Management frequency and extinction risk

GMSE: an R package for generalised management strategy evaluation (Supporting Information 6)

A. Bradley Duthie[1][3], Jeremy J. Cusack[1], Isabel L. Jones[1], Jeroen Minderman[1], Erlend B. Nilsen[2], Rocío A. Pozo[1], O. Sarobidy Rakotonarivo[1], Bram Van Moorter[2], and Nils Bunnefeld[1]

[1] Biological and Environmental Sciences, University of Stirling, Stirling, UK [2] Norwegian Institute for Nature Research, Trondheim, Norway [3] alexander.duthie@stir.ac.uk

The individual-based approach of default GMSE sub-models

The default sub-models of GMSE (`resource`, `observation`, `manager`, `user`) are individual-based (also called 'agent-based'), meaning that they model discrete individuals (resources or agents), which in GMSE are represented by individual table rows (as in `RESOURCES`, `AGENTS`, and `OBSERVATION`) or layers of three-dimensional arrays (as in `COST` and `ACTION`). Individual-based models (IBMs) have been a useful approach in ecology for decades (Uchmański and Grimm, 1996; Grimm, 1999), providing both a pragmatic tool for the mechanistic modelling of complex populations and a powerful technique for theoretical investigation. A key advantage of the individual-based modelling approach is the discrete nature of individuals, which allows for detailed trait variation and complex interactions among individuals. In GMSE, some of the most important traits for resources include types, ages, demographic parameter values, locations, etc., and for agents (manager and users), traits include different types, utilities, budgets, etc. The traits that resources and managers have can potentially affect their interactions, and default GMSE sub-models take advantage of this by simulating interactions explicitly on a landscape (see SI7 for an introduction to GMSE default data structures).

Replicate simulations as a tool for model inference

Mechanistically modelling complex interactions among discrete individuals typically causes some degree of stochasticity in IBMs (in the code, this is caused by the sampling of random values, which determine probabilistically whether or not events such as birth or death occur for individuals), reflecting the uncertainty that is inherent to complex systems. We can see a simple example of this by calling `gmse_apply` under the same default conditions twice.

```r
rand_eg_1 <- gmse_apply();
print(rand_eg_1);
```

```
## $resource_results
## [1] 1118
##
## $observation_results
## [1] 1451.247
##
## $manager_results
##           resource_type scaring culling castration feeding help_offspring
## policy_1             1      NA      45         NA      NA             NA
##
## $user_results
##          resource_type scaring culling castration feeding help_offspring
## Manager             1      NA       0         NA      NA             NA
## user_1              1      NA      22         NA      NA             NA
## user_2              1      NA      22         NA      NA             NA
## user_3              1      NA      22         NA      NA             NA
## user_4              1      NA      22         NA      NA             NA
##         tend_crops kill_crops
## Manager         NA         NA
## user_1          NA         NA
## user_2          NA         NA
## user_3          NA         NA
## user_4          NA         NA
```

Although a second call of `gmse_apply` has identical initial conditions, because resource demographics (e.g., birth and death) and agent decision making (e.g., policy generation and user actions) is not deterministic, a slightly different result is obtained below.

```
rand_eg_2 <- gmse_apply();
print(rand_eg_2);
```

```
## $resource_results
## [1] 1090
##
## $observation_results
## [1] 839.0023
##
## $manager_results
##          resource_type scaring culling castration feeding help_offspring
## policy_1             1      NA      66         NA      NA             NA
##
## $user_results
##         resource_type scaring culling castration feeding help_offspring
## Manager             1      NA       0         NA      NA             NA
## user_1              1      NA      15         NA      NA             NA
## user_2              1      NA      15         NA      NA             NA
## user_3              1      NA      15         NA      NA             NA
## user_4              1      NA      15         NA      NA             NA
##         tend_crops kill_crops
## Manager         NA         NA
## user_1          NA         NA
## user_2          NA         NA
## user_3          NA         NA
## user_4          NA         NA
```

