

A Rapid Assessment Agent Based Model for Natural Resource Management

Boschetti, F, D. McDonald and M. Brede

CSIRO - Marine and Atmospheric Research, Australia
Email: [Fabio.Boschetti,David.Mcdonald, Markus.Brede][@csiro.au](mailto:[Fabio.Boschetti,David.Mcdonald, Markus.Brede]@csiro.au)

Keywords: *ABM, Multiple Use Management, Human-Computer Interaction, Optimisation, Inversion*

EXTENDED ABSTRACT

The multiple use management of a natural resource involves finding a compromise which reasonably accounts for the economic interest of the parties involved and for the sustainable conservation of the resource. Conceptually, we can envisage an abstract space, in which all possible available strategies exist; resource managers can then search this space in order to find a desired compromise.

When numerical modelling is used as a resource management tool, this abstract space becomes somehow less 'abstract', since its (albeit simplified) representation coincides with the set of model input parameters. The numerical model then maps points in this input parameter space into an output data-space where projected outcomes of management decisions can be viewed and analysed.

Seen from this perspective, a resource management problem appears similar to many engineering or applied mathematical problems, in which a numerical optimisation tool can be used to map a desired policy outcome (from the outcome space) back into the strategy space, thereby easily deducing a set of reasonable, if not optimal, strategies.

Researchers and practitioners experienced with resource management problems are well aware that this black-box, purely algorithmic approach is unlikely to work in practise, for several reasons:

- 1) optimisation routines need to convert the judgement of the suitability of a strategic outcome into a numeric value, which is then minimised or maximised. Since often this judgement needs to account for contradictory requirements, a numerical expression is often very difficult to implement and may be misleading.
- 2) The objectives of a strategy outcome need to be stated clearly a priori. In real cases,

however, defining such requirements is the result of negotiations and forms part of the purpose and outcome of the modelling exercise itself.

- 3) As a consequence of point 2, the goal of the resource management may change during the process, when more information becomes available and negotiations proceed; external interventions which change the context of a modelling exercise cannot be implemented in a black-box, purely algorithmic fashion.

These perceived shortcomings are unfortunate, because they may prevent use of the considerable technological development and expertise that has been accumulated for decades in the field of numerical optimisation and which could surely be valuable in the field of resource management and modelling.

In this paper we describe an approach which attempts to overcome the limitations described above; the purpose is to enable a manager (or a management team) to use a numerical optimisation routine (a Genetic Algorithm in this case) as an *optional addition* to the more traditional running of a numerical models as tools in resource management. Furthermore, at each stage of the process, the manager is in full control of the operation of the system and able to interact with it at different levels. This interaction results in a) a very flexible way to define the suitability of a strategy outcome, b) no need to define the goals of a resource management a-priori and c) the possibility of changing goals at any stage of the process as a result of the information accumulated so far. Basically we aim to see the goals of the resource management emerging as a result of the interaction between the system and the manager.

1. INTRODUCTION – THE TEST CASE

Rather than describing the tool first and showing an example afterwards, in this paper we take the reader through an example and we use this example to describe the tool's components and use.

Our test case is a simplified representation of the fishery in the Ningaloo Marine Park, in Western Australia. As described in Figure 1, we assume the park includes two fishing zones, each characterised by its own fish species and each containing a target species. At each iteration a fishing fleet composed of 50 vessels decides which fishing zone to target and fishes an amount of fish determined by the quota allowed in the fishing zone. The numerical modelling is carried out with the model described in the next section. This is the first step of an ongoing research project; the data we used as initialisation to the code are not calibrated yet, and consequently the results are meaningful only for demonstration purposes.

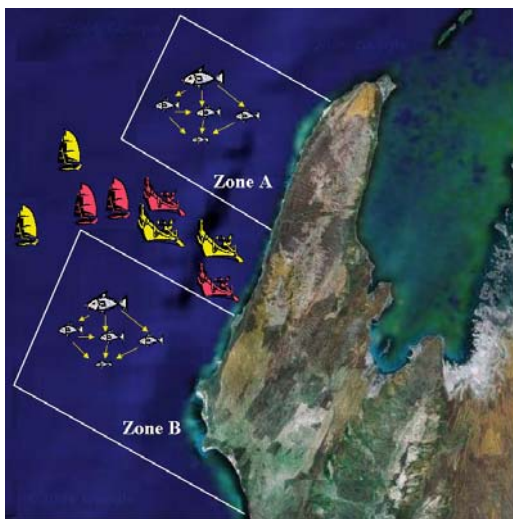


Figure 1. The test case used in our demonstrational modelling exercise.

