



**HAL**  
open science

## Accountable flexibility in the reserve site selection problem with exact optimisation methods

Adrien Brunel, Sophie Lanco Bertrand, Jérémy Omer

### ► To cite this version:

Adrien Brunel, Sophie Lanco Bertrand, Jérémy Omer. Accountable flexibility in the reserve site selection problem with exact optimisation methods. 2022. hal-03519381v1

**HAL Id: hal-03519381**

**<https://hal.science/hal-03519381v1>**

Preprint submitted on 10 Jan 2022 (v1), last revised 6 Dec 2022 (v3)

**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.

# Accountable flexibility in the reserve site selection problem with exact optimisation methods

Adrien Brunel<sup>a\*</sup>, Jérémy Omer<sup>b</sup> and Sophie Lanco Bertrand<sup>a</sup>

<sup>a</sup>IRD, MARBEC (Univ. Montpellier, CNRS, Ifremer, IRD), Sète, France

<sup>b</sup>Univ Rennes, INSA Rennes, CNRS, IRMAR - UMR 6625, Rennes, France

## Abstract

Marine protected areas are at the heart of current global policies against the erosion of ocean biodiversity. 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 with a minimum impact on human activities. Flexibility is a solver ability to provide relevant alternative solutions on the basis of which stakeholders negotiations can be carried out. The widely used Marxan deploys a metaheuristic algorithm which produce alternatives mostly based on repeating random processes. While several works already demonstrated how exact methods outperform metaheuristic approaches both in computation time and optimality, flexibility is still raised as a key feature in favour of metaheuristics. We propose in this work several algorithms to explore the optimal solution space with exact optimisation methods using simple recursive procedures based on explicit criteria. This work shows that the flexibility offered by metaheuristics can be outstripped by an accountable (explicit, transparent and replicable) flexibility of exact solutions. Allowing to understand precisely why and how the result was arrived at, the framework presented here should contribute to a more equitable negotiation process among stakeholders engaged in marine policy.

**Keywords:** optimal reserve site selection; marine protected area; marine spatial planning; integer linear programming; Marxan; decision support tool.

---

\*Corresponding author. Email: [adrien.brunel.pro@protonmail.com](mailto:adrien.brunel.pro@protonmail.com); Postal address: IRD, MARBEC (Univ. Montpellier, Ifremer, CNRS, IRD), Avenue Jean Monnet, 34203 Sète Cedex, France

# 1 Introduction

## 1.1 Decision Support Tools for Marine Spatial Planning

Acknowledging the destruction of habitats and the erosion of biodiversity at sea, marine spatial planning (MSP) is a governance framework that proposes to regulate the uses of the marine environment through a spatial zoning. MSP seeks to meet both ecological and socio-economic objectives, eventually targeting a sustainable development (Ehler and Douvère 2009). Within this framework, marine protected areas (MPAs) are increasingly becoming a key player. The global benefits they were shown to bring to biodiversity, e.g. (Stolton and Dudley 2010), have made them the tool of choice for conservation policies. The International Union for Conservation of Nature (IUCN) established a 30% protection target of each marine ecoregions by 2030, reaffirming United Nations (UN) commitment of Aichi Target 11. Similarly and more recently, the European Green Deal aims to protect 30% of the European seas. Thus, an important step of MSP approaches consists in identifying suitable areas to be dedicated to conservation, i.e., nature reserves. Consequently, conservation science and especially reserve design knows a strong appeal among decision-makers and institutions. Systematic reserve site selection procedures and associated decision support tools (DSTs) are thus a strategic issue.

The reserve site selection optimisation problem, in its *minimum set* formulation, aims at finding the collection of sites covering given levels of conservation features at a minimum socio-economic cost. For more mathematical details, see Section 2.1. Note that MPAs are a particular case of reserve sites. Yet, this work is also generalisable to reserve design on land. In this work, we used MPAs and reserve sites interchangeably. MPAs are seen as an essential tool to mitigate biodiversity erosion. In particular, it is assumed that species have more chance to persist in a strict protected area rather than in the unprotected ocean due to human activity impacts (Claudet et al. 2008). In a modelling perspective, we assumed that an MPA resulted in the strict closure of a marine area. Site protection led to a loss of potential profits (e.g. economic activity, space) quantified through an arbitrary cost from a human perspective. The general idea was thus to find an MPA covering every species<sup>1</sup> in a sufficient amount to ensure the long term subsistence of specie populations at minimum cost. In the case study, we did not have direct access to the spatial distribution of every conservation feature of interest. Consequently, we often estimated a surrogate for the conservation feature. We then assumed the protection of the surrogate was sufficient for the conservation feature protection. Historically, the reserve site selection problem was addressed through ranking and then iterative ranking algorithms of a previously computed conservation value (Tans 1974; Kirkpatrick 1983). Then, identified as a conservation resource allocation problem, an optimisation framework was proposed (Cocks and Baird 1989) and triggered fundamental developments (Possingham et al. 1993; Margules and Pressey 2000) for systematic reserve site selection. For a more detailed literature review, see (Sarkar 2004). Technically speaking, the reserve site selection problem is a discrete optimisation problem. This problem can be expressed as an integer linear program where real instances can be solved in reasonable time.

The literature of the reserve site selection optimisation problem often opposed metaheuristic and exact solving approaches. Metaheuristics are optimisation algorithms constituted of generic heuristics search, i.e., a set of empirical rules which quickly leads towards a sufficiently good solution. On the other hand, exact optimisation methods find an optimal solution if the problem respects certain mathematical properties (e.g., linear, convex formulations) and can be

---

<sup>1</sup>For simplicity purposes, we use the term *species* to stand for *conservation features*, i.e., any element judged worth protecting for the ecosystem sake.

time-consuming for large NP-hard problems (Garey and Johnson 1979). Metaheuristics propose quickly suboptimal solutions of any optimisation problem while exact algorithms provide, even if time-consuming, an optimal solution for a class of optimisation problems. Metaheuristics owe their speed to the fact that the stopping criterion do not wait for an optimality proof of the solution, unlike exact methods.

Despite Church et al. (1996) claims in favour of exact methods, Pressey et al. (1996) still raised concerns about their capacity to solve large realistic problems in this early debate of the conservation literature. Practically speaking, methodological advances and conservation practices were notably influenced by Marxan DST developments (Ball et al. 2009). In particular, Marxan chose to implement a simulated annealing (SA) metaheuristic algorithm to tackle the reserve site selection problem. This choice was justified in (Ardrón et al. 2010) where metaheuristics' strengths are stated: computational speed, efficiency on large and complex problems and flexibility of choice among provided suboptimal solutions. However, the first two arguments were discarded in (Rodrigues and Gaston 2002) thanks to a comparison of several reserve site selection study cases. More recently, this solving paradigm seemed to change in favour of exact algorithms within the conservation biology literature (Beyer et al. 2016; Hanson et al. 2019; Schuster et al. 2020). By nature, the objective values of solutions are better with exact methods; what changed is the reasonable computational time taken by exact solvers nowadays. Regarding formulation restriction, metaheuristics can definitely tackle non-linear formulation and more generally do not impose any restriction unlike exact methods. Yet, the reserve site selection problem includes only quadratic terms, which can be linearised when the decision variables are binary (Billionnet 2007). Finally, it left only the flexibility claim as a major advantage of metaheuristics over exact optimisation approaches.

