

Spatial territory game of Hawk and Dove

F. Takasu
20170414

```
SetDirectory["/Users/takasu/Desktop/Doanh TLU/Spatial Hawk Dove  
[ディレクトリの設定  
game/spatial_hawk_dove/DerivedData/spatial_hawk_dove/Build/Products/Debug/  
"]  
/Users/takasu/Desktop/Doanh TLU/Spatial Hawk Dove  
game/spatial_hawk_dove/DerivedData/spatial_hawk_dove/Build/Products/Debug
```

Payoff matrix for the classical Hawk-Dove game

```
mat = {{v - (1 + a) / 2 s, v}, {v - s, v - s / 2}}  
  
{ {- 1 / 2 (1 + a) s + v, v}, {- s + v, - s / 2 + v}}  
  
{fitnessH, fitnessD} = mat.{p, 1 - p}  
  
{ (1 - p) v + p (- 1 / 2 (1 + a) s + v), p (- s + v) + (1 - p) (- s / 2 + v)}  
  
Solve[fitnessH == fitnessD, p]  
[解く  
  
{ {p -> 1 / a}}  
  
2  
2
```

Point pattern dynamics: Red is Hawk. Blue is Dove.

Definition

```
drawIndividuals[x_List, opt___] :=  
Block[{len = Length[x], data = {}, state, size}, Do[state = x[[i]][3];  
[ブロック [長さ [反復指定  
size = 0.01;  
If[state == 0,  
[If文  
AppendTo[data, {RGBColor[1, 0, 0], Circle[{x[[i]][1], x[[i]][2]}, size]}]];  
[追加割当て [RGBカラー [円  
If[state == 1, AppendTo[data, {RGBColor[0, 0, 1],  
[If文 [追加割当て [RGBカラー  
Circle[{x[[i]][1], x[[i]][2]}, size]}]], {i, 1, len}];  
[円  
Return[Show[Graphics[data], opt]]  
[戻る [示す [グラフィックス  
]
```

```
data =
  ReadList["hd-territory-F=2-RANDOM-ONE-alpha=GAUSS", Real, RecordLists → True];
  [式を読みリストで返す] [実… [記録のサブリスト [真
```

```
data = Map[ Partition[#, 4] &, data];
  [適用 [重複しないサブリストに分割
```

```
Length[data]
```

```
[長さ
```

```
100
```

```
data[[1]]
```

```
{{0.25, 0.25, 0., 0.000314}, {0.75, 0.75, 1., 0.000314}}
```

```
data[[3]]
```

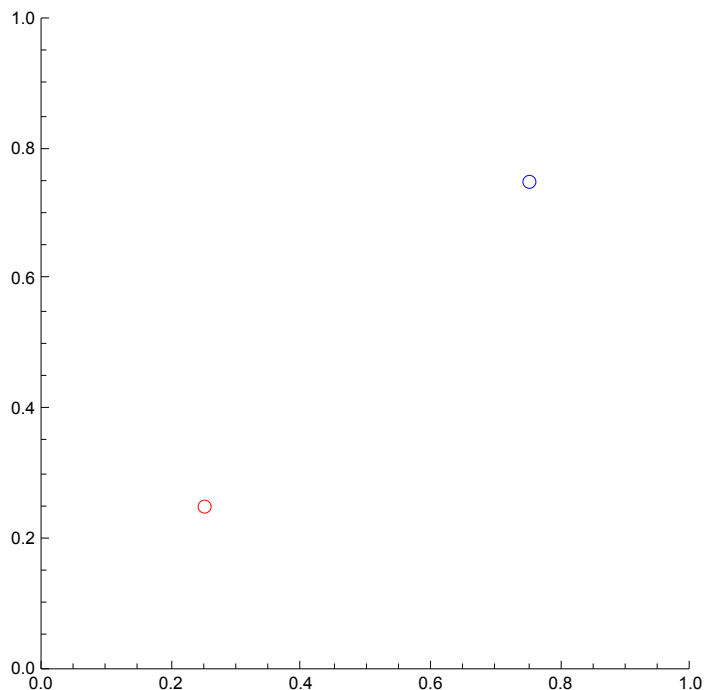
```
{{0.267472, 0.286162, 0., 0.000314}, {0.241675, 0.227892, 0., 0.000314},
 {0.268781, 0.213765, 0., 0.000314}, {0.248872, 0.253564, 0., 0.000314},
 {0.697762, 0.743859, 1., 0.000314}, {0.761237, 0.748911, 1., 0.000314},
 {0.839436, 0.736331, 1., 0.000314}, {0.824438, 0.740223, 1., 0.000314}}
```

```
drawIndividuals[data[[1]], AspectRatio → 1,
```

```
[縦横比
```

```
  Axes → True, PlotRange → {{0, 1}, {0, 1}}]
```

