Are equilibration fix points global optima? Testing a hypothesis

This notebook analysis an ensmble where initial commitments have been varied randomly.

```
In[•]:= SetDirectory[$HomeDirectory];
If[! MemberQ[$Path, #], AppendTo[$Path, #]] &[
  FileNameJoin[{"git", "DialecticalStructures"}]];
If[! MemberQ[$Path, #], AppendTo[$Path, #]] &[
  FileNameJoin[{"git", "ReflectiveEquilibrium"}]];
<< DialecticalStructures`BasicTDS`;
<< DialecticalStructures`InductiveReasoning`;
<< DialecticalStructures`CoherenceMeasures`;
<< DialecticalStructures`PositionsAnalytics`;
<< ReflectiveEquilibrium`ReflectiveEquilibrium`;
```

Setting up the scene

Get data from first case.

```
In[•]:= ensembleDir = "2018_09_17-0002";
data = Get[FileNameJoin[{
      NotebookDirectory[],
      "results",
      ensembleDir,
      ensembleDir <> "#" <> IntegerString[1, 10, 6] <> ".m"
     }]];
senIDs = Cases[data, {"senIDs", _}][[1, 2]];
tau = Cases[data, {"tau", _}][[1, 2]];
param = Cases[data, {"parameters", _}][[1, 2]];
```

Get initial commitments for all cases.

In[•]:=
```
initialComs = Module[{
   n
  },
  n = Length[FileNames[ensembleDir <> "#*.m", FileNameJoin[{
       NotebookDirectory[],
       "results",
       ensembleDir}]]];
  Table[
   Lookup[
    Cases[
      Get[FileNameJoin[{
         NotebookDirectory[],
         "results",
         ensembleDir,
         ensembleDir <> "#" <> IntegerString[i, 10, 6] <> ".m"
        }]],
      {"posEvolution", _}][[1, 2, 1]],
    "COM"],
   {i, n}
  ]
 ]
```

Out[•]=
```
{1026, 1464, 2166, 70, 653, 341, 294, 804, 579, 1482, 48, 178, 1192, 150, 1150, 419,
 26, 1405, 1273, 1568, 1303, 1374, 1285, 754, 1581, 1473, 1986, 1632, 901, 1399,
 329, 1865, 652, 118, 44, 2154, 1139, 147, 26, 1095, 574, 700, 1026, 1435, 1409,
 838, 1039, 1348, 6, 1574, 1890, 2138, 504, 531, 2070, 536, 1033, 241, 1449,
 567, 82, 1162, 53, 11, 2029, 907, 681, 885, 1134, 580, 330, 1079, 1240, 1397,
 453, 216, 1596, 1242, 712, 1047, 1852, 462, 2051, 487, 1808, 105, 1795, 1161,
 2065, 1165, 1335, 255, 797, 154, 187, 130, 538, 1583, 619, 62, 1090, 854, 1744,
 1572, 1169, 280, 592, 471, 16, 417, 1630, 1504, 1381, 2126, 320, 193, 1416, 639,
 1019, 929, 2174, 788, 63, 534, 901, 740, 2174, 1683, 931, 1406, 1554, 666, 533,
 668, 1137, 1510, 140, 771, 2066, 1705, 806, 2166, 732, 1285, 648, 1144, 828,
 1333, 820, 1903, 881, 246, 1930, 55, 1364, 1637, 1940, 2158, 2110, 223, 1896,
 1787, 1747, 596, 329, 2077, 1421, 1523, 1417, 1149, 2004, 333, 2042, 1950, 288,
 1685, 753, 1061, 1166, 1264, 2068, 955, 158, 1637, 1133, 1568, 2123, 1924,
 1962, 1369, 172, 1972, 2085, 1436, 167, 760, 665, 683, 2047, 140, 410, 220,
 1486, 128, 672, 168, 1998, 342, 1766, 1309, 1097, 1061, 1201, 1993, 1258, 353,
 766, 1108, 2097, 1972, 1665, 391, 2007, 1444, 280, 1537, 1289, 898, 133, 2104,
 185, 1691, 1035, 14, 376, 598, 984, 815, 1600, 1386, 442, 750, 46, 358, 1013,
 1959, 1928, 1631, 2043, 812, 165, 1851, 400, 1630, 606, 973, 1511, 214, 1818,
 1442, 1470, 2004, 1700, 773, 1820, 1703, 1421, 1653, 1644, 1325, 1283, 1529,
 2021, 895, 2025, 40, 909, 635, 751, 357, 1589, 1438, 507, 305, 2136, 1655, 1436,
 176, 1238, 73, 684, 148, 1971, 833, 390, 141, 1160, 1282, 131, 715, 2077, 328,
 201, 756, 695, 2010, 1052, 1417, 1675, 994, 72, 1875, 2133, 1548, 1775, 429,
 1, 411, 1242, 91, 482, 1532, 2100, 1356, 710, 649, 921, 468, 640, 26, 605, 697,
 820, 2087, 140, 50, 1217, 1229, 339, 1770, 1540, 719, 910, 2042, 1243, 316,
 1035, 275, 992, 521, 869, 2066, 557, 953, 1055, 1655, 124, 1839, 185, 1467,
 101, 1121, 1877, 210, 558, 659, 1850, 1170, 1611, 901, 1128, 2022, 1837, 2078,
 1355, 2041, 610, 436, 1139, 492, 1388, 744, 813, 1824, 2035, 1669, 819, 439,
 231, 454, 1040, 1594, 1768, 252, 222, 645, 1493, 2092, 1513, 1176, 432, 1605,
 165, 191, 2065, 1652, 1392, 1944, 1429, 1172, 972, 1793, 2145, 835, 550, 1498,
 2096, 1663, 1939, 1446, 1053, 1407, 140, 1549, 1225, 1011, 1357, 1194, 1371,
 65, 1180, 316, 838, 1441, 1067, 819, 2023, 1693, 1828, 965, 250, 1923, 748,
 1929, 605, 1432, 1319, 765, 1457, 1324, 1035, 847, 963, 1813, 1926, 23, 1927,
 1179, 721, 329, 1145, 618, 2000, 1754, 1009, 1873, 2134, 459, 1775, 1626, 1547,
 601, 1258, 743, 2056, 351, 118, 54, 1560, 222, 39, 1418, 1007, 561, 2027, 1426,
 497, 1808, 964, 720, 1254, 1599, 1374, 906, 570, 403, 2007, 387, 126, 1903}
```