## 1.2 Flexibility requirement in the choice of MPAs

In conservation biology, the flexibility covered two realities: in a few cases, it referred to the optimisation model and its ability to be easily tuned (Rodrigues et al. 2000; Cocks and Baird 1989). This aspect was not discussed in our work. In the majority of cases, the flexibility referred to the ability of an algorithm to provide relevant alternative solutions. The latter was the subject of this work and was the main notion called when flexibility is mentioned in the conservation literature. For instance, in a survey realised among Marxan users (Ardrón et al. 2010), "flexibility of generating multiple solutions was by far the most commonly noted strength of Marxan" for 2/3 of the respondents. Flexibility was often depicted as the key strength of metaheuristics over exact approaches, as it satisfied the need for multiple alternative solutions, even if suboptimal (Pressey et al. 1993; Possingham et al. 2000; Possingham et al. 2006; Sarkar 2012).

Given the importance attached to this property, it was crucial to characterise precisely the flexibility provided by the SA algorithm and thus understand how suboptimal solutions were built. First, SA is a recursive procedure which starts from an initial random reserve and stops when a maximum number of iterations is reached. At each iteration, SA randomly considers a neighbour of the current reserve. Again, a random process decides whether to replace the current reserve by the neighbour. One particularity of SA is that it can choose reserve candidates with a worse objective value, thus preventing the procedure from being stuck at the first local optimum met. Also, a *temperature* parameter of SA allows to widely explore the solution space at the beginning of the iterations, while a thinner search is performed at the end. Finally, in order to produce a pool of solutions, SA is run as many times as requested. More details can be found in Appendix B-2.1 in (Serra-Sogas et al. 2020). As a consequence, SA builds the set of suboptimal solutions mostly in a random fashion without

using any user-specified criterion; nor providing any control or knowledge over the gap between the suboptimal solutions provided and the optimal solution. Here, we addressed the question of how to generate alternative suboptimal reserve solutions with exact solvers, using explicit tunable criteria instead of random processes. The objective was to improve the accountability and control over the flexibility provided in the reserve solutions. Few existing works had the same concern. Arthur et al. (1997) developed an algorithm to compute the exhaustive set of optimal solutions in the maximum coverage formulation with presence/absence data. In the same spirit, Önal (2004) described a branch and bound algorithm to show how suboptimal solutions, called the second-best solutions, can be derived with exact methods.

### 1.3 Contributions

In this work, we modified the optimisation problem in order to explore the solution space with exact solvers, seeking for relevant alternative solutions and thus provide decision makers a flexibility with a portfolio of reserve solutions. Unlike metaheuristics, this exploration was guided by explicit criteria included in the optimisation model. We modified the reserve site selection problem by overconstraining it or even by changing the objective function. Therefore, the flexibility was not produced by repeating random processes but rather according to explicit specifications. Practically, our work presents several methods to derive flexibility through a pool of alternative solutions. The code was written in Julia language, and is available in open access<sup>2</sup>. The developed methods were numerically illustrated on the real example of Fernando de Noronha Brazilian archipelago in the tropical Atlantic. In Section 2.2, we first discussed the flexibility itself as the ability to produce good and different solutions. It naturally led to define a diversity measure between reserves, a distance and a pseudo-distance, to help us build explicit selection criteria. Then, the modifications brought to the initial optimisation model in Section 2.1 are described in Section 2.3. We showed how to produce an alternative solution with an optimality gap constraint in Section 2.3.1, with a diversity constraint in Section 2.3.2, and by maximising a minimum pseudo-distance objective in Section 2.3.3. Finally, in Section 2.4, we developed three different algorithms that generated a pool of alternative solutions based on the aforementioned criteria. They provided (1) the exhaustive set of optimal solutions, (2) the pool of  $n$  best suboptimal solutions within a user-defined optimality gap in increasing order, (3) the pool of suboptimal solutions at a given pseudo-distance between each other and (4) the pool of alternatives maximising the minimum of the set of pseudo-distances with past iterates considering a fixed extra cost budget. The results we described in Section 3 are discussed in Section 4 where we also questioned the notion of flexibility and its relevance for reserve site selection.

---

<sup>2</sup><https://github.com/AdrienBrunel/flexibility.git>

## 2 Methods

### 2.1 The reserve site selection problem

In general, the spatial distributions of sampling are scattered and sparse. Consequently, we usually discretise the study area thanks to a grid to properly capture data. In our case, as a result of the discretisation process, the study area was divided into a set of  $J$  planning units within which a set of  $I$  conservation features were distributed. The amount of conservation feature  $i$  in the planning unit  $j$  was denoted  $a_{ij}$ . Each planning unit had a cost  $c_j$ . In the reserve site selection problem, one generally seeks to find the collection of planning units covering sufficient levels of considered features at minimum cost. The covering of conservation feature  $i$  is sufficient if it exceeds the user-defined level that we noted  $t_i$ . The decision is about to include the planning unit in the reserve. Thus,  $x_j = 1$  if a planning unit  $j$  was selected in the reserve and  $x_j = 0$  otherwise. Moreover, a compactness parameter  $\beta$  allows computing a more or less aggregated reserve, as it directly penalises the reserve perimeter within the objective function. The perimeter was 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  $j_2$  is denoted as  $b_{j_1j_2}$ . Mathematically speaking, the general problem of reserve site selection is expressed as the following minimum set optimisation problem  $P_0$  :

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

This combinatorial optimisation problem is a minimum set cover problem known to be NP-hard (Garey and Johnson 1979). 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 5.1 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 will denote  $x^*$  and  $z^*$  respectively the optimal solution and the associated objective value of the original minimum set cover problem  $P_0$ . Let  $\gamma(z) = \frac{z - z^*}{z^*}$  be the algebraic relative variation of score  $z$  with respect to  $z^*$ .

### 2.2 Flexibility in the selection of reserve sites

**Producing *good* and *different* solutions** Marxan user manual stipulated that systematic conservation planning DSTs did not provide *the answer*, but rather produced multiple *good* and *different* alternative solutions. This way, it allowed decision makers to do a choice integrating exogenous factors or even open negotiations. To fill in this need, Marxan generated multiple solutions in a mostly random fashion using a SA algorithm. The amount of randomness (starting point, neighbour construction, candidate acceptance) actually made it very unlikely to have the same solution between two SA runs and thus produced different suboptimal solutions. Marxan implicitly built a pool of different good solutions without controlling how different and good they are. However, the difference between solutions could be explicitly measured, for example thanks to a distance function. Also, the solution quality could be quantified with the gap to the global optimal value. We thus chose to produce a pool of alternative solutions thanks to explicit criteria and directly modify the model. In particular, we produced alternatives by overconstraining the problem or modifying the objective function.

**Diversity measure** As we sought to quantitatively capture the difference between two reserve solutions, we focused beforehand on how to build a diversity measure between them. We

wanted to find a function characterising the difference between two solutions. The first natural idea was to use the distance defined by the absolute-value norm:

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

When  $x, y \in \{0, 1\}^N$ , the distance  $D(x, y)$  is 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 preferred the following linear expression, strictly equivalent<sup>3</sup> when  $x, y \in \{0, 1\}^N$ :

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

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

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

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 chosen didactic examples where the distance  $D$  and pseudo-distance  $d$  are 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 understood why the red solution was simply a worse alternative to the green solution and that we wanted to avoid this case. Then, the distance between red and green reserves was  $D(\hat{x}, x) = 2$  while the pseudo-distance was  $d(\hat{x}, x) = 0$ . Consequently, according to the pseudo-distance  $d$ , the green and red reserve were 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 allowed us to characterise differences between two reserves as much as distance  $D$ .