To make meaningful model inferences, it is often necessary to replicate simulations under the same initial conditions to understand the range of predicted outcomes for a particular set of parameter values. This can be computationally intense, but it can also lead to a more robust understanding of the range of dynamics that might be expected within a system. Additionally, when parameter values are unknown but believed to be important, replicate simulations can be applied across a range of values to understand how a particular parameter might affect system dynamics. Below, we show how to use the `gmse_replicates` function to simulate a simple example of a managed population that is hunted by users. This function calls `gmse` multiple times and aggregates the results from replicate simulations into a single table.

For a single simulation, the `gmse_table` function prints out key information from a `gmse` simulation result. The example provided in the GMSE documentation is below.

```
gmse_sim  <- gmse(time_max = 10, plotting = FALSE);
```

```
## [1] "Initialising simulations ... "
```

```
sim_table <- gmse_table(gmse_sim = gmse_sim);
print(sim_table)
```

```
##       time_step resources  estimate cost_culling cost_unused act_culling
##  [1,]         1      1106 1088.4354           57          53          68
##  [2,]         2      1147 1179.1383           20          90         200
##  [3,]         3      1051 1043.0839           83          27          48
##  [4,]         4      1157  997.7324          110           0          36
##  [5,]         5      1302 1111.1111           33          77         120
##  [6,]         6      1572 1473.9229           10         100         400
##  [7,]         7      1397 1451.2472           10         100         400
##  [8,]         8      1237 1383.2200           10         100         400
##  [9,]         9      1006 1111.1111           32          78         124
## [10,]        10      1062  634.9206          110           0          36
##       act_unused harvested
##  [1,]          6        68
##  [2,]          0       200
##  [3,]          0        48
##  [4,]          2        36
##  [5,]          2       120
##  [6,]          0       400
##  [7,]          0       400
##  [8,]          0       400
##  [9,]          0       124
## [10,]          3        36
```

The above table can be saved as a CSV file using the `write.csv` function.

```
write.csv(x= sim_table, file = "file_path/gmse_table_name.csv");
```

Instead of recording all time steps in the simulation, we can instead record only the last time step in `gmse_table` using the `all_time` argument.

```
sim_table_last <- gmse_table(gmse_sim = gmse_sim, all_time = FALSE);
print(sim_table_last)
```

```
##    time_step    resources     estimate cost_culling  cost_unused
##      10.0000    1062.0000     634.9206     110.0000       0.0000
##  act_culling   act_unused    harvested
##      36.0000       3.0000      36.0000
```

The `gmse_replicates` function replicates multiple simulations `replicates` times under the same initial conditions, then returns a table showing the values of all simulations. This can be useful, for example, for testing how frequently a population is expected to go to extinction or carrying capacity under a given set of parameter values. First, we demonstrate the `gmse_replicates` function for simulations of up to 20 time steps. The `gmse_replicates` function accepts all arguments used in `gmse`, and also all arguments of `gmse_table` (`all_time` and `hide_unused_options`) to summarise multiple `gmse` results. Here we use default `gmse` values in replicate simulations, except `plotting`, which we set to `FALSE` to avoid plotting each simulation result. We run 10 replicates below.

```
gmse_reps1 <- gmse_replicates(replicates = 10, time_max = 20, plotting = FALSE);
print(gmse_reps1);
```

```
##       time_step resources  estimate cost_culling cost_unused act_culling
##  [1,]        20      1436 1473.9229           10         100         400
##  [2,]        20      1273 1269.8413           13          97         304
##  [3,]        20      1243 1269.8413           13          97         304
##  [4,]        20      1191 1088.4354           41          69          96
##  [5,]        20       849 1020.4082          109           0          36
##  [6,]        20      1616 1904.7619           10         100         400
##  [7,]        20      1039  997.7324          110           0          36
##  [8,]        20       968  907.0295          110           0          36
##  [9,]        20      1403 1587.3016           10         100         400
## [10,]        20       983  884.3537          110           0          36
##       act_unused harvested
##  [1,]          0       400
##  [2,]          2       304
##  [3,]          3       304
##  [4,]          2        96
##  [5,]          1        36
##  [6,]          0       400
##  [7,]          1        36
##  [8,]          1        36
##  [9,]          0       400
## [10,]          2        36
```

