Planning Habitat Restoration with Genetic Algorithms

Jana Brotankova
James Cook University
Townsville QLD 4811, Australia
jana.brotankova@jcu.edu.au

Tommaso Urli
Philip Kilby
NICTA / CSIRO Data61 and the Australian National University
Canberra ACT 2601, Australia
tommaso.urli@nicta.com.au
philip.kilby@nicta.com.au

ABSTRACT

Conservation is an ethic of sustainable use of natural resources which focuses on the preservation of biodiversity. The term *conservation planning* encompasses the set of activities, typically carried out by conservation managers, that contribute to the attainment of this goal. Such activities can be preventive, such as the establishment of conservation reserves, or remedial, such as the displacement (or *offsetting*) of the species to be protected or the culling of invasive species. This last technique is often referred to as *habitat restoration* and, because of its lower impact on economic activities, is becoming more and more popular among conservation managers. In this paper we present the original formulation of the habitat restoration planning (HRP) problem, which captures some of the decisions and constraints faced by conservation managers in the context of habitat restoration. Example scenarios are drawn from the insular Great Barrier Reef (QLD) and Pilbara (WA) regions of Australia. In addition to the problem formulation, we describe an optimisation solver for the HRP, based on genetic algorithms (GAs), we discuss the preliminary results obtained by our solver, and we outline the current and future directions for the project.

Keywords

Conservation Planning; Population Dynamics; Habitat Restoration; Genetic Algorithms; Combinatorial Optimisation

1. INTRODUCTION

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* activities seek to reach this goal by means of deliberate actions aimed at the protection (or restoration) of specific biodiversity features. Traditional conservation planning relies on the establishment of conservation areas as the principal means of enacting conservation. The main problem with this approach is that it often clashes with economic and development interests in the areas to be preserved, and thus is becoming more and more difficult to justify. For this reason, although there is no definitive consensus among the conservation managers *habitat restoration* and *biodiversity offsetting* activities have recently gained popularity [7], as they can compensate, at least partially, for the environmental effects of economical and businesses development without hindering the potential for earnings.

Conservation planning has been for a long time a manual task carried out by conservation managers with a broad expertise on the field. In the last three decades, however, in an attempt to make more informed and coordinated decisions at the regional and national scale, a number of software solutions have been developed to assist managers in their work. Most of these solutions focus on spatial prioritisation strategies, i.e., the establishment of new conservation areas, however relatively little effort has been spent so far in the context of habitat restoration and biodiversity offsetting.

Conservation planning, as intended here, consists largely of selecting *actions* to perform at particular locations, in order to reduce the level of one or more threats to a particular feature. For instance, poison baits may be placed in order to reduce the number of cats (threats) that are predating on migratory birds (features), or poison may be sprayed to reduce the occurrence of a specific weed, to encourage the return of a particular type of grassland. Given staffing and budget limits, the ordering of these actions is also important. Conservation planning can then be seen as a type of scheduling problem with an added complexity of selecting which actions to perform. An additional complexity in the model studied here is represented by population dynamics. Feature populations evolve in a non-linear fashion in response to threat populations over time. In turn, the spread of threats depends non-linearly by the activities carried out by conservation planners. This complexity is at the heart of habitat restoration planning problem, and must be taken into account in the evaluation of potential action plans. The population model used here was specifically designed for use in conservation planning, and has been developed in collaboration with conservation managers. It has relatively modest data requirements, which is appropriate in a setting where potentially hundreds of actions, features and threats might be considered.

A formal proof of complexity, derived from an extended formulation [14] of the one presented in this paper, shows that the generalised HRP problem is NP-hard by polynomially reducing the set covering problem (SCP) to it. The

GECCO '16, July 20 - 24, 2016, Denver, CO, USA

ISBN 978-1-4503-4206-3/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2908812.2908957

complexity results also hold for the formulation presented in this paper.

This research project was 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 to demonstrate to government the effect of different funding decisions. 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 eliminating a threat with much reduced probability of re-introduction. This gives long-term benefit to the protected features. And finally, islands are often of low economic (production) value, and hence the economic impact of restoration activities is limited.

The main contribution of this paper is the first description and formulation of the habitat restoration planning (HRP) problem, an optimisation problem based on the constraints and decisions involved in carrying out habitat restoration activities[1]. Our formulation addresses the problem of planning conservation actions over a set of independent environments (islands). A key feature is an embedded population dynamics model that mimics the evolution of the populations of threats and features over time in response to the environment state.