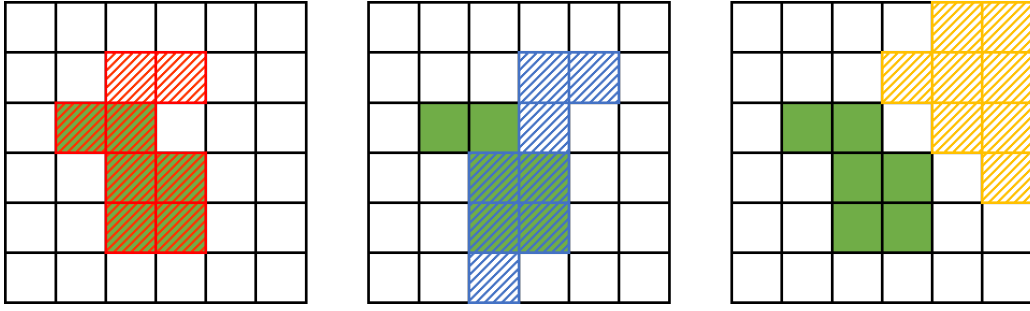
## 2.3 Modifying the model to produce an alternative solution

### 2.3.1 Optimality gap constraint

In this section, 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  $[(1 + \gamma_1)z^*, (1 + \gamma_2)z^*]$  relatively to the optimal score  $z^*$ .

---

<sup>3</sup>Compare truth tables of these two expressions to be convinced.



**Figure 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.

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

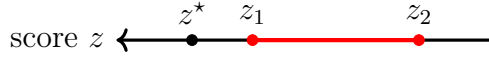
$$d(\hat{x}, x) = 2, d(x, \hat{x}) = 4, D(\hat{x}, x) = 6.$$

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

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

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

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  $\gamma_2$  (see illustration in Figure 2).



**Figure 2 – Illustration of constraint  $c_l(\gamma_1)$  and  $c_u(\gamma_2)$ .** The objective value  $z \geq z^*$  of optimisation problem  $P$  is constrained to fall within the a priori specified interval  $[z_1, z_2] = [(1 + \gamma_1)z^*, (1 + \gamma_2)z^*]$  in red.

### 2.3.2 Diversity constraint

In this section, we present a set of constraints which compels the reserve to differ from another given solution by at least  $\delta$  planning units. The constraint  $c_D(\hat{x}, \delta)$  makes the solution  $x$  to have at least  $\delta$  different planning units with respect to  $\hat{x}$  (cf. Figure 3). Importantly, for  $\delta = 1$ , we are simply forbidding  $x$  and  $\hat{x}$  to be strictly equal.

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

Similarly, we have the same constraint with pseudo-distance  $d$  instead of distance  $D$ . Indeed, the constraint  $c_d(\hat{x}, \delta)$  compels the reserve solution  $x$  to have at least  $\delta$  planning units selected in  $\hat{x}$  that are not included in  $x$  :

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

As discussed previously and illustrated in Figure 3, the pseudo-distance  $d$  allows forbidding the strict inclusion of the solution  $\hat{x}$  within the considered alternatives thanks to  $c_d(\hat{x}, \delta)$ .



$\hat{x}$	$x$	$D(\hat{x}, x)$	$d(\hat{x}, x)$
1	0	+1	+1
0	0	+0	+0
0	1	+1	+0
1	1	+0	+0
		$\geq \delta$	$\geq \delta$

**Figure 3 – Illustration of constraint  $c_D(\hat{x}, \delta)$  and  $c_d(\hat{x}, \delta)$ .** Example of distance  $D$  and pseudo-distance  $d$  construction between solutions  $\hat{x}$  and  $x$ .

### 2.3.3 Maximum pseudo-distance objective

We also proposed to directly modify the objective function value and thus write a new integer linear program in order to produce an alternative solution. Indeed, another idea could be to seek for the most pseudo-distant solution at a predefined maximum deterioration level of the objective value. Stated otherwise, we maximised the pseudo-distance from the optimal solution  $x^*$  under a fixed extra cost budget  $\gamma_2 z^*$ . This gave access to the solution most different from the optimum without exceeding a given extra impact on socio-economic activities. The resulting integer linear program is given by:

$$P_D : \begin{cases} \max_x & d(x, x^*) \\ \text{s.t.} & \sum_{j \in J} c_j x_j + \beta \sum_{j_1 \in J} \sum_{j_2 \in J} b_{j_1 j_2} x_{j_1} (1 - x_{j_2}) \leq (1 + \gamma_2) z^* \quad \forall j_1 \in J, \forall j_2 \in J \\ & \sum_{j \in J} a_{ij} x_j \geq t_i \quad \forall i \in I \\ & x_j \in \{0, 1\} \quad \forall j \in J \end{cases}$$

## 2.4 Producing a pool of alternative solutions

### 2.4.1 Imposing an optimality gap

We detailed in Algorithm 1 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  $\gamma_2$  or the maximum number of iteration  $n$  was reached. Importantly, if we set  $\gamma_2$  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  $\gamma_2$  were high enough, Algorithm 1 returned the  $n$  solutions with the smallest score. If  $n$  was chosen high enough, Algorithm 1 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.

**Algorithm 1:** Recursive search of  $n$  best alternative solutions with a relative optimality gap in  $[\gamma_1, \gamma_2]$  relatively to score  $z^*$  of solution  $x^*$  of problem  $P_0$ .

<b>input</b> : $P_0, x^*, z^*, n, \gamma_1, \gamma_2$	
<b>output:</b> $x^1, \dots, x^k$	
1	$P \leftarrow P_0; k \leftarrow 0; x^0 \leftarrow x^*$ ; <span style="float: right;">▷initialisation</span>
2	add $c_l(\gamma_1)$ and $c_u(\gamma_2)$ to $P$ ; <span style="float: right;">▷impose optimality gap</span>
3	<b>while</b> $P$ is feasible & $k < n$ <b>do</b> <span style="float: right;">▷stop when infeasible or enough solutions</span>
4	$k \leftarrow k + 1$ ;
5	add $c_D(x^{k-1}, 1)$ to $P$ ; <span style="float: right;">▷exclude previous solution</span>
6	solve $P$ : get an optimal solution $x^k$ or detect that $P$ is infeasible;

### 2.4.2 Imposing a pseudo-distance

Algorithm 2 is similar to Algorithm 1 except that we removed the constraints on the objective value  $c_l(\gamma_1)$  and  $c_u(\gamma_2)$ . Besides, the constraint  $c_d(x^{k-1}, \delta)$  replaced  $c_D(x^{k-1}, 1)$ . Indeed, in Algorithm 2, we compelled at each iteration the solution  $x^k$  to have at least  $\delta$  unselected planning units that were selected in every previous iterated solutions. In other words, we explicitly looked for a solution at least pseudo-distant of  $\delta$  planning units from every past iterates.

