{\Delta : \Delta \leq d(x^l, x)\}$ . The corresponding mixed integer linear program we solved at iteration  $k \geq 1$  is given by:

$$P_2^k : \left\{ \begin{array}{ll} \max_{x, \Delta} \Delta & \\ \text{s.t.} & \sum_{j \in J} c_j x_j + \beta \sum_{j_1 \in J} \sum_{j_2 \in J} b_{j_1 j_2} x_{j_1} (1 - x_{j_2}) \leq (1 + \gamma)z^* \\ & \sum_{j \in J} a_{ij} x_j \geq t_i \quad \forall i \in I \\ & \sum_{j \in J} x_j^l (1 - x_j) \geq \Delta \quad \forall l \in \llbracket 0, k - 1 \rrbracket \\ & x_j \in \{0, 1\} \quad \forall j \in J \\ & \Delta \in \mathbb{R}_+ \end{array} \right.$$

Algorithm [MaxDissimilarity](#) details the pseudocode of the recursive procedure we implemented to produce the presentation set. We provided more numerical details for this algorithm in Section [3.4](#).

---

**Algorithm MaxDissimilarity** Recursive search of  $n$  alternative solutions maximising the dissimilarity to the past solutions at a relative extra cost budget  $\gamma$ .

---

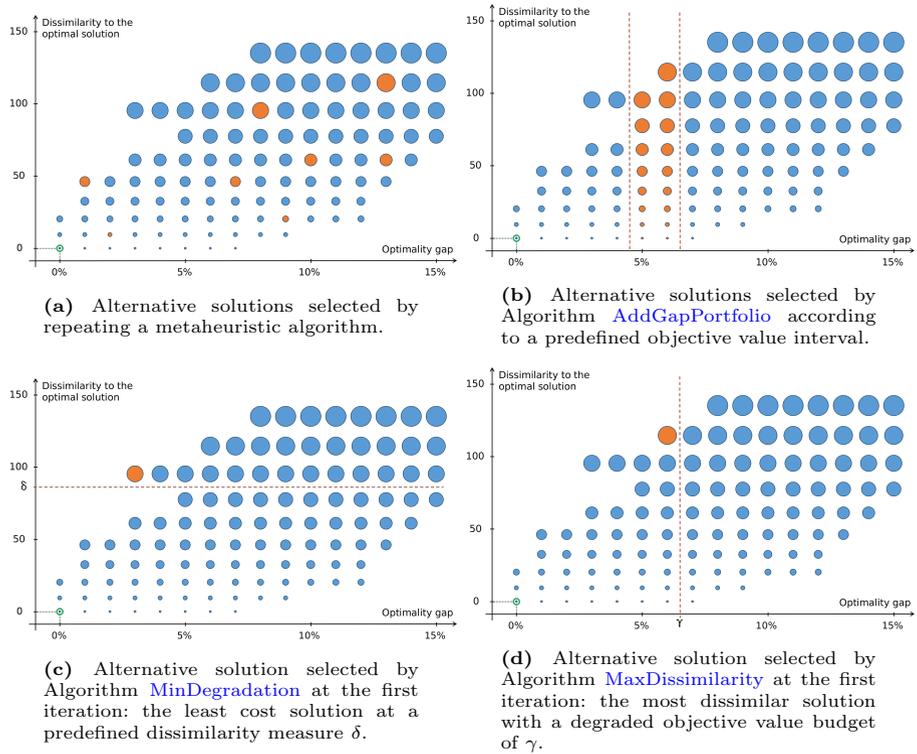
**Require:**  $P_2^1, x^*, z^*, n, \gamma$

**Ensure:**  $x^1, \dots, x^k$

- 1:  $k \leftarrow 0; P \leftarrow P_2^1; x^0 \leftarrow x^*$  ▷ initialisation
  - 2: **while** ( $P$  is feasible &  $k < n$ ) **do** ▷ stop when infeasible or enough solutions
  - 3:  $k \leftarrow k + 1$
  - 4: add  $c_d(x^{k-1}, \Delta)$  to  $P$  ▷ impose dissimilarities to the past solutions
  - 5: solve  $P$  ▷ get an optimal solution  $x^k$  or detect that  $P$  is infeasible
  - 6: **end while**
- 

## 2.4 Illustration of the approaches for computing a presentation set

Figure 2 illustrates schematically how the alternative solutions are selected by different methods. Solutions are mapped in a specific plane: the optimality gap versus the dissimilarity to the optimal solution of  $P_0$ . Figure 2a illustrates the alternative solutions produced by repeating a metaheuristic algorithm such as simulated annealing. These alternative solutions would be scattered in the considered plane. These are neither guaranteed to be close to optimality nor to be different from the optimal solution. Algorithm [AddGapPortfolio](#) selects the alternative solutions in a given objective value interval. Figure 2b shows the alternative solutions that would produce Algorithm [AddGapPortfolio](#). These solutions can be at any dissimilarity to the optimal solution, although their objective value belongs to a predefined interval by construction. Figure 2c shows how Algorithm [MinDegradation](#) would select the leftmost solution among the solutions above a predefined dissimilarity threshold. In other words, Algorithm [MinDegradation](#) would select the solution closest to the optimum at a fixed dissimilarity measure. Similarly, for Algorithm [MaxDissimilarity](#), the first alternative selected would be the solution with the most dissimilarity given a tolerated degradation of the objective value. In Figure 2d, this corresponds to the topmost solution among the solutions at the left of a predefined objective value threshold.



