Management frequency and extinction risk

GMSE: an R package for generalised management strategy evaluation (Supporting Information 5)

A. Bradley Duthie[13], 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 (`RESOURCES`, `AGENTS`, `OBSERVATION`) or layers of three-dimensional arrays (`COST`, `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

Modelling complex interactions among discrete individuals mechanistically 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] 1067
##
## $observation_results
## [1] 929.7052
##
## $manager_results
##          resource_type scaring culling castration feeding help_offspring
## policy_1             1      NA      56         NA      NA             NA
##
## $user_results
##         resource_type scaring culling castration feeding help_offspring
## Manager             1      NA       0         NA      NA             NA
```

```
## user_1              1      NA      17         NA      NA             NA
## user_2              1      NA      17         NA      NA             NA
## user_3              1      NA      17         NA      NA             NA
## user_4              1      NA      17         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 the second call to `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.

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

```
## $resource_results
## [1] 1098
##
## $observation_results
## [1] 1678.005
##
## $manager_results
##          resource_type scaring culling castration feeding help_offspring
## policy_1             1      NA      65         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 in 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 ... "
```

```
## [1] "Generation  7 of  10"
```

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

```
##       time_step resources  estimate cost_culling cost_unused act_culling
##  [1,]         1      1103 1360.5442           59          51          64
##  [2,]         2      1187 1179.1383           10         100         333
##  [3,]         3       971  725.6236          110           0          36
##  [4,]         4      1134 1111.1111           23          87         172
##  [5,]         5      1147 1020.4082          109           1          36
##  [6,]         6      1464 1564.6259           10         100         307
##  [7,]         7      1385 1360.5442           10         100         400
##  [8,]         8      1201 1632.6531           10         100         400
##  [9,]         9       977  793.6508          110           0          36
## [10,]        10      1165 1065.7596           45          65          88
##       act_unused harvested
##  [1,]          8        64
##  [2,]         66       333
##  [3,]          2        36
##  [4,]          4       172
##  [5,]          2        36
##  [6,]         89       307
##  [7,]          0       400
##  [8,]          0       400
##  [9,]          2        36
## [10,]          1        88
```

We can also only record the last time step in `gmse_table`.

```
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      1165.00      1065.76        45.00        65.00
##  act_culling   act_unused    harvested
##        88.00         1.00        88.00
```

The `gmse_replicates` function replicates multiple simulations under the same initial conditions, then returns a table showing the values of all simulations. This can be useful, for example, for testing how often 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 20 time steps; other parameter values are set to default values, except `plotting`, which we set to `FALSE` to avoid plotting each simulation result. We run 10 replicates below.

```
##       time_step resources  estimate cost_culling cost_unused act_culling
##  [1,]        20      1281 1247.1655           10         100         308
##  [2,]        20      1232 1519.2744           10         100         400
##  [3,]        20      1180 1088.4354           31          79         128
##  [4,]        20       846  997.7324          110           0          36
##  [5,]        20      1202 1383.2200           10         100         400
##  [6,]        20       924 1065.7596           31          79         128
##  [7,]        20       637  544.2177          109           1          36
##  [8,]        20      1094  839.0023          110           0          36
##  [9,]        20       716  748.2993          110           0          36
## [10,]        20       883  929.7052          110           0          36
##       act_unused harvested
```

```
##  [1,]         90       308
##  [2,]          0       400
##  [3,]          0       128
##  [4,]          3        36
##  [5,]          0       400
##  [6,]          0       128
##  [7,]          3        36
##  [8,]          1        36
##  [9,]          0        36
## [10,]          2        36
```

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

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`.

```
##       time_step resources  estimate cost_culling cost_unused act_culling
##  [1,]        20      1195  861.6780          108           2          36
##  [2,]        20      1349 1020.4082          110           0          36
##  [3,]        20       393  453.5147          110           0          36
##  [4,]        20      1015  816.3265          110           0          36
##  [5,]        20      1127 1156.4626           14          96         262
##  [6,]        20      1251 1179.1383           10         100         400
##  [7,]        20       737  975.0567          108           2          36
##  [8,]        20      1603 1473.9229           10         100         305
##  [9,]        20       401  362.8118          109           1          36
## [10,]        20      1453 1564.6259           10         100         400
##       act_unused harvested
##  [1,]          4        36
##  [2,]          0        36
##  [3,]          3        36
##  [4,]          2        36
##  [5,]         30       262
##  [6,]          0       400
##  [7,]          5        36
##  [8,]         95       305
##  [9,]          2        36
## [10,]          0       400
```

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

Note that while extinction is still non-existent 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) is 1281, with a maximum of 1009 and minimum of 637. While the median population size of `gmse_reps2` (management every two time steps) is 1603, with a maximum of 1161 and minimum of 393. We can then see what happens when management occurs only once

in every three time steps.

```
##       time_step resources  estimate cost_culling cost_unused act_culling
##  [1,]        20       927  952.3810          110           0          36
##  [2,]        20       546  226.7574          110           0          36
##  [3,]        20       230 1179.1383           10         100         400
##  [4,]        20      1097 1428.5714           10         100         400
##  [5,]        20       791 1224.4898           10         100         400
##  [6,]        20       588  476.1905          108           2          36
##  [7,]        20       755 1247.1655           10         100         400
##  [8,]        20      1175  770.9751          110           0          36
##  [9,]        20      1147 1541.9501           10         100         400
## [10,]        20       467  498.8662          108           2          36
##       act_unused harvested
##  [1,]          4        36
##  [2,]          0        36
##  [3,]          0       230
##  [4,]          0       400
##  [5,]          0       400
##  [6,]          2        36
##  [7,]          0       400
##  [8,]          2        36
##  [9,]          0       400
## [10,]          1        36
```

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

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.