A second contribution is a description of a model for the HRP, based on genetic algorithms (GAs). This model has been prototyped in Matlab, and tested on a scenario based on survey data from the insular environments of the Great Barrier Reef (QLD) region. Early results from this solver are presented.

The paper is organised as follows. In Section 2 we overview the previous work in the literature about computer-assisted conservation planning. In Section 3 we formally describe the habitat restoration planning problem in mathematical terms. In Sections 4 and 5 we describe our solution approach for the said problem, and provide some preliminary results based on survey data. Finally, in Section 6, we provide a summary of our contributions and we outline directions for future research.

2. RELATED WORK

In this section we review some of the most widely adopted software tools developed to assist conservation managers, pointing out their focus and their strengths. While we acknowledge the vast body of research, in particular regarding spatial prioritisation and reserve connectivity (see [16] for a survey), in the following we focus on projects that eventually led to the development of decision support tools for conservation managers.

Marxan [1] is possibly the most extensively used conservation planning tool in existence. Its focus is towards identifying a minimum cost *reserve network*, i.e., a set of conservation areas (or *planning units*), so that particular biodiversity feature targets are met. 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. Connectivity between areas is also taken into account. The problem is solved through a simulated annealing (SA) [5] approach.

C-Plan [9] is another conservation planning tool for reserve selection, seeking to solve the same set covering problem as Marxan, but using a measure of *irreplaceability*, i.e., the likelihood of needing any planning unit for achieving the targets, as a weight. C-Plan handles acquisition and potential development costs on a per-unit fashion, but it does not take into account issues such as connectivity, which is left as a choice to the decision maker. C-Plan includes a geographical information system (GIS) for visualisation, and can be used as an interface for Marxan through a plug-in.

In [6] the authors describe a mixed-integer programming model (MIP) and local search approach to solve the reserve selection problem. Unlike previous works, however, the focus is more on guaranteeing connectivity between the conservation areas, rather than meeting specific feature targets. This focus reflects a common trend in conservation sciences, that identifies connectivity as a prominent aspect, in reserve selection, to increase the resilience of populations, and increase their longevity. Moreover, the work builds on the previous literature on robust network design, and the solver tries to minimise the acquisition cost of the various conservation units.

RobOff [8] is a conservation planning tool explicitly designed for (robust) biodiversity offsetting. The goal of the software is to allocate resources to various conservation actions some of which can have uncertain effects on the biodiversity features. Unlike Marxan, C-Plan, and the approach presented here and in [14], the analysis carried out by RobOff is non-spatial.

In [14] an extended and richer version of the formulation presented here has been addressed using a Large Neighbourhood Search (LNS, [11]) approach based on a Constraint Programming (CP, [12]) model. The extended version generalises the concept of conservation action, allows a finer-grained handling of budgeting, and employs a robust optimisation technique to cope with uncertainty in the data, and aspect not covered in the present (original) formulation.

Our work shares some similarities with RobOff, in that it focuses on the efficient allocation of resources to conservation actions, rather than on spatial prioritisation. Unlike RobOff, however, we take into account spatial information about the planning areas, i.e., our environments evolve independently, and we capture the behaviour of populations over time using a population dynamics model.

3. PROBLEM FORMULATION

This section provides the first formal definition of the *habitat restoration planning* (HRP) problem. This problem formulation stems from the real-world need, for conservation managers in the Pilbara (WA) and Great Barrier Reef (QLD) regions of Australia, to plan habitat restoration activities in a set of independent insular environments.

[1]Since this formulation was originally defined, an extended and more general version of it [14] has been developed and addressed with a Large Neighbourhood Search (LNS, [11]) approach based on a Constraint Programming (CP, [12]) model. The extended version allows one to apply the same actions several times at the same location and time step, and includes means to acknowledge and handle uncertainty in the data.

3.1 Entities

The problem involves a number of distinct entities and parameters. We describe them in the following, and leave the discussion of their interactions to the next section.