**Algorithm 2:** Recursive search of  $n$  alternative solutions of problem  $P_0$  with at least  $\delta$  new planning units between successive iterated solutions.

<b>input</b> : $P_0, x^*, n, \delta$	
<b>output:</b> $x^1, \dots, x^k$	
1	$k \leftarrow 0; P \leftarrow P_0; x^0 \leftarrow x^*$ ; <span style="float: right;">▷initialisation</span>
2	<b>while</b> $P$ is feasible & $k < n$ <b>do</b> <span style="float: right;">▷stop when infeasible or enough solutions</span>
3	$k \leftarrow k + 1$ ;
4	add $c_d(x^{k-1}, \delta)$ to $P$ ; <span style="float: right;">▷impose pseudo-distance with past iterate</span>
5	solve $P$ : get an optimal solution $x^k$ or detect that $P$ is infeasible;

### 2.4.3 Maximising the minimum pseudo-distance between iterates

The most natural idea would have been to recursively produce the  $n$  most pseudo-distant alternative solutions from the optimal solution  $x^*$  at a fixed relative extra cost  $\gamma_2$  by successively excluding from the search space the solution found the iteration before. Indeed, to do so, we could have simply adapted Algorithm 1 by initialising the problem to  $P_D$  instead of  $P_0$  and removing the optimality gap constraints (line 2). 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  $x^*$ . 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  $l \in \llbracket 0, k-1 \rrbracket$ . Let  $x$  be a candidate solution at iteration  $k$  and  $\delta(x) = \min \{d(x^l, x), l \in \llbracket 0, k-1 \rrbracket\}$ . We proposed to search for a solution  $x^k$  that maximises  $\delta(x)$  among the solutions that do not exceed the degraded cost  $(1 + \gamma_2)z^*$ .

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

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

The objective value  $\delta$  would be the maximum of the minimum of the set of pseudo-distances with every past iterates. Variable  $\delta$  is constrained to be smaller than every pseudo-distances, so is the minimum. As we want to maximise  $\delta$ , it will logically take the value of one of the pseudo-distances, i.e. the smallest. Therefore, we maximise the minimum pseudo-distance with every past iterate. One should be careful as  $\delta$  is a decision variable and no longer a parameter as in previous algorithms. In order to better understand what Algorithm 3 is doing, we gave more numerical details in Section 3.4. The procedure is presented below :

**Algorithm 3:** Recursive search of  $n$  alternative solutions maximising the minimum pseudo-distance from every past iterates at a relative extra cost  $\gamma_2$ .

<p><b>input</b> : <math>P_D, x^*, z^*, n, \gamma_2</math>  <b>output</b>: <math>x^1, \dots, x^k</math></p> <p>1 <math>P \leftarrow P_D; k \leftarrow 0; x^0 \leftarrow x^*</math> ; <span style="float: right;">▷initialisation</span>  2 change the objective of <math>P</math> to <math>\delta</math> ;  3 <b>while</b> <math>P</math> is feasible &amp; <math>k &lt; n</math> <b>do</b> <span style="float: right;">▷stop when infeasible or enough solutions</span>  4     <math>k \leftarrow k + 1</math>;  5     add <math>c_d(x^{k-1}, \delta)</math> to <math>P</math> ; <span style="float: right;">▷<math>\delta</math> is a decision variable here</span>  6     solve <math>P</math>: get an optimal solution <math>x^k</math> or detect that <math>P</math> is infeasible;</p>
---

## 3 Results

In order to solve the reserve site selection problem, several generic integer programming solvers are available. In this work, we used Gurobi solver called through a dedicated code developed in Julia language (Bezanson et al. 2012; Bezanson et al. 2015) using the JuMP optimisation library (Dunning et al. 2017). Gurobi is a commercial solver available under a free academic license at <https://www.gurobi.com/>. We could have also solved the problem with the free and open-source integer programming solver Cbc (Forrest et al. 2018). The Julia language allowed us to directly express and customise the optimisation problem according to our need. Our code is open, free and available at <https://github.com/AdrienBrunel/flexibility.git>.

### 3.1 Case study data

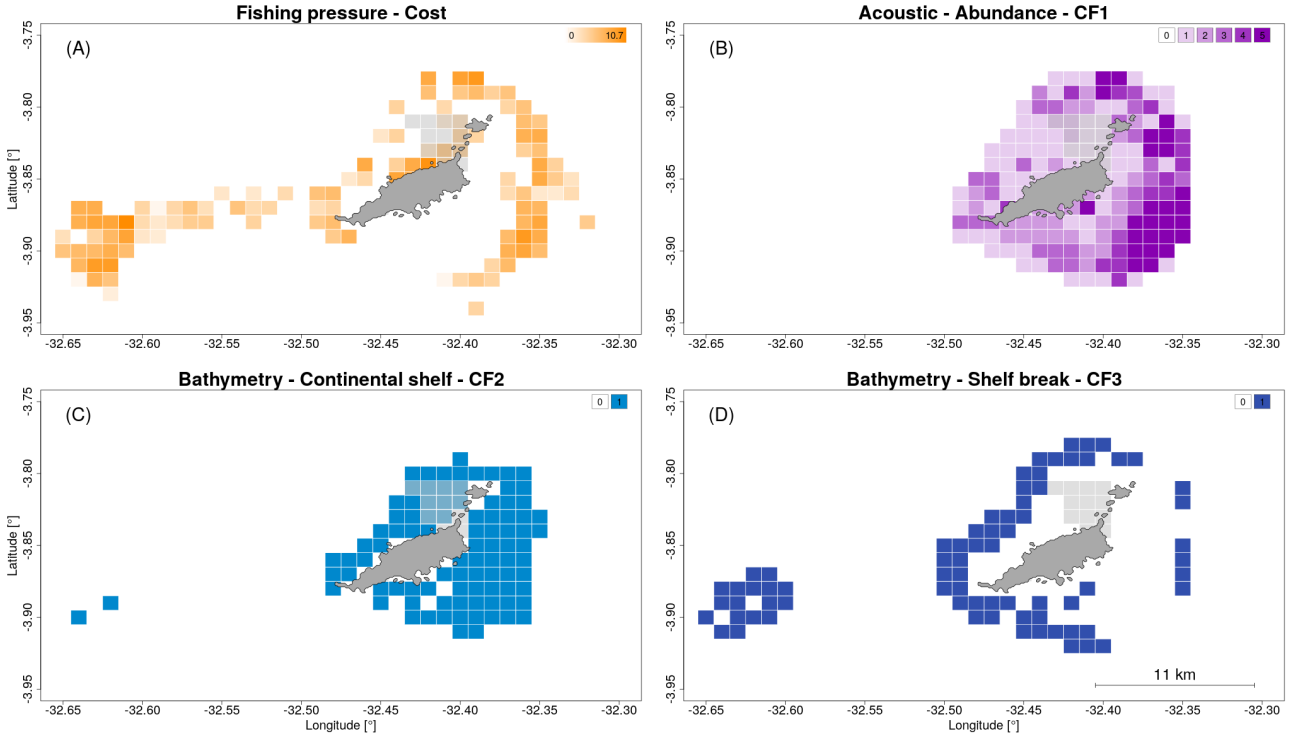
Fernando de Noronha is a small oceanic archipelago in the western tropical Atlantic, made up of 21 islands, islets and rocks with a total surface area of  $26 \text{ km}^2$ , and constituting a genuine Brazilian natural and cultural heritage. Fernando de Noronha is a conservation showcase in Brazil, but it also faces many interests (oil industry, tourism intensification, fisheries) which results in an open laboratory for MSP. 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 by Brunel et al. (2021). We only summarize the main characteristics of the dataset below.

