

# *Individual-Based Models of the Spread of Disease, Weeds, and Insects*

**Art Diggle  
Moin Salam  
Marta Monjardino**

Several simulations of the spread of organisms are presented using spatially explicit, individual-based stochastic models. These models have proven to be useful tools for predicting the behaviour of populations based on scientific understanding of the behaviour of individuals.

## ■ Introduction

We have constructed a number of simulation models of the spread of organisms across a landscape. These models are based on the spread of representative individual organisms. The models are quite simple but they are typically evaluated thousands or millions of times in a series of time-steps, often resulting in complex emergent behaviour. The models generally represent key features of the life cycle of the organism in question and one or more movement processes such as rain splash, dispersal by wind, human-mediated dispersal, or insect flight. In each spread event an individual moves along a path a distance that is chosen from a probability distribution. Both the path and the distribution of distances are functions of environmental or other parameters.

*Mathematica* is well suited to the development of this type of model because of its extensive function set, flexible programming style, and graphic capabilities. Generating complex probability surfaces or customised data structures is quite straightforward in *Mathematica*. It is often possible using functional programming approaches to produce elegant and efficient routines, but dyed-in-the-wool procedural programmers, such as ourselves, can always fall back on a procedural style when necessary.

## ■ Method

### □ Maps as Sparse Arrays

The state of these models at any time is represented as a *map*, which is a two-dimensional rectangular array of rectangular cells. Each cell contains information about the local environment and the population structure of the spreading organism. Usually the initial population of the organism in question is small but the area where spread may occur is large. Consequently it is desirable to represent the map as a sparse array. Data type definitions for a typical map follow.

The map includes a list of occupied cells, an array of indices that refer to the list of occupied cells, and a coordinate pair that specifies the physical size of the cells.

```
makeMap[m_mapArray, cells_cellList,  
        cellSize_realCoordinatePair] := map[m, cells, cellSize]
```

The size of the cell is defined by a pair of positive real numbers representing side lengths.

```
makeRealCoordinatePair[row_ /; 0 < row,  
                        column_ /; 0 < column] :=  
  realCoordinatePair[row, column]
```

**mapArray** is a rectangular array of nonnegative integers. Their positions in **mapArray** correspond to the physical location of that cell on the map. The integers in **mapArray** are the positions of the respective cells in **cellList**, and 0s denote unoccupied cells.

```
makeMapArray[  
  array_ /; MatrixQ[array,  
    IntegerQ[#] && NonNegative[#] &]] :=  
  mapArray @@ array
```

**cellList** is simply a list of cells. Cells are added to the cell list only when those cells become occupied.

```
makeCellList[cells___cell] := cellList[cells]
```

The cells contain the position of the cell in **mapArray** and information about the environment and population of the spreading organism. The example given here is for European House Borer, described in the Spread of Insects section. When cells are occupied, habitat information is generated based on a corresponding habitat map.

```
makeCell[coord_mapCoordinatePair, h_habitatProfile,  
          cohorts_cohortList, newMobileAdults_Integer] :=  
  cell[coord, h, cohorts, newMobileAdults]
```

The position of a cell in `mapArray` is defined by a pair of positive integers using the data type `mapCoordinatePair` which has special cases including `outOfBounds`. The functions `row` and `column` are selectors of both the `mapCoordinatePair` and `realCoordinatePair` data types.

```
makeMapCoordinatePair[row_Integer /; row > 0,
  column_Integer /; column > 0] :=
  mapCoordinatePair[row, column];

makeMapCoordinatePair[r_Integer, c_Integer,
  bounds_mapCoordinatePair] :=
  If[0 < r ≤ row[bounds] && 0 < c ≤ column[bounds],
    mapCoordinatePair[r, c],
    outOfBounds];

nowhere = mapCoordinatePair[-1, -1];
outOfBounds = mapCoordinatePair[0, 0];
```

The data type `cohortList` contains collections of the data type `cohort`.

```
makeCohortList[cohorts___cohort] := cohortList[cohorts];

cohort /: c1_cohort + c2_cohort := makeCohortList[c1, c2]
```

Upvalues for `cohortList` and `cohort` have been defined with respect to the operation `Plus`.

```
addCohort[cohorts_cohortList, c_cohort] :=
  Join[cohorts, makeCohortList[c]];

cohortList /: cList_cohortList + c_cohort :=
  addCohort[cList, c]

cohortList /: c_cohort + cList_cohortList :=
  addCohort[cList, c]
```

The `cohort` data type contains information on the time of infestation, number of habitats infested, total number of larvae present, and the number of adults that have been produced.

```
makeCohort[timeStepInfested_Integer,
  habitatsInfested_Integer, totalLarvae_Integer,
  exitHoles_Integer] :=
  cohort[timeStepInfested, habitatsInfested, totalLarvae,
    exitHoles]
```

In each time-step individuals will spread from each of the occupied cells. This type of data structure makes it easy to process only the occupied cells, ignoring the empty cells.

