

Intelligent Habitat Restoration Under Uncertainty

Tommaso Urli,¹ Jana Brotánková,² Philip Kilby,¹ and Pascal Van Hentenryck¹

¹ Optimisation Research Group, NICTA Canberra Research Lab,
7 London Circuit, Canberra, 2601 ACT, Australia

² ARC Centre of Excellence for Coral Reef Studies, James Cook University,
1 James Cook Drive, Townsville, 4811 QLD, Australia

Abstract

Conservation is an ethic of sustainable use of natural resources which focuses on the preservation of *biodiversity*, i.e., the degree of variation of life. *Conservation planning* seeks to reach this goal by means of deliberate actions, aimed at the protection (or restoration) of biodiversity features. In this paper we present an intelligent system to assist conservation managers in planning habitat restoration actions, with focus on the activities to be carried out in the islands of the Great Barrier Reef (QLD) and the Pilbara (WA) regions of Australia. In particular, we propose a constrained optimisation formulation of the habitat restoration planning (HRP) problem, capturing aspects such as population dynamics and uncertainty. We show that the HRP is NP-hard, and develop a constraint programming (CP) model and a large neighbourhood search (LNS) procedure to generate activity plans under budgeting constraints in a reasonable amount of time.

Introduction

Conservation planning has always relied on the establishment of natural reserves (*spatial prioritisation*) as the principal means of enacting conservation. However, the clash with the economic interests in the areas to be preserved, and the increasing scarcity of the latter, are rendering this practice more and more difficult to justify. For this reason, *habitat restoration* and biodiversity *offsetting* practices have recently gained popularity (Maron et al. 2012) among conservation managers. These approaches try to balance out the effects of development activities by promoting environmental recovery in other areas, alleviating at least partially the negative effects of economic development on biodiversity.

Habitat restoration consists of selecting actions to perform at particular locations in order to reduce specific threats to the biodiversity features. For instance, poison baits may be placed to reduce the number of cats preying on migratory birds, or herbicide may be sprayed to reduce the occurrence of a specific weed and thus encourage the return of a particular type of grassland. One of the main sources of complexity, when making conservation management decisions in this context, is represented by population dynamics. Populations

evolve over time according to complex laws depending on natural growth rates, carrying capacity of habitats, threat factors, etc. The variety of different species, the complexity of their interactions, and the ever stringent budget limitations, make it very difficult for a human decision maker to design an plan for the habitat restoration operations which takes into account all of the available data. As a consequence, conservation managers are calling for the development of intelligent decision support systems, aimed at assisting them in the planning of these activities. The ordering of actions is also important, as their efficacy depends on the population levels at the location and time at which they are applied.

This research project is instigated by, and conducted in cooperation with, two agencies: the Queensland Parks and Wildlife Service and the Western Australian Department of Parks and Wildlife. The main aim of the study is to replace ad-hoc decision making with a more disciplined and defensible approach. It also serves as a tool to investigate and show the effect of various budgeting scenarios on the conservation objectives. The focus of protection were islands off the coast of Australia, in Queensland (in the Southern part of the Great Barrier Reef) and Western Australia (near the Pilbara region). Islands are, in many ways, ideal locations for restoration activities. They are often remote, and so access to the areas can be restricted. Also, they offer the possibility of completely eradicating a threat with a reduced probability of re-introduction. This gives long-term benefits to the protected features. Finally, islands are often less attractive locations for development, and hence the economic impact of restoration activities is rather limited.

The main contributions of this paper are: *i*) a formulation of the habitat restoration planning (HRP) as a constrained optimisation problem, featuring a population dynamics model based on real-world data being currently elicited by the conservation managers involved in this project, *ii*) a formal proof of complexity, showing that the HRP problem belongs to the NP class, *iii*) a constraint programming (CP) model built upon robust optimisation principles, and *iv*) a large neighbourhood search (LNS) scheme to efficiently find solutions to the problem.

Related work

Conservation planning has been traditionally a manual task carried out by conservation managers. In the last three

decades, however, in an attempt to make more informed decisions, a lot of research has been devoted to the study conservation planning from an optimisation perspective. While we acknowledge the vast body of research, in particular regarding spatial prioritisation and reserve connectivity (see (Williams, ReVelle, and Levin 2005) for a survey), in the following, for space reasons, we decided to focus on projects that eventually led to the development of decision support tools for conservation managers.