2. THE NUMERICAL MODEL

In this work we employ the numerical model MacVitro, which is in the initial stage of development. At this stage, MacVitro includes 5 modules:

1) An ecological module. This accounts for trophic relations among different species; at present five

fish species within a 3-level food-web are modelled, but this can be extended or reduced easily, if needed. Trophic relations are implemented as differential equations and solved numerically.

2) A fishing module; this models the fishing behaviour, including access to the fishing zones, the sharing of the catch among vessels targeting the same zone and the choice of the target species.

3) An economic module; this models the fishers' decision making; fishers store their past record of catches and choose which fishing zone to target according to a prediction of what the most profitable zone might be in the next iteration. The prediction can be carried out either by trying to maximise the catch of each individual, or by accounting for the behaviour of the overall fleet (Boschetti, 2007).

4) A fishing regulation module; this defines the fishing regulations at each fishing zone; these include the maximum allowed quota, the beginning of the fishing season and its duration. The module can be extended easily to include other regulation criteria and instruments

5) A management tool; this controls the model runs and consists of a set of Graphic User Interfaces (GUIs) which allow the resource manager to a) decide what strategy to test b) evaluate and rank the strategic outcomes and c) employ the numerical optimisation tool, if required. The description of this module and its use are presented in the next section.

6) A numerical optimisation tool; with this, the manager can search the strategy parameter space. In this implementation we use a real-coded Genetic algorithm, whose specific implementation can be found in Boschetti et al. (1996).

It is important to notice here that the management module and the optimisation routine do not depend on the other specific modules (1 to 4); alternative ecological models may be substituted easily.

3. THE MANAGEMENT MODULE: MANAGING AND CONTROLLING THE SCENARIO TESTING

In this section we guide the reader through a hypothetical strategy testing exercise based loosely on the Ningaloo Marine Park, and, in doing so, explain the use of the management module.



Figure 2. Initialisation GUI.

When MacVitro is run, the user is first presented with the initialisation GUI shown in Figure 2. On the top part of the GUI we can see 4 panels named Scenario 1-4. These correspond to 4 management strategies the user will analyse and compare. Each scenario includes variables for each of the 2 fishing zones we model. Via this GUI, the user can provide MacVitro with the input parameters of the fishing regulation module, that is the allowed quota, start and duration of the fishing season for each fishing zone. Should the user decide not to input all required values, MacVitro assigns random numbers between specified ranges to the missing values. A smaller panel towards the bottom of the GUI allows specification of the seed of the random number generator, should the user wish to be able to re-run the same scenario testing starting from the same configuration.

Once the user is satisfied with the initialisation, MacVitro runs the 4 chosen scenarios and subsequently a new GUI appears, as shown in Figure 3. This GUI is divided into 2 main panels: a green panel on top and a red panel at the bottom.

At first only the top panel is used. The green panel includes 4 smaller panels, one for each of the scenarios tested. Each scenario panel provides information about both fishing zones. Let's consider the panel for Scenario 1. The top part shows a time series plot for fishing zone A. The red line shows the amount of biomass of the target species of interest, while the black line shows the overall catch in the same zone. Our simulation covers 3 years, with one fishing season per year, which explains why we have 3 peaks in the catch time series. Immediately on the right hand side of the plot, we have a window which shows 5 fields.

The 2 fields at the bottom display a summary of the time series. The 'Catch' field shows the integral of global catch on the fishing zone, while the 'Biomass' field shows the amount of biomass for the species of interest at the end of the run. Above these two fields, three further fields show the fishing regulation input values which produced the scenario outcome (quota, start and duration of the fishing season; these were input by the user via the initialisation GUI, or were randomly assigned). Notice that, unlike the previous two fields, these three fields can be edited. The lower part of the Scenario panel shows the same content for fishing zone B.