**Fig. 2: Schematic representation of alternative solutions selected by metaheuristics, Algorithm *AddGapPortfolio*, Algorithm *MinDegradation* and Algorithm *MaxDissimilarity*.** Each circle represents a reserve solution. The reserve solutions are located by the optimality gap and the dissimilarity to the optimal solution  $d(x^*, x)$ . The optimal solution  $x^*$  is circled in green at the bottom left of this plan. The bigger the circle, the greater the dissimilarities to the optimal solution. Alternative solutions that would select an algorithm are depicted with orange circles.

### 3 Case study

The code for this work is open, free and available<sup>3</sup>. We used the Gurobi solver under a free academic licence called through a code developed in Julia language [36, 37] using the JuMP optimisation library [38]. The developed methods were numerically tested on the real-world example of Fernando de Noronha composed of 3 conservation features and 756 planning units.

#### 3.1 Data preparation

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<sup>2</sup>. It constitutes a genuine Brazilian natural and cultural heritage and a conservation showcase in Brazil. But it also faces many interests (tourism intensification, fisheries), which results in an open laboratory for marine spatial planning and reserve site selection. We summarised below the main characteristics of the dataset detailed in [39].

The geographical area was discretised according to a rectangular grid made of  $N=36 \times 21=756$  planning units with longitude and latitude respectively in [32.65°W, 32.30°W] and [3.95°S, 3.75°S] ranges. Planning units located in Fernando de Noronha land and harbour were *a priori* excluded from potential reserve site candidates. The optimisation problem  $P_0$  considered through this work included three conservation features: the fish biomass, the continental shelf and shelf break habitats. Each conservation feature was given a targeted protection level of 50% of the total possible amount. The cost layer was made of the fishing pressure intensity. The  $\beta$  multiplier considered was set to  $\beta = 1$  for illustration purposes. Figure 3 shows the details of the input data involved in our case study.

For the first conservation feature, we estimated the fish biomass using acoustic data. Interpolating between sample data allowed producing a continuous distribution within the sampling area. Outside this area, although the actual distribution was unknown, values were set to 0. Then, ocean depth intervals were used as a surrogate for the two other conservation features: the continental shelf and shelf break habitats. The ocean bathymetry was obtained from the GEneral Bathymetric Chart of the Ocean (GEBCO) online platform<sup>4</sup>. Finally, a segmentation model was applied to fishermen's trajectories to derive the behavioural state for every GPS measure: fishing or travelling. This was then used to derive a quantitative proxy for the fishing pressure.

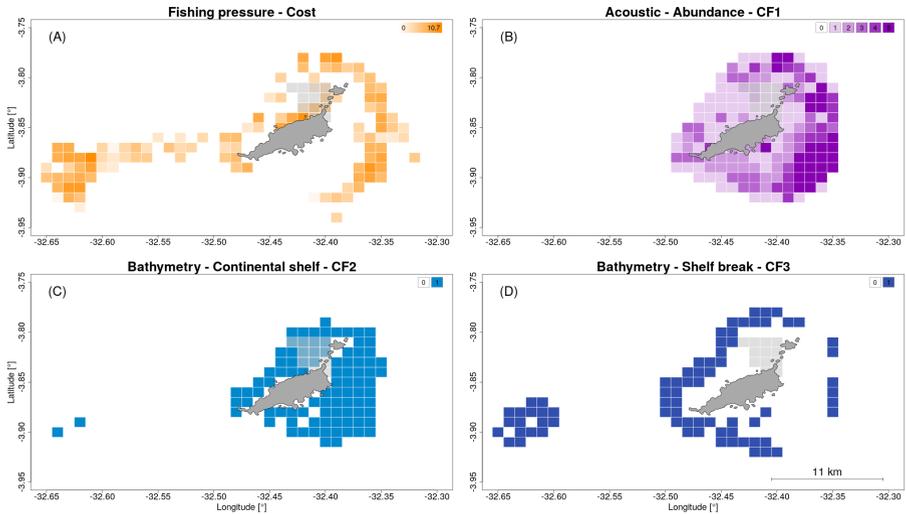
#### 3.2 Presentation set imposing an objective value interval

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

---

<sup>3</sup>GitHub repository at [https://github.com/AdrienBrunel/rssp\\_presentation\\_set.git](https://github.com/AdrienBrunel/rssp_presentation_set.git)

<sup>4</sup><https://download.gebco.net/>.



