

Interactive modelling for natural resource management

F. Boschetti^{1,3*}, A. de La Tour², E. A. Fulton¹ & L. R. Little¹

¹CSIRO Marine Research - Marine and Atmospheric Research, Australia

²INA-PG (AgroParisTech), MINA, 7b boulevard Jourdan, Paris, France

³School of Earth and Geographical Sciences at the University of Western Australia
35 Stirling Highway, Crawley 6009, Western Australia

* Corresponding author

Abstract

We present a modelling approach to rapidly assess the effect of management decisions on ecological problems and demonstrate its use in fishery management. Each stage of the approach is controlled by a graphical user interface which allows a team of non-expert modellers to compare the outcomes of multiple model simulations and to decide what further simulation is needed.

A distinguishing 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, as well as to change during the process in response to the information and the insight such modelling may provide. 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 potentially conflicting aspects of different model outcomes along the path to finding workable compromises. We discuss an application of the approach to the sustainable management of a recreational fishery in a marine park in Western Australia.

1 Introduction

The multiple use management of a natural resource involves finding a compromise between different types of natural resource use including extractive and non-extractive, and between the short and longer-term sustainable exploitation. Effective management of a natural resource involves planning and setting goals and objectives, designing strategies with which to achieve them, implementing the management strategies, and evaluating them to determine if they achieve the objective for which they were designed (Sainsbury et al. 2000).

Conceptually, we can envisage an abstract space, in which all possible available strategies (that is possible management interventions) exist; in principle, resource managers could then search this space in order to find a desired compromise. When simulation 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 simulation model then represents a mapping of points in input parameter space to 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 (Sainsbury et al., 2000; Doyen and Béné, 2003; Doyen et al., 2007; Martinet et al., 2007).

Researchers and practitioners experienced with resource management problems are well aware that this approach would be unlikely to work in practise, for several reasons:

1. optimisation routines need to convert the judgement of the suitability of a strategy outcome into a numeric value, which is then minimised or maximised. A meaningful numerical expression is very difficult to implement since most often a) different objectives are sought to be reached at the same time, b) the relative importance of these different objectives need to be weighted subjectively and c) these objectives may represent contradictory requirements.
2. Management objectives need to be stated clearly a priori. However, defining objectives is often the result of negotiations and forms part of the purpose and outcome of the modelling exercise itself.
3. As information becomes available to stakeholders through the modelling process management objectives may change, as stakeholders may realise they are unrealistic or incompatible with other objectives.

In this paper we describe an approach which attempts to overcome the limitations described above by coupling a computer simulation modelling tool with input derived from expert users and by allowing the human supervision to become an integral part of the modelling process. The purpose is to enable a manager or a management team to use a numerical optimisation routine as an *optional* addition to the more traditional running of a simulation model as a tool in resource management. Furthermore, at each stage of the approach, the user is in full control of the modelling process 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. In this manner, we aim to see the goals of the resource management emerge as a result of the interaction between the modelling exercise and the manager(s). The framework makes explicit the role of the simulation model in providing understanding of a system rather than simply predicting a possible outcome.

2 The approach

The approach involves a computer model, (referred to as Simulation Model in the rest of this paper) which captures the behaviour of an ecological and social system (in this specific case a fishery) in order to predict the environmental, social and economic effect a management strategy may have. This provides a mapping between the available management options and the estimated effects. This Simulation Model is embedded in a software package called ScenarioLab which also includes a Graphical User Interface (GUI) to facilitate the interaction between the model and the user(s) as well as an optimisation routine (a Genetic Algorithm in this case) which can optionally be used. ScenarioLab is designed to allow users (managers and stakeholders) with little modelling or ecological expertise to use the Simulation Model in order to explore management options and their effects. Figure 1 summarises the use of

ScenarioLab; this includes running the Simulation Model and analysing the output according to criteria which may be subjective, and not necessarily defined a priori (label 1 in the figure). This is followed by two options for the users: if they want to investigate the effect of different specific management options they can do so (label 2, this effectively represents a direct exploration of the model input space by the users; see also ‘forward modelling’ in Section 6). If the problem is more complicated and alternative management options are not obvious or the users want to ensure they are not disregarding less intuitive but still available management options, then they can pass their evaluation of the Simulation Model outputs to an optimisation routine (label 3) which automatically performs the exploration of the input space and provides new potential management scenarios to evaluate via the Simulation Model. Details of the approach and software implementation are given while the reader is guided through an example application.

3 An example application: fisheries management in a marine park

Our test case is a simplified representation of the fishery in the Ningaloo Marine Park, a 300 km long fringing coral reef in the northwest coast of Western Australia (see Figure 6). For a number of decades the park has undergone increased fishing and tourism pressure despite being located 1200 km from the closest large city (Perth). Our study aims at devising a set of fishing regulations that will ensure a sustainable future for the park; recreational fishing currently represents one of the main drivers for local tourism. Previous initiatives have established a number of sanctuary zones and imposed a set of recreational fishing regulations, as well as a complete ban on any form of commercial fishing within the park. The evaluation of the effectiveness of current regulations and sanctuary zones and the assessment of the possible need to adjust or redefine management objectives, are the main purposes of our analysis.

The studied area, which includes the park and surrounding region, has been divided into a number of zones in accordance with both ecological and administrative constraints and is displayed in Figure 6 (where the boundaries of the Ningaloo Marine Park and the existing sanctuary zones are given for reference). The boundary of the park roughly separates the shallow inshore waters from a deeper marine environment along a 40 metres depth contour and it can also represent the margin between two different ecological environments. A description of the modelled species is given in the next section.

4 The Simulation Model

The purpose of the Simulation Model is to estimate the expected effect of a number of fishery regulation options which are available to the management team. These management options control the number of dispensed fishing licences, the total areas reserved as sanctuary zones, daily catch limits, and minimum and maximum legal lengths for two species of interest. Once such regulations are defined, the Simulation Model mimics the behaviour of a fishing fleet (in this case of recreational fishers) and the effect of their catch and by-catch on the modelled foodwebs. The output is then presented to the user in the form of time series of biomass and catch for the different species.

For each regulation option the model is run a number of times under different model settings, characterised by different ecological inputs. The purpose is to capture, to some extent, the uncertainty in the model parameterization resulting from the lack of precise biological data as well as the inherent uncertainty of biological and ecological processes. This choice aims to inform the users of the range of variability which they need to expect in the modelled response even given precisely defined fishery regulations. In order to keep the computation load acceptable, in this application we decided to run the model under three different model settings, which try to capture 'pessimistic', 'neutral' and 'optimistic' system states.

A detailed description of the Simulation Model implementation can be found in the Appendices; here we briefly describe the main modules.

The ecological module accounts for trophic relations among different species; at present five fish species within a 3-level food-web are modelled for each environment/zone (Figure 2). This model includes:

- 1) a lower trophic level prey, representing the basic food source for the entire foodweb.
- 2) 3 intermediate species; these include two species targeted by recreational fishing, whose dynamics are the focus of the study, as well as a competitor which undergoes possible by-catch;
- 3) a top predator, which is also accounted for as by-catch.

The size of the foodweb is constrained for the sake of computational speed.

Two types of foodwebs are used in order to discriminate between the different ecological zoning. A detailed description of the ecological module is given in Appendix C .