- **Features** (or **biodiversity features**), a set $\mathbf{F}$ of (animal and plant) species that constitute the subjects of conservation. Each feature $ft \in \mathbf{F}$ has a *target* σ_{ft}, the desired relative increase in population relative to the initial population. For instance, a target of 1.3 represent a 30% increase over the initial population.
- **Threats**, a set $\mathbf{T}$ of species, often introduced accidentally in the ecosystem, e.g., pests or weeds, that constitute unnatural threats to the features, and that must be reduced in order to allow the populations of features to recover. Given a feature $ft \in \mathbf{F}$, the set $\mathbf{T}_{ft}$ represents the set of threats that affect the population of ft. Symmetrically, the set $\mathbf{F}_{th}$ represents the set of features endangered by a particular threat $th \in \mathbf{T}$.
- **Locations**, a set $\mathbf{L}$ of independent environments (islands) that can host features and threats. For each location $l \in \mathbf{L}$, and each feature $ft \in \mathbf{F}$, we are given a *minimum level* $\underline{x}_{ft,l}$ and a *saturation level* $\bar{x}_{ft,l}$ of ft at l. The current population level reflects the necessary assumption that, without any human intervention, each environment is in a state of equilibrium, where feature populations are at their minimum, and the threat populations are at their maximum. The rationale behind this assumption is that in absence of human intervention, the threats thrive because there is nothing stopping them and the features suffer because threats prey on them. Because of incomplete census data about threats, we can only identify presence or absence of threats. Therefore we quantify threat populations as fractions; for each threat $th \in \mathbf{T}$ and each location $l \in \mathbf{L}$, we have a *maximum level* $\bar{y}_{th,l} = 1$ of th at l, and a *minimum level* $\underline{y}_{th,l} = 0$ (complete eradication of the threat).
- **Actions**, a set $\mathbf{A} = \{a_{th} | th \in \mathbf{T}\}$ of conservation actions, one for each threat $th \in \mathbf{T}$, that can be executed at the managed locations, in order to reduce the amount of a given threat. Each action $a_{th} \in \mathbf{A}$ has a *cost* which depends on the location where the action is executed, denoted by $c_{a_{th},l}$. The fact that actions have different costs depending on the location where they are executed reflects the cost of travelling to a given location; since we are dealing with insular environments, that can be located far away from the coast, such costs can be significant.
- **Time**, actions can be planned over a horizon $\mathbf{H} = \{0, \dots, h-1\}$ of length h. The same discretisation is used, consistently, to describe the evolution in time of the populations of threats and features according to population mechanics. We will sometimes refer to a time unit in the horizon as a *time step*.

3.1.1 Growth rates

Each feature has a characteristic *growth rate* $\alpha_{ft} \in \mathbb{R}^+$, which is the speed at which the population grows 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* $\beta_{th} \in \mathbb{R}^+$, which is the speed at which the threat population grow in absence of conservation actions.

3.1.2 Threat impact and action effect

The effect of threat $th \in \mathbf{T}$ on feature $ft \in \mathbf{F}$ is described with an *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 (amount) of the threat. We can now define $\mathbf{T}_{ft} = \{th \in \mathbf{T} \mid \iota_{th,ft} > 0\}$.

Similarly, the *effect* $\epsilon_{a_{th}} \in [0,1]$ of an action $a_{th} \in \mathbf{A}$ on its target threat $th \in \mathbf{T}$ is the fraction of population of th that the action is expected to eliminate in a single time step. The reduction of the threat only depends on the action application.

3.1.3 Thresholds

When a conservation action $a_{th} \in \mathbf{A}$ is applied at some location $l \in \mathbf{L}$, the population of th decreases according to $\epsilon_{a_{th}}$. As a consequence, the populations of features $\mathbf{F}_{th}$ may start to recover, however this is not an immediate effect. In our ecosystem representation, this kind of behaviour is modelled through *thresholds* $\theta_{ft,th} \in [0,1]$ that are given for each $\langle ft, th \rangle$ pair. 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.

Each feature ft can be threatened by multiple species $\mathbf{T}_{ft}$; in order to see an increase in its population, the relation $y_{th,l} < \theta_{ft,th}$ must hold for all $th \in \mathbf{T}_{ft}$. Figure 1, obtained by running a simulation of the population dynamics, exemplifies this kind of interaction. 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}. It can be seen that it is not until both threats descend below their respective thresholds, that the population of ft starts to grow. As the level of the blue threat grows, e.g., because $t_{th} \in \mathbf{A}$ is not applied at some time steps, the recovery rate in the population of ft starts to decrease, and it inverts as soon as the threat level gets higher than the threshold. Finally, when both threat levels decrease significantly, the population can recover, until the carrying capacity of the location is reached.

The increase in the population of ft at a given time step t depends on the distances (Equation 1) of the threat levels from their relative ft-threshold

$$\theta_{ft,th} - y_{th,l,t} \tag{1}$$

This aspect will be described in the next section, where we formally define the rules controlling the evolution of features and threats populations.