**Fig. 3:** Data used for the reserve site selection optimisation problem. (A) Fishery-based cost layer in a continuous orange colour gradient. (B) Fish Biomass conservation feature surrogate in a discrete purple colour gradient. (C) Continental shelf and (D) Shelf break habitat conservation feature surrogates in light and deep blue respectively. Transparent grey pixels are the planning units a priori excluded from the solution.

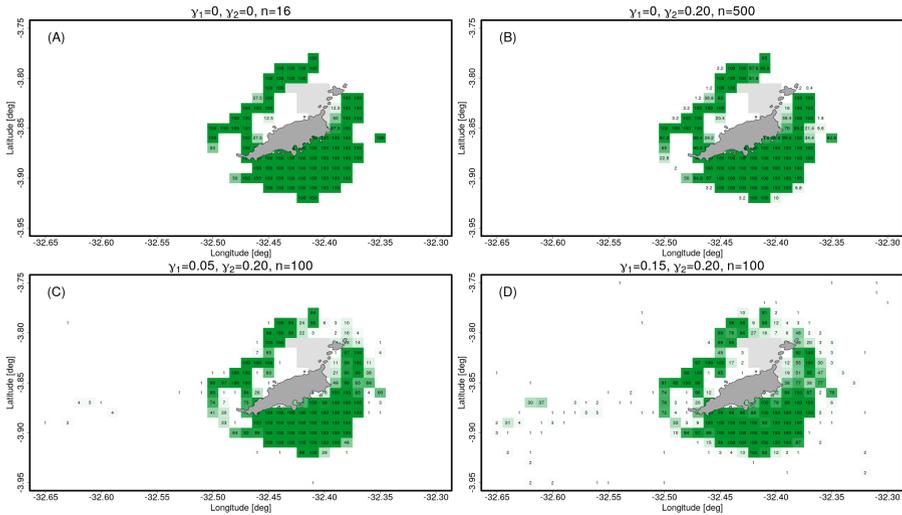
Algorithm [AddGapPortfolio](#) returns all the solutions whose relative optimality gaps belong to the interval  $[\gamma_1, \gamma_2]$ . By setting  $\gamma_1 = \gamma_2 = 0$ , we were thus able to compute the exhaustive set of optimal solutions. Here, we have 16 optimal solutions with an objective value  $z^* = 197.71$ . Panel A of Figure 4 shows the selection frequency among optimal solutions, *i.e.* the percentage of time a planning unit was selected among the 16 optimal solutions. We observe a low variability since 84 over 93 planning units were selected at a 100% frequency. The nine planning units with a selection frequency below 100% have all a cost of 1 and have similar amounts of conservation features. The 16 optimal solutions are composed of combinations of these nine planning units of similar characteristics that still meets targets.

### *Alternative solutions by increasing optimality gaps ( $\gamma_1 = 0, \gamma_2 > 0$ )*

We computed the  $n = 500$  following suboptimal solutions from best to worst optimality gap. We set  $\gamma_2$  to a high enough value to be certain Algorithm [AddGapPortfolio](#) produces the  $n$  alternative solutions. The objective value of the last and worst solution returned by the algorithm was 198.98, which corresponded to an optimality gaps of 0.64%. Any value of  $\gamma_2$  greater than 0.64 would have led to the same 500 alternative solutions. As above, Panel B of Figure 4 illustrates the selection frequency of these 500 alternative solutions. Again, a low variability is observed, because many planning units have similar characteristics and are interchangeable. The visual impression is thus similar to the exhaustive set of optimal solutions.

### Alternative solutions within an objective value interval ( $\gamma_1 > 0$ , $\gamma_2 > 0$ )

We set  $\gamma_1 > 0$  to get suboptimal solutions with an optimality gap of at least  $\gamma_1$ . We chose  $\gamma_2$  high enough to have  $n = 100$  alternative solutions. Panel C and D in Figure 4 respectively show results for  $\gamma_1 = 0.05$  and  $\gamma_1 = 0.15$ . We observe a greater variability than for the two previous presentation sets. However, when comparing the conservation features distribution in Figure 3, many planning units selected in the alternative solutions do not increase the amount of conservation features in the reserve nor decrease its perimeter. These planning units are only selected to deteriorate the objective value and thus satisfy the constraint of the objective value interval. Although the variability appears greater in Panel C and D compared to other panels, the core of the reserve is still globally similar to the optimal solutions.