Marxan (Ball, Possingham, and Watts 2009) is possibly the most widespread conservation planning tool in existence. It aims at identifying a *minimum cost reserve network*, i.e., a set of conservation areas (or conservation *units*), so that certain biodiversity feature targets are met. The cost being minimised is a combination of economic, social, and potential development costs. From a computational standpoint, Marxan solves a weighted set covering problem, where each set has multiple weights representing the amount of features in a specific area. C-Plan (Pressey et al. 2009) is a conservation planning tool for reserve selection, seeking to solve the same set covering problem addressed by Marxan. C-Plan achieve this by using a measure of *irreplaceability*, i.e., the likelihood of needing any planning unit for meeting the targets. Irreplaceability tries to explain why a set of units should be selected, and is a dynamic measure: the irreplaceability score of each unit changes as more units are added or removed to the reserve. RobOff (Poulzols and Moilanen 2013) is a conservation planning tool explicitly designed for biodiversity offsetting. Its goal is to allocate resources to various conservation actions, some of which can have *uncertain effects* on biodiversity over time. Unlike Marxan and C-Plan, the analysis carried out by RobOff is non-spatial. In (LeBras et al. 2013), the authors describe a MIP and a local search approach to solve the reserve selection problem. Unlike previous works, however, the focus is more on guaranteeing connectivity between the conservation areas, which is believed to be a fundamental requirement for preserving biodiversity in the long term. The work builds on the previous literature on robust network design, and aims at connecting areas which are of interest to certain species.

Most of the approaches in literature focus their attention in identifying promising areas for prioritisation. To the best of our knowledge little effort has been spent so far in the context of habitat restoration. Furthermore, at the time of writing, none of the available tools for spatial conservation planning embed complex population dynamics in their optimisation models.

Problem

In this section we first provide a formal definition for the *habitat restoration planning* (HRP) problem in terms of problem entities, population dynamics, and objectives. We then provide some complexity results and discuss the role of uncertainty.

Entities

The following entities reflect the data being currently elicited by the conservation managers involved in this project.

Features (or biodiversity features). The features are represented by a set \mathbf{F} of (animal and plant) species that must be preserved. Each feature $ft \in \mathbf{F}$ has a *target* σ_{ft} , the desired relative increase in population.

Threats. The threats are a set \mathbf{T} of species, often introduced accidentally in the ecosystem that constitute unnatural dangers to the biodiversity features and must be reduced in order to allow the populations of features to recover.

Locations. The locations are represented by a set \mathbf{L} of independent environments that can harbour features and threats. For each location $l \in \mathbf{L}$ and each feature $ft \in \mathbf{F}$, we are given an *initial level* $x_{ft,l}$ and a *saturation level* $\bar{x}_{ft,l}$ of ft at l . The initial level represents the minimum possible amount of features at location l and reflects the assumption that, before any habitat restoration activity is carried out, the environment is in a state of equilibrium, where the populations of features are at their minimum and the threats at their maximum. Conversely, the saturation level represents what is called in biology the *carrying capacity*, i.e., the maximum amount of individuals that an enclosed environment is able to sustain in terms of resources (food, water, soil). Because of incomplete census data about threats, we can only quantify threat populations as fractions; in particular, for each threat $th \in \mathbf{T}$ and each location $l \in \mathbf{L}$, we have an *initial level* $y_{th,l} \in [0, 1]$ of th at l . The minimum and maximum levels for threat are assumed to be 0 and 1 respectively.

Actions. We are also given a set \mathbf{A} of conservation actions that target threats and can be planned at specific locations, e.g., shooting or baiting. Each action $a \in \mathbf{A}$ has a *unit cost*, which is dependent on the location $l \in \mathbf{L}$ where it is to be applied and is denoted by $c_{a,l}$. Intuitively, the cost should aggregate information about travelling to a location, operator salaries, and materials. As such, it serves as an approximation to the real cost of performing the action.

Time. Actions can be planned along a planning horizon $\mathbf{H} = \{1, \dots, h\}$ of length h . The same discretisation is used, consistently to describe the evolution of feature and threat populations in time according to population dynamics (see below). While actions can be planned on every time step $t \in \mathbf{H}$, we consider an additional starting step 0 (the state of the environment before the planning) for populations. We will refer to the full time horizon as $\{0\} \cup \mathbf{H}$ whenever needed. We denote the features and threats levels at location $l \in \mathbf{L}$ and time step $t \in \{0\} \cup \mathbf{H}$ with $x_{ft,l,t}$ and $y_{th,l,t}$ respectively.

Budget. We are given an overall maximum budget B that we can spend on actions, expressed in the same unit of measurement as the actions costs. In addition to this, we allow a per-time step budget $B_t (t \in \mathbf{H})$, since funding for conservation is often made available in a “staggered” way.