Calculate sigma.

In[•]:=
```
PrintTemporary["Creating sigma..."];
sigma = Lookup[
  param,
  "sigma",
  Sigma[tau, True, senIDs]
  ];
PrintTemporary["...done."];
```

Calculate nPrinciples.

In[•]:=
```
PrintTemporary["Creating nPrinciples..."];
nPrinciples = Lookup[
  param,
  "nPrinciples",
  NPrinciples[sigma, senIDs]
  ];
PrintTemporary["...done."];
```

Calculate closedPosition.

In[•]:=
```
PrintTemporary["Creating closedPositions..."];
closedPositions =
  DeleteCases[DialecticallyClosedPositions[tau, True, senIDs], 1];
PrintTemporary["...done."];
```

We say that **P** is the set of all permissible theory-commitment pairs <T,C>, where such a pair is permissible iff C is minimally consistent and T is consistent and closed. (Claus is right in saying that we’ve defined the notions of theory and commitments such that <T,C>-pairs are permissible.) Furthermore, let **T** be the set of all consistent and closed theories, and **C** be the set of all minimally consistent commitments. **C** is the list **Range[3^7]** (integer-representation of partial positions). **T** is a subset of **C**. **P** is hence a list of integer-pairs.

In[•]:=
```
allTCPairs = Tuples[{closedPositions, Range[3^Length[senIDs]]}];
```

We further calculate

- for every T in **T**: Simplicity[T] and store the results in SparseArray **Simp**. (Simplicity[T] is stored in **Simp** at position T.)

In[•]:=
```
Simp =
  SparseArray[# -> Simplicity[#, nPrinciples[[#]], senIDs] & /@ closedPositions];
```

- for every C in **C** and for every C_0^i in initialComs: Closeness[C,C_0] and store the results in SparseArray **Clos[[i]]**. (Closeness[C,C_0^i] is stored in **Clos[[i]]** at position C.)

In[•]:=
```
Clos = Map[
   Function[
    initialCom,
    SparseArray[# → Closeness[#, initialCom, senIDs, KeyTake[param,
          {"ConflictPenalty", "ContractionPenalty", "ExpansionPenalty"}]] &
      /@ Range[3^Length[senIDs]]
    ]
   ],
   initialComs
  ];
```

- for every <T,C> in **P**: Account[T,C] and store the results in SparseArray **Acco**. (Account[T,C] is stored in **Acco** at position {T,C}.)

In[•]:=
```
Account = AccountFunction[param];
```

In[•]:=
```
Acco = Monitor[
   SparseArray[
    Flatten[Table[
      {closedPositions[[i]], c} →
       Account[c, closedPositions[[i]], sigma, senIDs],
      {i, Length[closedPositions]}, {c, 3^Length[senIDs]}
     ],
     1
    ]
   ],
   ProgressIndicator[i, {1, Length[closedPositions]}]
  ];
```

Evaluating fixed points

In[•]:=
```
With[
  {
   alpha = Lookup[param, "alpha"],
   beta = Lookup[param, "beta"]
  },
  {
   wa = (alpha * beta) / (alpha + beta - alpha * beta);
   ws = (beta - alpha * beta) / (alpha + beta - alpha * beta);
   wc = 1 - (wa + ws);
  }
 ];
AchievementFunction[tcp_, caseindex_] :=
  ws * Simp[[First[tcp]]] +
   wc * Clos[[caseindex, Last[tcp]]] + wa * Acco[[First[tcp], Last[tcp]]];
```

In[•]:=
```
(* TEST *)
AchievementFunction[
 {closedPositions[[4]], 68},
 200
]
```

