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.

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

```
## $resource_results
## [1] 1117
##
## $observation_results
## [1] 1179.138
##
## $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
```

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] 1115
##
## $observation_results
## [1] 1111.111
##
## $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.

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

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

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

```
##       time_step resources  estimate cost_culling cost_unused act_culling
##  [1,]         1      1102 1496.5986           64          46          60
##  [2,]         2      1171 1111.1111           22          88         180
##  [3,]         3      1155  997.7324          110           0          36
##  [4,]         4      1302 1428.5714           10         100         312
##  [5,]         5      1166 1088.4354           22          88         180
##  [6,]         6      1289 1519.2744           10         100         398
##  [7,]         7      1051 1156.4626           13          97         304
##  [8,]         8       885  861.6780          107           3          36
##  [9,]         9      1011  748.2993          110           0          36
## [10,]        10      1187 1383.2200           10         100         309
##       act_unused harvested
##  [1,]         10        60
##  [2,]          3       180
##  [3,]          1        36
##  [4,]         85       312
##  [5,]          2       180
##  [6,]          2       398
##  [7,]          2       304
##  [8,]          6        36
##  [9,]          2        36
## [10,]         91       309
```

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

```r
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.

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

```
##    time_step    resources     estimate cost_culling  cost_unused
##        10.00      1187.00      1383.22        10.00       100.00
##   act_culling   act_unused    harvested
##       309.00        91.00       309.00
```

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.

```r
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      1100  952.3810          110           0          36
## [2,]        20      1028 1065.7596           45          65          88
## [3,]        20      1225  861.6780          110           0          36
## [4,]        20      1432 1315.1927           10         100         400
## [5,]        20      1066 1315.1927           10         100         400
## [6,]        20      1318 1315.1927           10         100         297
## [7,]        20      1282 1337.8685           10         100         303
## [8,]        20      1401 1428.5714           10         100         310
## [9,]        20       937  929.7052          110           0          36
## [10,]       20      1444 1904.7619           10         100         400
##       act_unused harvested
## [1,]           3        36
## [2,]           1        88
## [3,]           1        36
## [4,]           0       400
## [5,]           0       400
## [6,]         101       297
## [7,]          96       303
## [8,]          89       310
## [9,]           4        36
## [10,]          0       400
```

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      1542 1836.7347           10         100         311
## [2,]        20      1343 1111.1111           23          87         172
## [3,]        20       842 1020.4082          110           0          36
## [4,]        20       667  748.2993          110           0          36
## [5,]        20       322  362.8118          110           0          36
## [6,]        20      2204 2403.6281           10         100         298
## [7,]        20      1090 1224.4898           10         100         400
## [8,]        20       637  997.7324          109           1          36
## [9,]        20       547  453.5147          110           0          36
## [10,]       20      1033 1020.4082          110           0          36
##       act_unused harvested
## [1,]          89       311
## [2,]           4       172
## [3,]           0        36
## [4,]           3        36
## [5,]           1        36
## [6,]         102       298
## [7,]           0       400
```

```
##  [8,]          1        36
##  [9,]          4        36
## [10,]          3        36
```

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 1253.5, with a maximum of 1444 and minimum of 937. The median population size of the newly simulated `gmse_reps2` (management every two time steps) is 937.5, with a maximum of 2204 and minimum of 322. 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       912 1179.1383           10         100         400
##  [2,]        20      1276  861.6780          110           0          36
##  [3,]        20       685  612.2449          110           0          36
##  [4,]        20      1503 1451.2472           10         100         400
##  [5,]         9         0    0.0000          108           2          36
##  [6,]        20      1340 1609.9773           10         100         400
##  [7,]        20       612 1269.8413           10         100         400
##  [8,]        20       564 1133.7868           17          93         232
##  [9,]        20      1435 1882.0862           10         100         400
## [10,]        20      1165 1360.5442           10         100         400
##       act_unused harvested
##  [1,]          0       400
##  [2,]          3        36
##  [3,]          3        36
##  [4,]          0       400
##  [5,]          4         0
##  [6,]          0       400
##  [7,]          0       400
##  [8,]          2       232
##  [9,]          0       400
## [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 1038.5, with a maximum of 1503 and minimum of 0. The number of extinctions observed in these replicate populations was 1. 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         0   0.00000          110           0          36
##  [2,]        20       711 702.94785          110           0          36
##  [3,]        20       179 226.75737          110           0          36
##  [4,]        20        44  22.67574          110           0          36
##  [5,]        20       205 181.40590          110           0          36
##  [6,]        20       222 340.13605          110           0          36
##  [7,]        13         0  22.67574          110           0          36
##  [8,]        20       650 634.92063          110           0          36
```