Growth rates. Each feature has a natural *growth rate* $\alpha_{ft} \in \mathbb{R}^+$, which is the speed at which the population would grow at each time step in complete absence of threats. For instance, $\alpha_{ft} = 0.6$ means that, at time step $t + 1$, the population of ft will be 60% larger than the population at time t . Similarly, each threat has a *growth rate* $\beta_{th} \in \mathbb{R}^+$. Note that both α_{ft} and β_{th} represent maximal growths; the actual

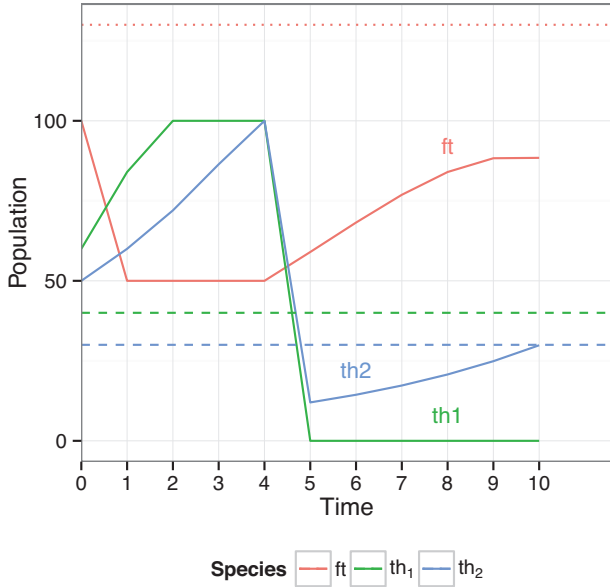


Figure 1: Population dynamics example.

increase in populations depend on the threat levels (for features) and on the applied actions (for threats).

Threat impact and action effect. Given a feature $ft \in \mathbf{F}$ and a threat $th \in \mathbf{T}$, the effect of th on ft is described with a *impact rate* $\iota_{th,ft} \in [0, 1]$, to be interpreted as the maximum fraction of population of ft eliminated by th at each time step, where the real fraction also depends on the level of the threat. Similarly, the *effect* $\epsilon_{a,th} \in [0, 1]$ of an action $a \in \mathbf{A}$ on a specific threat $th \in \mathbf{T}$ is the fraction of population of th that the action is expected to eliminate in a single time step.

For simplicity of notation, we define the set of threats impacting a given feature ft as $\mathbf{T}_{ft} = \{th \in \mathbf{T} \mid \iota_{th,ft} > 0\}$ and the set of actions impacting a given threat th as $\mathbf{A}_{th} = \{a \in \mathbf{A} \mid \epsilon_{a,th} > 0\}$.

Thresholds. When a conservation action $a \in \mathbf{A}_{th}$ is applied, the population of th decreases according to the action effect $\epsilon_{a,th}$. As a consequence, the populations of features \mathbf{F}_{th} may start to recover. In nature, this *rebound* is usually not immediate. We model this behaviour with *thresholds* $\theta_{ft,th} \in [0, 1]$ that are given for each suitable $\langle ft, th \rangle$ pair. In particular, the threshold $\theta_{ft,th}$ is the level of threat th below which we can start to see an increase in the population of ft . Since each feature ft can be threatened by multiple species \mathbf{T}_{ft} , $y_{th,l} < \theta_{ft,th}$ must hold for all $th \in \mathbf{T}_{ft}$ in order to see an increase in its population. Moreover, the amount of increase in the population of ft at a given time step t depends on the additive and multiplicative distances of the threat levels from their relative ft -thresholds. Figure 1, obtained by running the solver on an island with one feature and two threats, exemplifies the thresholding mechanism. The red curve represents the evolution of feature $ft \in \mathbf{F}$ at a location $l \in \mathbf{L}$, while the blue and green curves represent the levels of two threats $th_1, th_2 \in \mathbf{T}_{ft}$ at the same location. The dashed lines depict the thresholds θ_{ft,th_1} and θ_{ft,th_2} . At the

beginning of the planning horizon no actions are applied, the threat levels grow until they reach the carrying capacity of the island and the feature population reaches its minimum. It is not until some actions are applied, and both threats descend below their respective thresholds, that the population of ft starts to grow. As the level of th_2 grows, the increase rate in the population of ft starts to decrease, and it goes flat as soon as the threat level reaches the threshold.

Population Dynamics

The population dynamics models the evolution of populations, over time at each considered location. In the following, we will use the generic term *population* to refer both to features and threats, although their evolution follows slightly different rules. Moreover, the populations of features and threats must always lie within their respective minimum and maximum levels. Such “clamping” is easy and is excluded from the following equations for simplicity.

Threats update. In the population dynamics of threats, there are two competing processes: growth and elimination by means of actions. At each time step $t \in \mathbf{H}$, at each location $l \in \mathbf{L}$, the population of a specific threat $th \in \mathbf{T}$ evolves according to the following update rule

$$y_{th,l,t} = y_{th,l,t-1} \left(1 - \sum_{a \in \mathbf{A}_{th}} \epsilon_{a,th} \cdot z_{a,l,t} \right) \times \beta_{th}, \quad (1)$$

where $y_{th,l,t}$ is the level of threat th at location l and time t , and $z_{a,l,t}$ is the number of actions of type $a \in \mathbf{A}_{th}$ applied at location l and time t . Note that the growth applies to the population of the threat after it is reduced by the actions.