3.2 Population dynamics

Population dynamics describe the evolution of populations over time at each considered location. We will use the generic term *population* to refer both to features and threats, although their evolution follows slightly different rules. Moreover, the population of a feature $ft \in \mathbf{F}$ in a location $l \in \mathbf{L}$ must lie between its minimum and maximum levels at all times, i.e., $x_{ft,l,t} \in \left[\underline{x}_{ft,l}, \bar{x}_{ft,l}\right] \forall\ ft \in \mathbf{F}, l \in \mathbf{L}, t \in \mathbf{H}$. Similarly, $y_{th,l,t} \in [0,1] \forall\ th \in \mathbf{T}, l \in \mathbf{L}, t \in \mathbf{H}$. Such minimum/maximum bounding has been omitted from the following equations for the sake of clarity.

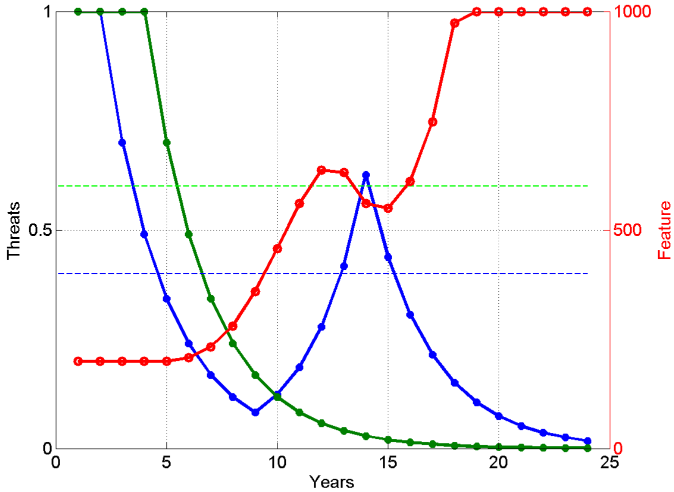

Figure 1: Population dynamics in presence of threats.

3.2.1 *Threats update*

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

$$y_{th,l,t} = y_{th,l,t-1}\left(1 + \beta_{th} - \epsilon_{a_{th}} \cdot z_{a_{th},l,t}\right), \quad (2)$$

where $y_{th,l,t}$ is the level of threat th at location l and time t, and the binary variable $z_{a_{th},t,l}$ models whether action $a_{th} \in \mathbf{A}$ is applied at location l and time t. The existence of the $z_{a_{th},t,l}$ decision variable inside the formulation is necessary to take into account the temporal dimension of the problem, since the state of an environment at a given time step depends on the actions applied at the previous one.

3.2.2 *Features update*

The population dynamics of the features is slightly more complex than that of the threats, and the update rule can be broken down in two cases.

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

$$x_{ft,l,t} = x_{ft,l,t-1}\left(1 + \alpha_{ft} \cdot \prod_{th\in\mathbf{T}_{ft}}\left(1 - \frac{y_{th,l,t}}{\theta_{ft,th}}\right)\right), \quad (3)$$

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, and the update is executed for all $t \in \{0, \ldots, h\}$.
If the threat populations are exactly the threshold, then the fraction is 1, and the increase in the population of ft is zero. As the threat populations descend below the threshold, the populations of features start to increase.

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

$$x_{ft,l,t} = x_{ft,l,t-1} \times \left[1 - min\left(1, \sum_{th\in\mathbf{T}_{th,l,t}} \frac{y_{th,l,t} - \theta_{ft,th}}{1 - \theta_{ft,th}} \cdot \iota_{th,ft}\right)\right], \quad (4)$$

where $\mathbf{T}_{ft,l,t} = \{th \in \mathbf{T}_{ft} \mid y_{th,l,t} > \theta_{ft,th}\} \subseteq \mathbf{T}_{ft}$ denotes the set of threats, among the ones that affect ft, that are above their threshold 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 levels of threats.

If the threat populations are exactly on the threshold, then the decline in the population of ft is zero, but as soon as they increase, the population of ft starts to decrease.

Note that no ecosystem model is able to precisely represent the very complex interactions between populations and environment that occur in nature. The aim of this model is, however, to provide a useful approximation to make more informed decisions in absence of such representation.

Moreover, the non-linearity of the relations that define the population dynamics makes it difficult to obtain exact solutions to this problem.