Out[•]= 0.931354

In[•]:=
```
diagnostics = Module[{
    initialCom, finalstate, n, allAchievementValues
   },
   n = Length[FileNames[ensembleDir <> "#*.m", FileNameJoin[{
        NotebookDirectory[],
        "results",
        ensembleDir}]]];
   Table[
    PrintTemporary["Analysing ensemble-member " <> ToString[i]];
    initialCom = Lookup[
      Cases[
        Get[FileNameJoin[{
           NotebookDirectory[],
           "results",
           ensembleDir,
           ensembleDir <> "#" <> IntegerString[i, 10, 6] <> ".m"
          }]],
        {"posEvolution", _}][[1, 2, 1]],
      "COM"];
    finalstate = Lookup[
      Cases[
        Get[FileNameJoin[{
           NotebookDirectory[],
           "results",
           ensembleDir,
           ensembleDir <> "#" <> IntegerString[i, 10, 6] <> ".m"
          }]],
        {"posEvolution", _}][[1, 2, -1]],
      {"THE", "COM"}];
    allAchievementValues = (AchievementFunction[#, i] &) /@ allTCPairs;
    {
     i,
     First[finalstate] == Last[finalstate],
     Max[allAchievementValues] - AchievementFunction[finalstate, i]},
    {i, n}
   ]
  ];
```

