Estimating values

Defining function

```
ClearAll["Global`*"];

bestSResponse[uInitial_, cInitial_] := {

  rR = 100;
  (* Constant value for the energetic budget R *)

  u = uInitial;
  c = cInitial * rR;

  nRel[s_, sHat_] := (c + sHat) / (c + s);

  fitness[sHat_, nHat_, s_, n_] :=
```

$$
\begin{aligned}
&\texttt{nRel[s, sHat]} * \left(\frac{1}{\texttt{nHat}} * \left(1 + \sum_{x=2}^{\texttt{nHat}} \left(\left(\left(\prod_{i=1}^{x-1} \frac{i*(1-u)}{1+(i-1)*(1-u)}\right) * s\right) \Big/ \left(\left(x - (x-1)*u - \right.\right.\right.\right.\\
&\qquad\left.\left.\left.\left.\prod_{i=1}^{x-1} \frac{i*(1-u)}{1+(i-1)*(1-u)}\right) * \texttt{sHat} + \left(\prod_{i=1}^{x-1} \frac{i*(1-u)}{1+(i-1)*(1-u)}\right) * s\right)\right)\right) + \\
&\quad\sum_{\texttt{role}=2}^{\texttt{nHat}} \left(\frac{1}{\texttt{nHat}} * \left(\frac{s}{(\texttt{role} - (\texttt{role}-1)*u - 1)*\texttt{sHat} + s} + \right.\right.\\
&\qquad\sum_{x=\texttt{role}+1}^{\texttt{nHat}} \left(\left(\left(\prod_{i=\texttt{role}}^{x-1} \frac{i*(1-u)}{1+(i-1)*(1-u)}\right) * s\right) \Big/ \right.\\
&\qquad\quad\left(\left(x - (x-1)*u - \prod_{i=\texttt{role}}^{x-1} \frac{i*(1-u)}{1+(i-1)*(1-u)}\right) * \texttt{sHat} + \right.\\
&\qquad\qquad\left.\left.\left.\left.\left.\left(\prod_{i=\texttt{role}}^{x-1} \frac{i*(1-u)}{1+(i-1)*(1-u)}\right) * s\right)\right)\right)\right)\right);
\end{aligned}
$$

```
  If[FractionalPart[rR / c] == 0, ns = Range[1, (rR / c) - 1], ns = Range[1, (rR / c)]];
  (* If the maximum possible number of mating events would consume all energy
   budget, then there would be no remaining energy to invest in ejaculates,
  which is biologically unrealistic. Therefore, in such situations,
  the maximum integer value possible is (R/C)-1; otherwise,
  the maximum possible number of matings is just R/C *)
```

$$\texttt{ss} = \texttt{Flatten}\left[\texttt{N}\left[\frac{\texttt{rR} - \texttt{c} * \texttt{ns}}{\texttt{ns}}\right]\right];$$

```
  (* Create the set of possible
   strategies to be played by mutant males, given C. *)

  bestS = {};
```

```
bestN = {};

(* Loop for all possible initial male strategy condition *)
For[nHati = 1, nHati < (Length[ns] + 1), nHati++, {
  bestResponseN = {0, ns[[nHati]]};
  bestResponseS = {0, ss[[nHati]]};

  k = 2;
  While[bestResponseS[[k]] ≠ bestResponseS[[k - 1]],
   (nHatInitial = bestResponseN[[k]];
    sHatInitial = bestResponseS[[k]];
    (* Set the wild-type male reproductive strategy,
    given by nHat and sHat *)

    fit = fitness[sHat = sHatInitial, nHat = nHatInitial, s = ss, n = ns];
    (* Calculated the fitness for every mutant strategy given
     all males in the population plays the initial condition *)

    If[Length[Position[fit, Max[fit]]] == 1,
     {AppendTo[bestResponseN, Extract[ns, Position[fit, Max[fit]][[1]]]];
      AppendTo[bestResponseS, Extract[ss, Position[fit, Max[fit]][[1]]]]},
     (* If there is only one strategy that yields the highest
       fitness given the in the wild-type male population conditions,
     add to the list of bestResponseN and bestResponseS the values
      of n and s adopted by the mutant, since they will be used
       in the next iteration as the wild-type male strategy *)

     {Print[{"c = ", c, "u = ", u, "intercept = ", intercept,
        "MAX FIT = ", Max[fit], "n=", Extract[ns, Position[fit, Max[fit]]],
        "s=", Extract[ss, Position[fit, Max[fit]]], "beta=",
        Extract[beta, Position[fit, Max[fit]]]}], Abort[]}];
    (* Otherwise, report an error message with the set of
     conditions to be evaluated later *)


    If[k == 10, Break[], k++](*}]*))];
  (* By previous observations,
  the stable response is reached after just 3 or 4 iterations. Therefore,
  this part only accounts for stopping the loop when they
   get stuck back and forth between neighbor solutions *)

  (* At this point, bestResponseN and bestResponseS are vectors containing the
   trajectory of strategy substitution since the initial conditions *)
  If[k < 10,
   {AppendTo[bestN, bestResponseN[[Length[bestResponseN]]]];
    AppendTo[bestS, bestResponseS[[Length[bestResponseS]]]]},
   (* If the while loop stopped before the 10th iteration,
   it means it stabilized in the ESS. So, add the last strategies. *)

   {AppendTo[bestN, DeleteDuplicates[bestResponseN[[Range[6, 10]]]]];
    AppendTo[bestS, DeleteDuplicates[bestResponseS[[Range[6, 10]]]]]
    (* Otherwise, add the list of non duplicated strategies
     in the four last iterations. For this particular study,
    usually ESS was found in the first 3 or 4 iterations. Therefore,
    additional iterations normally cicled between two neighbor solutions.*)
```

```
      }]
     (* At this point, bestN and bestS incorporated the best response strategy
       (or pair of best response strategies) when the initial wild-
      type males adopted bestResponseN[[nHati]] and bestResponseS[[nHati]]. The
       next step is checking the consistency for the best
       response strategy across different initial conditions *)

   }];

  DeleteDuplicates[bestN],
  DeleteDuplicates[N[bestS]]

  (* The outcome is a vector including at position [[1]] the non-
   duplicated bestN and at position [[2]], the non-duplicated bestS,
  when the population-level parameters u=uInitial and c=cInitial. *)
 }
```