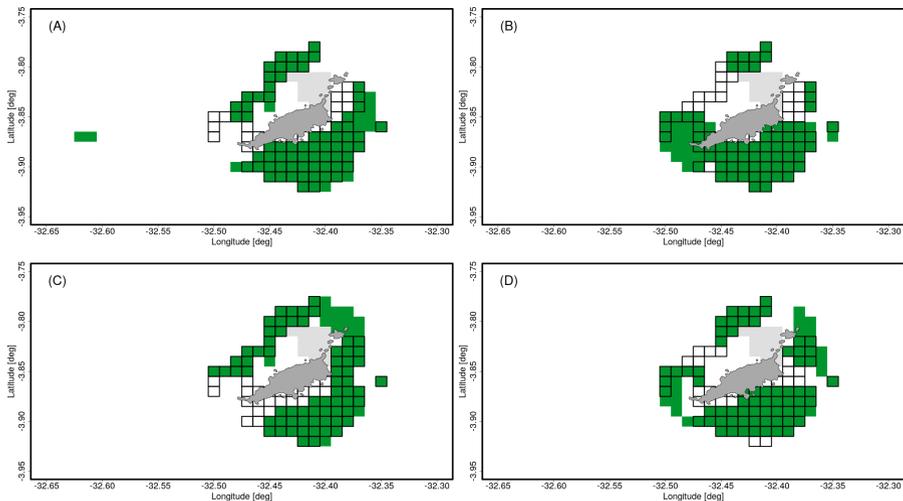


**Fig. 4:** Selection frequency among alternative solutions obtained with Algorithm [AddGapPortfolio](#). Selected planning units of alternative reserve solutions are represented with a green transparency gradient according to the selection frequency expressed in percentage (black number inside planning unit).

### 3.3 Presentation set imposing a dissimilarity measure

In this section, we applied Algorithm [MinDegradation](#) to our case study. Figure 5 shows  $n = 4$  alternative solutions found by the recursive procedure for  $\delta = 20$ . We first observe that, as expected, at least 20 planning units selected in the optimal solution  $x^*$  are not found in the alternative solutions. The dissimilarity to the optimal solution appears more clearly than the alternatives produced by Algorithm [AddGapPortfolio](#). The alternative solutions proposed in Figure 5 cover different regions of the archipelago, although limited by the

fact that non-zero amounts of conservation feature are aggregated around the main island. In particular, the southern region is privileged in Panel B whereas the north and east of Fernando de Noronha are preferred in Panel C. Panel A shows a solution similar to the optimal one, although two planning units are selected at the extreme west of the study area. Panel D displays a solution cut into several pieces all around the main island.

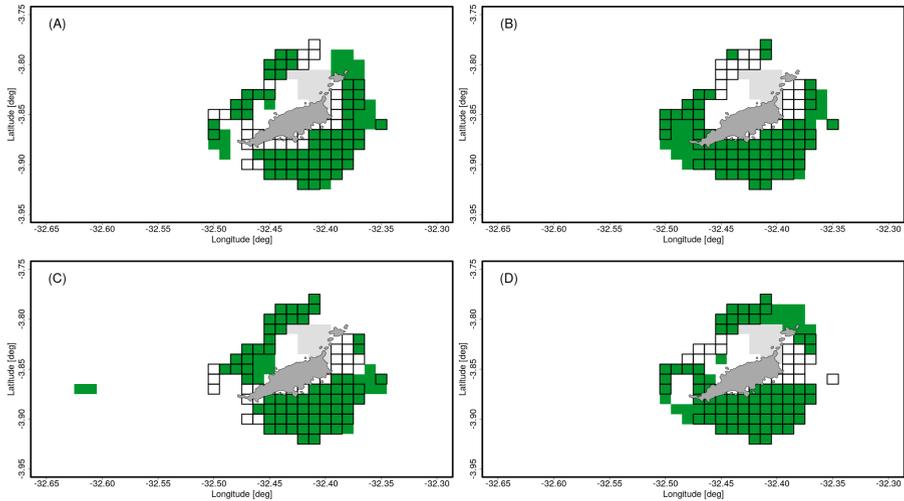


**Fig. 5:** Example of alternative solutions obtained with Algorithm *MinDegradation* for  $\delta = 20$ . The alternative reserve solution is represented in green, while the optimal solution  $x^*$  is depicted with planning units delimited by a thick black border. The number of white planning units with a thick black border corresponds to the dissimilarity measure between the optimal solution and the alternative solution.