3.3 Budget constraints

Habitat restoration activities are often subject to budget constraints. These arise from the limited availability of funding for conservation, and can be expressed 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 other words, the sum of the costs of all actions applied throughout the planning horizon and across all the locations must be within the given allotted budget B.

In our formulation, we also consider the possibility that, even if the total budget is large, the maximum expenditure at a given time step is limited due to the staggered availability of funding. This can be expressed as

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

where B_t is the maximum allowed expenditure at time step t. Note that this second type of budget constraint is a tightening of the previous one, as it can only reduce the search space.

3.4 Objective

The habitat restoration planning problem has two conflicting goals. On the one hand, we want to reach the conservation targets for all the features. This goal could be formalised with the following constraint

$$\sum_{l\in\mathbf{L}} x_{ft,l,h} \geq \sigma_{ft} \cdot \sum_{l\in\mathbf{L}} x_{ft,l,0}, \quad \forall ft \in \mathbf{F}. \quad (7)$$

where $h = max(\mathbf{H})$ is the latest time step in the planning horizon and σ_{ft} are the desired relative increases in the feature populations (targets).

On the other hand, we may not exceed the budget allocated for conservation activities. This goal is expressed by constraints 5 and 6.

Unfortunately, fully satisfying both constraints at the same time is rarely possible, therefore we must treat one as a *soft constraint*. Since budget is the most stringent constraint in the context of conservation, we consider Equations 5 and 6 as *hard constraints* and transform Equations 7 into an objective function.

Aside from the obvious advantage of relaxing the problem, this approach also allows us to produce least-worst solutions when, regardless of the quality of a plan, the conservation targets cannot be possibly met. This situation can occur under several conditions, including

- insufficient funding to protect all the features,
- unavailability of actions affecting a threat that preys on one of the features be protected,
- an horizon too short to allow for an effective plan.

Our objective function, inspired by Equation 8 is composed of two terms.

$$\textbf{minimise} \;\; w_{cons} \sum_{ft \in \mathbf{F}} \left[max\left(0, \Delta_{ft}\right)^2 \right] - w_{impr} \sum_{ft \in \mathbf{F}} \left[max\left(0, -\Delta_{ft}\right) \right], \quad (8)$$

The first term (*conservation*) represents the *distance from satisfaction* for Equations 7. This term is based on the value of Δ_{ft}, which represents the difference between the target σ_{ft} and the ratio between the initial ($t = 0$) and the final ($t = H$) amount of *ft* across all the locations.

$$\Delta_{ft} = \sigma_{ft} - \frac{\sum_{l \in \mathbf{L}} x_{ft,l,h}}{\sum_{l \in \mathbf{L}} x_{ft,l,0}} \quad (9)$$

A positive Δ_{ft} means that the target for *ft* has not been met, a negative Δ_{ft} means that the target has been met and, possibly, topped. Because we take the maximum between zero and Δ_{ft}, the *conservation* term contributes to the cost if and only if Δ_{ft} is positive. This term is quadratic, meaning that the penalty for a population far from the target is much bigger than the penalty for a population close to the target. The second term (*improvement*) promotes solutions that not only meet the targets, but also exceed them, and is active if and only if $\Delta_{ft} < 0$. This term is linear. Because our priority is meeting the targets, the two terms are also weighted differently, a common technique in soft computing. In our setup $w_{cons} = 10^4$ and $w_{impr} = 1$. Note that these values are rather arbitrary and represent only one possible prioritisation of the two terms. In fact, w_{cons} and w_{impr} constitute "knobs" of the formulation, that can be adjusted by conservation managers to request a specific prioritisation policy.

4. OPTIMISATION

We propose a solution approach for the HRP problem based on genetic algorithms (GAs, see [4, 3, 13]), a class of bio-inspired global optimisation algorithms based on the principle of the survival of the fittest.

In a genetic algorithm, an optimisation problem is solved by iteratively evolving a population of candidate solutions by means of operators that draw a parallel from evolutionary processes. Such operators are applied in turn until the stopping condition is met, and are

- **selection**, which allows to choose which individuals will generate the new population,
- **crossover** (sometimes **recombination**), which combines the binary representation of two solutions in order to yield one or more new solutions (*individuals* or *offspring*), and
- **mutation**, which perturbs a solution in order to provide diversification and generate new genetic material.

The objective function (described by Equation 8) is encoded in a **fitness** measure that provides a criterion for the implementation of the selection operator.

Genetic algorithms have been a prolific research fields for more than 40 years. In this span of time, many different approaches to implement the crossover, selection, and mutation operators have been proposed. In the following we describe our choices, highlighting the involved parameters, and the implementation details.