Features update. The update rule for features can be broken down in two cases

- If all threats $th \in \mathbf{F}_{ft}$ are **sub-critical**, i.e., all their populations are below their thresholds $\theta_{ft,th}$, then

$$x_{ft,l,t} = x_{ft,l,t-1} \left(1 + \alpha_{ft} \cdot \prod_{th \in \mathbf{T}_{ft}} 1 - \frac{y_{th,l,t}}{\theta_{ft,th}} \right), \quad (2)$$

where the growth of the population of ft takes into account both the natural growth rate α_{ft} of ft and the levels of threats to ft . If the threat populations are exactly the threshold, then the fraction is 1 and the population of ft does not change.

- If at least one threat $th \in \mathbf{F}_{ft}$ is **super-critical**, i.e., its population is above the threshold $\theta_{ft,th}$, then

$$x_{ft,l,t} = x_{ft,l,t-1} \times \left[1 - \min \left(1, \sum_{th \in \mathbf{T}_{ft,l,t}} \frac{y_{th,l,t} - \theta_{ft,th}}{1 - \theta_{ft,th}} \cdot \iota_{th,ft} \right) \right] \quad (3)$$

where $\mathbf{T}_{ft,l,t} = \{th \in \mathbf{T}_{ft} \mid y_{th,l,t} > \theta_{ft,th}\}$ denotes the set of threats, among the ones that affect ft , that are super-critical at location $l \in \mathbf{L}$ and time $t \in \mathbf{H}$, and the decrease of the population of ft takes into account both the impact $\iota_{th,ft}$ and the level $y_{th,l,t}$ of each threat. If the threat populations are exactly on the threshold, then population of ft does not change.

Model soundness. No ecosystem model can provide a completely accurate representation of the complex dynamics occurring in nature. Indeed, our goal is not to represent the environment faithfully, but rather to provide a useful approximation that enables more informed decisions. Our model relies on a number of parameters to control the simulation, i.e., thresholds, growth rates, impacts, and effects. Of these, thresholds are an artificial parameter, introduced to model the delayed rebound of feature populations upon the reduction of threats. Growth rates, impacts, and effects can be estimated by historical data and expert consultation. Given that this information can be noisy, our model acknowledges and explicitly addresses uncertainty in form of confidence intervals (see section “Uncertainty”).

In order to assess the soundness of our model, we consulted two experts in population dynamics, and asked them to review it. While the experts agreed that our model represents a solid example of ecological system modelling, they both identified some shortcomings that should be addressed in the future, in particular: *i*) the potential difficulty in eliciting the necessary data, *ii*) the necessity of modelling more complex interaction between species (competition, meso-predator release, etc.), and *iii*) the necessity to address geographical aspects (connectivity, reintroduction, dispersal, etc.). The elicitation of the population data is an ongoing process, and our model is consistent with the data currently available to us. Having said that, we expect that minor adaptations to the model will be necessary once the final data is available. Modelling more complex interactions between species requires a radical redesign of the population dynamics, and is the subject of our future research. Regarding geographical aspects, our model is designed to plan activities to be carried out in an insular context, therefore reintroduction and dispersal are relatively low-probability phenomena, however this probability is not zero. We will discuss these aspects in the “Conclusions and future work” section.

Objective

The habitat restoration planning problem has two conflicting goals. On the one hand, we would like to reach the conservation targets for all the features. On the other hand, we would like not to exceed the conservation budget. The first goal can be formalised with the constraint

$$\sum_{l \in \mathbf{L}} x_{ft,l,h} \geq \sigma_{ft} \cdot \sum_{l \in \mathbf{L}} x_{ft,l,0}, \quad \forall ft \in \mathbf{F}. \quad (4)$$

where $h - 1 = \max(\mathbf{H})$ is the latest time step in the planning horizon and σ_{ft} are the desired relative increases in the feature populations. Similarly, the second goal can be stated as

$$\sum_{t \in \mathbf{H}} \sum_{l \in \mathbf{L}} \sum_{a \in \mathbf{A}} c_{a,l} \cdot z_{a,l,t} \leq B. \quad (5)$$

where c_a is the unit cost of each action, B is the total budget, and $z_{a,l,t}$ represents the number of times action a was applied at time t and location l . In this formulation, we also consider the possibility that, because of the staggered availability of funding, at each time step t , the maximum amount of budget to spend is limited by a per-time step budget B_t .