Brunel et al. (2021) discretised the geographical area according to a rectangular grid made of  $N=36 \times 21=756$  planning units with longitude and latitude respectively in  $[32.65^\circ\text{W}, 32.30^\circ\text{W}]$  and  $[3.95^\circ\text{S}, 3.75^\circ\text{S}]$  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, Brunel et al. (2021) first estimated the values of fish biomass 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<sup>4</sup> 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.

---

<sup>4</sup>GEneral Bathymetric Chart of the Ocean, <https://download.gebco.net/>



**Figure 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.

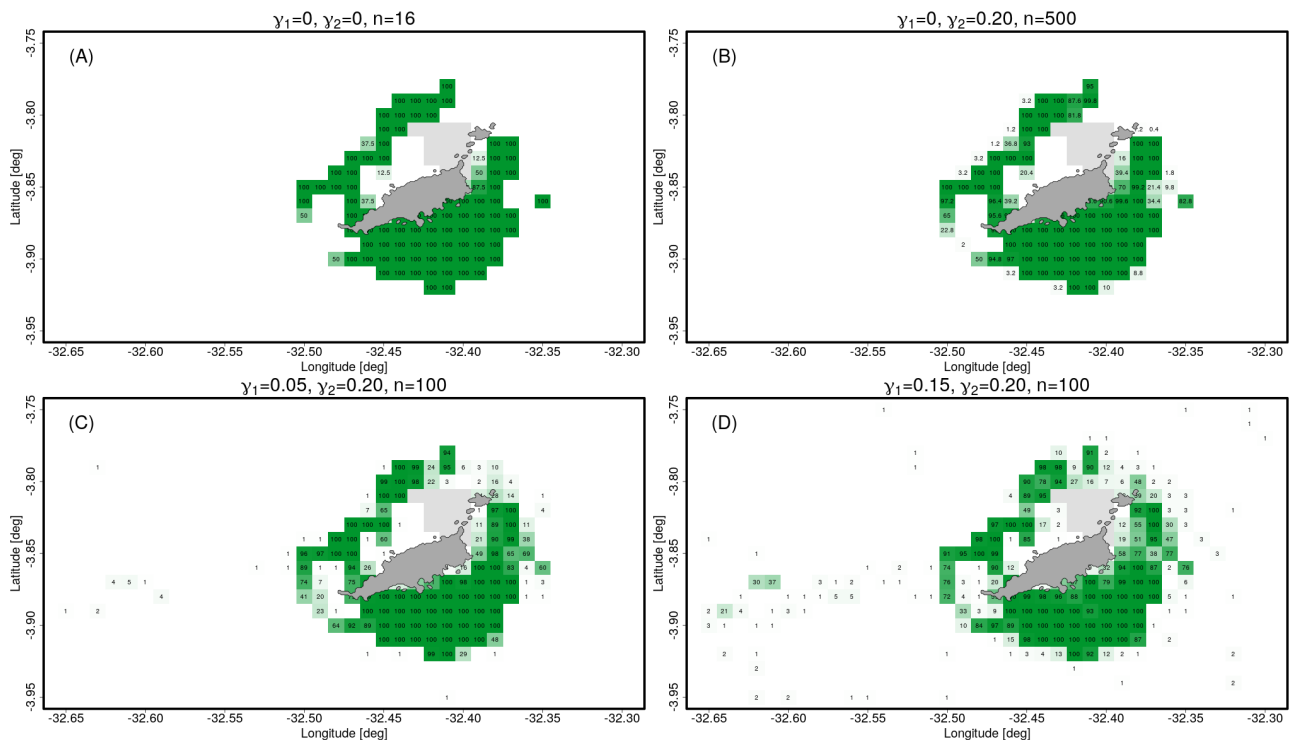
### 3.2 Alternative solutions imposing an optimality gap

**Exhaustive set of optimal solutions ( $\gamma_1 = \gamma_2 = 0$ )** We applied Algorithm 1 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  $z^* = 197.71$ . 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 1 was either stopped by  $\gamma_2$  criterion or when the maximum amount of alternative solutions  $n$  was reached. To get  $n$  alternative solutions with this algorithm, we thus set  $\gamma_2$  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  $\gamma_2$  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<sup>5</sup>. 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.

<sup>5</sup>among which we had 16 optimal solutions

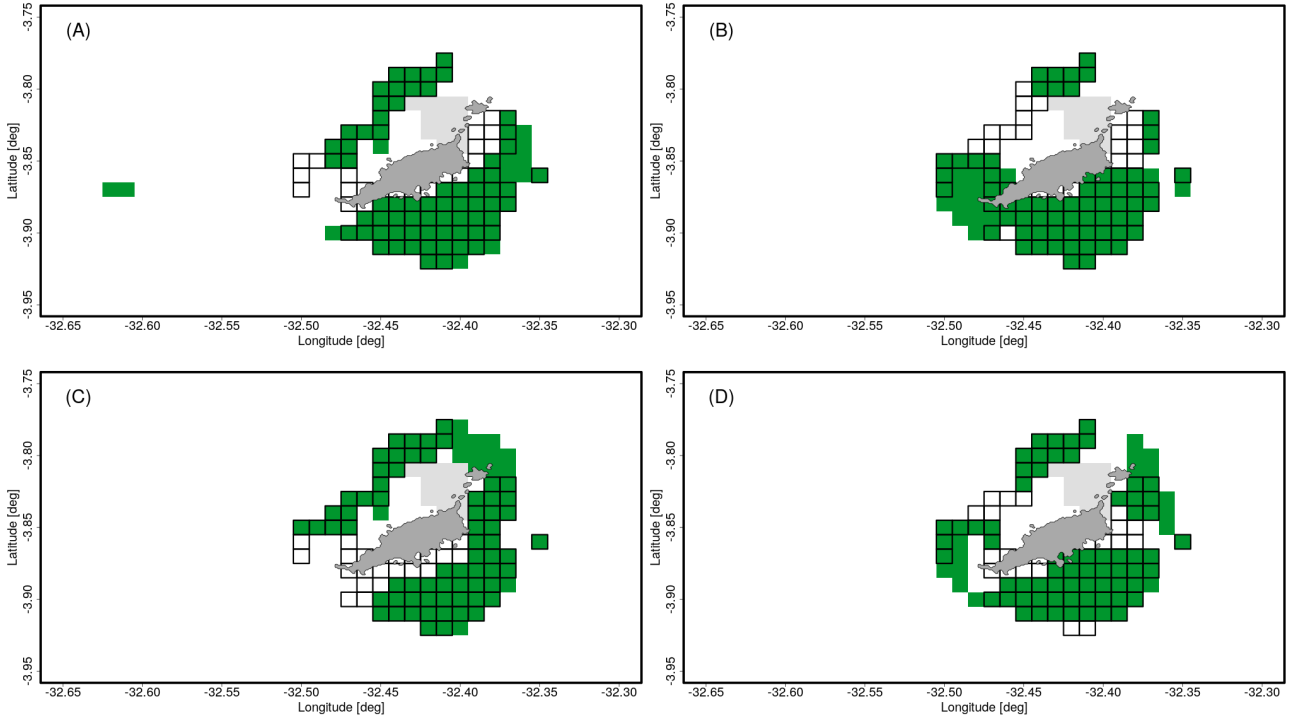
**Alternative solutions within a targeted optimality gap** ( $\gamma_1 > 0$ ,  $\gamma_2 > 0$ ) We set  $\gamma_1 > 0$  voluntarily to get suboptimal solutions where the relative optimality gap was at least  $\gamma_1$ . We chose  $\gamma_2$  high enough to have  $n = 100$  alternative solutions. Panel 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.