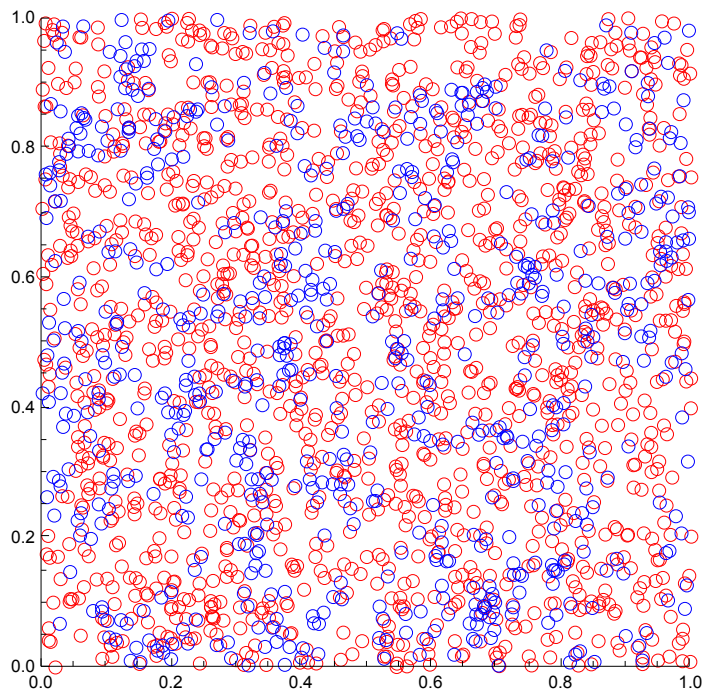
```
[軸… [真 [プロット範囲
```



```

drawIndividuals[Last[data], AspectRatio → 1,
               縦横比
               Axes → True, PlotRange → {{0, 1}, {0, 1}}]
               軸… 真 プロット範囲