This constraint strengthens the previous one and can be expressed with

$$\sum_{l \in \mathbf{L}} \sum_{a \in \mathbf{A}} c_{a,l} \cdot z_{a,l,t} \leq B_t \quad \forall t \in \mathbf{H}. \quad (6)$$

Since budget is the most stringent constraint in the context of conservation, we decided to consider Equations 5 and 6 as hard constraints and to transform Equation 4 into an objective function to minimise.

Our objective function is a weighted sum of a quadratic term penalising solutions that fail to reach the conservation targets (*conservation* term) and of a linear term promoting solutions that improve over the conservation targets (*improvement* term)

$$\begin{aligned} \text{minimise} \quad & w_{cons} \sum_{ft \in \mathbf{F}} [\max(0, \Delta_{ft})^2] \\ & - w_{impr} \sum_{ft \in \mathbf{F}} [\max(0, -\Delta_{ft})], \end{aligned} \quad (7)$$

where

$$\Delta_{ft} = \sigma_{ft} - \frac{\sum_{l \in \mathbf{L}} x_{ft,l,h}}{\sum_{l \in \mathbf{L}} x_{ft,l,0}}$$

is the overall relative increase in the population of $ft \in \mathbf{F}$ since the start. The terms of the objective function are weighted differently, so to prefer meeting the targets for all features, rather than exceeding the targets only for some of them; w_{cons} is the relative importance of reaching the targets (σ_{ft}), and w_{impr} is the relative importance to improve on the targets. In our setup $w_{cons} = 10^4$ and $w_{impr} = 1$.

Uncertainty

An issue in conservation planning is that information is often uncertain. Population levels, for instance, are usually only known as confidence intervals. For this reason, we *allow* (but not *require*) some of the input data of our problem formulation to be expressed as intervals, including the population levels ($x_{ft,l,t}$ and $y_{th,l,t}$), the effect of actions on threats ($\epsilon_{a,th}$), and the impacts of threats on features ($\iota_{th,ft}$).

We will show that our constraint-programming model takes this kind of uncertainty into account and that the generated solutions acknowledge uncertainty explicitly. We also discuss an approach that we initially explored to represent uncertainty and the limitations that motivated why it was not considered appropriate. Figure 2 shows a solution for a problem involving uncertainty. The uncertainty (at step 0 and in the action effects) is propagated along the planning horizon to enclose all the values that the population levels can take as various decisions are taken. In practice, we maintain population levels for the *best-* and *worst-case* scenarios; all the intermediate values are possible realisations of the populations. Optimisation is carried out on the worst-case scenario, nevertheless taking into account all the possible outcomes at each step allows us to provide managers with an informative representation of the effect of a given conservation plan.

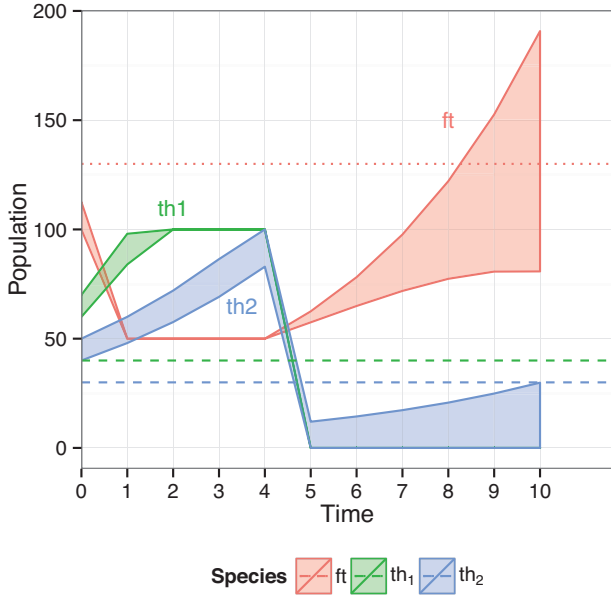


Figure 2: Population dynamics example with uncertainty.

Complexity

We briefly show that the HRP problem, in its decision variant, i.e., the problem of finding out whether it is possible to reach given feature targets subject to budget constraints, is a complex computational task, since it is NP-hard. We sketch the proof that reduces the Set Covering Problem (SCP), whose decision variant is known to be NP-complete (Karp 1972), to a special case of the HRP problem.

