A Genetic Algorithm Solver for Pest Management Control in Island Systems

Jana Brotankova[1], Marcus Randall[2], Andrew Lewis[3], Bob Pressey[1], Amelia Wenger[1]

[1] ARC Centre of Excellence for Integrated Coral Reef Studies, James Cook University, Townsville, Australia,
[2] Faculty of Business, Bond University, Gold Coast, Australia,
[3] School of Information and Communication Technology, Griffith University, Brisbane, Australia

Abstract. Optimising the management of national parks and reserves is a truly multidisciplinary issue. This problem requires considerable knowledge of the characteristics of the ecosystems and species and their interactions. This then can be translated into a highly complex knapsack problem. Essentially, within a limited budget, a manager needs to select the conservation actions according to expected payoffs (in terms of protecting or restoring desired species) versus cost (the amount of resources/money) required for the actions. This paper presents a formulation and optimisation techniques to solve the problem. From this method, decision-support software is being developed, tailored to meet the needs of pest control on islands, in close cooperation with conservation managers. The solver uses a Genetic Algorithm (GA), and incorporates a model of the conservation problem (a simplified model of the ecosystem). The results indicate the solver is able to derive strategies that reduce the number of threats in island ecosystems, allowing the preservation of desired plants and animals. However, the problem model needs further refinement in order to derive truly realistic options for conservation managers.

1 Introduction

Systematic conservation planning began to emerge as a discipline just over 30 years ago [10] and is now extensively applied and influential worldwide [11]. As well as guiding the design of new spatial management for conservation, systematic methods can be applied within established reserves to inform zoning [2] or, potentially, the allocation of actions to mitigate threats. Given the number of introduced species that act as threats to native species and ecosystems and limited budgets, prioritising of conservation actions in space and time is necessary to identify the most cost-effective programmes for the eradication of pests.

Existing conservation planning software systems have been used extensively for conservation-related research, but also for real-world applications that have led to gains in conservation. However, these tools are limited in various ways in addressing the full complexity of day-to-day management decisions. Marxan [14], C-Plan [12], and Zonation [6], are mainly targeted to selection of new reserve networks. RobOff [8], [9] does work with conservation actions including pest eradication, but is non-spatial.

Furthermore, most of the research on multiple-action planning (as opposed to planning generic "conservation" applications) has been at a spatial resolution too coarse to be useful for managers on the ground (e.g., Carwardine et al. [1] and Wilson et al. [15]). For our purposes, we need data, and software capable of analysing those data, at a very fine resolution (e.g., typically 1–10 ha) and across hundreds of sites.

Given the limitations of capability of existing software systems and the challenges, in terms of acquisition and analysis of data for fine-resolution applications, a solver system, based around the Genetic Algorithms (GA) optimisation is created to address these limitations. GA was selected as it is a well-established technique that has a long track-record of producing good solutions, and which is naturally formulated to work with 0-1 problems.

Additionally, the software behind the solver must have an intuitive and attractive visual and spatial interface to facilitate its use by conservation managers. The new software is being designed initially for use by managers of islands in north-western Western Australia (the Pilbara region) and the southern part of the Great Barrier Reef, both regions having biodiversity values of national and global significance.

The remainder of this paper is organised as follows. Section 2 explains the mathematical formulation of the problem, detailing the choice of objective function as well as how it becomes a knapsack polytope. Section 3 describes general the mechanics of a GA as well as setting forth the way in which it is applied to this problem. The computational experiments, results and analyses of these are given in Sections 4 and 5. Finally, along with the conclusions, the direction for developing both the problem definition and solver techniques is outlined in Section 6.

2 Formulation of the Problem

Conservation managers face the following problem when planning pest eradication measures for islands: they need to optimise the selection of actions on the islands in the most cost-effective way considering the evolution of the system in a given time scale. As such, in the first instance, a model needs to be developed that can describe the ecosystem, at least in simple terms, and the changes due to the actions taken.

Several input tables are required to describe the ecosystem. First are the abundances or the extent of features (native species and ecosystems) and threats on the islands. Features are the native animals and plants that need protecting. Threats are the invasive species (animals or weeds) that adversely affect the features. Capturing the interactions amongst the features and threats is a non-linear problem that includes thresholds and depends on the ratios of the predators and prey. In terms of testing the optimisation process, a simplified model was used that considers linear relations among threats and species: each individual of the threats reduces a constant number of each (impacted) feature per year.