```



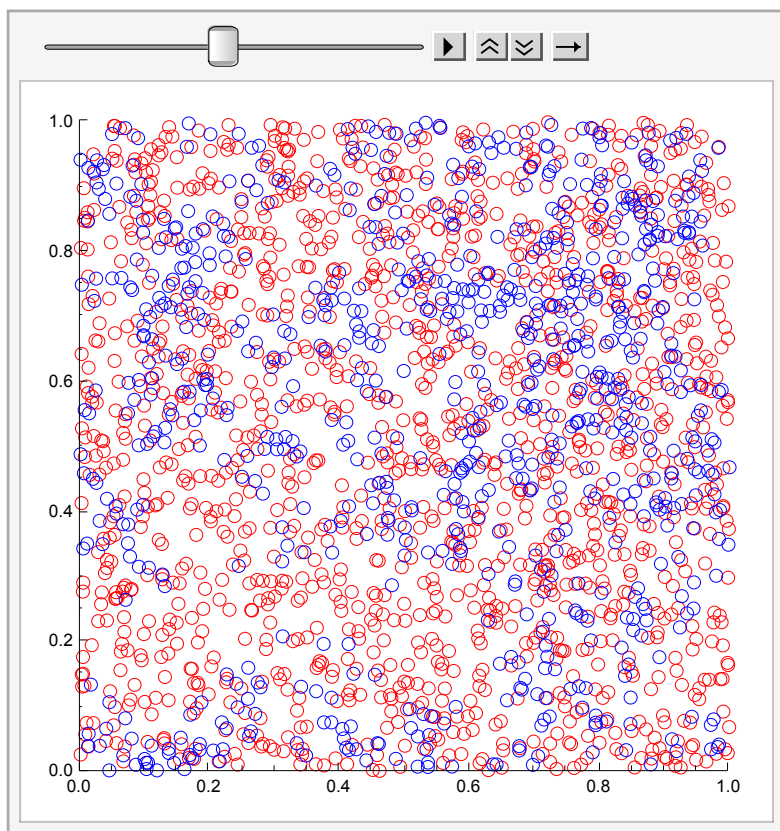
```

glist = {};
Do[
  反復指定
  g = drawIndividuals[data[[i]],
    AspectRatio → 1, Axes → True, PlotRange → {{0, 1}, {0, 1}}];
    縦横比 軸… 真 プロット範囲
  PrintTemporary[i];
  一時的にセルを出力
  Print[g];
  出力表示
  AppendTo[glist, g], {i, 1, Length[data], 1}
  追加割当て 長さ
]

```

```
ListAnimate[glist]
```

リストからアニメーション



```
Export["SHD.mov", glist]
```

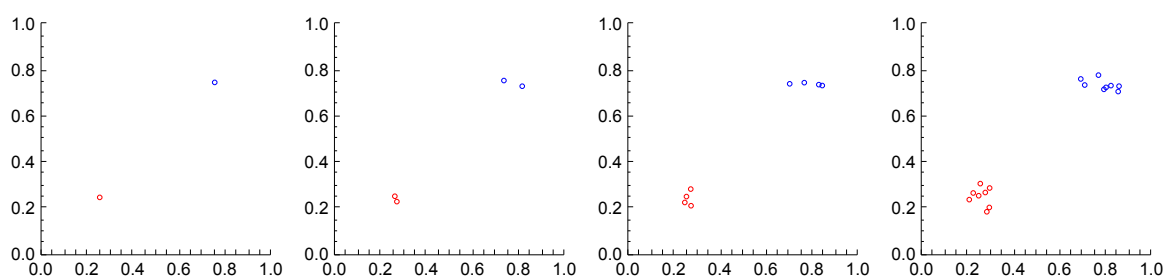
エクスポート

SHD.mov

```
Show[GraphicsRow[{glist[[1]], glist[[2]], glist[[3]], glist[[4]]}], ImageSize -> 600]
```

示す 1行のグラフィックス

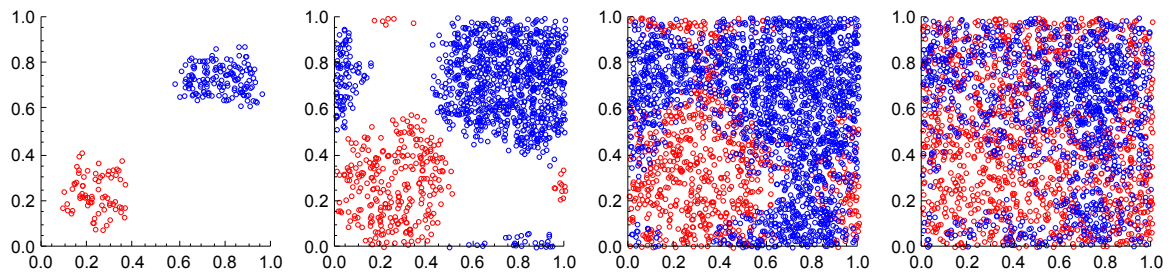
画像サイズ



```
Show[GraphicsRow[{glist[[10]], glist[[20]], glist[[30]], glist[[40]]}], ImageSize -> 600]
```