4.1 Solution encoding and initialisation

In order to solve an optimisation problem with genetic algorithms, one needs to specify how a solution is encoded in a *chromosome*, i.e., a string of (often binary) values, that is used as "genetic material" for the evolutionary operators to work with.

In our implementation, each chromosome is composed by $n = |\mathbf{L}||\mathbf{H}||\mathbf{A}|$ binary decision variables, each one representing whether an action $a_{th} \in \mathbf{A}$ will be applied at location $l \in \mathbf{L}$ at time $t \in \mathbf{H}$. In other words, each *gene* encodes one of the $z_{t,l,a}$ decision variables appearing in Equations 2. Because $z_{t,l,a}$ are binary decisions, i.e., we can either decide to apply an action at a given location a time step or not, the mapping to a binary chromosome is straightforward, and allows to implement mutation using simple bit flips.

The initial population of solution is generated by setting each variable to a value in $\{0, 1\}$ chosen uniformly at random.

4.2 Control parameters

As in many other meta-heuristics, the behaviour and performance of our solver are controlled through a set of parameters that can be tuned. The parameters are the following

- **chromosomes** m, the number of individuals in the evolved population,
- **generations** g, the number of iterations of the solver, i.e., the number of evolutionary steps,
- **elite chromosomes** e, the percentage of chromosomes (sorted by decreasing fitness) that carry over to the next population,
- **crossover probability** p_c, probability of combining a selected pair of chromosomes to yield a new chromosome (with probability $1 - p_c$ the pair of chromosomes will carry over to the next generation without recombination),
- **mutation probability** p_m, probability of mutating each gene in a chromosome.

We fixed the above parameters through a preliminary experimental analysis. The identified configuration is the following: $m = 50$, $g = 150$, $e = 10\%$, $p_c = 0.9$, and $p_m = 0.1$. These values were used for the experiments shown in Figure 2. Expectedly, these values will have to be tuned again, possibly using an automated parameter tuning procedure, once the definitive survey data will be available.

4.3 Evolutionary operators

We now describe how our selection, crossover, and mutation operators are implemented. Note that these are not problem specific, and represent quite simple instantiation of a classic genetic algorithm.

4.3.1 Selection

At each iteration we perform the following steps. First, all the unfeasible chromosomes, i.e., the solutions with a cost exceeding the allotted budget B and the period-specific budgets B_t, are discarded. Second, the fitness of each surviving chromosome, which represents a feasible plan, is measured by simulating the outcome of the plan according to Equations 2, 3 and 4, and using the final populations to compute the objective (Equation 8). Finally, we copy the elite chromosomes to the new population based on their fitness value and the percentage of elite chromosomes (e) parameter. The remainder of individual in the new population is generated according to the crossover and mutation strategies described below.

4.3.2 Crossover

We generate pairs of chromosomes according to a roulette selection scheme, where the probability of being selected is proportional to the fitness of the chromosomes. Then, with probability p_c we perform a *one-point crossover* where the cut position is chosen uniformly at random. If the probability test fails, the selected pair of chromosomes is copied verbatim to the new population.

4.3.3 Mutation

Once the new population has been constructed, with probability p_m we flip each bit in each chromosome. We employ an *elitist* selection scheme, where the single best chromosome in the population is exempted from mutation, so that the best solution is never lost.

5. EXPERIMENTAL RESULTS

As a proof of concept, and to validate the viability of our formulation, we implemented the genetic algorithm in MATLAB. We then tested the solver on a master instance based on real-world survey data. The data describes 13 islands, collectively harbouring 46 features and 19 threats. Our problem formulation has been developed to match survey data which will only be completely available in the course of the next year. The data which is currently missing, specifically the impact of actions on threats $\iota_{a,th}$, and the effect of threats on features $\epsilon_{th,ft}$, has been replaced with synthetic values.

The solver has been run with the parameters in Section 4.2, and with budgets $B \in \{1000, 5000, 10000, 50000, 100000, 200000, \ldots, 1000000\}$, the period-specific budgets B_t were not used in this analysis. In this section we describe our preliminary results.