## □ Spread Along a Path

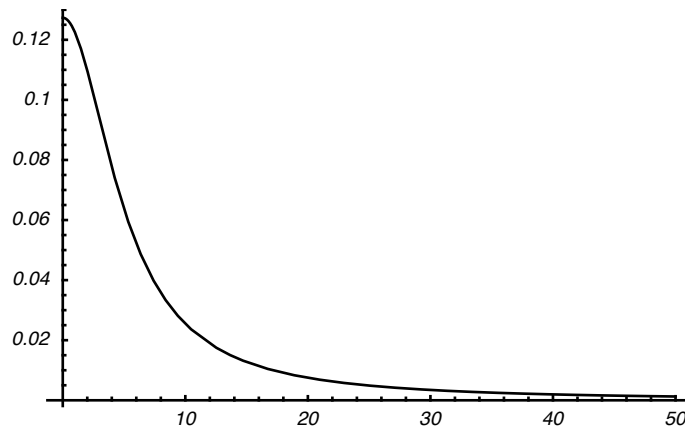
Movement of individual organisms occurs along a path and the distance travelled is chosen from a probability distribution. We often find it useful to use a half-Cauchy distribution for this purpose. The Cauchy distribution has a history of use in this role [1, 2, 3].

Spread distance for each organism that is moving is generated using the following function where `medianDistance` is the median spread distance.

```
spreadDistance[medianDistance_] :=  
  Abs[Random[CauchyDistribution[0, medianDistance]]]
```

The Cauchy distribution is a fat-tailed bell-shaped distribution.

```
Plot[2 PDF[CauchyDistribution[0, 5], x], {x, 0, 50},  
  PlotRange → All]
```



For the Cauchy distribution the probability that a spreading organism will come to rest per unit of distance, decreases with distance. For the following example where the median spread distance is 5 units, more than 6% of the organisms travel a distance greater than 50 units and more than 0.06% a distance greater than 5000 units.

```
TableForm[Table[  
  {5 × 10x,  
    N[1 - 2 (CDF[CauchyDistribution[0, 5], 5 × 10x] - 1 / 2)]},  
  {x, 0, 3}]]
```

5	0.5
50	0.063451
500	0.00636599
5000	0.00063662

This contrasts markedly with spread generated by an exponential distribution, where the probability that a spreading organism will come to rest per unit of distance is constant.

```
TableForm[Table[
  { $5 \times 10^x$ , N[1 - CDF[ExponentialDistribution[1 / 8],  $5 \times 10^x$ ]]},
  {x, 0, 3}]
```

```
5      0.535261
50     0.00193045
500     $7.18778 \times 10^{-28}$ 
5000    $3.68086 \times 10^{-272}$ 
```

For a straight radial path the following function produces the new location as a `realCoordinatePair` from the starting location, radial distance travelled, and angle of travel. This function is a straightforward conversion from polar to rectangular coordinates.

```
radialIncrement[origin_realCoordinatePair,
  radius_ /; 0 ≤ radius, angle_ /; -∞ ≤ angle ≤ ∞ :=
makeRealCoordinatePair[
  row[origin] + radius Cos[angle Degree],
  column[origin] + radius Sin[angle Degree]
```

The angle is in degrees.

```
row[realCoordinatePair[row_, column_]] := row

column[realCoordinatePair[row_, column_]] := column
```

## □ Structure of a Model

Models are typically structured as loops that cycle once per movement event. The loops execute procedures which operate on the map to spread organisms, advance the life cycle of organisms, or alter the environment. The example given here is for European House Borer which migrates once per generation of the insect.

```
aRun[generations_Integer /; generations ≥ 0] :=
Do[spreadAdults[populationMap, spreadParam, maxFlights,
  habitatDetectabilityParameter, effectiveEggsPerAdult,
  optimumMigrationSurvival];
populationMap = layEggs[populationMap, g,
  effectiveEggsPerAdult],
  {g, generations}]
```

populationMap is the map; spreadParam, maxFlights, habitatDetectabilityParameter, effectiveEggsPerAdult, and optimumMigrationSurvival are predefined parameters that affect the rate of buildup of populations and movement of individuals.