Variables are defined as the actions on islands, e.g., shooting goats, trapping or poisoning cats, spraying weeds. One action on one island is one variable. Every action has a defined efficiency (percentage of the targeted threat knocked down), and cost (how much must the managers pay to perform this action on this island).

The ecosystem model gives two knapsack formulations referred as Scenario 1 and Scenario 2:

1. Minimise total cost of actions
   s.t. The number of features is greater than a defined threshold
2. Maximise the number of features
   s.t. The total cost is less than a budget amount

Knapsack problem belongs to problems in combinatorial optimization, dating to late 20th century [5]. The task is to find the best combination of items with given parametres, meeting given requirements.

The whole model must run over certain time interval, typically 3–5 years, with a time step of one year. The funding in conservation usually works on a year-to-year budget scheme, so this paper will focus on the second knapsack formulation.

To capture the dynamics of the system, a model was developed that predicts abundances of features and threats on each island in each of the given time slices. Each time slice is connected to the previous and the following one via coefficients of birth and mortality. Two basic equations are needed for the model: the time evolution of Threats and of Features.

2.1 Threats

The threats breed on the islands and are killed by the selected actions. In the first simplified approach, we consider only one action for one threat. Thus, the population y_t of Threat t in time c on island i can be written as:

$$y_{c,i,t} = y_{c-1,i,t} + \beta_t \times y_{c-1,i,t} - y_{c-1,i,t} \times z_{c,i,a} \tag{1}$$

where the index $c-1$ depicts the previous year, β_t is the coefficient of breeding (growth rate) of the threat t, and $z_{c,i,a}$ is the action a on island i performed in time c. $z_{c,i,a} = 1$ or 0 if the action is or is not performed, respectively. The growth rate is defined as an increment of the threat population per year in an absence of any predation.

2.2 Features

A similar equation can be written for features, with the only difference being that they are killed by the threats:

$$x_{c,i,f} = x_{c-1,i,f} + \alpha_f \times x_{c-1,i,f} - \sum_{t=1}^{n_{i,f}} \sigma_{f,t} \times y_{c-1,i,t} \quad (2)$$

Here $x_{c,i,f}$ is the population of feature f at time c on island i; α_f is the growth rate of feature f; $\sigma_{f,t}$ is the coefficient of interaction between feature f and threat t, i.e., how many features f are predated per year by threat t. The last term on the right is summed over $n_{i,f}$ which is the number of all the present threats that reduce feature f on island i. If the sum is greater than the actual abundance of this feature on the island, the whole population is set to 0.

2.3 Method

The variable z represents actions on the islands, as mentioned previously. The solver suggests a combination of a set of these z values for each year and each combination of actions on islands. This will be an input to the model, together with the abundances of threats and features, as well as the coefficients. The model takes into account abundances of threats in year 0 (before any actions are taken) and calculates abundances of threats in year 1. Then it calculates abundances of features in year 1, using the input abundances and information about the threats reducing the features. These two steps are repeated until the last year. The cost of each solution z is calculated from the combination of zeros and ones.

3 Genetic Algorithm implementation

Genetic algorithms belong to a broader class of evolutionary algorithms. Their origins date back to the 1960s [4] and were inspired by the natural selection of species choosing fitter individuals and allowing them to breed and produce new solutions. The standard operations of GAs are selection, mutation, and crossover [3].

Each individual (a chromosome) represents one solution to the problem. Chromosomes consist of a combination of genes taking the values of either 0 or 1. All the chromosomes have equal length. The genetic information propagates to the next generations either directly, or by crossover. The point is to find the best solution by improving the genetic information with respect to the desired criterion translated to a fitness function.

The mechanics of a standard GA implementation are as follows. At the beginning, a random set of individuals is generated as the initial population. Their quality iis determined by the fitness function and in the case of the test problem, the number of features of concern over all the islands. The individuals are ranked and assigned mating probabilities. The best few individuals ("elite individuals") are transferred directly to the new generation. Crossover is performed on the rest of the individuals. That is, chromosomes are selected randomly using the roulette wheel selection mechanism to "mate". "Mating" means that the two parent chromosomes are split at a random point(s), and exchange parts of themselves with one another. After that, mutation is performed with a small probability. In this, some of the genes will randomly change value. Mutation is vital to avoid search stagnation, which can lead to locally optimum solutions.