The fishing module models the fishing behaviour, the sharing of the catch among vessels targeting the same fishing zone, the effect of gear selection and the choice of the target species. A detailed description of the fishing module is given in Appendix D

The economic module 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 is carried out by attempting to maximise the catch of each individual vessel while accounting for the behaviour of the overall fleet (Boschetti, 2007). The economic module is described in detail in Appendix E

A fishing regulation module; this defines the fishing regulations at each fishing zone and represents the management strategy options (input space of the Simulation Model). These include the extent of sanctuary zones, the number of fishing licences allowed, and the bag limits, and legal minimum and maximum length for two species of interest: Spangled Emperor (*Lethrinus nebulosus*) and Chinaman Cod (*Epinephelus rivulatus*). The module can be extended easily to include other regulatory and assessment criteria.

The scenario management module; this includes the Graphical User Interfaces which allow the resource manager to a) decide what strategy to test; b) evaluate and rank the strategy outcomes; and c) employ the numerical optimisation tool, if required. This module is described in detail in the next section while more information about the technical implementation of the Graphical User Interface and the software flowchart are given in Appendix A

The numerical optimisation tool which allows to search the strategy parameter space. In this implementation we use a real-coded Genetic Algorithm (Davis, 1991). Details about this module can be found in Appendix B

5 The Scenario Management Module

At the core of this method lies the belief that the aim of the management program may not need to be set a priori. In the following we assume that a team including managers and stakeholders (rather than a single expert user) may jointly undertake the exercise, not only for decision making but also as an avenue for communication. This leads to exploration and discussion of the conflicting aspects of different model outcomes.

When ScenarioLab is launched the management team is first presented with the Initialisation GUI shown in Figure 7. On the top left hand side a panel called “Parameter Ranges”, lists the parameter values of eight specific regulation items the management team can use to control fishing in the park. Two of these parameters are global: the extent of the area closed to fishing (expressed as a multiplying factor of total area of the current sanctuary zones) and the number of fishing licences dispensed. The other parameters refer to the specific species of current management concerns: Spangled Emperor and Chinaman Cod. For each species, the management team can impose regulations on the maximum daily catch as well as minimum and maximum legal lengths. The “Parameter Ranges” panel is editable and allows for the definition of the minimum and maximum value allowed for each parameter; that is the ‘reasonable’ range within which we search for suitable management options. These values define the bounds of the Simulation Model input parameter space and they are naturally required by most numerical optimisation routines.

At the bottom left of Figure 7 a panel called “Initial Values” contains four smaller panels called Specification 1-4. These are scenario specifications that correspond to a management strategy and are defined by a set of eight numbers chosen randomly between the ranges defined in the “Parameter Ranges” panel (in Figure 7 only five fields can be seen, the two global parameters and the quota and minimum and maximum legal length for the spangled emperor; the parameters for the Chinaman Cod can be selected via the drop-down menu in the middle of each Scenario specification panel). Each field is editable so the management team can modify the random values should it want to analyse a given scenario. Once the team is happy with the chosen values these input data are passed to the Simulation Model to run the simulations, at the end of which the Evaluation GUI (shown in Figure 8) is displayed.

The evaluation GUI shows three panels, a green one on top, a red one to the lower left and a blue one to the lower right. Within the green panel are four smaller grey panels, one for each scenario specification. Each panel displays the output of an independent

run of the Simulation Model. As an example, we analyse Specification 1. The plot displays two sets of time series: for a given species, the red time series show the evolution of the biomass through time, while the black time series indicate the catch. Each set of time series consists of three individual time series (dashed, thick and thin lines) describing the behaviour of the species under different model settings ('pessimistic', 'neutral' and 'optimistic' system states) in order to capture the inherent uncertainty in the model parameterization and process, as described in Section 4. By using different drop-down menus the time series of all modelled species in all modelled zones can be analysed.

At this stage, the management team has two options: it can either direct further modelling by defining new scenarios to test or it can use the optimisation module to help the process.

5.1 User Guided Procedure

The team can decide what scenario should be tested next (this corresponds to arrow 2 in Figure 5). Should the team detect any feature in a scenario outcome worth further analysis it can modify some of the fishing regulations for that scenario via the editable field on the right hand side of the plot (which currently shows the specifications of the individual run which generated the plotted time series). The Simulation Model is then run and the new output plotted in the refreshed panel for the team to examine the effect of new regulation.

5.2 Assisted Search

Alternatively, the user may decide to employ the Genetic Algorithm (GA) to facilitate the search in the scenario parameter space. This corresponds to arrow 3 in Figure 5. 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 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.

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 ones (the equivalent of 'objective function evaluation' in the parlance of numerical optimisation). The user can provide this feedback via the 'ranking' drop-down menu (currently displaying 'Not ranked' in Figure 8), which the management team can use to instruct the GA on whether the outcome of a specific scenario is the 'best' in comparison to the other outcomes, the 'worst' or somewhere in between, in which case a numerical rank can be provided.

This feedback can be thought of as training the GA. The GA then takes this feedback into account for its own internal functioning (Davis, 1991) and generates a new set of scenarios. Through this menu the management team can provide a subjective and relative evaluation of the suitability of the model outcome. The evaluation is relative in the sense that the model outcome is compared to the outcomes in the other three scenarios. The evaluation is also subjective because it is obtained by assessing various aspects of the model results and depends on the management team as well as on the purpose of the modelling exercise. In the case of multiple users we expect that

the ranking of the model outcomes will involve discussions with exchange of ideas, information, views and opinions.

Finally, in the red panel at the bottom left another scenario specification and output panel is displayed representing the best outcome among all models run so far. This panel has two roles: first it ensures the best result obtained so far is not lost and second it provides a link, and thus continuity, between the different iterations in ScenarioLab.

The process then iterates with the management team choosing between the “User Guided Procedure” and the “Assisted Search” until a set of satisfactory regulations is obtained or enough information has been collected.

6 Relation to other modelling approaches

Computer models can be used for different purposes and consequently in different modes. For the sake of this discussion we group these modes roughly into four classes:

- 1) Forward modelling; a model can be used in order to reproduce an observed behaviour or to explore the model behaviour. This is usually done by modifying some input parameters and then observing and classifying the variation in the model output. In this mode the user implicitly performs an optimisation (if he/she aims to reproduce the observed behaviour) and a sensitivity analysis (by observing the impact of a change in the input parameters), however no computer routine is explicitly used to facilitate these tasks. At times the user may not even be aware of his/her mentally carrying out these two mathematical processes; this approach is called ‘forward modelling’ in some disciplines. This is probably the most widespread use of computer modelling and some users can become extremely proficient at it. The main drawback of this approach is that a considerable amount of time is required to develop a deep knowledge of a problem and of a specific numerical model.
- 2) Traditional numerical optimisation; in this mode an optimisation routine is used to explore the model parameter space in order to match a target model behaviour or response. This is a typical use in engineering or applied mathematics. It is a way to formalise and speed up the forward modelling approach but, as discussed in the Introduction, depends crucially in the definition of the cost function which determines quantitatively how well the target behaviour has been reconstructed. This cost function needs to be given a priori, thus changes in the purpose of the optimisation are not allowed without re-writing the cost function and restarting the process.
- 3) Teaching or education purposes; here the model is used to train users to carry out a certain task or to learn a specific skill. The ‘flight simulators’ are a well known example, but similar ideas can be developed to teach high-school students about dynamical processes (Hogan and Thomas, 2001) and to train managers to address complex tasks (Dorner, 1996).
- 4) Participatory modelling; here modelling is used as an avenue for communication aimed at addressing a problem, reaching an agreement or

highlighting the issues which need to be better modelled in the next phase of the project.