In[•]:=
```
diagnostics
```

Out[•]=
```
{{1, False, 0.}, {2, True, 0.}, {3, False, 0.}, {4, True, 0.}, {5, True, 0.},
 {6, True, 0.}, {7, True, 0.}, {8, True, 0.}, {9, True, 0.}, {10, True, 0.},
 {11, True, 0.}, {12, True, 0.}, {13, True, 0.}, {14, True, 0.}, {15, False, 0.},
 {16, False, 0.}, {17, True, 0.}, {18, True, 0.}, {19, True, 0.}, {20, True, 0.},
 {21, True, 0.}, {22, True, 0.}, {23, True, 0.}, {24, True, 0.}, {25, False, 0.},
 {26, True, 0.}, {27, False, 0.}, {28, True, 0.}, {29, True, 0.}, {30, False, 0.},
 {31, True, 0.}, {32, False, 0.}, {33, True, 0.}, {34, True, 0.},
 {35, True, 0.}, {36, False, 0.00185529}, {37, True, 0.}, {38, True, 0.},
```

```
{39, True, 0.}, {40, True, 0.}, {41, True, 0.}, {42, True, 0.}, {43, False, 0.},
{44, True, 0.}, {45, False, 0.}, {46, True, 0.}, {47, True, 0.}, {48, True, 0.},
{49, True, 0.00185529}, {50, True, 0.}, {51, False, 0.}, {52, False, 0.},
{53, True, 0.}, {54, True, 0.}, {55, True, 0.}, {56, True, 0.}, {57, True, 0.},
{58, True, 0.}, {59, False, 0.00185529}, {60, False, 0.}, {61, True, 0.},
{62, True, 0.}, {63, True, 0.}, {64, True, 0.00185529}, {65, True, 0.},
{66, True, 0.}, {67, False, 0.}, {68, True, 0.}, {69, True, 0.}, {70, True, 0.},
{71, True, 0.}, {72, False, 0.}, {73, True, 0.}, {74, False, 0.}, {75, False, 0.},
{76, True, 0.}, {77, True, 0.}, {78, False, 0.}, {79, True, 0.}, {80, True, 0.},
{81, False, 0.}, {82, True, 0.}, {83, True, 0.}, {84, True, 0.}, {85, False, 0.},
{86, True, 0.}, {87, True, 0.}, {88, False, 0.}, {89, True, 0.}, {90, True, 0.},
{91, True, 0.}, {92, True, 0.}, {93, True, 0.}, {94, True, 0.}, {95, True, 0.},
{96, True, 0.}, {97, True, 0.}, {98, True, 0.}, {99, True, 0.}, {100, True, 0.},
{101, True, 0.}, {102, True, 0.}, {103, True, 0.}, {104, True, 0.},
{105, False, 0.}, {106, True, 0.}, {107, True, 0.}, {108, True, 0.},
{109, True, 0.00185529}, {110, True, 0.}, {111, True, 0.}, {112, True, 0.},
{113, True, 0.}, {114, False, 0.}, {115, True, 0.}, {116, True, 0.},
{117, False, 0.}, {118, True, 0.}, {119, False, 0.}, {120, True, 0.},
{121, False, 0.00185529}, {122, True, 0.}, {123, True, 0.}, {124, True, 0.},
{125, True, 0.}, {126, True, 0.}, {127, False, 0.00185529}, {128, True, 0.},
{129, False, 0.}, {130, True, 0.}, {131, True, 0.}, {132, True, 0.},
{133, True, 0.}, {134, True, 0.}, {135, True, 0.}, {136, True, 0.}, {137, True, 0.},
{138, True, 0.}, {139, True, 0.}, {140, True, 0.}, {141, True, 0.},
{142, False, 0.}, {143, True, 0.00185529}, {144, True, 0.}, {145, False, 0.},
{146, True, 0.}, {147, True, 0.}, {148, True, 0.}, {149, True, 0.},
{150, False, 0.00185529}, {151, True, 0.}, {152, True, 0.}, {153, False, 0.},
{154, True, 0.}, {155, False, 0.}, {156, True, 0.}, {157, False, 0.00185529},
{158, False, 0.}, {159, False, 0.}, {160, True, 0.}, {161, False, 0.},
{162, True, 0.}, {163, False, 0.}, {164, True, 0.}, {165, True, 0.},
{166, True, 0.}, {167, True, 0.}, {168, True, 0.}, {169, False, 0.},
{170, True, 0.}, {171, True, 0.}, {172, True, 0.}, {173, True, 0.}, {174, True, 0.},
{175, True, 0.}, {176, False, 0.}, {177, True, 0.}, {178, True, 0.},
{179, False, 0.}, {180, False, 0.}, {181, True, 0.}, {182, True, 0.},
{183, True, 0.}, {184, True, 0.}, {185, False, 0.}, {186, True, 0.},
{187, True, 0.}, {188, False, 0.}, {189, True, 0.}, {190, True, 0.},
{191, True, 0.}, {192, True, 0.}, {193, True, 0.}, {194, False, 0.},
{195, True, 0.}, {196, True, 0.}, {197, True, 0.}, {198, True, 0.},
{199, False, 0.}, {200, True, 0.}, {201, True, 0.}, {202, True, 0.},
{203, True, 0.}, {204, True, 0.}, {205, False, 0.}, {206, True, 0.},
{207, False, 0.}, {208, True, 0.}, {209, False, 0.}, {210, True, 0.},
{211, True, 0.}, {212, True, 0.}, {213, True, 0.}, {214, False, 0.},
{215, True, 0.}, {216, True, 0.}, {217, True, 0.}, {218, True, 0.},
{219, False, 0.}, {220, True, 0.}, {221, False, 0.}, {222, True, 0.},
{223, False, 0.}, {224, False, 0.}, {225, True, 0.}, {226, True, 0.},
{227, False, 0.}, {228, True, 0.}, {229, True, 0.}, {230, False, 0.},
{231, False, 0.}, {232, False, 0.}, {233, True, 0.}, {234, True, 0.},
{235, True, 0.}, {236, True, 0.}, {237, True, 0.}, {238, True, 0.}, {239, True, 0.},
```

```
{240, True, 0.}, {241, True, 0.}, {242, True, 0.}, {243, True, 0.}, {244, True, 0.},
{245, False, 0.}, {246, False, 0.}, {247, False, 0.00185529}, {248, True, 0.},
{249, True, 0.}, {250, True, 0.}, {251, True, 0.}, {252, True, 0.}, {253, True, 0.},