```
##  [9,]         9        0   0.00000          109            1           36
## [10,]        20      212 317.46032          110            0           36
##       act_unused harvested
##  [1,]          3         0
##  [2,]          2        36
##  [3,]          3        36
##  [4,]          3        36
##  [5,]          3        36
##  [6,]          3        36
##  [7,]          2         0
##  [8,]          3        36
##  [9,]          4         0
## [10,]          3        36
```

Now note from the first column of `gmse_reps4` above that 2 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          109           1          36
##  [2,]         5         0        0          110           0          36
##  [3,]         5         0        0          110           0          36
##  [4,]         5         0        0          109           1          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          110           0          36
##       act_unused harvested
##  [1,]          3         0
##  [2,]          0         0
##  [3,]          3         0
##  [4,]          3         0
##  [5,]          1         0
##  [6,]          3         0
##  [7,]          1         0
##  [8,]          2         0
##  [9,]          2         0
## [10,]          2         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);
```

```
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)
```

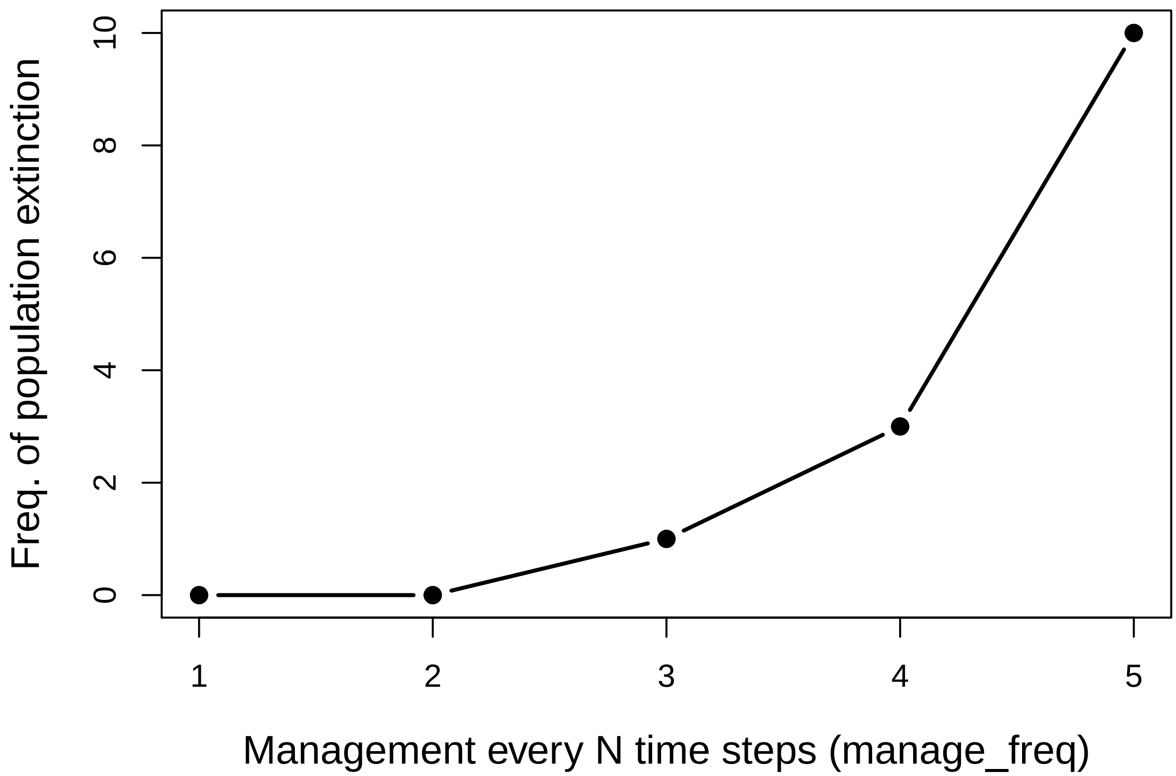

Figure 1: Extinction risk given an increasing number of time steps between updating policy decisions for culling costs in a simulated population. Higher values on the x-axis correspond to more time passing before a new policy is set. For each point, a total of 10 replicate simulations were run.

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.