The implementation details of the GA for this problem are described in Algorithm 1. One chromosome represents one combination of variables Y. The variables can take values of either 1 or 0 which represent the actions that were or were not performed on a particular island.

Algorithm 1 Genetic algorithm

```
Generate the initial population of random knapsack solutions
for 1 to the number of iterations do
  Assess feasibility: feasible individuals are placed in the mating pool
  Assess the quality of the chromosomes, based on the fitness function
  Transfer the elite individuals into the next generation
  repeat
    Roulette wheel selection: allocate probabilities of selection based on quality of each individual in the mating pool
    Crossover: fill the next generation by two offspring of two randomly selected parents from the mating pool
  until the new generation is filled
  Mutation: randomly change some of the genes with small probability (not for the last run)
  if the best individual in this generation is better than the best one recorded so far then
    Make it the best individual
  end if
end for
end
Output the best individual and its objective cost
```

For Scenario 1, the fitness function is the total cost of actions, subject to the constraint of the given desired final population levels. For Scenario 2, the fitness function is the number of features, subject to the constraint that the collective cost of the selected actions is below a particular budget amount. For each feature, the initial population is taken as 100%; they are summed at the end. Thus, if the population of a feature doubled in abundance during the total time of the model, its contribution was 200%, while if its population halved, its contribution was 50%.

The task for the solver is to select the best combination of actions on islands that still meets the objectives and constraints. For example, for scenario 2 the solver selects the combination with the best fitness function (the maximum of protected features) out of the solutions under the budget level.

Until now, the budget was the only constraint: only individuals with costs smaller or equal to the budget passed into the mating pool. For complex cases, it may be useful to introduce more constraints. For example, apart from the budget, there may be need to focus on a particular island or species. The problem is then while keeping within the budget, achieve at least 150% species *a* increase, and remove all weeds *b* from island *c*. Obviously, this may lead to the situation in which all the constraints cannot be met. Also, the managers may formulate the objectives in a more tentative way: certain things would be good to do, but if it is too hard, they are not performed. To cover this complexity, *hard* and *soft* constraints can be introduced. If a chromosome does not meet a hard constraint (e.g., becuse of the size of the budget), it will be excluded from breeding. If it does not satisfy the soft constraints, it will be excluded with a certain probability (e.g., 80% if we really want to meet it or 30% if we do not). Refining the handling of these constratints will be examined in later detail in future work.

4 Computational experiment

The algorithm has been tested on a laptop with an Intel$^{®}$ Core^{TM} i7-4600U CPU @ 2.10 GHz 2.70 GHz processor, with 16 GB RAM and 64-bit operating system under Windows 7. Matlab is used to implement the solver. The trial dataset included only 13 islands, 32 features, and 19 threats, as a simple subset intended to demonstrate the feasibility of the proposed approach. In future work investigations will be extended to use the full set of islands (numbering one hundred and twenty for the Great Barrier Reef, or even more in the Pilbara).

An example of the performance of the solver is given in Figure 1. The top panel shows the convergence of the solution: the average and the best value of the fitness function. The bottom panel shows the percentage of feasible solutions.

Another information about the solver performance needs to be derived from the average and standard deviation. In future work, the solver will be run multiple times by varying the random seed.

An important part of the process is to interpret and visualise the output. A key aspect of visualisation will be a spatial interface in a geographic information system, showing which actions are allocated to which islands, with capability for querying the spatial output by pulling up tables and graphics. The spatial interface is still in development, but some of the background graphics are shown here.

The results need to be shown in the context of the abundance of the features. For this, the model needs to be run again with the final combination of the actions. An example of the temporal trends in abundances of features and threats is shown in Figure 2. The top panels show the trends in threats, the bottom panels trends in features. The left panels are population levels of each single threat/feature on a single island, while the right panels are the sums of each threat/feature across all islands. The reductions in the threats (see in the top left panel) indicate that an action was performed. The indicative budget was set at $200,000.