Obviously the above classes should not be considered as rigid, since mixed approaches are possible. To a certain extent, ScenarioLab allows inclusion of most of the above modes within a single platform and a single approach. The forward modelling mode is represented by arrow 2 in Figure 5 and ScenarioLab can be used purely in this fashion if the user so wishes. Compared to carrying out forward modelling without the Evaluation GUI, this approach provides some minor advantages in the automated visualisation and the automated storage of the tested input parameter sets. More importantly however, the Evaluation GUI allows manual modification of sets of parameters at the same time (four in the example given in this paper) and to compare the different results without having to retrieve stored model output. The current ‘best’ outcome (red panel in Figure 8) also provides a reference to the best scenario so far generated, which is something that is easy to lose track of when tens or hundreds of scenarios have been run manually.

The numerical optimisation mode is represented by arrow 3 in Figure 5; as discussed at length above, in the current implementation the subjective evaluation of the user replaces the numerical evaluation of the cost function, which is standard in traditional optimisation approaches. While the rationale for doing so has been discussed in the Introduction, an option for traditional optimisation could obviously be included in ScenarioLab should the need arise and provided the limitations of the approach are well understood. This could be done, for example, by providing a number of pre-defined standard cost functions in a further GUI window as well as an option for choosing real data or sets of expected behaviours to match.

We believe that one of the important features of ScenarioLab is the possibility to combine the subjective, human-driven optimisation *with* the forward modelling approach which, in the optimisation jargon, provides for both *exploration* and *exploitation* of the input parameter space: that is, it allows the users to alternate between a global optimisation stage (human-driven optimisation) in which novel behaviours are *explored*, and a local optimisation stage (forward modelling) in which the current behaviours are locally explored (*exploited*, in optimisation jargon).

Teaching as well as a participatory modelling are also possible in ScenarioLab; an instructor can guide the analysis of how the model outputs relate to the input parameters for teaching purposes and a facilitator can assist a discussion leading to the subjective judgement of the model output in a participatory setting. In the latter case, the validity and usefulness of the model can also be discussed and the ranking of the model outputs can highlight potentially diverging views on the interpretation of model outcome and the purpose of the management exercise itself.

Once again, we believe that the most powerful component of the approach lies in the possibility of incorporating all these approaches into a unified framework, thereby allowing for analysis, exploration, learning and discussion to occur as components of a single process.

7 Generalising ScenarioLab to other modelling problems

One of ScenarioLab's features is that the criterion according to which a model run is judged is not fixed and can change during the iterations, that is, it can change as a result of the insights and learning which occur during the modelling exercise itself. This is an extremely powerful feature, both conceptually and practically, which goes beyond the classic view of a model as a fixed and closed system. It enables the users to do what they are best at, which is to provide judgement, insight, intuition and even ethical concerns and to make the most of the only thing a computer does well: fast computation.

Technically this is possible by passing to a GA the subjective ranking provided by the users in place of the commonly expected numerical evaluation of a cost function. This idea is at the core of what is called Interactive Evolutionary Computation (IEC) (Takagi and Iba, 2005; Takagi and Ohsaki, 2007). (Takagi, 2001a) describes the technique, together with an exhaustive list of software-engineering applications. In recent years we have gained a considerable experience with the application of IEC to several complex numerical problems (Boschetti and Moresi, 2001; Wijns et al., 2003a; Wijns et al., 2003b; Boschetti, 2005) which 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. In these experiments interactive optimisation has been carried out by students, modellers, practitioners with no specific modelling expertise and experts in numerical optimisation (so far we have never tested the method with teams of multiple users). Our experience with these applications is that the combination of the GA search and the feedback provided by the expert user can be extremely powerful, allowing the discovery of acceptable solutions in very few iterations (Wijns et al., 2003a) and in considerably less time than required by traditional forward modelling (Boschetti and Moresi, 2001).

To users familiar with numerical optimisation and with traditional GA this convergence speed can be surprising. Our conjecture is that this enhanced performance is due to at least three main reasons. First, the expert feedback provided by the user effectively makes the mapping between input parameters and outcome space smoother than the one provided by traditional numerical cost functions (Boschetti, 2005). For example, a user may disregard minor numeral variations in an image pixel values or minor oscillations in a time series which do not impact their overall problem-specific interpretation; a traditional numerical optimisation routine may spend several iterations optimising these minor variations with no apparent advantage to the user. At times these minor features may even result in the numerical optimisation routine being misled towards unpromising areas of the parameter space, which may result in a final solution of poor quality or in further computational effort to rediscover the more promising area. Second, a user can switch very quickly the attention of his/her analysis between different features of the model output; this results in implicitly giving different weights to different features in the solution, which may be relevant at different stages of the optimisation process. In other words, the user's subjective judgement 'adapts' to the different stages of the optimisation process while a numerical optimisation would be insensitive to them. Third, a user is in general more capable to discriminate between global and local improvement in a solution: the user subjective judgement allows for the search to rapidly jump between

different areas of the solution space without getting trapped for many iterations in the proximity of a single local minimum (Kishi and Takagi, 1999; Boschetti and Moresi, 2000). Rather than spending several iterations trying to fine tune a solution, a user is capable to judge whether a different area of the solution space should be investigated (and thus abruptly switch the solutions' rank) or to terminate the optimisation given that a satisfactory solution has been achieved.

Obviously, the requirement for a human intervention in the process also carries a number of drawbacks: we can't expect the users to be able to control and possibly manually tune the several hundred input parameters that 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. In principle, automated tools could be designed to support the user's task. As an example, an algorithm could be trained to 'understand' and 'replicate' the user implicit choices as we tested in (Kaltwasser et al., 2005). An algorithm of this sort could take over from the user once a sufficient training has been reached, after which the user could simply supervise the algorithm by checking that the ranking is consistent and/or restarting the training once he/she realises that new ranking criteria should be employed. Research into Artificial and Computation Intelligence is likely to provide tools of this kind and future applications of interactive optimisation may benefit from this.

Also, it might be argued that the subjectivity implied in the human intervention is exactly what we wish to avoid by using numerical models in the first place. There are several problems in industrial applications and applied mathematics to which this comment surely applies. However, these are problems for which the choice of the cost function to optimise is 'natural' in the sense that it clearly represents the purpose of the study and reliably 'measures' the level to which the purpose is achieved. As discussed in the Introduction this rarely applies to ecological or social problems in which subjectivity and uncertainty is inherent in the definition of the problem. Designing a cost function in numerical form to use with a traditional, rather than interactive, optimisation routine would merely hide, not remove, this subjectivity.

Another important drawback is that the computational effort required by the ecological models will affect the speed with which the users can interact with the search process. With very large and slow models, the users may be able 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. 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 trial-and-error exploration of the same parameter space (provided a single user was in charge of the modelling exercise and consequently could easily schedule the interaction with the model (Boschetti and Moresi, 2001)). However, it is unlikely that the same would hold true when a team of multiple users wishes to employ the method, since scheduling regular meetings would be impractical and the momentum of the discussion generated by the insights gained by the modelling would probably be lost.

This raises the question of what simulation models can be used in this approach. If, as we suggest, this tool is most suitable for 'small' and 'fast' simulation models, could it still have a role in addressing problems which require heavier computation? The answer may lie in a two stage approach in which two simulation models of the same

process are used to address a complex problem. ScenarioLab could then be treated as a demonstrational, educational and exploration tool in combination with the simpler of the two simulation models. In this setting its role is to introduce non-modellers (resource managers or stakeholders) to modelling and allow them to experience in first person what the modelling exercise involves and to understand what reasonable outcome it can generate. It also allows the same users to highlight, thanks to the insights the modelling can generate, what specific issues need further, more detailed analysis and which questions deserve further inspection.