### 3.4 Presentation set maximising the dissimilarity measure

We applied Algorithm *MaxDissimilarity* to find the  $n = 4$  alternative solutions maximising the minimum dissimilarity to the past solutions. Illustrations are presented in Figure 6 for a relative extra cost budget of  $\gamma = 10\%$ . Interestingly, a clear visual difference between the four alternative reserves appears in Figure 6. Panel A proposes a solution cut into 4 pieces, favouring the east of the archipelago. Panel B shows a clear preference for the south of the island. Panel C is the most similar to the optimal solution, although two planning units are found at the extreme west of the main island. Panel D presents a reserve in two pieces, one in the north and one in the south. The dissimilarity measure between the reserve in Panel A and the optimal one is equal to 27 planning units. The first iteration simply maximises the dissimilarities to the optimal solution. The minimum dissimilarity measure between the reserve in Panel B to past solutions is 22: the dissimilarity to the optimal solution

and to the solution in Panel A are both equal to 22. By definition of Algorithm [MaxDissimilarity](#), there is no other solution such that the dissimilarity measure from those two past solutions are both greater than 22.

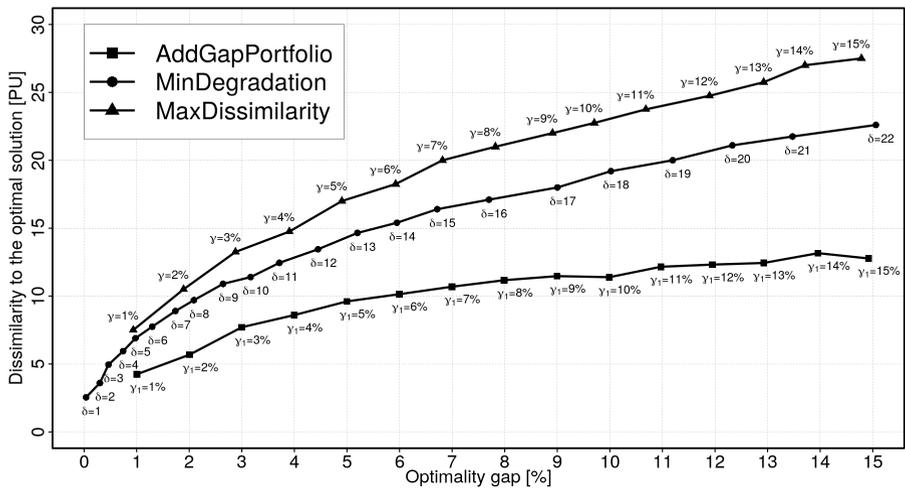


**Fig. 6:** Example of alternative solutions maximising the minimum dissimilarity measure to the past solutions at a fixed extra cost. We show four successive alternatives with an extra cost budget  $\gamma = 10\%$ . The alternative reserve solution is represented in green, whereas the optimal solution  $x^*$  is depicted with planning units delimited by a thick black border.

### 3.5 Optimality gap versus dissimilarity measure

We compared the alternative solutions obtained with Algorithm [AddGapPortfolio](#), [MinDegradation](#) and [MaxDissimilarity](#) for various values of the parameters involved. To do so, we represented the mean optimality gap of the alternative solutions and the dissimilarity to the optimal solution. We repeatedly applied Algorithm [AddGapPortfolio](#), [MinDegradation](#) and [MaxDissimilarity](#) by respectively setting the  $\gamma_1$ ,  $\delta$  and  $\gamma$  parameter to increasing values.

All curves in Figure 7 increase. As expected, the top curve is obtained with Algorithm [MaxDissimilarity](#) because it explicitly sought to maximise the dissimilarity to the past solutions. Since the dissimilarity measure was not considered at all in Algorithm [AddGapPortfolio](#) but only the targeted objective value interval, the corresponding curve is the lowest and is not strictly increasing. The curve obtained with Algorithm [MinDegradation](#) is in between the two others because it explicitly accounted for the dissimilarity measure but did not seek to maximise it.



**Fig. 7: Dissimilarity to the optimal solution versus the optimality gap.** Algorithm [AddGapPortfolio](#) for  $\gamma_1 \in [1\%, 15\%]$  and  $n = 100$ , Algorithm [MinDegradation](#) for  $\delta \in [1, 22]$  and  $n = 20$ , Algorithm [MaxDissimilarity](#) for  $\gamma \in [1\%, 15\%]$  and  $n = 4$ . Optimality gaps and dissimilarity measures are averaged over the alternative solutions composing the presentation set obtained with the considered algorithm.

## 4 Discussion

We proposed two procedures to produce a diverse set of near-optimal solutions using exact solving methods. The presentation set was 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 nominal optimisation problem. We observed that using the natural distance as the dissimilarity measure leads to alternative solutions that strictly includes the optimal one. Considering that it was not a valuable alternative but only a degraded solution, our dissimilarity measure allowed to discriminate such cases, which is new in conservation biology. The Algorithm [MinDegradation](#) provides the least cost alternative solutions that are sufficiently different from each other according to a given dissimilarity threshold. The Algorithm [MaxDissimilarity](#) provides the most different solutions from each other at a fixed degradation of the cost. These procedures implied the formulation of mixed integer linear programs solved using exact methods. Another important contribution is the comparative analysis of these two procedures among them and with existing methods.