The solver efficiency needs to be tested. One of the tests is shown in Figure 3. The dashed curve with stars shows the percentage of the budget used, e.g., our budget (on the x-axis) is \$250,000, but our solver found a solution of \$225,000, so the dashed line shows 90%. The fluctuations in the dashed line can indicate two things: i) that a local minimum was found which may be a bit far from the global one (we do not know where the global maximum is but we assume that it corresponds to 100% of protected features), or ii) there is no action that would be "cheap enough" to be taken within the budget, while being efficient enough to make a difference. To understand this better, the vertical dashed black line shows the maximal cost of the actions, where every possible action is taken on every island. Naturally, after crossing the 100% limit, the dashed line decreases as the maximal cost becomes lower than the budget.

The full line with circles in Figure 3 indicates the cost-effectiveness of the whole conservation process, which is interesting information, highly relevant to the conservational managers. It shows a typical curve in conservation science: the effect of conservation actions rises fast at the beginning, but there are long-established diminishing marginal returns [13]. In this case, the fast rise at the beginning is extreme, which is caused by the simplifying assumptions in the model.

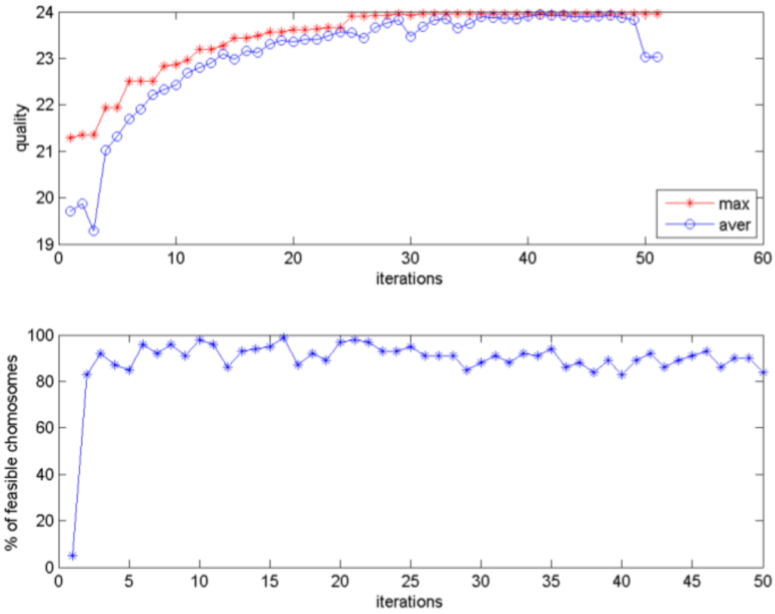

Fig. 1. Example of the convergence of the GA solver: the line with stars shows the quality of the best chromosome, while the one with circles shows the average (top), and the percentage of feasible solutions (bottom).

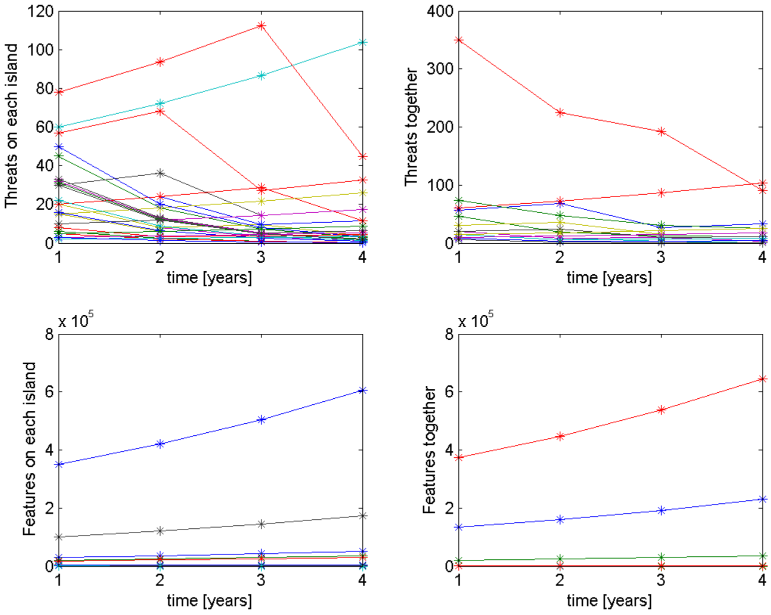

Fig. 2. Temporal trends in abundances of threats and features. Top panels: threats; bottom panels: features. Left panels: each line represents one threat/feature on one island; right panels: one line represents the sum of one threat/feature across all the islands.