Note from the results above that resources in all simulations persisted for 20 time steps, which means that extinction never occurred. We can also see that the population in all simulations never terminated at a density near the default carrying capacity of `res_death_K = 2000`, and was instead consistently near the target population size of `manage_target = 1000`. If we wish to define management success as having a population density near target levels after 20 time steps (perhaps interpreted as 20 years), then we might assess this population as successfully managed under the conditions of the simulation. We can then see what happens if managers only respond to changes in the social-ecological system with a change in policy once every two years, perhaps as a consequence of reduced funding for management or increasing demands for management attention elsewhere. This can be done by changing the default `manage_freq = 1` to `manage_freq = 2`.

```
gmse_reps2 <- gmse_replicates(replicates  = 10, time_max = 20, plotting = FALSE,
                              manage_freq = 2);
print(gmse_reps2);
```

```
##       time_step resources  estimate cost_culling cost_unused act_culling
##  [1,]        20      1100 1065.7596           55          55          72
##  [2,]        20       625  476.1905          110           0          36
##  [3,]        20       864  952.3810          110           0          36
##  [4,]        20      1034  929.7052          108           2          36
##  [5,]        20      1370 1519.2744           10         100         400
##  [6,]        20      1255 1451.2472           10         100         400
##  [7,]        20      1036 1315.1927           12          98         332
##  [8,]        20       932  997.7324          110           0          36
##  [9,]        20      1439  997.7324          110           0          36
## [10,]        20      1118 1043.0839           84          26          44
##       act_unused harvested
##  [1,]          2        72
##  [2,]          2        36
##  [3,]          2        36
##  [4,]          5        36
##  [5,]          0       400
##  [6,]          0       400
##  [7,]          0       332
##  [8,]          1        36
##  [9,]          2        36
## [10,]          7        44
```

Note that while extinction still does not occur in these simulations, when populations are managed less frequently, they tend to be less close to the target size of 1000 after 20 generations. The median population size of `gmse_reps1` (management in every time step) was 1217, with a maximum of 1616 and minimum of 849. The median population size of the newly simulated `gmse_reps2` (management every two time steps) is 1068, with a maximum of 1439 and minimum of 625. We can now see what happens when management occurs only once in every three time steps.

```
gmse_reps3 <- gmse_replicates(replicates = 10, time_max = 20, plotting = FALSE,
                              manage_freq = 3);
print(gmse_reps3);
```

```
##       time_step resources  estimate cost_culling cost_unused act_culling
##  [1,]        20       800  702.9478          110           0          36
##  [2,]        20      1309  839.0023          110           0          36
##  [3,]        20       991  884.3537          109           1          36
##  [4,]        20       669  498.8662          110           0          36
##  [5,]        20      1074 1179.1383           20          90         200
##  [6,]        20      1221 1315.1927           12          98         332
##  [7,]        20      1061  748.2993          110           0          36
##  [8,]        20      1183  975.0567          110           0          36
##  [9,]        20      1197  725.6236          108           2          36
## [10,]        20       520 1360.5442           10         100         400
##       act_unused harvested
##  [1,]          1        36
##  [2,]          2        36
##  [3,]          0        36
##  [4,]          0        36
##  [5,]          0       200
##  [6,]          0       332
##  [7,]          2        36
##  [8,]          1        36
##  [9,]          2        36
## [10,]          0       400
```