{254, True, 0.}, {255, True, 0.}, {256, True, 0.00185529}, {257, True, 0.},
{258, True, 0.}, {259, False, 0.}, {260, False, 0.}, {261, True, 0.},
{262, True, 0.}, {263, False, 0.00185529}, {264, True, 0.}, {265, True, 0.},
{266, True, 0.}, {267, True, 0.}, {268, False, 0.}, {269, False, 0.},
{270, True, 0.}, {271, False, 0.}, {272, True, 0.}, {273, False, 0.00185529},
{274, True, 0.}, {275, False, 0.}, {276, True, 0.}, {277, True, 0.},
{278, True, 0.}, {279, True, 0.}, {280, True, 0.}, {281, False, 0.},
{282, False, 0.}, {283, True, 0.}, {284, True, 0.}, {285, False, 0.},
{286, True, 0.}, {287, False, 0.}, {288, True, 0.}, {289, False, 0.},
{290, True, 0.}, {291, True, 0.}, {292, True, 0.}, {293, False, 0.},
{294, True, 0.}, {295, True, 0.}, {296, True, 0.}, {297, False, 0.00185529},
{298, True, 0.}, {299, True, 0.}, {300, False, 0.}, {301, True, 0.},
{302, True, 0.}, {303, True, 0.}, {304, True, 0.}, {305, True, 0.},
{306, False, 0.}, {307, False, 0.}, {308, False, 0.}, {309, True, 0.},
{310, True, 0.}, {311, True, 0.}, {312, False, 0.}, {313, False, 0.00927644},
{314, True, 0.}, {315, False, 0.}, {316, True, 0.}, {317, True, 0.},
{318, True, 0.}, {319, False, 0.}, {320, True, 0.00185529}, {321, False, 0.},
{322, False, 0.}, {323, False, 0.}, {324, True, 0.}, {325, True, 0.},
{326, True, 0.}, {327, True, 0.}, {328, True, 0.}, {329, True, 0.}, {330, True, 0.},
{331, True, 0.}, {332, False, 0.}, {333, True, 0.}, {334, True, 0.},
{335, True, 0.}, {336, True, 0.}, {337, True, 0.}, {338, True, 0.}, {339, True, 0.},
{340, True, 0.}, {341, True, 0.}, {342, True, 0.}, {343, True, 0.}, {344, True, 0.},
{345, True, 0.}, {346, True, 0.}, {347, True, 0.}, {348, True, 0.}, {349, True, 0.},
{350, True, 0.}, {351, True, 0.}, {352, True, 0.}, {353, True, 0.},
{354, False, 0.}, {355, True, 0.}, {356, True, 0.}, {357, True, 0.},
{358, True, 0.}, {359, False, 0.}, {360, True, 0.}, {361, True, 0.},
{362, True, 0.}, {363, False, 0.00185529}, {364, True, 0.}, {365, True, 0.},
{366, True, 0.}, {367, False, 0.}, {368, False, 0.00927644}, {369, False, 0.},
{370, True, 0.}, {371, True, 0.}, {372, False, 0.}, {373, True, 0.},
{374, True, 0.}, {375, True, 0.}, {376, True, 0.}, {377, True, 0.}, {378, True, 0.},
{379, True, 0.}, {380, True, 0.}, {381, True, 0.}, {382, True, 0.}, {383, True, 0.},
{384, False, 0.}, {385, True, 0.}, {386, False, 0.00185529}, {387, True, 0.},
{388, True, 0.}, {389, True, 0.}, {390, False, 0.}, {391, True, 0.},
{392, True, 0.}, {393, True, 0.}, {394, True, 0.}, {395, True, 0.}, {396, True, 0.},
{397, True, 0.}, {398, False, 0.}, {399, True, 0.}, {400, False, 0.},
{401, True, 0.}, {402, True, 0.}, {403, True, 0.}, {404, True, 0.}, {405, True, 0.},
{406, False, 0.}, {407, False, 0.}, {408, False, 0.00185529}, {409, False, 0.},
{410, False, 0.}, {411, False, 0.}, {412, True, 0.}, {413, False, 0.},
{414, True, 0.}, {415, True, 0.}, {416, True, 0.}, {417, True, 0.}, {418, True, 0.},
{419, False, 0.}, {420, False, 0.}, {421, True, 0.}, {422, False, 0.},
{423, True, 0.}, {424, True, 0.}, {425, True, 0.}, {426, True, 0.}, {427, True, 0.},
{428, True, 0.}, {429, True, 0.}, {430, True, 0.}, {431, False, 0.},
{432, True, 0.}, {433, True, 0.}, {434, False, 0.}, {435, True, 0.},
{436, True, 0.}, {437, False, 0.}, {438, False, 0.}, {439, False, 0.},
```

```
{440, False, 0.}, {441, True, 0.}, {442, True, 0.}, {443, True, 0.},
{444, False, 0.}, {445, True, 0.}, {446, True, 0.}, {447, False, 0.},
{448, True, 0.}, {449, False, 0.00185529}, {450, True, 0.}, {451, True, 0.},
{452, True, 0.}, {453, False, 0.}, {454, True, 0.}, {455, False, 0.},
{456, True, 0.}, {457, False, 0.}, {458, False, 0.00185529}, {459, False, 0.},
{460, True, 0.}, {461, False, 0.}, {462, False, 0.}, {463, True, 0.},
{464, False, 0.}, {465, True, 0.}, {466, False, 0.}, {467, True, 0.},
{468, False, 0.00927644}, {469, False, 0.}, {470, True, 0.}, {471, True, 0.},
{472, True, 0.}, {473, True, 0.}, {474, True, 0.}, {475, True, 0.},
{476, True, 0.}, {477, True, 0.}, {478, True, 0.}, {479, True, 0.},
{480, True, 0.}, {481, True, 0.}, {482, False, 0.}, {483, True, 0.},
{484, True, 0.}, {485, True, 0.}, {486, False, 0.00185529}, {487, True, 0.},
{488, False, 0.}, {489, True, 0.}, {490, False, 0.}, {491, True, 0.},
{492, True, 0.}, {493, True, 0.}, {494, True, 0.}, {495, True, 0.}, {496, True, 0.},
{497, False, 0.}, {498, True, 0.}, {499, True, 0.}, {500, False, 0.00185529}}
```