A better understanding of the model can be obtained by comparison of the “no actions anywhere” and “all actions everywhere” cases (in other words, the chromosome consists of zeroes only or ones only, respectively). This is given in Figure 4. The top panels again show threats, while the bottom ones show features. The left panels show the time evolutions when nothing was done, the middle is the solution from the solver, and the right when everything was done. The bottom panels are zoomed in. Abundant features (out of the scale; see in Figure 2) practically do not see the impact of threats, while low populated features respond to the actions.

The left four panels show the temporal trends in threats and features when no actions were done anywhere. The right four panels, on the contrary, show when all possible actions were performed on all islands. These data were used for establishing the boundaries (minimum and maximum of possible impact of the actions) in Figure 3.

The bottom right panels in each group show how great a difference we can actually make. In this particular case, the initially most populous features show virtually no impact, so the contribution of the actions looks minor. However, the populations of a large number of features with smaller initial populations (that

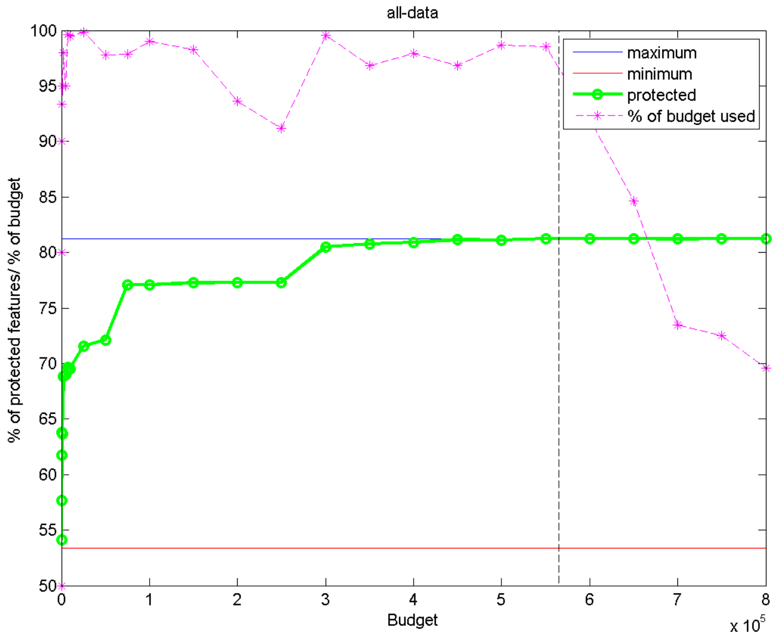

Fig. 3. Efficiency test of the model. The full line (circles) indicates the percentage of protected features achieved, for different budget expenditure. The populations achieved for maximum and minimum impact of actions are depicted by the horizontal full lines. The dashed line (stars) shows the percentage of the budget that was used. The dashed black line shows the maximal cost: when all actions are performed on all islands.

could be considered "more at risk") have survived when actions were taken, instead of being completely obliterated when no action was taken.

5 Discussion

The first model of the conservation problem includes several important simplifications. One reason is that the aim of this paper is to test and implement the solver rather than making a highly realistic model. Furthermore, the precise model requires very specific data which are being currently collected as part of a larger research project [1]. The process of elicitation is very complicated, time-consuming and ongoing, and needs to be considered in the model development.

The advantage of this simplified model is its straight-forwardness, so it is not difficult as a test of the solver. There are some simplifications of the present work that will be overcome by the subsequent development of the model:

[1] See the Acknowledgements for details of the project

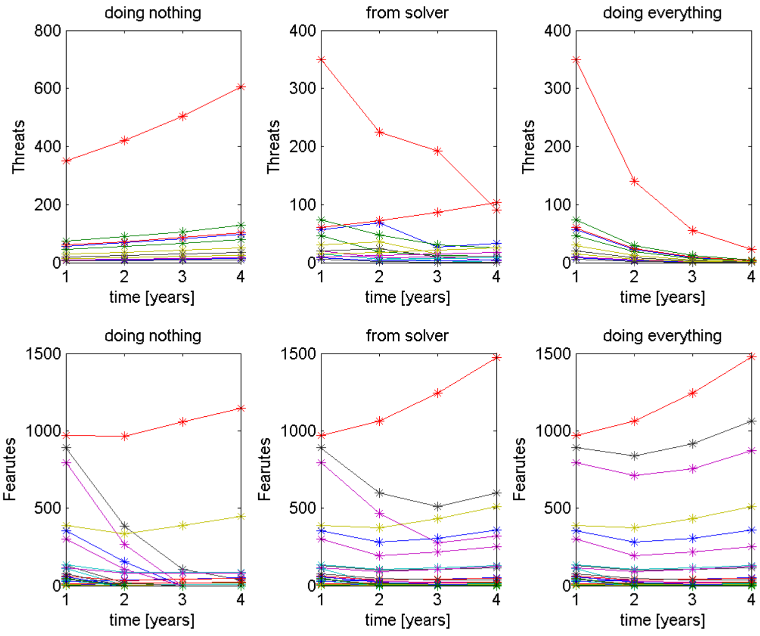

Fig. 4. Temporal trends in abundances of threats and features. Top panels: threats; bottom panels: features (zoomed in). Left panels: nothing was done; middle: solution from solver; right: everything was done.

- The relation among threats and features is not as simple as a constant. We obviously cannot say, for example, “one cat eats three curlews a day”. Predation depends on the abundance of the threat and also of the prey. We also need to distinguish the case when a threat is established on an island, and is in a stable state. This is the case where there have been no actions performed in the past couple of years. In that case, we can consider the abundance of the threats constant with no intervention. The abundance of features may still be declining.
- Recovery functions (recovery of features after the threat is suppressed) are not uniform. Each island has a “capacity” to carry certain populations of native species and growth curves for populations are typically logistic in form.
- The response of the features to the elimination of threats is usually not linear. It has thresholds, e.g., we need to eliminate at least 2/3 of goats to see the native vegetation recover.
- The model considers one action to suppress one threat. In reality, the actions break down into several methods: shooting, baiting, trapping, spraying or pulling weeds etc. Individual methods have different efficacies, which also

depend on the repetition: e.g., aerial shooting of goats will only be efficient at the beginning; after one or two fly-overs, the goats will learn to hide.

- The response of threats to actions is not linear [16]. There is a rule of thumb in pest management: eliminating the last 1–10% of a pest may cost the same as reducing the first 90–99% [7]; the most expensive is to remove the last threat (e.g., goat/cat/rat) from the island. This needs to be captured in the model, because it is an important factor for deciding whether eradication or control is chosen.
- The model considers that one action has an impact on one threat only. In reality, one action can influence more threats, and also many features. For example, if we decide to bait rats, cats may also be impacted, but some native birds will eat the baits too, or die from side effects (such as eating a poisoned rat).
- The costs of actions will break down into several items, like fuel for the boat, wages of the staff, and the baits or other material. Fuel costs may be a significant overhead, so once staff are already there, it may be efficient to do other things that would not normally be considered as worth the cost of the overheads. Thus, actions may be clustered.
- We do not consider the reintroduction of threats on some islands (the contra-action may be quarantine) or reintroduction of features by migration (especially birds). This is again an important factor when we decide between eradication or control strategies.
- The budget is not the only constraint we can expect, as mentioned above. Managers in conservation science need to take decisions which are of social-political origin and/or based on their experience which is not captured by the model. The advantage of the constraints model approach is that these special requirements can be implemented by simply adding other constraints. This feature will enable the system to work interactively, which is a vital condition for the new software.

The real-world extent of the problem is 120 islands (for Great Barrier Reef) or roughly 400 islands (for the Pilbara), 40 features, 25 threats, and 30 actions. The heuristic solver can enable managers to explore different options of the system, with different objectives and budgets. The point is not to get the best possible solution, but several near-optimal options to get an idea of the system. To explore several options it needs to be fast (in the order of tens of seconds or minutes) but it may be possible to compromise on accuracy.

6 Conclusions

The management of invasive species on island systems is an ongoing challenge for conservation managers and decision makers alike. Automated decision making tools have been developed in the past, but have had only limited applicability and success. In this paper, a new model is developed of the problem that is based on optimisation of a form of the well-known knapsack problem in which every potential action on an island is a potential item in the knapsack. Using a genetic algorithm, it was shown that the solver was able to derive strategies that reduced the number of threats in island ecosystems, allowing the preservation of desired plants and animals.