The SCP is stated as follows. Given a set of elements $U = \{1, 2, \dots, m\}$, called the *universe*, a family $S = \{s_1, s_2, \dots, s_n\}$, $s_i \subseteq U$ of n sets such that $\bigcup_{i \in \{1, \dots, n\}} s_i = U$, and an integer k , find a sub-family $C \subseteq S$ of cardinality at most k , so that the union of the sets in C is the universe. Given an instance of the SCP, we can reduce it to an instance of HRP in the following way

- we consider a horizon $H = \{1\}$ of 1 time step ($\cup \{0\}$),
- for every set $s \in S$ we instantiate a location $l_s \in L$,
- for every element $e \in U$ we instantiate a feature $ft_e \in F$, whose growth rate is $\alpha_{ft_e} = 1$, i.e., at each time step, in absence of threats, the population doubles,
- at every location $l_s \in L$, at time step $t = 0$, we set the population of each feature $ft_e \in F$ to the minimum ($\underline{x}_{ft_e, l_s}$) which we define as

$$\underline{x}_{ft_e, l_s} = \begin{cases} 1 & \text{if } e \in s \\ 0 & \text{otherwise} \end{cases}$$

- saturation levels \bar{x}_{ft_e, l_s} are set to 2, although it is impossible for the populations to grow by more than one in one time step. Also, as a consequence, it does not make sense to apply an action more than once for each location,
- we consider a single threat $th \in T$, whose impact rate ι_{th, ft_e} on each feature $ft_e \in F$ is 1, and whose threshold is $\theta_{ft_e, th} = 1$,

- we consider a single action $a \in A$, with cost $c_{a, l} = 1 \forall l \in L$ and effect $\epsilon_{a, th} = 1$, i.e., when applied at a location $l_s \in L$, the action completely eradicates th at that location,
- for every $ft_e \in F$, the target σ_{ft_e} is defined by

$$\sigma_{ft_e} = 2,$$

that is, we want the population of each feature to double.

- we finally set the budgets $B = B_1 = k$.

Because of Equation 1, applying the action at a specific location l_s , brings the level of threat to 0, at the cost of 1 budget unit. This allows all the populations of the features at l_s to develop at their maximum potential, according to Equation 2. Since $\alpha_{ft_e} = 1$ and $x_{ft_e, l_s, 0} = \underline{x}_{ft_e, l_s} = 1, \forall ft_e \in F, l_s \in L$, in absence of threat th , all the populations of features ft_e at l_s becomes $x_{ft_e, l_s, 1} = 2$ at the next time step, reaching the target. Given the budget constraint, we can “weed out” at most k locations, which corresponds to choosing k sets to cover the universe. We have therefore shown that, by using the above reduction, we can use an HRP solver to solve any SCP instance. The above mapping, together with the main theorem in (Karp 1972), concludes the proof that the HRP problem is NP-hard.

Our approach

In the following we outline a constraint programming (CP) model for the HRP problem. We then introduce a large neighbourhood search (LNS) scheme to obtain solutions from it in a practical amount of time. The choice of CP was mainly driven by the non-linearity of the constraints, which rules out approaches based on linear optimisation, e.g., LP or MIP.

Model

Our CP model derives immediately from the presented problem formulation, therefore we will avoid an extensive discussion of the constraints, and focus on the handling of uncertainty.

Variables. The decisions variables of this problem are the number of actions of each type $a \in A$ to be executed at each location $l \in loc$ at each time step $t \in H$, and are denoted by $z_{a, l, t}$. Such variables are integral and upper-bounded by a value

$$\min\left(\frac{\min(B, B_t)}{c_{a, l}}, \lceil \epsilon_{a, l}^{-1} \rceil\right),$$

dependent on the budget available at time t and on the smallest effect $\epsilon_{a, l}$ an action a has on any threat at location l , defined as

$$\epsilon_{a, l} = \min_{th \in T | y_{th, l, 0} > 0} \epsilon_{a, th}.$$

This bounding is conservative, i.e., it doesn't exclude solutions, and greatly reduces the search space, avoiding the exploration of solutions where an action is applied a lot of times but without effect.

The population levels, except for the ones at time step zero which are fixed to the input values, are auxiliary variables

updated by the solver through constraint propagation. In order to properly handle uncertainty, each population level $x_{ft,l,t}$ (resp. $y_{th,l,t}$) is represented by an interval, i.e., two variables $\underline{x}_{ft,l,t}$ and $\bar{x}_{ft,l,t}$ (resp. $\underline{y}_{th,l,t}$ and $\bar{y}_{th,l,t}$) denoting the minimum and maximum population levels at each step.

In a previous attempt, we tried to model uncertainty using a single variable for each population level, treating its domain as a confidence interval, and transforming the update rules in inequalities. Unfortunately, the semantics of constraint propagation is unsuitable for use in this context. To understand why, imagine a search procedure where the cost is upper-bounded by adding constraints dynamically, e.g., branch & bound. The propagation triggered by the update in the upper bound of the cost updates the minimum feature levels. However, because the update rules are inequalities, the minimum feature levels can be raised until they reach the maximum feature levels without any changes in the threat levels. In other words, the mechanical explanation for the minimum feature levels is lost, subsumed by the new cost bound.

