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 submodels

The default submodels 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 submodels 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] 1093
##
## $observation_results
## [1] 816.3265
##
## $manager_results
##          resource_type scaring culling castration feeding help_offspring
## policy_1             1      NA      74         NA      NA             NA
##
## $user_results
##         resource_type scaring culling castration feeding help_offspring
```

```
## Manager             1      NA       0          NA      NA             NA
## user_1              1      NA      13          NA      NA             NA
## user_2              1      NA      13          NA      NA             NA
## user_3              1      NA      13          NA      NA             NA
## user_4              1      NA      13          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] 1166
##
## $observation_results
## [1] 1360.544
##
## $manager_results
##          resource_type scaring culling castration feeding help_offspring
## policy_1             1      NA      64         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      1075 1201.8141           60          50          64
##  [2,]         2      1071  907.0295          110           0          36
##  [3,]         3      1123 1156.4626           13          97         281
##  [4,]         4       907  861.6780          109           1          36
##  [5,]         5       981 1133.7868           17          93         232
##  [6,]         6       933  884.3537          110           0          36
##  [7,]         7      1019 1269.8413           10         100         301
##  [8,]         8       816  770.9751          109           1          36
##  [9,]         9       914  884.3537          110           0          36
## [10,]        10      1001  816.3265          110           0          36
##       act_unused harvested
##  [1,]          7        64
##  [2,]          3        36
##  [3,]         32       281
##  [4,]          3        36
##  [5,]          2       232
##  [6,]          2        36
##  [7,]         95       301
##  [8,]          1        36
##  [9,]          4        36
## [10,]          4        36
```

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");
```

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
##      10.0000    1001.0000     816.3265     110.0000       0.0000
##  act_culling   act_unused    harvested
##      36.0000       4.0000      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.

```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       1089 1020.4082           110           0           36
##   [2,]        20        793  725.6236           110           0           36
##   [3,]        20        701  748.2993           110           0           36
##   [4,]        20       1183  748.2993           108           2           36
##   [5,]        20       1583 1836.7347            10         100          400
##   [6,]        20        805  952.3810           110           0           36
##   [7,]        20       1168 1156.4626            13          97          304
##   [8,]        20       1149 1655.3288            10         100          400
##   [9,]        20        968 1179.1383            10         100          400
##  [10,]        20       1125 1247.1655            10         100          400
##        act_unused harvested
##   [1,]          4        36
##   [2,]          3        36
##   [3,]          2        36
##   [4,]          6        36
##   [5,]          0       400
##   [6,]          3        36
##   [7,]          3       304
##   [8,]          0       400
##   [9,]          0       400
##  [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        939  816.3265           108           2           36
##   [2,]        20        764  702.9478           110           0           36
##   [3,]        20        697  884.3537           109           1           36
##   [4,]        20       1247 1541.9501            10         100          302
##   [5,]        20       1131 1201.8141            10         100          286
##   [6,]        20        994  816.3265           110           0           36
##   [7,]        20       1493 1519.2744            10         100          298
##   [8,]        20        959 1224.4898            10         100          302
##   [9,]        20        485  408.1633           109           1           36
##  [10,]        20       1500 1473.9229            10         100          292
##        act_unused harvested
##   [1,]          4        36
##   [2,]          2        36
##   [3,]          4        36
##   [4,]         98       302
##   [5,]        114       286
##   [6,]          0        36
##   [7,]        101       298
```

```
##   [8,]         97       302
##   [9,]          3        36
## [10,]        104       292
```