Analyse diagnostics

```
In[•]:= diagnostics = {{1, False, 0.`}, {2, True, 0.`}, {3, False, 0.`}, {4, True, 0.`},
   {5, True, 0.`}, {6, True, 0.`}, {7, True, 0.`}, {8, True, 0.`}, {9, True, 0.`},
   {10, True, 0.`}, {11, True, 0.`}, {12, True, 0.`}, {13, True, 0.`},
   {14, True, 0.`}, {15, False, 0.`}, {16, False, 0.`}, {17, True, 0.`},
   {18, True, 0.`}, {19, True, 0.`}, {20, True, 0.`}, {21, True, 0.`},
   {22, True, 0.`}, {23, True, 0.`}, {24, True, 0.`}, {25, False, 0.`},
   {26, True, 0.`}, {27, False, 0.`}, {28, True, 0.`}, {29, True, 0.`},
   {30, False, 0.`}, {31, True, 0.`}, {32, False, 0.`}, {33, True, 0.`},
   {34, True, 0.`}, {35, True, 0.`}, {36, False, 0.0018552875695733162`},
   {37, True, 0.`}, {38, True, 0.`}, {39, True, 0.`}, {40, True, 0.`},
   {41, True, 0.`}, {42, True, 0.`}, {43, False, 0.`}, {44, True, 0.`},
   {45, False, 0.`}, {46, True, 0.`}, {47, True, 0.`}, {48, True, 0.`},
   {49, True, 0.0018552875695733162`}, {50, True, 0.`},
   {51, False, 0.`}, {52, False, 0.`}, {53, True, 0.`}, {54, True, 0.`},
   {55, True, 0.`}, {56, True, 0.`}, {57, True, 0.`}, {58, True, 0.`},
   {59, False, 0.0018552875695733162`}, {60, False, 0.`}, {61, True, 0.`},
   {62, True, 0.`}, {63, True, 0.`}, {64, True, 0.0018552875695733162`},
   {65, True, 0.`}, {66, True, 0.`}, {67, False, 0.`}, {68, True, 0.`},
   {69, True, 0.`}, {70, True, 0.`}, {71, True, 0.`}, {72, False, 0.`},
   {73, True, 0.`}, {74, False, 0.`}, {75, False, 0.`}, {76, True, 0.`},
   {77, True, 0.`}, {78, False, 0.`}, {79, True, 0.`}, {80, True, 0.`},
   {81, False, 0.`}, {82, True, 0.`}, {83, True, 0.`}, {84, True, 0.`},
   {85, False, 0.`}, {86, True, 0.`}, {87, True, 0.`}, {88, False, 0.`},
   {89, True, 0.`}, {90, True, 0.`}, {91, True, 0.`}, {92, True, 0.`},
   {93, True, 0.`}, {94, True, 0.`}, {95, True, 0.`}, {96, True, 0.`},
   {97, True, 0.`}, {98, True, 0.`}, {99, True, 0.`}, {100, True, 0.`},
   {101, True, 0.`}, {102, True, 0.`}, {103, True, 0.`}, {104, True, 0.`},
   {105, False, 0.`}, {106, True, 0.`}, {107, True, 0.`}, {108, True, 0.`},
```

```
{109, True, 0.0018552875695733162`}, {110, True, 0.`}, {111, True, 0.`},
{112, True, 0.`}, {113, True, 0.`}, {114, False, 0.`}, {115, True, 0.`},
{116, True, 0.`}, {117, False, 0.`}, {118, True, 0.`}, {119, False, 0.`},
{120, True, 0.`}, {121, False, 0.0018552875695733162`}, {122, True, 0.`},
{123, True, 0.`}, {124, True, 0.`}, {125, True, 0.`}, {126, True, 0.`},
{127, False, 0.0018552875695733162`}, {128, True, 0.`}, {129, False, 0.`},
{130, True, 0.`}, {131, True, 0.`}, {132, True, 0.`}, {133, True, 0.`},
{134, True, 0.`}, {135, True, 0.`}, {136, True, 0.`}, {137, True, 0.`},
{138, True, 0.`}, {139, True, 0.`}, {140, True, 0.`}, {141, True, 0.`},
{142, False, 0.`}, {143, True, 0.0018552875695733162`}, {144, True, 0.`},
{145, False, 0.`}, {146, True, 0.`}, {147, True, 0.`}, {148, True, 0.`},
{149, True, 0.`}, {150, False, 0.0018552875695733162`}, {151, True, 0.`},
{152, True, 0.`}, {153, False, 0.`}, {154, True, 0.`}, {155, False, 0.`},
{156, True, 0.`}, {157, False, 0.0018552875695733162`}, {158, False, 0.`},
{159, False, 0.`}, {160, True, 0.`}, {161, False, 0.`}, {162, True, 0.`},
{163, False, 0.`}, {164, True, 0.`}, {165, True, 0.`}, {166, True, 0.`},
{167, True, 0.`}, {168, True, 0.`}, {169, False, 0.`}, {170, True, 0.`},
{171, True, 0.`}, {172, True, 0.`}, {173, True, 0.`}, {174, True, 0.`},
{175, True, 0.`}, {176, False, 0.`}, {177, True, 0.`}, {178, True, 0.`},
{179, False, 0.`}, {180, False, 0.`}, {181, True, 0.`}, {182, True, 0.`},
{183, True, 0.`}, {184, True, 0.`}, {185, False, 0.`}, {186, True, 0.`},
{187, True, 0.`}, {188, False, 0.`}, {189, True, 0.`}, {190, True, 0.`},
{191, True, 0.`}, {192, True, 0.`}, {193, True, 0.`}, {194, False, 0.`},
{195, True, 0.`}, {196, True, 0.`}, {197, True, 0.`}, {198, True, 0.`},
{199, False, 0.`}, {200, True, 0.`}, {201, True, 0.`}, {202, True, 0.`},
{203, True, 0.`}, {204, True, 0.`}, {205, False, 0.`}, {206, True, 0.`},
{207, False, 0.`}, {208, True, 0.`}, {209, False, 0.`}, {210, True, 0.`},
{211, True, 0.`}, {212, True, 0.`}, {213, True, 0.`}, {214, False, 0.`},
{215, True, 0.`}, {216, True, 0.`}, {217, True, 0.`}, {218, True, 0.`},
{219, False, 0.`}, {220, True, 0.`}, {221, False, 0.`}, {222, True, 0.`},
{223, False, 0.`}, {224, False, 0.`}, {225, True, 0.`}, {226, True, 0.`},
{227, False, 0.`}, {228, True, 0.`}, {229, True, 0.`}, {230, False, 0.`},
{231, False, 0.`}, {232, False, 0.`}, {233, True, 0.`}, {234, True, 0.`},
{235, True, 0.`}, {236, True, 0.`}, {237, True, 0.`}, {238, True, 0.`},
{239, True, 0.`}, {240, True, 0.`}, {241, True, 0.`}, {242, True, 0.`},
{243, True, 0.`}, {244, True, 0.`}, {245, False, 0.`}, {246, False, 0.`},
{247, False, 0.0018552875695733162`}, {248, True, 0.`}, {249, True, 0.`},
{250, True, 0.`}, {251, True, 0.`}, {252, True, 0.`}, {253, True, 0.`},