Given a management frequency of once every three time steps, the median population size of `gmse_reps3` (management in every time step) is 1067.5, with a maximum of 1309 and minimum of 520. The number of extinctions observed in these replicate populations was 0. Below we change the management frequency to once every four time steps.

```
gmse_reps4 <- gmse_replicates(replicates = 10, time_max = 20, plotting = FALSE,
                              manage_freq = 4);
print(gmse_reps4);
```

```
##       time_step resources   estimate cost_culling cost_unused act_culling
##  [1,]        20      2027 1904.76190           10         100         400
##  [2,]        20       203  204.08163          110           0          36
##  [3,]        20      1904 1791.38322           10         100         400
##  [4,]        13         3   22.67574          110           0          36
##  [5,]        11         0   68.02721          110           0          36
##  [6,]        15         0 1405.89569           10         100         400
##  [7,]        20      1948 1950.11338           10         100         400
##  [8,]        20       834  975.05669          110           0          36
##  [9,]        20         29    0.00000          110           0          36
## [10,]        20       1833 1473.92290           10         100         400
##       act_unused harvested
##  [1,]          0       400
##  [2,]          0        36
##  [3,]          0       400
##  [4,]          2         3
##  [5,]          2         0
##  [6,]          0         0
##  [7,]          0       400
##  [8,]          1        36
##  [9,]          1        29
## [10,]          0       400
```

Now note from the first column of `gmse_reps4` above that 3 populations did not persist to the 20th time step; i.e., 3 populations went to extinction (note that GMSE has a minimum resource population size of 5). This has occured because managers cannot respond quickly enough to changes in the population density, and therefore cannot increase the cost of culling to maintain target resource levels if population size starts to decrease. We can see the extinction risk increase even further if management only occurs once every 5 time steps.

```
gmse_reps5 <- gmse_replicates(replicates = 10, time_max = 20, plotting = FALSE,
                             manage_freq = 5);
print(gmse_reps5);
```

```
##       time_step resources estimate cost_culling cost_unused act_culling
##  [1,]         5         0        0          110           0          36
##  [2,]         5         0        0          110           0          36
##  [3,]         5         0        0          110           0          36
##  [4,]         5         0        0          110           0          36
##  [5,]         5         0        0          110           0          36
##  [6,]         5         0        0          110           0          36
##  [7,]         5         0        0          110           0          36
##  [8,]         5         0        0          110           0          36
##  [9,]         5         0        0          110           0          36
## [10,]         5         0        0          109           1          36
##       act_unused harvested
##  [1,]          2         0
##  [2,]          1         0
##  [3,]          1         0
##  [4,]          2         0
##  [5,]          1         0
##  [6,]          1         0
##  [7,]          0         0
##  [8,]          0         0
##  [9,]          2         0
## [10,]          1         0
```

When a manager can only make policy decisions once every five time steps, extinction occurs in 10 out of 10 simulated populations before year 20. If we wanted to summarise these results, we could plot how extinction risk changes with increasing `manage_freq`.

```
ext_risk1 <- sum(gmse_reps1[,2] < 20);
ext_risk2 <- sum(gmse_reps2[,2] < 20);
ext_risk3 <- sum(gmse_reps3[,2] < 20);
ext_risk4 <- sum(gmse_reps4[,2] < 20);
```

```r
ext_risk5 <- sum(gmse_reps5[,2] < 20);
y_var     <- c(ext_risk1, ext_risk2, ext_risk3, ext_risk4, ext_risk5);
x_var     <- 1:5;
plot(x = x_var, y = y_var, type = "b", pch = 20, lwd = 2, cex = 1.5,
     xlab = "Management every N time steps (manage_freq)",
     ylab = "Freq. of population extinction", cex.lab = 1.25)
```