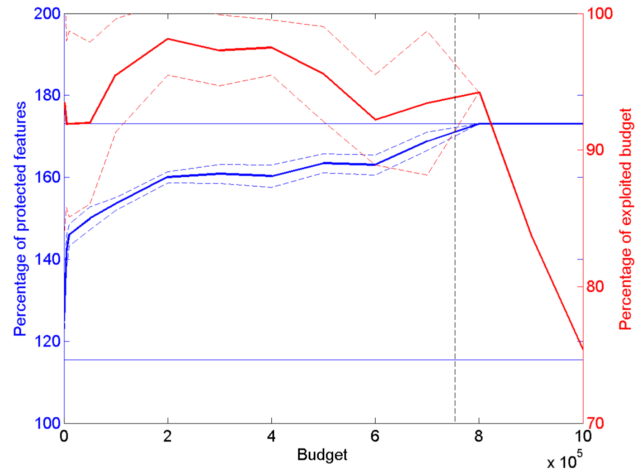

Figure 2: Performance (blue) and budget-efficiency (red) of the solver as a function of the maximum allotted budget.

5.1 Performance

Our first set of experiments is aimed at verifying that our algorithm is able to produce good quality feasible plans, and that it is able to exploit increasingly large budgets.

The outcome of this set of experiments can be observed in Figure 2, which shows the impact of different budget allowances on the overall quality of the found solutions. The plot has been generated by averaging 20 runs of the solver. In the plot, the solid red and blue lines represent, respectively, the percentage of exploited budget and the percentage of protected features[2]. The dashed lines above and below the solid ones represent the standard deviation from the corresponding measure, computed across all 20 optimisation runs. The vertical black dashed line shows the maximum needed budget needed to apply all actions at all locations at all time steps. Similarly, the blue horizontal lines shows the minimum and maximum impact we can make, i.e., the amount of features we can obtain, respectively, by not applying any actions, and by applying all actions everywhere.

The plot has the typical shape observed in conservation planning (for example see [10]), and reveals a diminishing returns effect. At the beginning, the percentage of protected features increases along with the increased budget, but then slows down as it reaches the carrying capacity. This happens when the budget is large enough to allows executing all the most relevant actions, and executing the remaining ones does not help significantly.

Another interesting observation on this set of experiments is that, when the budget is low, there are only a few sets of actions that can be applied successfully, because applying a small number of actions is not sufficient to reduce the threats population enough to allow the features to recover. As a consequence, a rather large portion of the budget remains unconsumed. As the budget grows, this effect disappears, and the solver achieves a higher budget efficiency. When the budget grows larger than the maximum budget needed

[2]We chose to visualise the percentage of protected features instead of the objective function value, since this measure is easier to interpret visually.

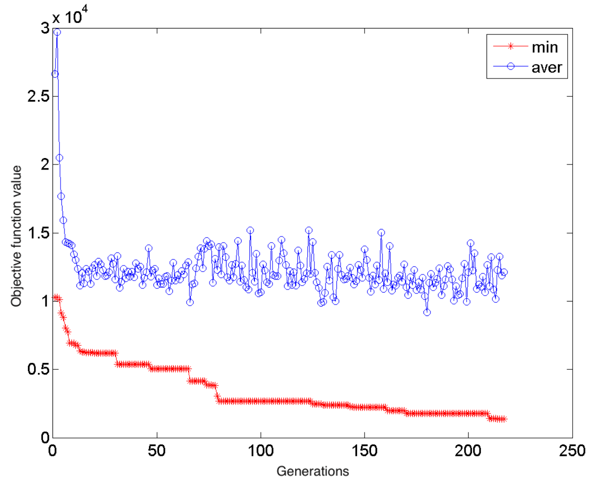

Figure 3: Convergence of the objective function on a single run of the solver.

to protect all the features, the budget usage measure drops again, as all the actions are being applied already, and there is nothing else to be done.

5.2 Convergence

Our second set of experiments aims at showing the convergence properties of the algorithm. This is represented in Figure 3, which shows the evolution of the objective value (Equation 8) during a single optimisation run[3]. In the plot, the red line represents the fitness of the best chromosome, while the blue one represents the average fitness of the whole population.

Again, we observe a rather typical behaviour in optimisation algorithms, where the decrease in the objective function value is faster at the beginning, but slows down as the quality of the solution increases. Note that this plot represents the evolution of the global objective function: on an location to location level the objective function might look different, as some of the islands are left untouched by the plan, which focuses on the ones generating a higher gain.