{254, True, 0.`}, {255, True, 0.`}, {256, True, 0.0018552875695733162`},
{257, True, 0.`}, {258, True, 0.`}, {259, False, 0.`}, {260, False, 0.`},
{261, True, 0.`}, {262, True, 0.`}, {263, False, 0.0018552875695733162`},
{264, True, 0.`}, {265, True, 0.`}, {266, True, 0.`}, {267, True, 0.`},
{268, False, 0.`}, {269, False, 0.`}, {270, True, 0.`}, {271, False, 0.`},
{272, True, 0.`}, {273, False, 0.0018552875695733162`}, {274, True, 0.`},
{275, False, 0.`}, {276, True, 0.`}, {277, True, 0.`}, {278, True, 0.`},
{279, True, 0.`}, {280, True, 0.`}, {281, False, 0.`}, {282, False, 0.`},
{283, True, 0.`}, {284, True, 0.`}, {285, False, 0.`}, {286, True, 0.`},
```

```
{287, False, 0.`}, {288, True, 0.`}, {289, False, 0.`}, {290, True, 0.`},
{291, True, 0.`}, {292, True, 0.`}, {293, False, 0.`}, {294, True, 0.`},
{295, True, 0.`}, {296, True, 0.`}, {297, False, 0.0018552875695733162`},
{298, True, 0.`}, {299, True, 0.`}, {300, False, 0.`}, {301, True, 0.`},
{302, True, 0.`}, {303, True, 0.`}, {304, True, 0.`}, {305, True, 0.`},
{306, False, 0.`}, {307, False, 0.`}, {308, False, 0.`},
{309, True, 0.`}, {310, True, 0.`}, {311, True, 0.`}, {312, False, 0.`},
{313, False, 0.009276437847866359`}, {314, True, 0.`}, {315, False, 0.`},
{316, True, 0.`}, {317, True, 0.`}, {318, True, 0.`}, {319, False, 0.`},
{320, True, 0.0018552875695733162`}, {321, False, 0.`}, {322, False, 0.`},
{323, False, 0.`}, {324, True, 0.`}, {325, True, 0.`}, {326, True, 0.`},
{327, True, 0.`}, {328, True, 0.`}, {329, True, 0.`}, {330, True, 0.`},
{331, True, 0.`}, {332, False, 0.`}, {333, True, 0.`}, {334, True, 0.`},
{335, True, 0.`}, {336, True, 0.`}, {337, True, 0.`}, {338, True, 0.`},
{339, True, 0.`}, {340, True, 0.`}, {341, True, 0.`}, {342, True, 0.`},
{343, True, 0.`}, {344, True, 0.`}, {345, True, 0.`}, {346, True, 0.`},
{347, True, 0.`}, {348, True, 0.`}, {349, True, 0.`}, {350, True, 0.`},
{351, True, 0.`}, {352, True, 0.`}, {353, True, 0.`}, {354, False, 0.`},
{355, True, 0.`}, {356, True, 0.`}, {357, True, 0.`}, {358, True, 0.`},
{359, False, 0.`}, {360, True, 0.`}, {361, True, 0.`}, {362, True, 0.`},
{363, False, 0.0018552875695733162`}, {364, True, 0.`}, {365, True, 0.`},
{366, True, 0.`}, {367, False, 0.`}, {368, False, 0.009276437847866359`},
{369, False, 0.`}, {370, True, 0.`}, {371, True, 0.`}, {372, False, 0.`},
{373, True, 0.`}, {374, True, 0.`}, {375, True, 0.`}, {376, True, 0.`},
{377, True, 0.`}, {378, True, 0.`}, {379, True, 0.`}, {380, True, 0.`},
{381, True, 0.`}, {382, True, 0.`}, {383, True, 0.`}, {384, False, 0.`},
{385, True, 0.`}, {386, False, 0.0018552875695733162`},
{387, True, 0.`}, {388, True, 0.`}, {389, True, 0.`}, {390, False, 0.`},
{391, True, 0.`}, {392, True, 0.`}, {393, True, 0.`}, {394, True, 0.`},
{395, True, 0.`}, {396, True, 0.`}, {397, True, 0.`}, {398, False, 0.`},
{399, True, 0.`}, {400, False, 0.`}, {401, True, 0.`}, {402, True, 0.`},
{403, True, 0.`}, {404, True, 0.`}, {405, True, 0.`}, {406, False, 0.`},
{407, False, 0.`}, {408, False, 0.0018552875695733162`},
{409, False, 0.`}, {410, False, 0.`}, {411, False, 0.`}, {412, True, 0.`},
{413, False, 0.`}, {414, True, 0.`}, {415, True, 0.`}, {416, True, 0.`},
{417, True, 0.`}, {418, True, 0.`}, {419, False, 0.`}, {420, False, 0.`},
{421, True, 0.`}, {422, False, 0.`}, {423, True, 0.`}, {424, True, 0.`},
{425, True, 0.`}, {426, True, 0.`}, {427, True, 0.`}, {428, True, 0.`},
{429, True, 0.`}, {430, True, 0.`}, {431, False, 0.`}, {432, True, 0.`},
{433, True, 0.`}, {434, False, 0.`}, {435, True, 0.`}, {436, True, 0.`},
{437, False, 0.`}, {438, False, 0.`}, {439, False, 0.`}, {440, False, 0.`},
{441, True, 0.`}, {442, True, 0.`}, {443, True, 0.`}, {444, False, 0.`},
{445, True, 0.`}, {446, True, 0.`}, {447, False, 0.`}, {448, True, 0.`},
{449, False, 0.0018552875695733162`}, {450, True, 0.`}, {451, True, 0.`},
{452, True, 0.`}, {453, False, 0.`}, {454, True, 0.`}, {455, False, 0.`},
{456, True, 0.`}, {457, False, 0.`}, {458, False, 0.0018552875695733162`},
{459, False, 0.`}, {460, True, 0.`}, {461, False, 0.`}, {462, False, 0.`},
```

```
{463, True, 0.`}, {464, False, 0.`}, {465, True, 0.`}, {466, False, 0.`},
{467, True, 0.`}, {468, False, 0.009276437847866359`},
{469, False, 0.`}, {470, True, 0.`}, {471, True, 0.`}, {472, True, 0.`},
{473, True, 0.`}, {474, True, 0.`}, {475, True, 0.`}, {476, True, 0.`},
{477, True, 0.`}, {478, True, 0.`}, {479, True, 0.`}, {480, True, 0.`},
{481, True, 0.`}, {482, False, 0.`}, {483, True, 0.`}, {484, True, 0.`},
{485, True, 0.`}, {486, False, 0.0018552875695733162`},
{487, True, 0.`}, {488, False, 0.`}, {489, True, 0.`}, {490, False, 0.`},
{491, True, 0.`}, {492, True, 0.`}, {493, True, 0.`}, {494, True, 0.`},
{495, True, 0.`}, {496, True, 0.`}, {497, False, 0.`}, {498, True, 0.`},
{499, True, 0.`}, {500, False, 0.0018552875695733162`}};
```