Fine scale *c* - range [from 0.5% to 5%]

```
Clear[c, u, dummy, rR, fitness, bestN, bestResponseN,
  bestResponseS, bestS, ns, ss, nHat, nHatInitial, sHat, sHatInitial];

DistributeDefinitions[c, u, dummy, rR, fitness, bestN, bestResponseN, bestResponseS,
  bestS, bestSResponse, ns, ss, nHat, nHatInitial, sHat, sHatInitial];

dummy = {0.1, 0.9};

Timing[

 For[h = 1, h < 3, h++, {

   nStar = {};
   sStar = {};

   c = Range[0.5 / 100, 5 / 100, 0.2 / 100];
   u = PadLeft[{}, Length[c], dummy[[h]]];

   m = Parallelize[MapThread[bestSResponse, {uInitial = u, cInitial = c }]];

   For[j = 1, j < 26, j++, {
     AppendTo[nStar, m[[j]][[1]]];
     AppendTo[sStar, m[[j]][[2]]]}];

   Print[{"u =", dummy[[h]]}];
   Print[{"n* = ", nStar}];
   Print[{"s* = ", sStar}]
  }]
]
```

Fine scale *u* - [*c* = 0.5, 1, and 5%]

```
Clear[c, u, dummy, rR, fitness, bestN, bestResponseN,
  bestResponseS, bestS, ns, ss, nHat, nHatInitial, sHat, sHatInitial];

DistributeDefinitions[c, u, dummy, rR, fitness, bestN, bestResponseN, bestResponseS,
  bestS, bestSResponse, ns, ss, nHat, nHatInitial, sHat, sHatInitial];

dummy = {0.5 / 100, 1 / 100, 5 / 100};

Timing[

 For[h = 1, h < 4, h++, {

   nStar = {};
   sStar = {};

   u = Range[0.01, 1, 0.04];
   c = PadLeft[{}, Length[u], dummy[[h]]];

   m = Parallelize[MapThread[bestSResponse, {uInitial = u, cInitial = c }]];

   For[j = 1, j < 26, j++, {
     AppendTo[nStar, m[[j]][[1]]];
     AppendTo[sStar, m[[j]][[2]]]}];

   Print[{"c =", dummy[[h]]}];
   Print[{"n* = ", nStar}];
   Print[{"s* = ", sStar}]
  }]
]
```

Calculating fertilization success

Defining functions

```
ClearAll["Global`*"];
```

$$\text{fitness[n\_, role\_, u\_] :=} \sum_{x=\text{role}}^{n} \left(\frac{\prod_{i=\text{role}}^{x-1} \left(\frac{i*(1-u)}{1+(i-1)*(1-u)} \right)}{(x - (x-1)*u)} \right);$$

Immediate fertilization success (i.e., paternity)

```
roleSeq = Range[1, 20, 1];
nSeq = Range[1, 20, 1];
dummy = {0.1, 0.9};

For[j = 1, j < 3, j++, {
  uSeq = PadLeft[{}, Length[roleSeq], dummy[[j]]];
  m = MapThread[fitness, {n = nSeq, role = roleSeq, u = uSeq}];
  Print["u = ", dummy[[j]], m]
 }]



uSeq = Flatten[{0.01, Range[0.05, 0.95, 0.05], 0.99}];
dummy = {2, 3, 4};

For[j = 1, j < 4, j++, {
  roleSeq = PadLeft[{}, Length[uSeq], dummy[[j]]];
  nSeq = roleSeq;
  m = MapThread[fitness, {n = nSeq, role = roleSeq, u = uSeq}];
  Print["role = ", dummy[[j]], m]
 }]
```

Success across all clutches of the same female

```
dummy = {0.1, 0.9};

For[j = 1, j < 5, j++, {
  nSeq = Range[j, 30, 1];

  For[i = 1, i < 3, i++, {
    u = dummy[[i]];
    m = fitness[n = nSeq, role = j, u = u];
    Print["role = ", j, ", u = ", u, "   ", m]
   }]
 }]
```