示す 1行のグラフィックス

画像サイズ



```
seqPopSize = Map[Length, data];
```

適用 長さ

```
ListPlot[seqPopSize, Joined -> True, PlotRange -> All, AxesLabel -> {"t", "Pop size"}]
```

リストプロット

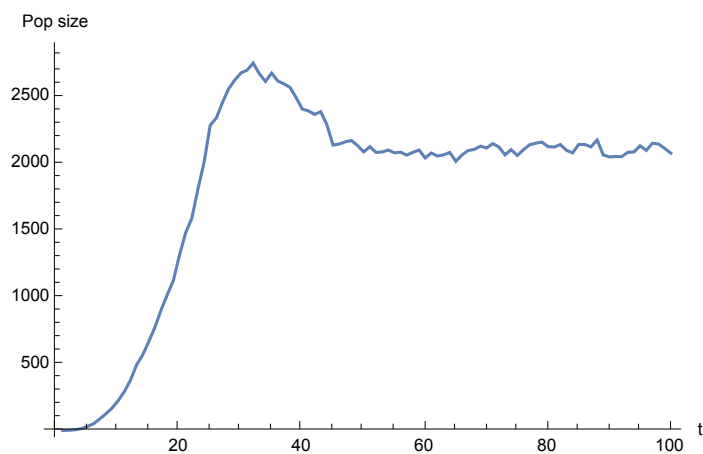
点の結合

真

プロット範囲

...

軸のラベル



```
N[1 / (Pi 0.01^2)]
```

数値 円周率

3183.1

```
N[1 / (4 × 0.01^2)]
```

数値

2500.

Define a function to pick up a designated strategy

```
pickUpAStrategy[data_List, strategy_] := Module[{len = Length[data], tmp = {}},
```

モジュール

長さ

```
Do[
```

反復指定

```
  If[data[[i]][[3]] == strategy, AppendTo[tmp, data[[i]]], {i, 1, len}
```

If文

追加割当て

```
];
```

```
Return[tmp];
```

戻る

```
]
```

```
pickUpAStrategy[data[[1]], 1.]
```

```
{{0.75, 0.75, 1., 0.000314}}
```

```
dataHawk = Map[pickUpAStrategy[#, 0.] &, data];
```

[適用]

```
dataDove = Map[pickUpAStrategy[#, 1.] &, data];
```

[適用]

```
seqHawk = Map[Length, dataHawk];
```

[適用] [長さ]

```
seqDove = Map[Length, dataDove];
```

[適用] [長さ]

```
ListPlot[seqHawk + seqDove, Joined → True, PlotRange → All]
```

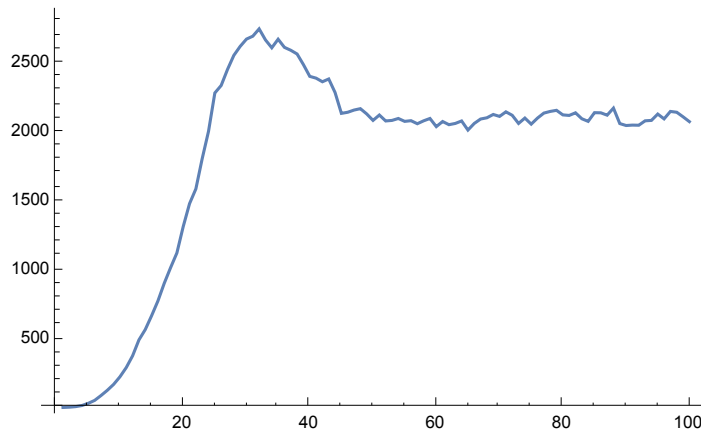
[リストプロット]

[点の結合]

[真]

[プロット範囲]

[すべて]



```
ListPlot[{seqHawk, seqDove}, Joined → True, PlotRange → All,
```

[リストプロット]

[点の結合]

[真]

[プロット範囲]

[すべて]

```
PlotStyle → {{Thickness[0.005], RGBColor[1, 0, 0]},
```

[プロットスタイル] [太さ]