Constraints. The constraints encapsulate the update rules in Equations 1, 2, and 3 but are adapted to deal with the two-variable modelling of uncertainty. In particular, we maintain the worst- and the best-case scenarios using two variants of each constraint, differing in what extreme of each uncertainty interval is used. The best-case scenario variant of the constraints are generated from the update rules by replacing $x, y, \epsilon,$ and ι with $\bar{x}, \bar{y}, \bar{\epsilon},$ and $\bar{\iota}$. The worst-case scenario variant of the constraints are generated from the update rules by replacing $x, y, \epsilon,$ and ι with $\underline{x}, \underline{y}, \underline{\epsilon},$ and $\underline{\iota}$. Moreover, all population levels are clamped to lie within the location limits. The objective function is the worst-case scenario variant of Equation 7.

Search

Large neighbourhood search (LNS) (Shaw 1998; Ropke and Pisinger 2006) is a neighbourhood search meta-heuristic based on the idea of iteratively relaxing a subset of the decision variables of a solution (*destroy* step), and re-optimising them until a better solution is generated (*repair* step). The *large neighbourhood* is represented by all the possible solutions that can be reached by re-optimising a relaxed one. In order to make the exploration efficient, LNS is often coupled with filtering techniques aimed at containing the size of the neighbourhood by culling low quality or unfeasible solutions. We use the presented CP model as a filtering engine, a technique successfully applied to a range of combinatorial optimisation problems, also in the field of computational sustainability (Di Gaspero, Rendl, and Urli 2015).

Initial solution. LNS requires starting from a feasible initial solution. In order to identify it we perform a tree-search in the search space of the above CP model, using a rather simple branching strategy which chooses the first unassigned decision variable and assigns it the lowest value in its domain.

Destroy step. At each iteration, we relax all the decision variables related to δ islands selected uniformly at random,

where δ (the *destruction rate*) is updated during the search according to the following adaptive strategy. At the beginning of the search δ is set to 1; if the solver spends i_{max} iterations without finding a better solution, where i_{max} is a parameter of the solver, δ is increased by 1. If an improving solution is found, or if the solver has spent i_{max} iterations at $\delta = \delta_{max}$, where δ_{max} is the number of locations, δ is reset to 1. In the latter case, the search is restarted from a new initial solution but the cost is upper bound by the cost of the current incumbent.

Repair step. We re-optimize the relaxed solution by running a branch & bound tree search with a random branching strategy, subject to the constraint that the cost of any new solution must be smaller than the cost of the incumbent (best solution found so far). Each branch & bound search has a time budget of $n \cdot t_{var}$, where n is the number of relaxed variables, and t_{var} is a parameter of the solver.

Experimental results

Our solver was implemented in GECODE 4.4.0 (Schulte, Tack, and Lagerkvist 2015), and has been tested by exploring the effects of different budgeting scenarios on a *master* instance involving 13 islands, 46 features, and 19 threats. The chosen horizon for this validation was 10 periods. The instance is generated from incomplete survey data that will be fully available only during the next year, and represents only about 10% of the extent of the real problem faced by conservation managers. The missing data, namely the impact of actions on threats $\epsilon_{a,th}$, and the effect of threats on features $\iota_{th,ft}$, have been replaced with synthetic values. The validation experiments were run on a Linux cluster of 2×2.8 GHz AMD 6-Core Opteron 4184 nodes with 64 GB of RAM each, using a time limit of 1 hour per optimisation run.

The solver parameters were tuned automatically through an *F-Race* (Birattari et al. 2010). The tuning experiments were run on 800 random extracts (subsets) of the master instance, involving 70 – 100% of the islands and budgets $B \in \{0.5, 1, 2, 5\}$ millions. The tuning procedure fixed the parameters to $i_{max} = t_{var} = 50$. The detail of the tuning process are beyond the scope of this paper. The tuning experiments were run on a 2.9 GHz Intel Core i7 with 8 GB of RAM running MacOS 10, using a time limit of 5 minutes.

Figure 3 shows some preliminary results of this validation. From the plot, the correlation between the budget increase and the quality of the produced plans is quite clear. In particular, when the budget is large enough, i.e., $> 1M$, the solver manages to produce plans that meet almost always all the conservation targets. The experiments reveal a *diminishing returns* effect when the budget becomes larger, suggesting that the solver can be also used to estimate the needed conservation budget, a useful information to justify funding requests.