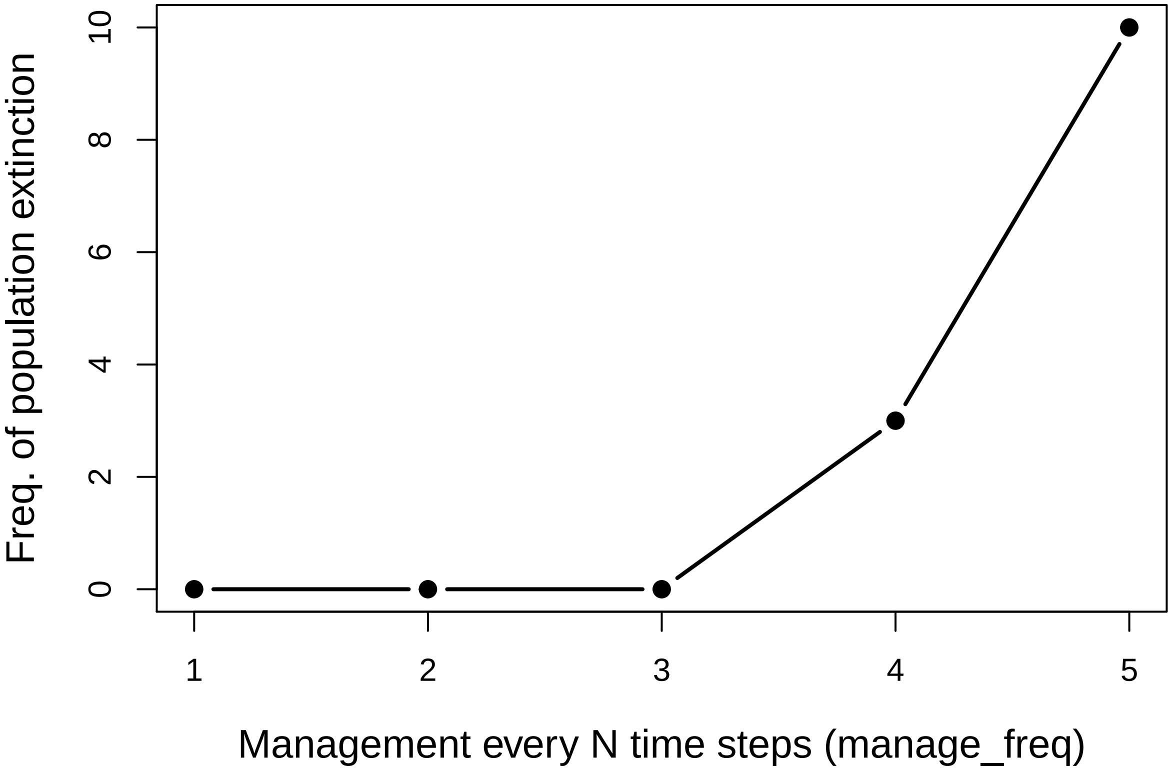

Figure 1: Extinction risk given an increasing number of time steps between updating policy decisions for culling costs in a simulated population. Higher values on the x-axis correspond to more time passing before a new policy is set. For each point, a total of 10 replicate simulations were run.

The above plot and the simulations from which it was derived illustrates a greatly simplified example of how GMSE might be used to assess the risk of extinction in a managed population. A comprehensive analysis would need more than 10 replicate simulations to accurately infer extinction risk, and would require careful pararmeterisation of all sub-models and a sensitivity analysis where such parameters are unknown. A benefit of this approach is that it allows for the simulation of multiple different scenarios under conditions of uncertainty and stochasticity, modelling the range of outcomes that might occur within and among scenarios and facilitating the development of social-ecological theory. Future expansion on the complexity of individual-based default sub-models of GMSE will further increase the realism of targeted case studies.

References

Grimm, V. (1999). Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological Modelling*, 115(2-3):129–148.

Uchmański, J. and Grimm, V. (1996). Individual-based modelling in ecology: what makes the difference? *Trends in Ecology & Evolution*, 11(10):437–441.