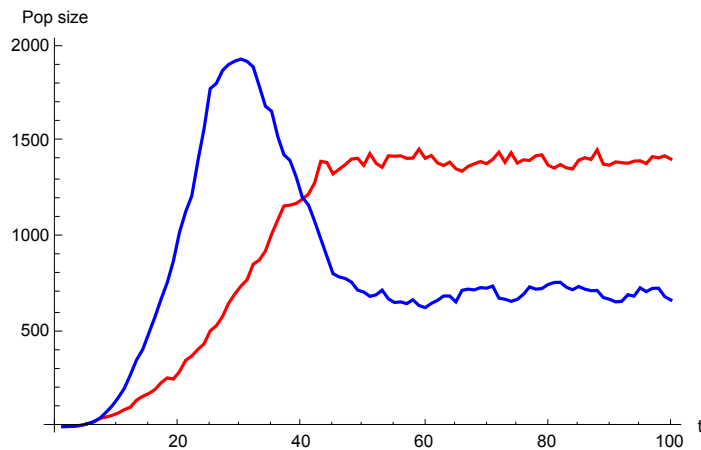
[RGBカラー]

```
{Thickness[0.005], RGBColor[0, 0, 1]}}, AxesLabel → {"t", "Pop size"}]
```

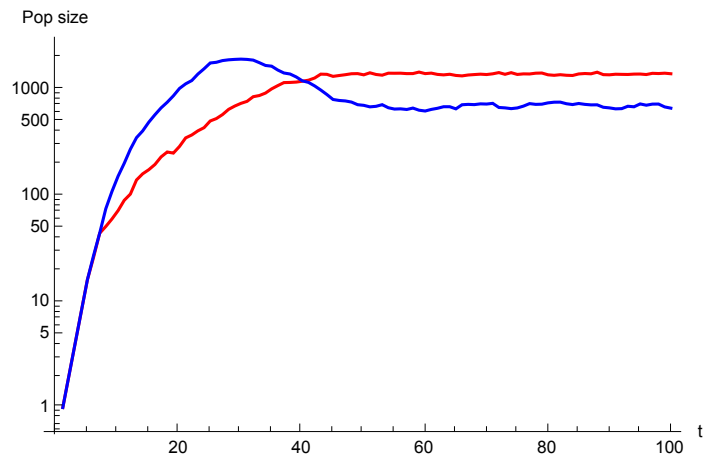
[太さ]

[RGBカラー]

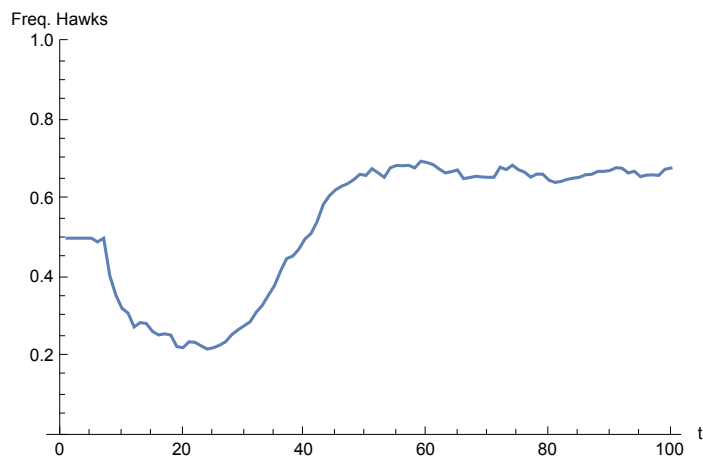
[軸のラベル]



```
ListLogPlot[{seqHawk, seqDove}, Joined → True, PlotRange → All,
  リストの対数プロット 点の結合 真 プロット範囲 すべて
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}, AxesLabel → {"t", "Pop size"}]
  プロットスタイル RGBカラー RGBカラー 軸のラベル
```



```
ListPlot[ $\frac{\text{seqHawk}}{\text{seqDove} + \text{seqHawk}}$ , Joined → True,
  リストプロット 点の結合 真
  PlotRange → {0, 1}, AxesLabel → {"t", "Freq. Hawks"}]
  プロット範囲 軸のラベル
```