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 dissimilarity measures could be used in our recursive procedures to assess the differences between solutions in the same line as in [\[30\]](#). These differences only depends on the definition of a dissimilarity measure, and can be adapted according to the application case. 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 solving methods, we 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. The user can then choose exactly the trade-off between the diversity of the set of alternative solutions and the optimality gap. The sensitivity analysis conducted in [Section 3.5](#) is an illustration of this trade-off for each approach. Regarding the weaknesses, the proposed approaches are mostly limited by the computation time required. It can be large for some instances and it increases with the number of alternatives requested. However, we did not focus on improving the computation time in this work. In the current state of the algorithms, we can provide orders of magnitude for the computation time with a personal computer (Intel Core i7-8850H CPU @ 2.60GHz) when 4 alternatives need to be computed. To do so, we performed tests with several instances randomly generated according to the process described in [Appendix C](#). We observed the following computational times:

- about 2-3 minutes for 500 planning units and 3 conservation features with Algorithm [MaxDissimilarity](#)

- about 10-60 minutes for 1000 planning units and 5 conservation features with Algorithm [MaxDissimilarity](#)
- about 10-20 seconds for 500 planning units and 3 conservation features with Algorithm [MinDegradation](#)
- about 2-15 minutes for 1000 planning units and 5 conservation features with Algorithm [MinDegradation](#)

These computation times must be put into perspective. If we are not necessarily looking for a proof of optimality, they can be much lower. 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, producing only 4 alternatives is a meaningful choice, because they are really different alternatives that do not require additional statistical analyses.

In conclusion, unlike what was commonly stated in the conservation literature [16, 20], our work showed that exact optimisation methods used for the reserve site selection problem can also be advantageous to produce a range of alternative solutions. As a consequence, it is not true that metaheuristics are the only methods that are able to produce a presentation set. Besides, the inclusion of an explicit dissimilarity criterion directly within the optimisation model allowed to build a more controlled and transparent presentation set. By seeking significantly different solutions, we increased the chance to address objectives that are not necessarily modelled, such as socio-political or management objectives. 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. 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 [23, 40]. Considering this linearisation but also locked-in and locked-out planning units, we have the full mathematical optimisation problem  $P_0^f$  of reserve site selection:

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

We also accounted for the correction of the  $\beta$  multiplier undesirable edge effect[39], leading to the introduction of  $b^*$  where:

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

$$b_{j, N+1}^* = \begin{cases} 1, & \text{if pixel } j \text{ shares a single side with the outer boundary} \\ 2, & \text{if pixel } j \text{ shares 2 sides with the outer boundary} \\ 0, & \text{otherwise} \end{cases}$$

## Appendix B Imposing an objective value interval

We show how we produced the presentation set composed of alternative solutions located at a predefined objective value interval. We here developed our own algorithm although the function *add\_gap\_portfolio* of PrioritizR allows to generate the same set of alternative solutions.

### B.1 Objective value constraints

Let  $\gamma_1 \in \mathbb{R}^+$  and  $\gamma_2 \in \mathbb{R}^+$ , such as  $\gamma_1 \leq \gamma_2$ , be the boundaries of the objective value interval relatively to the optimal value  $z^*$ . The constraints  $c_l(\gamma_1)$  and  $c_u(\gamma_2)$  are imposing the objective value to belong to the predefined interval  $[(1 + \gamma_1)z^*, (1 + \gamma_2)z^*]$ :

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

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

If  $\gamma_1 = \gamma_2 = 0$ , we explore only the optimal solutions set. For  $\gamma_1 > 0$ , we explore alternative solutions that are strictly suboptimal.

### B.2 Distance constraint

The constraint  $c_D(y, \delta)$  impose the solution  $x$  to have at least  $\delta$  different planning units with respect to  $y$ :

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

Importantly,  $\delta = 1$  forbids  $x$  and  $y$  to be strictly equal.

### B.3 Generate the presentation set

Practically, we first add to the optimisation problem the constraints  $c_l(\gamma_1)$  and  $c_u(\gamma_2)$  which must be satisfied at every iteration. Then, to derive a pool of alternative solutions, we excluded at iteration  $k \geq 1$  the solution  $x^{k-1}$  derived the iteration before. The addition of constraint  $c_D(x^{k-1}, 1)$  guarantee this. Indeed, this constraint prevents the searched solution at iteration  $k \geq 1$  to be exactly  $x^{k-1}$ .