Finally, immediately above the top plot, slightly on the right hand side, an editable field allows the user to provide an evaluation of the scenario outcome. At this stage it displays the writing 'Not yet ranked'; we call this the 'ranking' field and we will discuss its use below.

Via this GUI the user can now analyse the outcome of each of the four scenarios and in particular compare these outcomes against one another. This is clearly the most important part of the overall process and need not necessarily be carried out by a single user. Rather it can provide an avenue for discussion among a management team, including parties with conflicting interests.

At this stage, the user has two options: he can either control further modelling, by providing new scenarios to test or he can use the optimisation module to help the process. We discuss both options in the following sections.

3.1. User Guided Procedure

The user can decide what scenario should be tested next. In Figure 3, for example, we see that Scenario 3 resulted in a population crash in both fishing zones A and B; in this case the user can modify the fishing regulation parameters for both zones (or only one if the population crash occurred in a single zone). Once this is done, MacVitro re-runs only the scenarios which have been modified and updates the plots in the corresponding panels. The user can now compare the new scenarios to the scenarios previously run. Obviously, had the user modified all scenarios, or input completely new parameters for all scenarios, all plots would be updated.

3.2. Assisted Search

Alternatively, the user may decide to employ the Genetic Algorithm to facilitate the search in the scenario parameter space. The rationale for doing

so is that optimisation routines like GA are tailored to search parameter spaces; in highly non linear problems (like ecological modelling), models may generate counter-intuitive results and a well designed search may reveal combinations of input parameters resulting in scenario outcomes that even an expert user may not predict; it is this possibility of generating unexpected results which makes optimisation routines useful in these kinds of problems.

optimisation). The user can provide this feedback via the 'ranking' field described above. Via a drop-down button, the user can instruct the GA on whether the outcome of a specific scenario is a) the best in comparison to the other outcomes under display, b) the worst or c) somewhere in between, in which case a definite numerical rank can be provided. In the Artificial Intelligence parlance, this feedback can be thought of as training which the user provides to the GA. The GA then takes this feedback into account for its own internal