2

2

Population dynamics of Hawks and Doves for CSR

Hawks and Doves are distributed as Completely Spatial Randomness

```
fnHawk = F (1 - 2 Pi R^2 (nH - 1)) nH;
  円周率
```

```
fnDove = F (1 - Pi R^2 / 2 (nD - 1) - Pi R^2 nH) nD;
  円周率 円周率
```

```
fnList = {fnHawk, fnDove}
```

```
{F nH (1 - 2 (-1 + nH) pi R^2), F nD (1 - 1/2 (-1 + nD) pi R^2 - nH pi R^2)}
```

```

para = {F → 2, R → 0.01};
fnListN = fnList /. para
{2 (1 - 0.000628319 (-1 + nH)) nH, 2 nD (1 - 0.00015708 (-1 + nD) - 0.000314159 nH)}

next[{nHx_, nDx_}] := fnListN /. {nH → nHx, nD → nDx}
next[{2, 2}]
{3.99749, 3.99686}

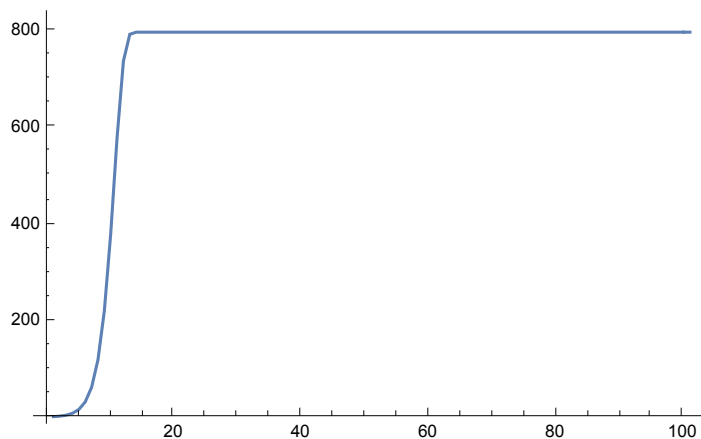
seq = NestList[next, {1, 1}, 100];
      リストのネスト
Last[seq]
      最後
{seqHawks, seqDoves} = Transpose[seq];
      転置
{796.775, 1590.55}

```

```

ListPlot[seqHawks, Joined → True, PlotRange → All]
      リストプロット      点の結合      真      プロット範囲      すべて

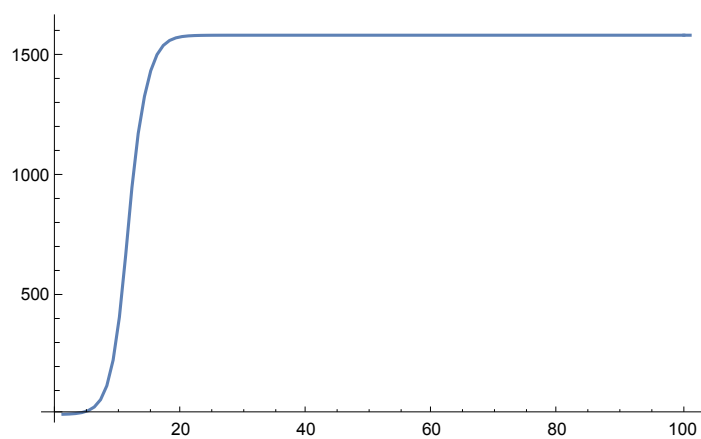
```



```

ListPlot[seqDoves, Joined → True, PlotRange → All]
      リストプロット      点の結合      真      プロット範囲      すべて

```




```
ListPlot[seqHawks / (seqHawks + seqDoves), Joined → True, PlotRange → {0, 1}]
```

リストプロット

点の結合

真

プロット範囲