**Figure 5** – Selection frequency among alternative solutions obtained with Algorithm 1. 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 Alternative solutions imposing a pseudo-distance

In this section, we applied Algorithm 2 to our case study. Therefore, we directly constrained successive alternative solutions to be pseudo-distant of at least  $\delta$  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  $x^*$  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  $x^*$  were not found in the current solution. Those correspond to the white planning units with a thick black border. Indeed, Algorithm 2 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 1. 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.



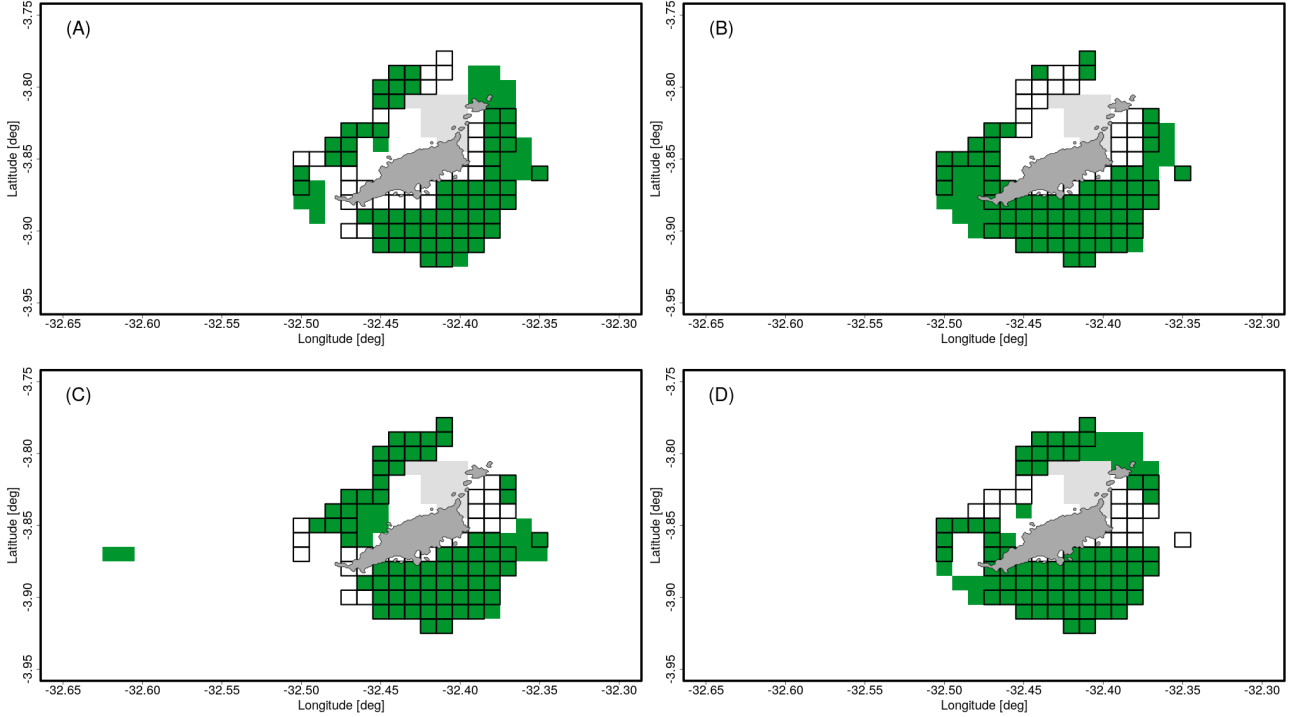
**Figure 6** – Example of alternative solutions obtained with Algorithm 2 for  $\delta = 20$ . The alternative reserve solution is represented in green, while the optimal solution  $x^*$  is depicted with planning units delimited by a thick black border.

### 3.4 Alternative solutions maximising the minimum pseudo-distance between iterates

In this numerical application, we applied Algorithm 3 to find the  $n = 4$  alternative solutions maximising the minimum pseudo-distance from past iterates. To do so, the integer linear program  $P_D^k$  was solved recursively. An example of derived reserve solutions is illustrated in Figure 7 for a relative extra cost budget  $\gamma_2 = 10\%$ . 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.

Find below more numerical details on what is realised in Algorithm 3:

- Reserve in Panel A is pseudo-distant of 27 planning units to the optimal solution  $x^*$ . The first iteration simply solves  $P_D$  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.



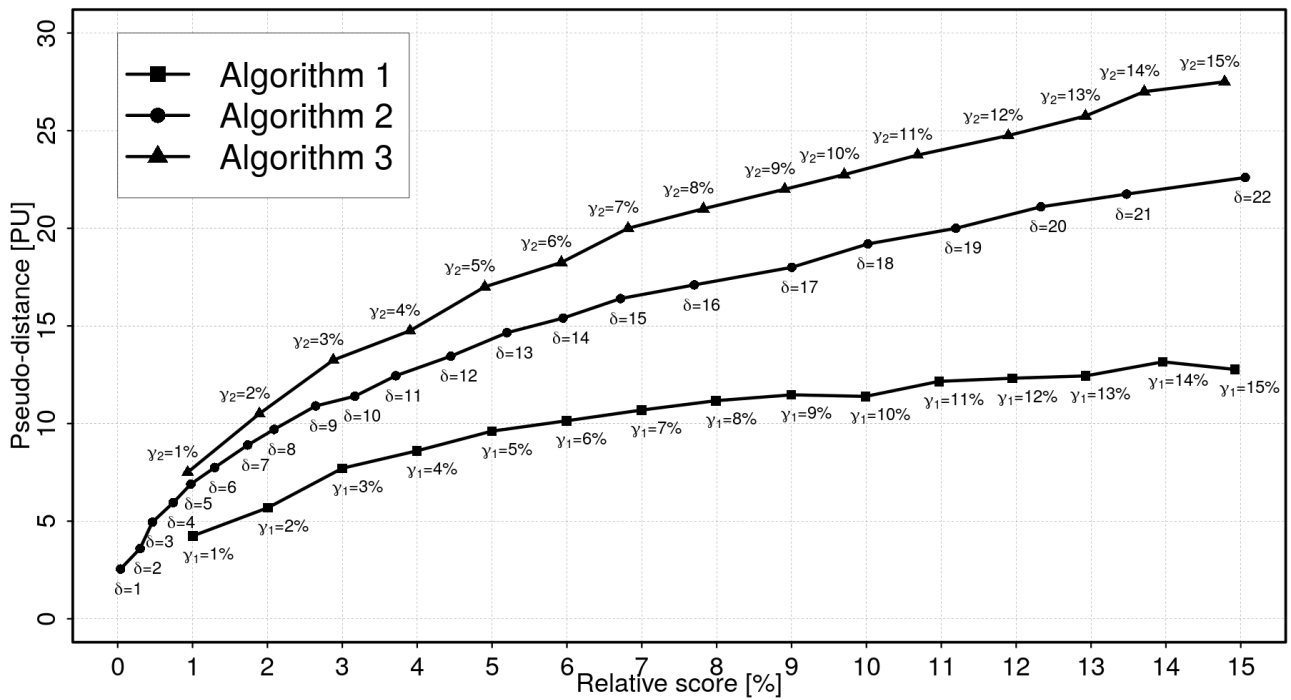
**Figure 7 – Example of alternative solutions maximising the minimum pseudo-distance to past iterates at a fixed extra cost.** We show four successive alternatives with an extra cost budget  $\gamma_2 = 10\%$ . The alternative reserve solution is represented in green, whereas the optimal solution  $x^*$  is depicted with planning units delimited by a thick black border.