Practically, the integer linear program solved at iteration  $k \geq 1$  is  $P_3^k$  such as:

$$P_3^k : \left\{ \begin{array}{ll} \min_x & \sum_{j \in J} c_j x_j + \beta \sum_{j_1 \in J} \sum_{j_2 \in J} b_{j_1 j_2} x_{j_1} (1 - x_{j_2}) \\ \text{s.t.} & \sum_{j \in J} c_j x_j + \beta \sum_{j_1 \in J} \sum_{j_2 \in J} b_{j_1 j_2} x_{j_1} (1 - x_{j_2}) \leq (1 + \gamma_2) z^* \\ & \sum_{j \in J} c_j x_j + \beta \sum_{j_1 \in J} \sum_{j_2 \in J} b_{j_1 j_2} x_{j_1} (1 - x_{j_2}) \geq (1 + \gamma_1) z^* \\ & \sum_{j \in J} a_{ij} x_j \geq t_i & \forall i \in I \\ & \sum_{j \in J} x_j (1 - x_j^l) + x_j^l (1 - x_j) \geq 1 & \forall l \in \llbracket 0, k-1 \rrbracket \\ & x_j \in \{0, 1\} & \forall j \in J \end{array} \right.$$

The constraints  $c_l(\gamma_1)$  and  $c_u(\gamma_2)$  used in  $P_3^k$  are not linear. We linearised these constraints exactly as we did for the model  $P_0^f$  described in Appendix A. Algorithm [AddGapPortfolio](#) details the pseudocode of the recursive procedure we implemented to produce the presentation set. The procedure stops if the problem becomes infeasible or the maximum number of iterations  $n$  is reached. Infeasibility is reached when the objective value of the alternative solution exceeds the upper bound  $\gamma_2$ . If the user wants a larger presentation set, they may choose a greater threshold  $\gamma_2$ . For instance, if  $\gamma_1 = 0$  and  $\gamma_2$  is high enough, Algorithm [AddGapPortfolio](#) returns the  $n$  solutions with the smallest objective value. If  $n$  is chosen high enough, Algorithm [AddGapPortfolio](#) returns the exhaustive set of solutions with an objective value relatively to the optimal value within the interval  $[\gamma_1, \gamma_2]$ . Unlike metaheuristics where the optimality gap is unknown, we *a priori* established it using this algorithm. We thus offer users more control over the presentation set provided.

---

**Algorithm AddGapPortfolio** Recursive search of  $n$  best alternative solutions whose objective values relatively to the optimal value  $z^*$  of solution  $x^*$  of problem  $P_0$  belongs to the predefined interval  $[\gamma_1, \gamma_2]$ .

---

**Require:**  $P_0, x^*, z^*, n, \gamma_1, \gamma_2$

**Ensure:**  $x^1, \dots, x^k$

- 1:  $k \leftarrow 0; P \leftarrow P_0; x^0 \leftarrow x^*$  ▷ initialisation
  - 2: add  $c_l(\gamma_1)$  and  $c_u(\gamma_2)$  to  $P$
  - 3: **while**  $P$  is feasible &  $k < n$  **do** ▷ stop when infeasible or enough solutions
  - 4:      $k \leftarrow k + 1$
  - 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

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 choose 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 using 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  $\alpha_i$  is a predefined parameter for each conservation feature  $i \in I$ . The parameter  $\alpha_i$  controls the dispersion of the mean values 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  $\mu_i$  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  $\sigma_{ij}$  of the Gaussian distribution is such as  $\sigma_{ij} = \sigma_i m_{ij}$  where  $\sigma_i$  is a chosen parameter for each conservation feature  $i \in I$ . The code used to generate data is available in open access<sup>5</sup>. The instance is characterised by the rectangular grid size  $N_x$  and  $N_y$  and the number of conservation features  $N_{cf}$ .

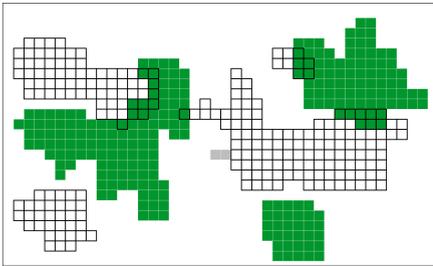
### C.2 Scenarios

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

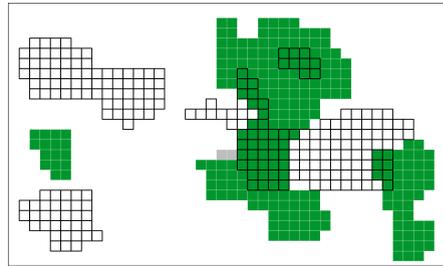
---

<sup>5</sup>[https://github.com/AdrienBrunel/data\\_generation](https://github.com/AdrienBrunel/data_generation)

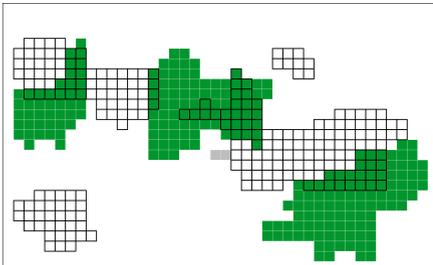




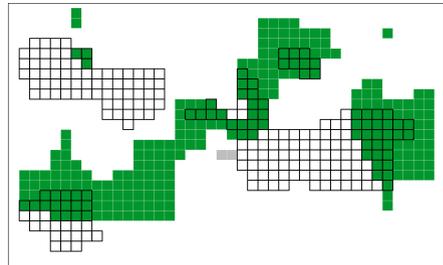
(a) Alternative reserve solution found at iteration 1 with Algorithm *MaxDissimilarity*.