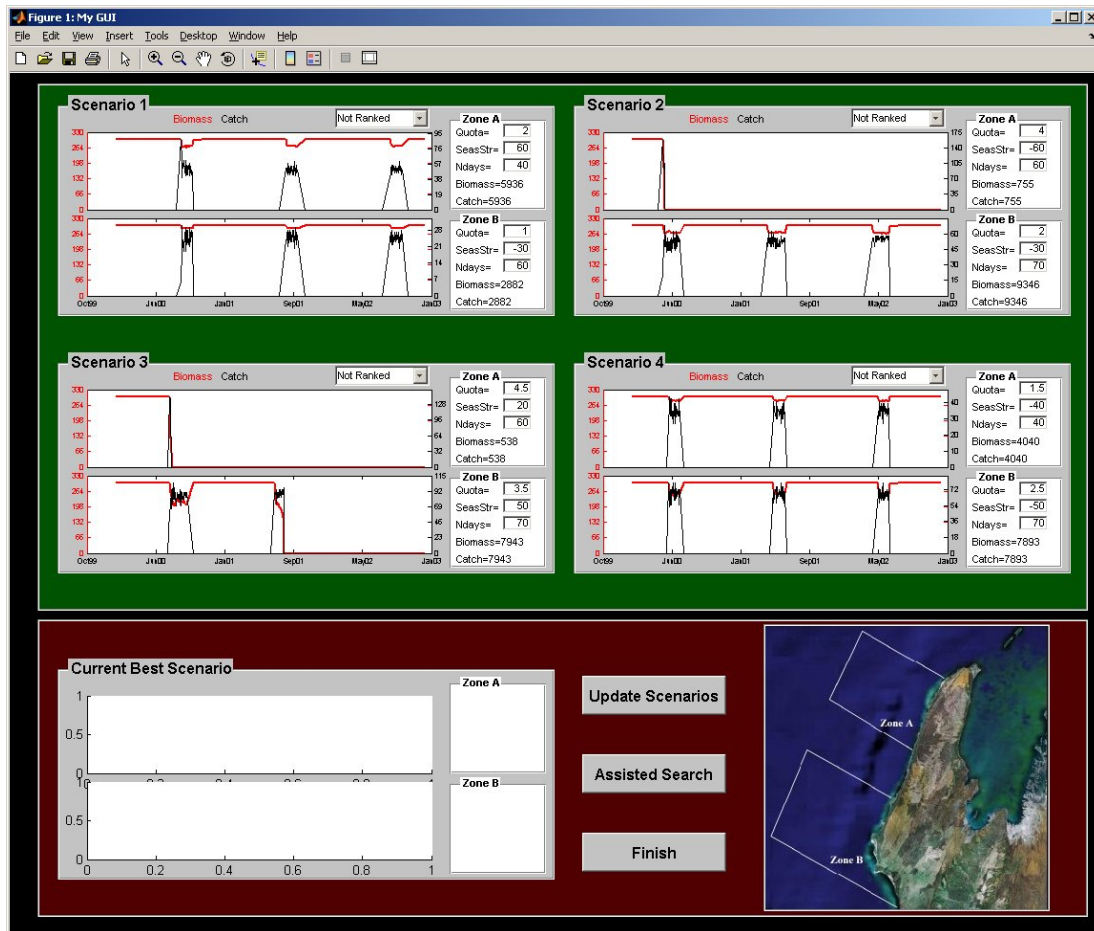


Figure 3. In the green panel, the outcomes of the scenarios chosen by the user are displayed, together with a number of editable text fields, allowing the user to either modify the scenario parameters, or provide a feedback on the outcomes acceptability, which is used to train a Genetic Algorithm.

In order to carry out this search, the GA needs some indication from the user of what scenario resulted in good outcomes and which resulted in bad outcomes (the equivalent of 'objective function evaluation' in the parlance of numerical

functioning (see Davies, 1991 or Boschetti et al., 1996 for explanation of the GA process) and generates a new set of scenarios. MacVidro then runs these scenarios, which appear in an updated GUI, as shown in Figure 4.

Here the green panel at the top includes the four scenarios recently generated. Unlike in Figure 3, in Figure 4 the red panel at the bottom is also used, where the 'best' scenario found so far is shown. This has two purposes: first, the user does not risk losing the information of the best outcome should he decide to proceed with the search in order to explore further options. Second, the user can evaluate whether the GA search displays improvement in the scenario outcomes or whether it is reasonable to believe that the best scenario has been found and no further modelling is needed.

At this point the user is free to choose either to proceed via the User Guided Procedure or via the Assisted Search. In either case, the best scenario found so far is included in the process, which means that it also can be modified and ranked.

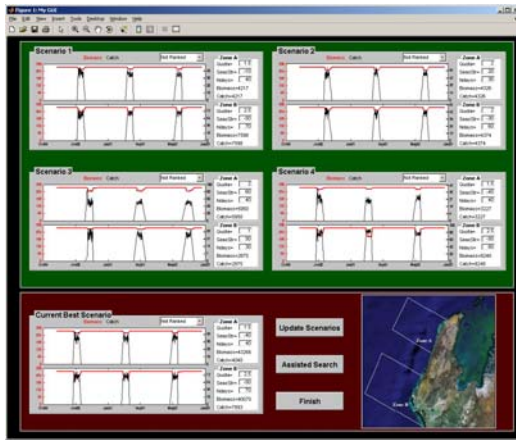


Figure 4. Outcome of the use of the GUI in conjunction with the run of a Genetic Algorithm. In the bottom panel, the best solution the user has found so far is displayed, together with the outcomes of the new runs generated by the GA in the red panel.

4. DISCUSSION

Unfortunately, at this stage of our research we are unable to show any real-world application of the method. The test case, based on the Ningaloo Research Program (NRP), is still at an evaluation stage; real data will soon be available and a first assessment of the method's potential will be one of the next steps in our research.