5.3 Other approaches

An extended formulation of the habitat restoration planning (HRP) problem, has been recently developed [14] to overcome some of the limitations of the present (original) formulation. This is, to date, the only other approach for solving the HRP problem. The extended formulation includes the possibility to execute more than one action of a given type at each location and time step, thus introducing a measure of *intensity* of an action, and considers *uncertainty*, in terms of intervals of confidence, as part of the solving process. Such formulation has been solved using a constraint programming (CP) and large neighbourhood search (LNS) approach. Because of the differences in the formulation, a direct comparison with it would be unfair, however we plan on supporting the extended formulation in the future, and to carry out a comparison of the two approaches. In this regard, we also plan to implement our approach in a more complete GA framework, so as to be able to experiment with other evolutionary operators.

[3]Note that the parameters here might be different from the ones used for the first set of experiments

6. CONCLUSIONS

We have presented the original formulation of the habitat restoration planning (HRP) problem, and proposed a solving technique based on genetic algorithms (GAs). Our formulation includes a rather sophisticated population dynamics model, that tries to approximate the interactions between species in natural environments. Our solver generates plans that aim at reaching given conservation targets for a set of species to be protected, while respecting hard budget constraints. The results obtained by our solver are promising, however the current formulation does not take into account some aspects which are likely to become relevant as the solver is deployed in a real-world context.

First, conservation data is intrinsically uncertain: surveys report populations as confidence intervals, rather than as precise figures. Furthermore, the effect of actions on threats is, itself, uncertain. While these problems reside in the data, and can be only solved by improving the accuracy of survey operations, a more robust model could take uncertainty into account, and reason about a plan's effectiveness in terms of confidence intervals, allowing to analyse the worst case and the average case scenarios. This is one of the most challenging future directions of this project, and has been partly addressed by [14].

Second, although some targeted conservation actions can be carried out, the assumption that one action can only affect one type of threat is not completely realistic. As a counter example, an action such as baiting or placing traps can affect multiple threats at once, and also, to some extent, the features we wish to protect in the first place. Similarly, some features that we might want to protect, e.g., goats, might affect other features, e.g., lantana, and some threats might contribute in keeping other threats under control (mesopredator release). All of these aspects will likely push us to generalise the population dynamics, so that features and threats populations are subject to the same update rules.

Third, our objective function is currently quite simplistic, as it treats all species interchangeably, which is usually not the case. We therefore intend to analyse different objective function formulations, e.g., weighting features by their endangerment level, or using a *minimax* formulation, in which one tries to maximise the minimum increase in the features population, and analyse their amenability for practical purposes. A very interesting research avenue, in this regard, is evolutionary multi-objective optimisation (EMOO). In particular, the objective function in Equation 8 is actually a scalarisation of a naturally multi-objective problem, where the conservation of each feature is a different objective. Leveraging the abundant research in the field of evolutionary multi-objective optimisation, we could directly target the original problem using variants of NSGA [2] or AGE [15].

Finally, we expect more conservation data to be available in the future. This calls for an extensive evaluation of our solver's scalability, and likely the exploration of different optimisation techniques, also to handle aspects such as uncertainty, which could be better represented by techniques which operate on variable domains.

7. REFERENCES

[1] I. R. Ball, H. P. Possingham, and M. E. Watts. Marxan and relatives: Software for spatial conservation prioritisation. In A. Moilanen, K. A. Wilson, and H. P. Possingham, editors, *Spatial conservation prioritisation: Quantitative methods and computational tools*, pages 195–195. Oxford University Press, Oxford, UK, 2009.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[3] D. E. Goldberg et al. *Genetic algorithms in search optimization and machine learning*, volume 412. Addison-Wesley Reading Menlo Park, 1989.

[4] J. H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* University of Michigan Press, 1975.

[5] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simmulated annealing. *Science*, 220(4598):671–680, 1983.

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

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

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

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

[10] A. Rebelo and W. Siegfried. Where should nature reserves be located in the cape floristic region, south africa? models for the spatial configuration of a reserve network aimed at maximizing the protection of floral diversity. *Conservation Biology*, 6(2):243–252, 1992.

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

[12] F. Rossi, P. Van Beek, and T. Walsh. *Handbook of constraint programming.* Elsevier, 2006.

[13] M. Srinivas and L. M. Patnaik. Genetic algorithms: A survey. *Computer*, 27(6):17–26, 1994.

[14] T. Urli, J. Brotankova, P. Kilby, and P. Van Hentenryck. Intelligent habitat restoration under uncertainty. 2016.

[15] M. Wagner, K. Bringmann, T. Friedrich, and F. Neumann. Efficient optimization of many objectives by approximation-guided evolution. *European Journal of Operational Research*, 243(2):465–479, 2015.

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