- 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 Distance

We compared the mean relative scores and pseudo-distances of alternative solutions obtained with Algorithm 1, 2 and 3 for various values of parameters involved. More precisely, we performed a sensitivity analysis on  $\gamma_1$  for Algorithm 1,  $\delta$  for Algorithm 2 and  $\gamma_2$  for Algorithm 3. 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 3 because it explicitly sought to maximise the minimum pseudo-distance to past iterates. Similarly, as pseudo-distance was not considered at all in Algorithm 1 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 2 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  $n$ .





**Figure 8 – Relative score  $\gamma$  versus pseudo-distance  $d(x^*, \cdot)$ .** Algorithm 1 for  $\gamma_1 \in \llbracket 1\%, 15\% \rrbracket$  and  $n = 100$ , Algorithm 2 for  $\delta \in \llbracket 1, 22 \rrbracket$  and  $n = 20$ , Algorithm 3 for  $\gamma_2 \in \llbracket 1\%, 15\% \rrbracket$  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 find relevant alternative solutions using exact solvers in the reserve site selection optimisation problem. We proposed three procedures to build pools of alternative reserve solutions according to specific criteria. We also numerically illustrated the proposed methods on the real application example of Fernando de Noronha.

Our first algorithm allowed us to list the exhaustive set of optimal solutions, in the same spirit as in (Arthur et al. 1997). Also, our algorithm gave the  $n$  best solutions in increasing order, from best to worse, which is close to what was done in (Önal 2004). Our algorithm went further, because it was able to enumerate alternative solutions at a user-defined optimality gap starting from the lower bound. We benefited from the fact that exact solvers are able to quantify the optimality gap, while metaheuristics cannot. The exploration of the solution space based on the value of the optimality gap was thus possible. However, this flexibility built by explicitly fixing the optimality gap close to the optimal score showed a weak variability in the reserve solutions. 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 *good* and *different* alternatives. That is why, we introduced a diversity measure to build a criterion explicitly based on how different solutions are, i.e. a mathematical distance. As we wished to avoid alternative reserves where the optimal solution is encompassed, we discarded the natural absolute-value norm and preferred a pseudo-distance. We directly included in our second and third algorithm the notion of pseudo-distance in the optimisation problem to produce more diversity within the pool of alternative solutions. These algorithms effectively generated alternative solutions which were visually different and seemed to answer the need of flexibility. In addition, the flexibility was completely controlled thanks to two parameters that allowed the user to compromise between distance from an optimal solution and the optimality gap. The sensitivity analysis illustrated this compromise and thus quantifies the relevance of an alternative solution.

In our numerical application, input data feeding the optimisation problem were globally concentrated around the main island, so the diversity between alternatives was not easily caught with the naked eye. The methodology we proposed has the strength to be easily customised by any user to define other parameterisations. In particular, it could be interesting to build a diversity measure accounting for a geographical metric. For example, we could compute a diversity measure based on the true geographical distance between planning units of both reserves.

In conclusion, although metaheuristics have been historically preferred to address the reserve site selection optimisation problem, several works (Beyer et al. 2016; Schuster et al. 2020) demonstrated to what extent exact methods outperform metaheuristics in reserve score and showed they can tackle realistic instances in reasonable time. The last argument standing in favour of metaheuristics was the apparent absence of flexibility when exact approaches are applied, as it provides a unique solution. However, our work suggested several methods showing how it is possible to build a relevant flexibility through exact methods. Indeed, unlike what is commonly stated in the conservation literature (Possingham et al. 2000; Ardron et al. 2010), our work showed that exact solvers used for the reserve site selection problem can also be used to produce a pool of alternative solutions. This ability is not inherent to metaheuristics. Besides, the inclusion of explicit criteria (optimality, diversity, etc.) directly within the optimisation model allowed to build a more controlled and transparent, i.e. an accountable flexibility. By

doing so, we know the additional selection criterion at the root of the alternative solutions pool. We think this methodology opens wide perspectives for better modifications of the model, thus building a flexibility on other criteria and perhaps more relevant alternative solutions. The proposed algorithms however can be time-consuming on larger instances if many alternative solutions are required. The time consumption can be mitigated thanks to a decrease in the number of iterations. In addition, we think the number of alternatives provided should not be greater than a few units to avoid unnecessary noise in the decision-making process. For instance, if plenty of similar alternative solutions are provided, a post-processing effort is needed to sort and select a few relevant ones. Unlike Marxan, our procedures, and especially our last algorithm, needed only a few iterations to directly produce a small set of alternative solutions both good and different from each other.

The concept of flexibility itself has not been extensively discussed in the conservation literature. The flexibility was always described as a key principle and a requirement. This concept is mainly motivated by the possibility to select the final solution among a range of solutions following untold objectives such as management, geometry, socio-political priorities, etc (Pressey et al. 1993; Possingham et al. 2000; Ardron et al. 2010). Yet, these external factors should be clearly stated at the very beginning of the decision-making process. Once explicitly stated, we could attempt to incorporate such factors within the optimisation model and better support the decision. These discussions held beforehand could help to build a more adapted optimisation model revolving around the socio-ecological question, finding better input data, simulating more relevant scenarios, etc. Discussing the criteria involved in the definitive choice of a reserve would improve the accountability of the solution (Rodrigues et al. 2000). Indeed, if transparency is improved, we could potentially mitigate the ocean grabbing that can be involved when DSTs are to be used in MSP (Queffelec et al. 2021).