This subsequent, more detailed analysis can then be carried out via the more complex simulation model. We expect that by using the simpler model to better define the crucial components of the problem, which would be far more time consuming to achieve directly with the larger model, this two-stage approach would allow a saving of time and resources. We also envisage that the involvement of end users and stakeholders from the beginning of the modelling exercise could result in a better definition and understanding of the expected results. This two-stage approach is currently being tested in a larger setting of the Ningaloo Park study and we will report on the results in the coming years.

Finally, models can be used to test the feasibility of proposed interventions, to find efficient ways to achieve specific goals but also to explore alternative options not yet considered. The latter can also be attempted via traditional numeral optimisation but designing an effective cost function for this purpose can be very difficult. This is where the user can be most valuable to the optimisation process by purposely choosing solutions with specific properties, for example being different from solutions already considered, thereby ‘biasing’ the optimisation towards unexplored areas of the solution space.

8 Conclusions

Two principles are at the basis of ScenarioLab. First, the evaluation of a modelling outcome is subjective and contextual: different users may judge an outcome differently depending on their expectations, needs, assumptions and expertise. Since assumptions, like expectations and expertise, may change as a result of the modelling results, the evaluation of a modelling outcome may change during a ScenarioLab session. Second, humans find it easier to express relative judgments (“model outcome 3 is worse than model outcome 7 but better than model outcome 2”) than absolute ones (“model outcome 3 is the fourth best among all runs”). These two principles have been the main driver for the development of ScenarioLab’s graphical user interfaces (GUI) which allows users to control and evaluate different model runs in parallel and to direct future modelling iterations.

For this to be possible the tool needs to a) be intuitive enough to be comfortably controlled by non-expert modellers, b) run reasonably fast so that the results can be computed in near real-time, c) provide a flexible way to define the suitability of a strategy outcome and d) allow modification of goals at any stage of the process, as a result of the information provided and the discussion it generated. We have described a number of implementation features and software tools which accommodate for such needs.

9 References

- Boschetti, F., 2005. Controlling and investigating cellular automaton behavior via interactive inversion and visualization of the search space. *New Generation Computing*, 23:157-169.
- Boschetti, F., 2007. Improving resource exploitation via collective intelligence by assessing agents' impact on the community outcome. *Ecological Economics*, 63:553-562.
- Boschetti, F., Dentith, M.C. and List, R.D., 1996. Inversion of seismic refraction data using genetic algorithms. *Geophysics*, 61:1715-1727.
- Boschetti, F. and Moresi, L., 2000. Comparison between interactive (subjective) and traditional (numerical) inversion by Genetic Algorithms. Congress on Evolutionary Computation CEC00. La Jolla, California, USA, La Jolla, California, USA.
- Boschetti, F. and Moresi, L., 2001. Interactive inversion in geosciences. *Geophysics*, 66:1226-1234.
- Brede, M., Boschetti, F. and McDonald, D., 2008. Strategies for resource exploitation. *Ecological Complexity*, 5:22-29.
- Davis, L., 1991. Handbook on genetic algorithms. Van Nostrand Reinhold.
- Dorner, D., 1996. *The Logic Of Failure: Recognizing And Avoiding Error In Complex Situations* Metropolitan Books, New York.
- Doyen, L. and Béné, C., 2003. Sustainability of fisheries through marine reserves: a robust modeling analysis. *Journal of Environmental Management*, 69:1-13.
- Doyen, L., De Lara, M., Ferraris, J. and Pelletier, D., 2007. Sustainability of exploited marine ecosystems through protected areas: A viability model and a coral reef case study. *Ecological Modelling*, 208:353-366.
- Hogan, K. and Thomas, D., 2001. Cognitive Comparisons of Students' Systems Modeling in Ecology. *Journal of Science Education and Technology*, 10:319-345.
- Kaltwasser, P., Boschetti, F. and Hornby, P., 2005. Measure of similarity between geological sections accounting for subjective criteria. *Computers & Geosciences*, 31:29-34.
- Kishi, K. and Takagi, H., 1999. Developing montage systems for study on Interactive evolutionary computation. SOFT 3rd workshop on Evaluation of Heart and Mind, Gamagoori, Japan.
- Martinet, V., Thebaud, O. and Doyen, L., 2007. Defining viable recovery paths toward sustainable fisheries. Elsevier.
- Sainsbury, K.J., Punt, A.E. and Smith, A.D.M., 2000. Design of operational management strategies for achieving fishery ecosystem objectives. pp. 731-741.
- Savit, R., Manuca, R. and Riolo, R., 1999. Adaptive Competition, Market Efficiency, and Phase Transitions. *Physical Review Letters*, 82:2203-2206.
- Takagi, H., 2001a. Interactive evolutionary computation as humanized computational intelligence technology. *Computational Intelligence: Theory and Applications*, Proceedings, 2206:1-1.
- Takagi, H., 2001b. Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the Ieee*, 89:1275-1296.
- Takagi, H. and Iba, H., 2005. Interactive evolutionary computation - Preface. *New Generation Computing*, 23:113-114.
- Takagi, H. and Ohsaki, M., 2007. Interactive evolutionary computation-based hearing aid fitting. *Ieee Transactions on Evolutionary Computation*, 11:414-427.

- Wijns, C., Boschetti, F. and Moresi, L., 2003a. Inverse modelling in geology by interactive evolutionary computation. *Journal of Structural Geology*, 25:1615-1621.
- Wijns, C., Poulet, T., Boschetti, F., Griffiths, C. and Dyt, C., 2003b. Interactive Methodology Applied to Stratigraphic Inverse Modelling. In: A.C.a.R.W. (Eds.) (Editor), *Geological Prior Information: Value and Quantification*. Geol. Soc. of London Special Publication, London.
- Zhang, Y.C., 1999. Modeling market mechanism with evolutionary games. *Europhysics News*, 29:51-53.

Figures

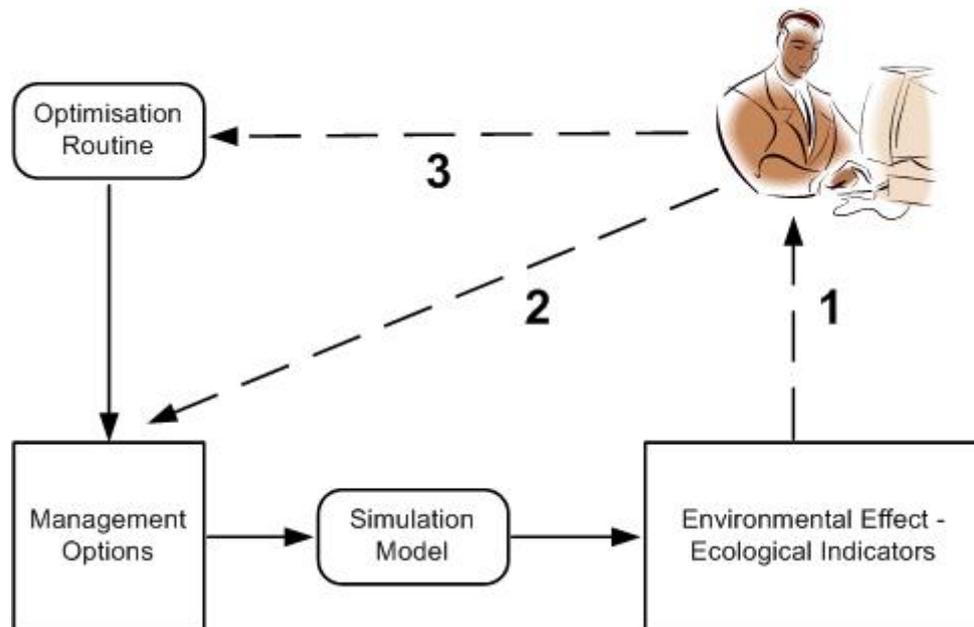


Figure 1 – Diagram of ScenarioLab workflow: ovals represent algorithms, rectangles input and output data, think arrows data transfer and dashed arrows user evaluation. The user controls every iteration through the process: after the Simulation Model has generate the output, the user evaluates the environmental effect (label 1 in the figure) and either chooses what further management option to test (2) or passes a subjective evaluation of the environmental effect to a optimisation routine (3) which then calls a further simulation.