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 1107, with a maximum of 1583 and minimum of 701. The median population size of the newly simulated `gmse_reps2` (management every two time steps) is 976.5, with a maximum of 1500 and minimum of 485. 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,]        10         0    0.0000          110           0          36
##   [2,]        20      1122 1451.2472           10         100         400
##   [3,]        20       344  385.4875          110           0          36
##   [4,]        20       630  476.1905          110           0          36
##   [5,]        20       880 1383.2200           10         100         400
##   [6,]        20      1395 2063.4921           10         100         400
##   [7,]        20       923 1768.7075           10         100         400
##   [8,]        20      1250 1043.0839           73          37          52
##   [9,]        20       964  884.3537          110           0          36
## [10,]        20      1003  816.3265          110           0          36
##       act_unused harvested
##   [1,]          2         0
##   [2,]          0       400
##   [3,]          2        36
##   [4,]          2        36
##   [5,]          0       400
##   [6,]          0       400
##   [7,]          0       400
##   [8,]          7        52
##   [9,]          1        36
## [10,]          4        36
```

Given a management frequency of once every three time steps, the median population size of `gmse_reps3` (management in every time step) is 943.5, with a maximum of 1395 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      1491 1179.13832           10         100         295
##   [2,]        20      2181 2018.14059           10         100         300
##   [3,]        20       108  204.08163          110           0          36
##   [4,]        20       212   45.35147          110           0          36
##   [5,]         9         0   90.70295          109           1          36
##   [6,]        20       437  680.27211          110           0          36
##   [7,]        10         0   68.02721          110           0          36
##   [8,]        20       155  317.46032          110           0          36
```

```
##  [9,]        14         0   68.02721          110            0            36
## [10,]         8         3    0.00000          110            0            36
##       act_unused harvested
##  [1,]        105       295
##  [2,]         98       300
##  [3,]          4        36
##  [4,]          3        36
##  [5,]          3         0
##  [6,]          4        36
##  [7,]          2         0
##  [8,]          4        36
##  [9,]          3         0
## [10,]          2         3
```

Now note from the first column of `gmse_reps4` above that 4 populations did not persist to the 20th time step; i.e., 4 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.

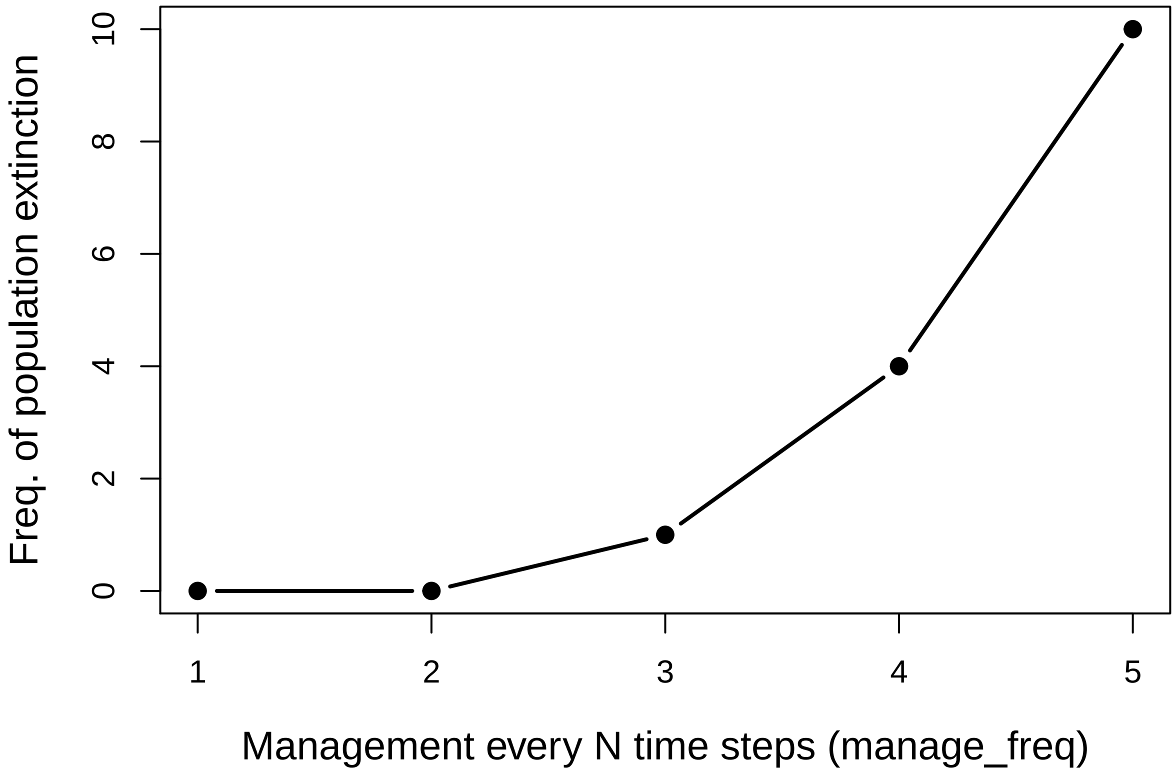
```
gmse_reps5 <- gmse_replicates(replicates = 10, time_max = 20, plotting = FALSE,
                              manage_freq = 5);
print(gmse_reps5);
```

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

When 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)
```



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 submodels 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.