Since none of the available decision support tools for conservation addresses the HRP problem as we formulated it, a quantitative comparison with other approaches is impossible. However, as the model evolves, we plan to compare different optimisation techniques. Moreover, a qualitative comparison with the other available tools will be carried out as a

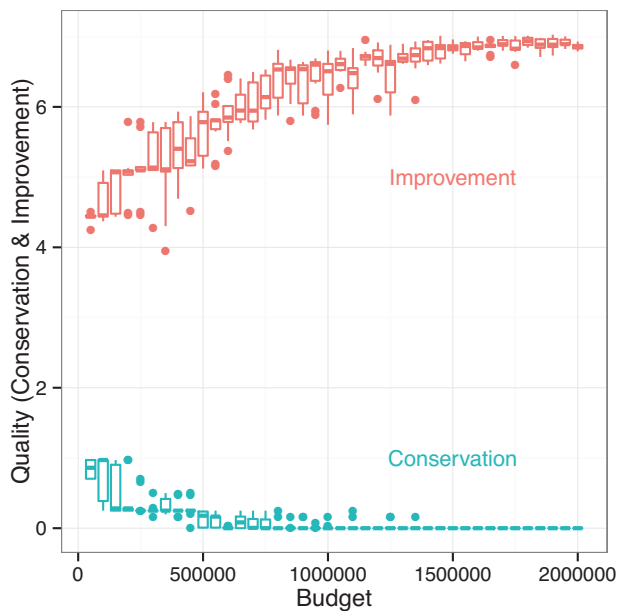


Figure 3: Effect of budgeting. The turquoise and the pink box plots represent the distributions, over 10 random independent runs, of the *conservation* and *improvement* (scaled by 0.05 for clarity) terms of the objective function.

part of our future research.

Conclusions and future work

We have presented a constrained optimisation formulation of the habitat restoration planning (HRP) problem and shown that it belongs to the NP complexity class. We have developed a constraint programming (CP) model which explicitly acknowledges and addresses uncertainty, and a large neighbourhood search (LNS) search scheme to produce feasible solutions in a reasonable amount of time. We have produced some preliminary results, showing the effect of different budgeting scenarios on the plan produced by our solver, and the general practicability of the approach.

Our population model has been evaluated by two experts in population dynamics, who have identified some aspects to consider for the future iterations of this project. In particular, we plan to introduce a unitary modelling for threats and features, which will allow us to model realistic interactions between species, such as competition and mesopredator releases, in a more natural way. Moreover, we plan on introducing reintroduction and dispersal probabilities that will especially affect the islands closer to the coast, e.g., Curtis Island. Finally, more real-world data should become available in the near future, enabling us to test the scalability of our approach on a real-scale scenario.

References

Ball, I. R.; Possingham, H. P.; and Watts, M. E. 2009. Marxan and relatives: Software for spatial conservation prioritisation. In Moilanen, A.; Wilson, K. A.; and Possingham, H. P., eds., *Spatial conservation prioritisation: Quantitative*

methods and computational tools. Oxford, UK: Oxford University Press. 195–195.

Birattari, M.; Yuan, Z.; Balaprakash, P.; and Stützle, T. 2010. F-Race and iterated F-Race: An overview. In *Experimental methods for the analysis of optimization algorithms*. Springer. 311–336.

Di Gaspero, L.; Rendl, A.; and Urli, T. 2015. Balancing bike sharing systems with constraint programming. *Constraints* 1–31.

Karp, R. M. 1972. *Reducibility among combinatorial problems*. Springer.

LeBras, R.; Dilkina, B. N.; Xue, Y.; Gomes, C. P.; McKelvey, K. S.; Schwartz, M. K.; Montgomery, C. A.; et al. 2013. Robust network design for multispecies conservation. In *AAAI*.

Maron, M.; Hobbs, R. J.; Moilanen, A.; Matthews, Jeffrey W. Christie, K.; Gardner, T. A.; Keith, D. A.; Lindenmayer, D. B.; and McAlpine, C. A. 2012. Faustian bargains? restoration realities in the context of biodiversity offset policies. *Biological Conservation* 155:141–148.

Poulzols, F. M., and Moilanen, A. 2013. Roboff: software for analysis of alternative land-use options and conservation actions. *Methods in Ecology and Evolution* 4:426–432.

Pressey, R. L.; Watts, M. E.; Barret, T. W.; and Ridges, M. J. 2009. The c-plan conservation planning system: Origins, application, and possible futures. In Moilanen, A.; Wilson, K. A.; and Possingham, H. P., eds., *Spatial conservation prioritisation: Quantitative methods and computational tools*. Oxford, UK: Oxford University Press. 211–234.

Ropke, S., and Pisinger, D. 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science* 40(4):455–472.

Schulte, C.; Tack, G.; and Lagerkvist, M. Z. 2015. Modeling. In Schulte, C.; Tack, G.; and Lagerkvist, M. Z., eds., *Modeling and Programming with Gecode*. Corresponds to Gecode 4.4.0.

Shaw, P. 1998. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming—CP98*. Springer. 417–431.

Williams, J. C.; ReVelle, C. S.; and Levin, S. A. 2005. Spatial attributes and reserve design models: A review. *Environmental Modeling & Assessment* 10(3):163–181.