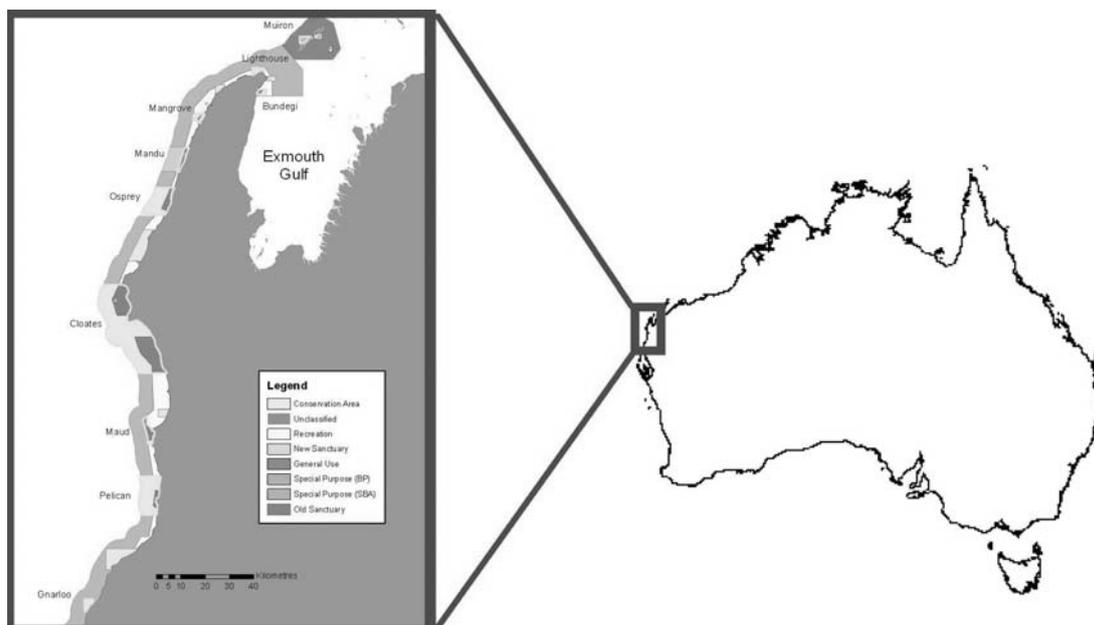


Figure 2 - The studied area and the current management zoning within the Ningaloo Marine Park.

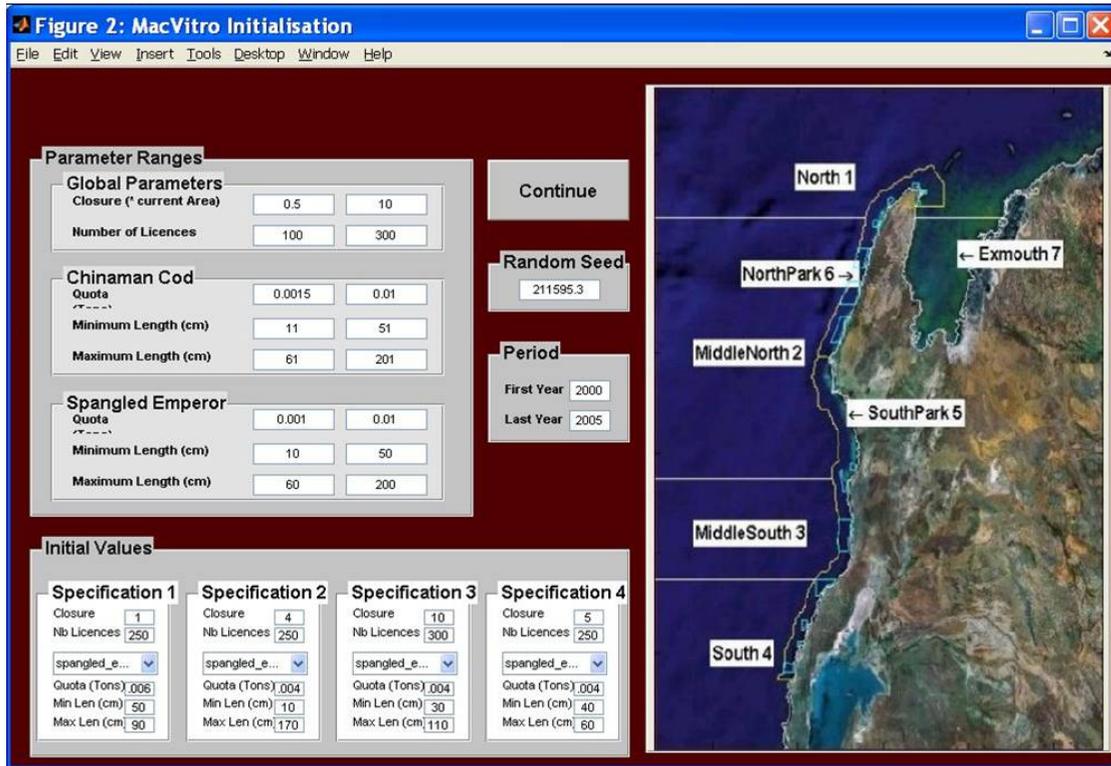


Figure 3 - ScenarioLab initialisation GUI containing the management parameters for the Simulation Model.

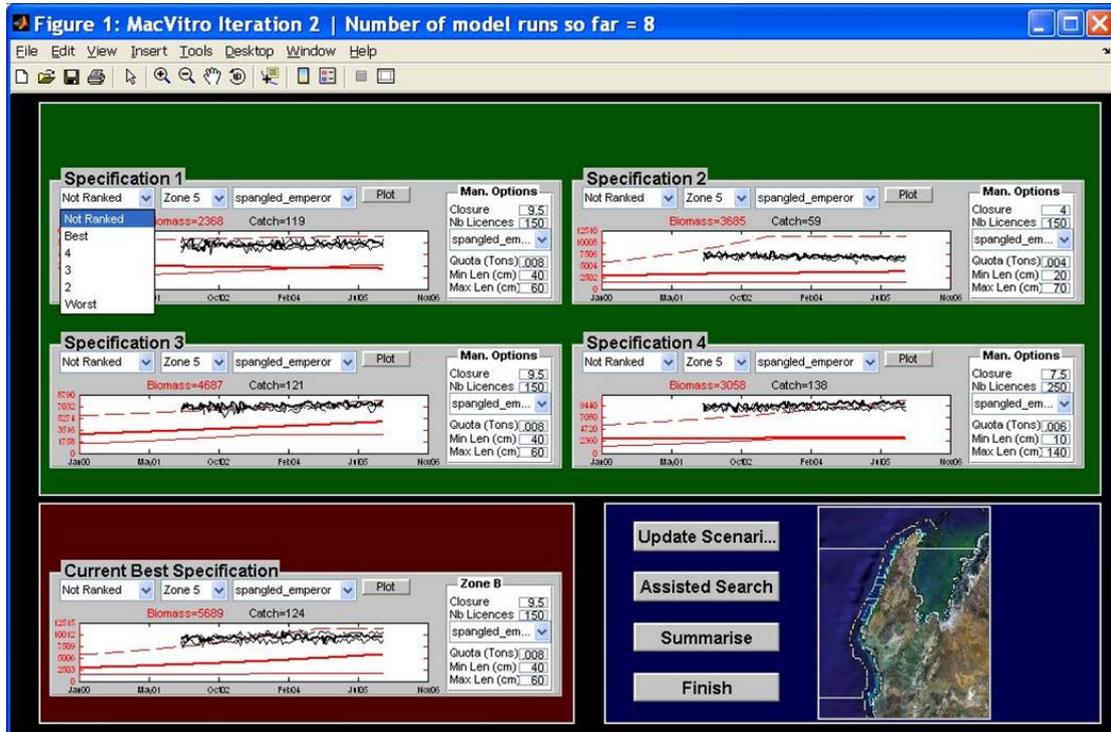


Figure 4. ScenarioLab Evaluation GUI.