Nevertheless, we have reasons to be mildly optimistic about its possible usefulness. The method we describe is a modification of what is called Iterative Evolutionary Computation (IEC, see Takagi, 1998). In Takagi (2001) a description

of the technique, together with an exhaustive list of soft-engineering applications can be found. The authors also have a considerable experience with the application of IEC to several geo-scientific problems (Wijns et al, 2003, 2003b, Boschetti et al, 2002). Geo-scientific applications share many features with ecological modelling: they are high-dimensional, highly non-linear, require approximate solutions to differential equations, are often spatially explicit and can include combinations of continuous and discrete computation. Our experience with these applications is that the combination of the Genetic Algorithm search and the feedback provided by the expert user can be extremely powerful, allowing the discovery of acceptable solutions in surprisingly few iterations. The conjecture behind such fast convergence property is that the expert feedback provided by the user effectively makes the mapping between input parameters and outcome space smoother than would be provided by traditional numerical cost function measures (Boschetti, 2004).

Obviously, the requirement for a human intervention in the process also carries a number of drawbacks. We can't expect the user to be able to control several hundred input parameters, as ecological models often require; a suitable selection of the important parameters must be done before the method can be used and it is inevitably problem specific. Also, the computational effort required by the ecological models will affect the speed with which the user can interact with the search process. When used in combination with very large models, the user may need to interact with the GUIs only once or twice a day, letting the model run in the background for hours in between. In this case, a few days may be needed in order to obtain an acceptable solution; however, in our experience, this cumbersome approach did not degrade the result of the experiments and still proved much faster and effective than fully human-driven exploration of the same parameter space (Boschetti and Moresi, 2001).

Ecological models also have features which are not shared by other modelling applications, like the interplay between physical, biological and human components; whether the method discussed in this paper can be useful for addressing these problems, and ultimately prove useful in the multiple-use management of renewable resource is something we will report on in the coming years.

5. CONCLUSION

We described a set of simple GUIs which allow a user to control multiple runs of an ecological model and to evaluate and compare their outcomes. At any point during the process, the user can choose new scenarios or modify previous runs. Alternatively, he can use a Genetic Algorithm as an aid in the search for suitable outcomes. The Genetic Algorithm does not need a numerical cost function evaluation, rather it is trained by the user who provides simple and approximate evaluation of the run outcomes.

The crucial feature of the method is that it allows the goal of the management strategies to arise as a result of the interaction between the user and the model, rather than being defined a priori.

We envisage that a management team, rather than a single user, may also employ the method as an avenue for communication, in order to discuss the conflicting aspects of different model outcomes along the path to finding workable compromises.

6. REFERENCES

- Boschetti, F., M. Dentith, and R. List (1996), Inversion of seismic refraction data using genetic algorithms. *Geophysics*, 61, 1715-1727.
- Boschetti F. and L. Moresi (2001), Interactive inversion in Geosciences, *Geophysics*, 64, 1226-1235.
- Boschetti F., C. Wijns, and L. Moresi, (2002), Effective exploration and visualisation of geological parameter space, *Geochemistry, Geophysics, Geosystems*, 4(10), 1086..
- Boschetti, F., (2004), Controlling and investigating Cellular Automata behaviour via interactive inversion and visualization of search space, *New Generation Computing*, Special Issues on Interactive Evolutionary Computation, 23(2), 157-170.
- Boschetti, F., (2007), Improving resource exploitation via Collective Intelligence by assessing agents' impact on the community outcome, *Ecological Economics*, 63, 553-562.
- Davis, L., 1991, Handbook on genetic algorithms: Van Nostrand Reinhold.
- Takagi, H., (2001), Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation, *Proceedings of the IEEE*, 89(9), 1275-1296.
- Takagi, H. (1998), Intercative evolutionary computation: system optimization based on human selective evaluation, *IEEE Int. Conf. on Intelligent Engineering Systems*, 1-6, Vienna, Austria, Sept 1998.
- Wijns, C., F. Boschetti, and L. Moresi (2003), Inverse modelling in geology by interactive evolutionary computation, *J. Struct. Geol.*, 25(10), 1615-1621, doi:10.1016/S0191-8141(03)00010-5.
- Wijns, C., T. Poulet, F. Boschetti, C. Dyt, and C.M Griffiths, (2003), Interactive inverse methodology applied to stratigraphic forward modelling, in: A. Curtis and R. Wood (Eds.), *Geological Prior Information*, Geol. Soc. of London Special Publication.