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



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



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

**Fig. C2: Presentation set computed with Algorithm *MaxDissimilarity*.** The considered scenario was made of  $40 \times 25$  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 Declarations

### Ethical approval

Not applicable.

### Authors' contributions

A.B. performed the formal analysis and wrote the first draft of the manuscript. All authors contributed to the conceptualisation and investigation involved in this work, reviewed and edited the manuscript. J.O. and S.LB. were in charge of supervision and validation. S.LB. was involved in material preparation, data curation, funding acquisition, resources provisioning and project administration. All authors read and approved the final manuscript.

### Competing interests

The authors declare they have no competing interests.

### Funding

A.B. is funded by Région Occitanie and Créocéan. 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).

### Consent to Participate

Not applicable.

### Consent for publication

Not applicable.

### Availability of data and material

The datasets generated and analysed during the current study are available at the following GitHub repository: [https://github.com/AdrienBrunel/data\\_generation](https://github.com/AdrienBrunel/data_generation).

### Acknowledgements

We thank Jeffrey Hanson for his thoughtful comments and suggestions on an earlier draft. We also thank the editor Matthew Holden and the two anonymous reviewers for their helpful suggestions.

## 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](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., Garcia-Charton, J.-A., Perez-Ruzafa, A., Badalamenti, F., Bayle-Sempere, J., Brito, A., Bulleri, F., Culioli, J.-M., Dimech, M., Falcon, J.M., Guala, I., Milazzo, M., Sanchez-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 final, 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] Bell, D., Gray, T., Haggett, C.: The ‘Social Gap’ in Wind Farm Siting Decisions: Explanations and Policy Responses. *Environmental Politics* **14**(4), 460–477 (2005). <https://doi.org/10.1080/09644010500175833>. Accessed 2022-07-27
- [14] Virtanen, E.A., Lappalainen, J., Nurmi, M., Viitasalo, M., Tikanmäki, M., Heinonen, J., Atlaskin, E., Kallasvuori, M., Tikkanen, H., Moilanen, A.: Balancing profitability of energy production, societal impacts and biodiversity in offshore wind farm design. *Renewable and Sustainable Energy Reviews* **158**, 112087 (2022). <https://doi.org/10.1016/j.rser.2022.112087>. Accessed 2022-07-27
- [15] 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)
- [16] 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](https://doi.org/10.1007/0-387-22648-6_17). [http://link.springer.com/10.1007/0-387-22648-6\\_17](http://link.springer.com/10.1007/0-387-22648-6_17) Accessed 2021-05-04
- [17] 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)
- [18] 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
- [19] 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

- [20] Ardron, J.A., Possingham, H.P., Klein, C.J.: *Marxan Good Practices Handbook*, (2010)
- [21] 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
- [22] 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
- [23] 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
- [24] 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](https://doi.org/10.1016/S0006-3207(02)00042-3). Accessed 2021-09-21
- [25] 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)
- [26] 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
- [27] 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
- [28] 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
- [29] 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

- [30] 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](https://doi.org/10.1016/S0304-3800(00)00249-0). Accessed 2022-04-08
- [31] 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)
- [32] 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](https://doi.org/10.1016/S0006-3207(03)00093-4). Accessed 2021-03-23
- [33] 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
- [34] Linke, S., Watts, M., Stewart, R., Possingham, H.P.: Using multivariate analysis to deliver conservation planning products that align with practitioner needs. *Ecography* **34**(2), 203–207 (2011). <https://doi.org/10.1111/j.1600-0587.2010.06351.x>. Accessed 2022-05-23
- [35] 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)
- [36] 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
- [37] 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
- [38] 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
- [39] Brunel, A., Davret, J., Trouillet, B., Bez, N., Salvétat, J., Gicquel, A., Bertrand, S.L.: Opening the Black Box of Decision Support Tools in Marine Spatial Planning: Shedding Light into Reserve Site Selection Algorithms for a Balanced Empowerment of Stakeholders. Preprint HAL (2021). <https://doi.org/https://hal.archives-ouvertes.fr/hal-03445922>. Accessed 2022-09-23

- [40] Billionnet, A.: Mathematical optimization ideas for biodiversity conservation. *European Journal of Operational Research*, 21 (2013)