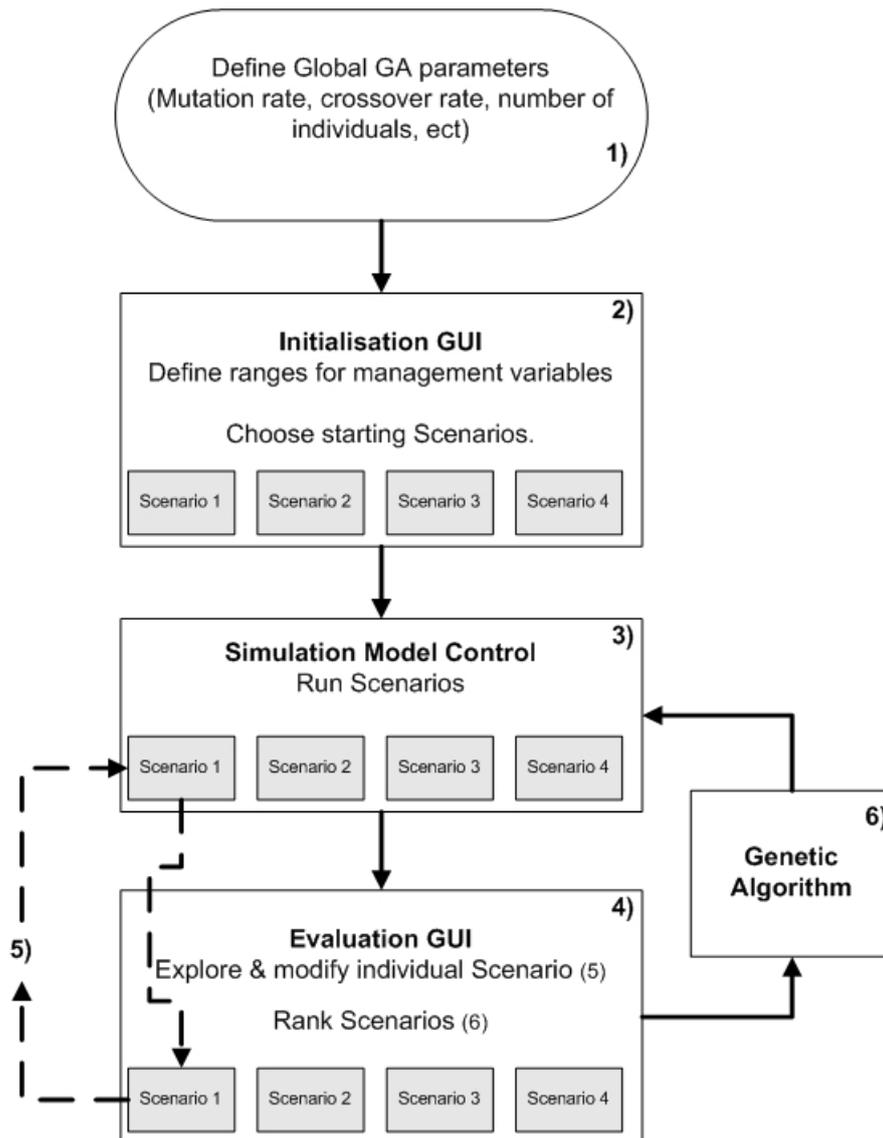


Figure 5. Diagram of the overall ScenarioLab method.

Appendix A – Graphical User Interface and software flowchart

Here we describe in more detail how the GUI communicates with the Genetic algorithm and the Simulation Model. This represents the high-level description of the software flowchart as summarised in Figure 5.

In the first step (block 1 in Figure 5) we need to define the global parameters required by the Genetic Algorithm, which include cross-over rate, mutation rate and number of individuals (which represent the number of scenarios in this work).

The second step consists in launching the Initialisation GUI (block 2 in Figure 5). Here the user can define the allowed ranges for the management parameters in the Genetic Algorithm search, which defines the volume of the input parameter space. Once this is done the initial Genetic Algorithm population is generated randomly. Each individual in the population represents the input parameter for one scenario, that is one Simulation Model run. The user also has the opportunity to modify the random values in any of the scenarios if a specific set(s) of input management parameters needs to be modelled.

Once these values are defined, they are passed to the Simulation Model Control module (block 3). This module runs all scenarios, that is, it applies the Simulation Model to each set of input parameters (four in the Figure) and generates 4 model outputs, one for each scenario.

The output from each scenario is then passed to the Evaluation GUI (block 4), where it is visualised as time series. Here the user has two choices. He/she can modify the input parameters of an individual scenario in order to explore the consequences of a different management strategy. In this case the modified scenario input is passed to the Simulation Model Control module which runs only the modified scenario, whose output is then automatically refreshed in the Evaluation GUI (path 5 in Figure 5, corresponding to arrow 2 in Figure 5).

Alternatively, he/she can decide to rank the scenarios' output as described in Section 5.2 and corresponds to arrow 3 in Figure 5. In this case the ranking is passed to the Genetic Algorithm (block 6), which process the information and generates a new set of four scenarios input. These are passed to the Simulation Model Control module, which runs all Scenarios before sending the new outputs for visualisation to the Evaluation GUI. These two options (modifying a single scenario input or using the Genetic Algorithm) can be iterated until the user is satisfied with the modelling outcome.

Appendix B – Interactive Genetic Algorithm

In an interactive optimisation (Takagi, 2001b) the traditional numerical measure of data mismatch is replaced by the user's subjective evaluation. Humans find it hard to express subjective judgment with absolute values, while they generally find it much easier to compare different instances of the same process and rank them

according to certain criteria. Consequently interactive inversion works by producing different possible solutions and presenting them to the user for judgment and ranking (Takagi, 2001a). The idea was originally developed in the field of Artificial Intelligence in order to support artistic creativity and has since then been applied to several industrial and scientific applications.

Genetic Algorithms (GAs) are a search method suitable for the optimisation of highly non-linear functions. Starting with a set of random solutions, these algorithms progressively modify the solution set by mimicking the evolutionary behaviour of biological systems (selection, cross-over and mutation), until an acceptable result is achieved. GAs are today an established technique, with wide range of applications to both theoretical and industrial problems. We refer the reader to (Davis, 1991) for basic description of genetic algorithms and to (Boschetti et al., 1996), for a more detailed description of the specific GAs implementation used in this work. The mathematically oriented reader can also refer to (Boschetti and Moresi, 2001) for a discussion on the implications of subjective evaluation in both the search space landscape and convergence speed of a GA.