How many final states are global optima?

```
In[•]:= Count[diagnostics, e_ /; (e[[3]] == 0)]
Out[•]= 474
In[•]:= N[474/500]
Out[•]= 0.948
```

How many final states are full RE states?

```
In[•]:= Count[diagnostics, e_ /; (e[[3]] == 0) && e[[2]]]
Out[•]= 356
In[•]:= N[356/500]
Out[•]= 0.712
In[•]:= N[356/474]
Out[•]= 0.751055
```

Initial and final commitments that yield fixed points that are no optima

In[•]:=
```
{
   IntegerToList[initialComs[[#]], senIDs],
   IntegerToList[
    Lookup[
     Cases[
       Get[FileNameJoin[{
          NotebookDirectory[],
          "results",
          ensembleDir,
          ensembleDir <> "#" <> IntegerString[#, 10, 6] <> ".m"
         }]],
       {"posEvolution", _}][[1, 2, -1]],
     "COM"],
    senIDs]
  } & /@ Cases[diagnostics, e_ /; (e[[3]] != 0)][[All, 1]]
```

Out[•]=
```
{{{4, !1, !3, !5, !6, !7}, {3, 4, 5, !1, !2, !6, !7}}, {{2, !1}, {!1}},
 {{3, 7, !1, !2, !4, !5, !6}, {3, 4, 5, 7, !1, !2, !6}}, {{1, 3}, {3}},
 {{3, !2}, {!2}}, {{1, 2, 3, !4, !5, !6, !7}, {1, 3, 5, !2, !4, !6, !7}},
 {{1, 2, 3, !4, !5, !6, !7}, {1, 2, 3, 4, 5, !6, !7}},
 {{7, !1}, {7}}, {{2, 3, 4, 6, !5, !7}, {1, 3, 4, 5, 6, !2, !7}},
 {{1, 2, 6, !3, !4, !5, !7}, {2, 5, 6, !1, !3, !4, !7}},
 {{1, 3, 6, !4, !5, !7}, {1, 3, 4, !2, !5, !6, !7}}, {{6, 7}, {7}},
 {{1, !2, !3, !4, !5, !7}, {1, 3, 4, !2, !5, !6, !7}},
 {{1, 2, !3, !4, !6, !7}, {1, 2, 3, 4, 5, !6, !7}},
 {{1, 6, 7, !2, !3, !5}, {2, 6, 7, !1, !3, !4, !5}},
 {{!1, !2, !3, !5, !6, !7}, {!2, !6, !7}}, {{3, 5}, {5}},
 {{1, 2, 3, 6, !5, !7}, {1, 2, 3, 4, 5, !6, !7}}, {{4, 6, 7, !1, !2, !5}, {4, 7, !2}},
 {{2, 4, !3, !5, !7}, {2, 6, 7, !1, !3, !4, !5}},
 {{6, !1, !2, !3, !4, !5, !7}, {5, 6, 7, !1, !2, !3, !4}},
 {{1, 7, !2, !3, !4, !5, !6}, {1, 4, 5, 7, !2, !3, !6}},
 {{3, 4, 6, 7, !1, !2, !5}, {1, 3, 4, 5, 6, 7, !2}},
 {{4, 6, !1, !2, !3, !5}, {4, 6, !1}},
 {{2, 4, 7, !3, !5, !6}, {1, 2, 3, 4, 5, 7, !6}},
 {{2, 3, 4, 6, !5, !7}, {2, 3, 4, 5, 6, 7, !1}}}
```