## 5 Appendix

### 5.1 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 (Billionnet 2007). 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 :

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

We also accounted for the correction of the compactness parameter  $\beta$  undesirable edge effect (Brunel et al. 2021), leading to the introduction of  $b^*$  where :

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

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

## References

- Ardron, Jeff A., H. P. Possingham, and Carissa J. Klein (2010). *Marxan Good Practices Handbook*.
- Arthur, Jeffrey L, Mark Hachey, and Kevin Sahr (1997). “Finding all optimal solutions to the reserve site selection problem: formulation and computational analysis”. en. In: *Environmental and Ecological Statistics*, p. 13.
- Ball, Ian R., Hugh P. Possingham, and Matthew E. Watts (2009). “Marxan and relatives: software for spatial conservation prioritisation”. In: *Spatial Conservation Prioritisation: Quantitative Methods and Computational Tools*. Vol. 14, pp. 185–196.
- Beyer, Hawthorne L. et al. (2016). “Solving conservation planning problems with integer linear programming”. en. In: *Ecological Modelling* 328, pp. 14–22. DOI: [10.1016/j.ecolmodel.2016.02.005](https://doi.org/10.1016/j.ecolmodel.2016.02.005).
- Bezanson, Jeff et al. (2012). “Julia: A Fast Dynamic Language for Technical Computing”. en. In: *arXiv:1209.5145 [cs]*. arXiv: 1209.5145.
- Bezanson, Jeff et al. (2015). “Julia: A Fresh Approach to Numerical Computing”. en. In: *arXiv:1411.1607 [cs]*. arXiv: 1411.1607.
- Billionnet, Alain (2007). *Optimisation Discrète, de la modélisation à la résolution par des logiciels de programmation mathématique*. Dunod.
- Brunel, Adrien et al. (2021). “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, <https://hal.archives-ouvertes.fr/hal-03445922>.
- Church, Richard L., David M. Stoms, and Frank W. Davis (1996). “Reserve selection as a maximal covering location problem”. en. In: *Biological Conservation* 76.2, pp. 105–112. DOI: [10.1016/0006-3207\(95\)00102-6](https://doi.org/10.1016/0006-3207(95)00102-6).
- Claudet, Joachim et al. (2008). “Marine reserves: size and age do matter”. en. In: *Ecology Letters* 11.5, pp. 481–489. DOI: [10.1111/j.1461-0248.2008.01166.x](https://doi.org/10.1111/j.1461-0248.2008.01166.x).
- Cocks, K.D. and I.A. Baird (1989). “Using mathematical programming to address the multiple reserve selection problem: An example from the Eyre Peninsula, South Australia”. en. In: *Biological Conservation* 49.2, pp. 113–130. DOI: [10.1016/0006-3207\(89\)90083-9](https://doi.org/10.1016/0006-3207(89)90083-9).
- Dunning, Iain, Joey Huchette, and Miles Lubin (2017). “JuMP: A Modeling Language for Mathematical Optimization”. en. In: *SIAM Review* 59.2. arXiv: 1508.01982, pp. 295–320. DOI: [10.1137/15M1020575](https://doi.org/10.1137/15M1020575).
- Ehler, Charles N. and Fanny Douvère (2009). “Marine Spatial Planning, A Step-by-Step Approach toward Ecosystem-based Management”. In: *Intergovernmental Oceanographic Commission and Man and the Biosphere Programme IOC Manual and Guides No. 53 ICAM Dossier No. 6*. UNESCO, Paris, pp. 99.
- Forrest, John et al. (2018). *Coin-Or/Cbc: Version 2.9.9*. DOI: [10.5281/ZENODO.1317566](https://doi.org/10.5281/ZENODO.1317566).
- Garey, Michael R. and David S. Johnson (1979). *Computers and intractability: a guide to the theory of NP-completeness*. en. 27. print. A series of books in the mathematical sciences. New York [u.a]: Freeman.
- Hanson, Jeffrey O. et al. (2019). “Optimality in prioritizing conservation projects”. en. In: *Methods in Ecology and Evolution* 10.10. Ed. by Dave Hodgson, pp. 1655–1663. DOI: [10.1111/2041-210X.13264](https://doi.org/10.1111/2041-210X.13264).
- Kirkpatrick, J.B. (1983). “An Iterative Method for Establishing Priorities for the Selection of Nature Reserves: An Example From Tasmania”. In: *Biological Conservation* 25.2, pp. 127–134.
- Margules, C. and R. L. Pressey (2000). “Systematic conservation planning”. In: *Nature* 405, pp. 243–253.

- Önal, Hayri (2004). “First-best, second-best, and heuristic solutions in conservation reserve site selection”. en. In: *Biological Conservation* 115.1, pp. 55–62. DOI: [10.1016/S0006-3207\(03\)00093-4](https://doi.org/10.1016/S0006-3207(03)00093-4).
- Possingham, H. P. et al. (2006). “Protected areas: Goals, limitations, and design”. In: *Principles of Conservation Biology*. 3rd ed., pp. 507–549.
- Possingham, H. P. et al. (1993). “The mathematics of designing a network of protected areas for conservation”. In: *National ASOR Conference*, pp. 536–545.
- Possingham, Hugh, Ian Ball, and Sandy Andelman (2000). “Mathematical Methods for Identifying Representative Reserve Networks”. en. In: *Quantitative Methods for Conservation Biology*. New York: Springer-Verlag, pp. 291–306. DOI: [10.1007/0-387-22648-6\\_17](https://doi.org/10.1007/0-387-22648-6_17).
- Pressey, R L et al. (1993). “Beyond Opportunism: Key Principles for Systematic Reserve Selection”. en. In: 8.4, p. 5.
- Pressey, R.L., H.P. Possingham, and C.R. Margules (1996). “Optimality in reserve selection algorithms: When does it matter and how much?” en. In: *Biological Conservation* 76.3, pp. 259–267. DOI: [10.1016/0006-3207\(95\)00120-4](https://doi.org/10.1016/0006-3207(95)00120-4).
- Queffelec, Betty et al. (2021). “Marine spatial planning and the risk of ocean grabbing in the tropical Atlantic”. en. In: *ICES Journal of Marine Science*. Ed. by Wesley Flannery, pp. 1–13. DOI: [10.1093/icesjms/fsab006](https://doi.org/10.1093/icesjms/fsab006).
- Rodrigues, Ana S., J. Orestes Cerdeira, and Kevin J. Gaston (2000). “Flexibility, efficiency, and accountability: adapting reserve selection algorithms to more complex conservation problems”. en. In: *Ecography* 23.5, pp. 565–574. DOI: [10.1111/j.1600-0587.2000.tb00175.x](https://doi.org/10.1111/j.1600-0587.2000.tb00175.x).
- Rodrigues, Ana S.L. and Kevin J. Gaston (2002). “Optimisation in reserve selection procedures—why not?” en. In: *Biological Conservation* 107.1, pp. 123–129. DOI: [10.1016/S0006-3207\(02\)00042-3](https://doi.org/10.1016/S0006-3207(02)00042-3).
- Sarkar, S. (2004). “Conservation Biology”. In: *The Stanford Encyclopedia of Philosophy*.
- Sarkar, Sahotra (2012). “Complementarity and the selection of nature reserves: algorithms and the origins of conservation planning, 1980–1995”. en. In: *Archive for History of Exact Sciences* 66.4, pp. 397–426. DOI: [10.1007/s00407-012-0097-6](https://doi.org/10.1007/s00407-012-0097-6).
- Schuster, Richard et al. (2020). “Exact integer linear programming solvers outperform simulated annealing for solving conservation planning problems”. en. In: *PeerJ* 8, e9258. DOI: [10.7717/peerj.9258](https://doi.org/10.7717/peerj.9258).
- Serra-Sogas, Norma et al. (2020). *Marxan User Manual: For Marxan version 2.43 and above*.
- Stolton, Sue and Nigel Dudley (2010). *Arguments for Protected Areas: Multiple Benefits for Conservation Use*.
- Tans, William (1974). “Priority ranking of biotic natural areas”. In: *Michigan Botanist*.