Formally, the modifications to a traditional Genetic Algorithm required to work interactively are minimal. Once a set of initial random potential solutions (management variables, in our case) is generated, this is fed to a forward model (the Simulation Model in our application). Then a set of outputs (time series of catches and fish biomass in our case) is produced. The outputs are visualized and the user ranks them according to his/her subjective judgement. The ranking is input to the next GA iteration which processes it by applying selection, cross-over and mutation to the current solution set in order to produce the next solution set to model.

There are two main reasons why GAs offer themselves naturally to be implemented within an interactive setting. First, GAs work by optimizing an ensemble of solutions, unlike other optimisation algorithms that process one single solution; since we want to present several instances of model outputs to a user in order to compare them and assess them in a relative, rather than absolute fashion, only population-based optimisation algorithms offer this option. Secondly, some non-interactive GA implementations naturally process the ranking of the solutions fitness rather than their exact numerical values. This is done in order to reduce the probability that the algorithm converging rapidly to a local minimum (see (Boschetti et al., 1996) for a technical discussion on this topic). Employing the user ranking of the models' output in the Evaluation GUI thus allows use of a standard GA algorithm without any need to pre-process the subjective evaluation. To our knowledge no other optimisation algorithm provides both these features. Once the rank has been established, a criterion needs to be chosen to process it; in a GA, this reduces to deciding how many copies of a solution should be passed to the next iteration as a function of the rank position. We used what in the GA literature is referred to as Linear Normalization selection, according to which the number of copies is inversely proportional to the rank position. Other criteria could be adopted but given that in an interactive optimisation very few solutions are processed, care should be taken to ensure that copies are not produced only from a few top rank solution(s).

From an implementation point of view, some work needs to be done in order to make the subjective ranking input 'user friendly' and spare the user from tedious

file editing. A user interface needs to be built that allows the user to view at the same time all the different solutions generated by the GA, rank them easily and proceed with the GA operations, possibly within a few mouse clicks. This is important in order to reduce the burden of examining numerous solutions for many generations, that could lead to a lack of attention and accuracy (particularly amongst non-scientific management audiences). These issues are referred to as ‘human fatigue’ in the Interactive Inversion literature (Kishi and Takagi, 1999). To overcome this potential problem, in our implementation we provide the user with the option of ranking only some of the models’ output (the ‘best’ and ‘worst’ output according to subjective evaluation) if so he/she wishes; in this case the unranked outputs are assigned a random rank. Because the most important information of the GA process is provided by the solutions with best and worst fitness, in our experience this options results in a faster analysis and less fatigue on the user without compromising noticeably the results ((Wijns et al., 2003a); also Takagi, personal communication).

Particularly interesting is the effect of the subjective judgement on the GA performance. In traditional optimisation the shape of the search landscape is determined by the choice of cost function, which is fixed, determined a priori and often hard to design. In an interactive inversion the human subjective judgement implicitly shapes the search landscape. Not only this is not fixed a prior (as discussed in the main body of the paper) but also inevitably changes when the user moves the focus of the analysis on different features (catches versus biomass estimation, or between different target species or different combinations of the above). This explains why satisfactory results are often obtained within very few function evaluations (Boschetti and Moresi, 2001; Wijns et al., 2003a). Also, the user may tune the search criteria during the inversion, depending on the knowledge he/she accumulated during the process, providing another layer of ‘interactivity’ to the approach.

Appendix C - The Ecological Module

Here we describe a) the spatial subdivision on the modelled area, b) the trophic relations within the simplified foodweb assigned to each zone and c) the migration between adjacent zones.

The modelled area is subdivided into 10 spatial zones. Each zone is assigned a distinct foodweb. Two types of foodweb are modelled, one for shallow and one for deeper waters. Each zone is assigned one of the two foodwebs according to its bathymetry and migration can occur between adjacent zones with the same foodweb type (see further details below). Within each zone the trophic relations described below are applied to the local foodweb.

Each species undergoes the following processes: spawning and recruitment, migration consumption and growth, natural mortality and losses due to predation and fishing. Consumption, growth, natural mortality and losses due to predation are modelled as continuous processes. Spawning and recruitment are modelled as discrete events. Fishing is modelled as a continuous process during the fishing season.

Each species in our model is divided into age classes ($N_{Classes}$ is set to 10 here) and the biomass ($B_{s,a}$) of each age class (a) of each species (s) is tracked. This is updated

simultaneously in each zone. To simplify the notation, for the rest of the document we have dropped the index for the zone.

C.1 Growth

The consumption-based growth of species s , in age class a , (excluding the basal trophic level) is modelled as:

$$B_{s,a} = B_{s,a} \left(1 + \varepsilon_s \cdot \sum_{p,b} G_{s,a}^{p,b} \right) \quad \text{Eq C-1}$$

where $G_{s,a}^{p,b}$ is the grazing term for species s and age a consuming prey p of age class b , defined by:

$$G_{s,a}^{p,b} = \frac{Q_{s,a}^{p,b} \cdot B_{p,b} \cdot C_{s,a}}{1 + \frac{C_{s,a}}{l_{s,a}} \cdot \varepsilon_s \left(\sum_{p,b} Q_{s,a}^{p,b} \cdot B_{p,b} \right)} \quad \text{Eq C-2}$$

where:

$C_{s,a}$ = maximum clearance rate of predator s of age a

ε_s = assimilation efficiency of predator s

$l_{s,a}$ = maximum growth rate of predator s of age a

$Q_{s,a}^{p,b}$ = availability of prey p of age b to predator s of age a

$B_{s,a}$ = biomass of species s of age a

The growth of the basal trophic level is controlled by a logistic equation rather than a grazing term:

$$B_{s,a} \leftarrow B_{s,a} (1 + G_a^s) \quad \text{Eq C-3}$$

where G_a^s is defined by:

$$G_a^s = l_{s,a} * \left(1 - \frac{\sum B_a}{K} \right) \quad \text{Eq C-4}$$

where:

$l_{s,a}$ = maximum growth rate of basal species s of age a

B_a = biomass of age class a of the basal species

K = Carrying capacity of the zone where the basal species is living

C.2 Migration and movements

Migration is modelled between zones sharing the same foodweb type. This is carried out by defining species biomass density for each zone and setting the migration rates as a function of the density gradient between zones:

$$B_j^{s,a} \leftarrow B_j^{s,a} + \sum_i (M_{i \rightarrow j}^{s,a} - M_{j \rightarrow i}^{s,a}) \quad \text{Eq C-5}$$

where $M_{i \rightarrow j}^{s,a}$ is the migration of a species s and age a from zones i to zone j and is calculated as follows:

$$M_{i \rightarrow j}^{s,a} = D_{s,a} Z_{ij} \left(\frac{B_j^{s,a}}{A_j} - \frac{B_i^{s,a}}{A_i} \right) \quad \text{Eq C-6}$$

where:

$M_{i \rightarrow j}^{s,a}$ = migration from zone i to zone j for species s and age a

$D_{s,a}$ = migration coefficient for species s and age a

Z_{ij} = length of the boundary between zones i and j

A_i = surface area of zone i

$B_j^{s,a}$ = biomass of species s and age a in zone j

C.3 Natural mortality

We model natural mortality due to senescence as:

$$B_{s,a} \leftarrow B_{s,a} - N_{s,a} \quad \text{Eq C-7}$$

where $N_{s,a}$ is the natural mortality term for species s and age a , defined by:

$$N_{s,a} = N_{L,s,a} B_{s,a} + N_{Q,s,a} (D_{s,a})^2 \quad \text{Eq C-8}$$

where:

$N_{L,s,a}$ = linear mortality coefficient of species s of age a

$N_{Q,s,a}$ = quadratic mortality coefficient of species s of age a

$B_{s,a}$ = biomass of species a at maximum age class

$D_{s,a}$ = density of species a at maximum age class

C.4 Loss due to predation

Loss due to predation is modelled via an equation similar to Equation Eq C-1 (with a sign changed):

$$B_{s,a} \leftarrow B_{s,a} \left(1 - \sum_{p,b} G_{s,a}^{p,b} \right) \quad \text{Eq C-9}$$

Notice that here (s,a) stands for the prey and (p,b) stands for the predator. $G_{p,b}^{s,a}$ is defined by:

$$G_{p,b}^{s,a} = \frac{A_{p,b}^{s,a} \cdot B_{s,a} \cdot C_{p,b}}{1 + \frac{C_{p,b}}{l_{p,b}} \cdot \varepsilon_{s,a} \cdot \left(\sum_{s,a} A_{p,b}^{s,a} \cdot B_{s,a} \right)} \quad \text{Eq C-10}$$

where:

$C_{p,b}$ = maximum clearance rate of predator p of age b

$\varepsilon_{s,a}$ = assimilation efficiency of prey s of age a

$l_{p,b}$ = maximum growth rate of predator p of age b

$Q_{p,b}^{s,a}$ = availability of prey s of age a to predator p of age b

$B_{s,a}$ = biomass of prey s of age a

C.5 Spawning and Recruitment

In this instance we use the Beverton-Holt formula for computing the biomass of the new recruits:

$$B_t = \frac{\alpha \cdot B_r}{\beta + B_r} \quad \text{Eq C-11}$$

where:

B_t = biomass of new recruits of the year (of age 1)

α = Beverton-Holt α for the species

β = Beverton-Holt β for the species

B_r = spawning biomass

and is defined as:

$$B_r = \sum_a f_a w_a \cdot B_a \quad \text{Eq C-12}$$

with:

B_a = biomass of fish of age a

w_a = percentage of reserve (gonadal) weight in the total weight of the fish

f_a = part of fish of age a that are matures

Finally for each species, the amount of biomass of age >1 recruits is calculated as follows

$$B_a = B_a + \frac{B_{a-1}}{\text{AgeClass}L_{a-1}} - \frac{B_a}{\text{AgeClass}L_a} \quad \text{for } a \geq 2 \text{ and } a \leq \text{Nclasses}-1 \quad \text{Eq C-13}$$

$$B_{\text{Nclasses}} = B_{\text{Nclasses}} + \frac{B_{\text{Nclasses}-1}}{\text{AgeClass}L_{\text{Nclasses}-1}} \quad \text{For } a = \text{Nclasses} \quad \text{Eq C-14}$$

Appendix D - The Fishing Module

Yield is represented as the biomass caught by the fishing vessels and total removals as the total amount of fish removed from the system (catch plus by-catch).

The biomass loss due to fishing by fleet j (n this case recreational fishers) is modelled as:

$$B_{s,a} \leftarrow B_{s,a} (1 - U_{s,a}^j) \quad \text{Eq D-15}$$

with:

$$U_{s,a}^j = 1 - \exp(-F_j^{s,a} \cdot \Delta t) \quad \text{Eq D-16}$$

where:

$U_{s,a}^j$ = proportion of biomass of species s at age class a killed by fleet j

$F_j^{s,a}$ = fishing mortality of species s age class a due to fishing pressure by fleet j

where:

$$F_j^{s,a} = V_j \cdot q_j^{s,a} \cdot Y_j^V \cdot X_j^{s,a} \quad \text{Eq D-17}$$

V_j = number of fishing vessels in fleet j

$q_j^{s,a}$ = catchability of species s age class a to fleet j

Y_j^l = Size specific gear selectivity for vessel type used by fleet j

$X_j^{s,a}$ = Effect of fleet j on species s age class a (this is the proportion of the catchable biomass that the gear kills, either because it is landed or because it is caught and discarded; it is set to 1 if a target species of legal size, otherwise it is set at a user defined value for bycatch species).

When catch limits or quotas are in place the catch is defined as the minimum of equation A-15 and the allowable catch (i.e. the bag limit or remaining quota).

Appendix E - Economic module

The fleet includes N fishing vessels $n=1..N$, which are modelled as agents. In particular, they are ‘economically rational agents’ who try to maximize the economic return of their action. At each fishing expedition the vessels choose where to fish among Z available zones (a vessel has to choose one and only one zone at each expedition). In each zone a certain amount of S targeted species is available $Fish_{z,s}, z=1..Z, s=1..S$. This amount changes in time and from zone to zone as a function of fishing as well as of trophic relations.

The amount of fish caught in a fishing zone is calculated via Eq A-15. This is shared equally among all the vessels which chose to operate in that specific zone, that is, we assume that all vessels have the same fishing capacity. Also, the fishing capacity of each vessel is constrained by the quota, which depends on the targeted species and is the same for each vessel and each fishing zone. Consequently, the catch of each vessel depends on the action of all other vessels: the larger the number of vessels which access an area, the more likely their individual catch will be small. This results in each vessels trying to avoid crowded fishing zones, that is trying to avoid the zones which most other vessels target. In the game-theoretical literature this setting is known as ‘minority game’ and is able to generate fairly complex dynamics (Savit et al., 1999; Zhang, 1999). For a discussion of the application of the minority game to fishery problems see (Boschetti, 2007; Brede et al., 2008)

The fishers’ decision making model can be summarised as follows:

- 1) Initialisation: assign a random strategy (W_z) to each vessel. W_z represents the expectation of what catch may be obtained at zone z .
- 2) At each iteration t and for each vessel, update the value of W_z based on the catches in the last T steps. This simulates the effort in analysing catch records in order to plan the fishing operation at $t+1$. The value of T can be seen as a measure of the length of the history available in the record, or the ‘memory’ of the vessel crew, or the effort which is employed in the planning. In particular

$$\text{we set } W_z = \sum_{t=1..T} \delta_z^t \text{Catch}_n^t \frac{T-t+1}{T}, \quad (6)$$

where $\delta_z^t = 1$ if the vessel fished in zone z at time t in the past, and $\delta_z^t = 0$ otherwise

- 3) Each vessel randomly chooses which zone to fish next, with $P_z \propto W_z$, where P_z is the probability of choosing zone z . Obviously, $\sum_z P_z = 1$.
- 4) Calculate the number of vessels which aims towards each zone.
- 5) Calculate the catch for each vessel according to Eq A-15.
- 6) Go back to point 2

This process can be seen as within an optimisation framework in which each vessels tries to optimise its own catch. It allows for a balance between *exploration* and *exploitation*. Exploitation is represented by W_z : the more a vessel expects a specific zone to provide a good catch, the more likely the zone will be chosen. Exploration is represented by the random pick which allows non optimal zones to be chosen with lower probability.

Figures captions

Figure 5 – Diagram of ScenarioLab workflow: ovals represent algorithms, rectangles input and output data, thick arrows data transfer and dashed arrows user evaluation. The user controls every iteration through the process: after the Simulation Model has generated the output, the user evaluates the environmental effect (label 1 in the figure) and either chooses what further management option to test (2) or passes a subjective evaluation of the environmental effect to an optimisation routine (3) which then calls a further simulation.

Figure 6 - The studied area and the current management zoning within the Ningaloo Marine Park.

Figure 7 - ScenarioLab initialisation GUI containing the management parameters for the Simulation Model.

Figure 8. ScenarioLab Evaluation GUI.

Figure 5. Software flowchart; relation between the Graphical User Interfaces, the Simulation Model and the Genetic Algorithm.