Other combinatorial-oriented techniques, such as Extremal Optimisation, will be tested in future versions. They will enable i) to show the quality of the solver in the terms of distance of the solution from the overall optimal one, ii) to gauge the difficulty of the problem and iii) to work out some of the terrain features (such as the extent of local optima).

This paper presents an initial and simplified version of the problem. The next stage of development will be to refine this model, elaborating on a number of conditions that were imposed for the purposes of developing a simplified, early prototype. Importantly, it will be necessary to account for the real-world interactions that actions have on threats and features to provide a realistic and comprehensive system that can provide conservation managers with reliable, effective and credible decision-making support.

7 Acknowledgements

The main support for this work has been from the project “Decision support for prioritizing and implementing biosecurity on Western Australia’s islands”, funded by the Gorgon Barrow Island Net Conservation Benefits Scheme. Key support also came from Project 9.3 “Prioritising management actions for Great Barrier Reef islands”, funded by the Tropical Ecosystem Hub of the National Environmental Research Program. We would like to acknowledge our many collaborators on our island projects in the Great Barrier Reef and Western Australia who have shaped the thinking reflected in this paper and are providing data for the analyses. Particular thanks go to Ian Craigie, Lesley Gibson, John Hicks, Cheryl Lohr, Keith Morris, John Olds and Malcolm Turner.

References

1. Carwardine, J., O’Connor, T., Legge, S., Mackey, B., Possingham, H., Martin, T.: Prioritizing threat management for biodiversity conservation. Conservation Letters **5**, 196–204 (2012)
2. Fernandes, L., Day, J., Kerrigan, B., Breen, D., De’ath, G., Mapstone, B., Coles, R., Done, T., Marsh, H., Poiner, I.: A process to design a network of marine no-take areas: lessons from the Great Barrier Reef. Ocean and Coastal Management **52**, 439–447 (2009)
3. Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Addison Wesley, Reading MA (1989)
4. Holland, J.: Outline for a logical theory of adaptive systems. Journal of the ACM **3**, 297–314 (1962)
5. Mathews, G.: On the partition of numbers. In: Proceedings of the London Mathematical Society 28, p. 486490 (1897)

6. Moilanen, A.: Landscape zonation, benefit functions and target-based planning: Unifying reserve selection strategies. Biological Conservation **4**, 571579 (2007)
7. Myers, J.: Eradication and pest management. Annual Review of Entomology **43**, 471491 (1998)
8. Pouzols, F., Burgman, M., Moilanen, A.: Methods for allocation of habitat management, maintenance, restoration and offsetting, when conservation actions have uncertain consequences. Biological Conservation **153**, 41–50 (2012)
9. Pouzols, F., Moilanen, A.: Roboff: software for analysis of alternative land-use options and conservation actions. Methods in Ecology and Evolution **4**, 426432 (2013)
10. Pressey, R.: The first reserve selection algorithm - a retrospective on jamie kirkpatrick's 1983 paper. Progress in Physical Geography **26**, 434–441 (2002)
11. Pressey, R., Bottrill, M.: Approaches to landscape- and seascape-scale conservation planning: convergence, contrasts and challenges. Oryx **43**, 464–475 (2009)
12. Pressey, R., Watts, M., Barrett, T., Ridges, M.: The C-Plan conservation planning system: origins, applications, and possible futures. In Spatial conservation prioritization: Quantitative methods and computational tools. Oxford University Press, Oxford (2009)
13. Rebelo, A., Siegfried, W.: 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**, 243–252 (1992)
14. Watts, M., Ball, I., Stewart, R., Klein, C., Wilson, K., Steinback, C., Lourival, R., Kircher, L., Possingham, H.: Marxan with zones: Software for optimal conservation based land- and sea-use zoning. Environmental Modelling & Software **24**, 1513–1521 (2009)
15. Wilson, K., Underwood, E., Morrison, S., Klausmeyer, K., Murdoch, W., Reyers, B., Wardell-Johnson, G., Marquet, P., Rundel, P., McBride, M.: Conserving biodiversity efficiently: What to do, where, and when. PLoS Biology **5**, 1850–1861 (2007)
16. Yokomizo, H., Possingham, H., Thomas, M., Buckley, Y.: Managing the impact of invasive species: the value of knowing the densityimpact curve. Ecological Applications **19**, 376386 (2009)