The functions spreadAdults and layEggs are both loops that operate on each occupied cell of the map.

```
spreadAdults[m_, spreadParam_, maxFlights_, detectability_,
multiplicationRate_Integer,
optimumMigrationSurvival_ /;
0 ≤ optimumMigrationSurvival ≤ 1] :=
Module[{originCell, origin, destination, emergenceData,
theMap = Evaluate[m], occupiedCells, cellCounter = 0},
occupiedCells = occupiedCellsOf[theMap];
Do[originCell = aCell[theMap, i];
origin = mapCoordinatePairOf[originCell];
emergenceData = adultEmergence[originCell,
multiplicationRate];
m = changeACell[theMap, i, cellOf[emergenceData]];
Do[destination =
spreadAnAdult[m, origin, spreadParam, maxFlights,
detectability];
If[inBounds[destination],
m = addAdults[Evaluate[m], destination, 1]],
{j, mobileAdultsOf[emergenceData]}];
theMap = Evaluate[m],
{i, occupiedCells}]]
```

spreadAdults has the attribute HoldFirst and modifies the map directly.

```
SetAttributes[spreadAdults, HoldFirst]
```

The function occupiedCellsOf is a selector of the map data type that returns the total number of occupied cells, aCell is a selector of the map data type that returns the cell in the cellList of the map corresponding to the index i, and mapCoordinatePairOf is a selector of the cell data type that returns the mapCoordinatePair of the cell. The function adultEmergence calculates the number of adult beetles that will emerge from the cell and updates the number of larvae living in the cell and the amount of remaining food resources in the cell. adultEmergence returns an instance of the data type adultEmergenceData which contains the updated cell and the number of mobile adults and has selectors cellOf and mobileAdultsOf.

```
makeAdultEmergenceData[c_cell,
mobileAdults_Integer /; mobileAdults ≥ 0] :=
adultEmergenceData[c, mobileAdults]
```

The function `changeACell` is an operator on the map data type that replaces the cell at position `i` in the `cellList` of the map. The function `spreadAnAdult` is evaluated for each mobile adult. It makes use of the function `flightPath` described in the Spread of Insects section and returns the `mapCoordinatePair` of the endpoint of the migration path. If the insect dies, `nowhere` is returned. New cells are added to the map during this process. The function `inBounds` returns `True` for any `mapCoordinatePair` that is not `nowhere` or `outOfBounds`. The function `addAdults` adds 1 to the `newMobileAdults` of the cell destination of the map.

The function `layEggs` calculates the number of remaining unoccupied habitats in each cell. The number of newly infested habitats is the least of the number of remaining uninhabited habitats and the number of mobile adults that have landed in the cell.

```
layEggs[theMap_map, when_Integer /; when ≥ 0,
  eggsPerAdult_Integer /; eggsPerAdult ≥ 0] :=
Module[{newCell, eggsLaid, successfulAdults,
  remainingHabitats,
  newMap = theMap, cellArea = areaOf[cellSizeOf[theMap]]},
Do[newCell = aCell[newMap, cellIndex];
If[newMobileAdultsOf[newCell] > 0,
  remainingHabitats =
    Round[cellArea habitatDensity[
      habitatProfileOf[newCell]]] -
    totalInfestedHabitats[cohortListOf[newCell]];
  successfulAdults = Min[newMobileAdultsOf[newCell],
    remainingHabitats];
If[succesfulAdults > 0,
  eggsLaid = succesfulAdults eggsPerAdult;
  newCell = makeCell[mapCoordinatePairOf[newCell],
    habitatProfileOf[newCell], cohortListOf[newCell] +
    makeCohort[when, succesfulAdults, eggsLaid, 0],
    0];
  newMap = changeACell[newMap, cellIndex, newCell]],
{cellIndex, occupiedCellsOf[newMap]}};
newMap]
```

The function `cellSizeOf` is a selector of the map data type. The function `areaOf` is a selector of the `realCoordinatePair` data type that returns the product of the coordinates. The functions `newMobileAdultsOf`, `habitatProfileOf`, and `cohortListOf` are selectors of the cell data type. The function `habitatDensity` is a selector of the `habitatProfile` data type, and the function `totalInfestedHabitats` is a selector of the `cohortList` data type that returns the sum of `habitatsInfested` from the cohorts.

## □ Output Using ListDensityPlot

The following is an example that illustrates the use of `ListDensityPlot` to produce plots of population density of an organism with an overlay of boundary lines imported from a Geographic Information System (GIS). The population density data represents a 300 by 350 grid of 50m square cells. The raw GIS Data was imported from an ASCII file using the function `Import["FILENAME.ASC", "Table"]`.

```
someRawGISData = {{6600}, {366257.42`, 6.29838801`*^6},
  {366282.65`, 6.29635546`*^6},
  {366307.33`, 6.29434453`*^6},
  {368316.41`, 6.29436477`*^6},
  {368292.21`, 6.29637653`*^6},
  {368269.52`, 6.29840962`*^6},
  {366257.42`, 6.29838801`*^6}, {6600},
  {372316.49`, 6.29824388`*^6},
  {372325.92`, 6.29744929`*^6},
  {374357.95`, 6.29747267`*^6},
  {374346.39`, 6.29847847`*^6},
  {373272.92`, 6.29846618`*^6},
  {373158.31`, 6.29844195`*^6},
  {373014.34`, 6.2983687`*^6},
  {372845.54`, 6.29824997`*^6},
  {372336.6`, 6.29824409`*^6},
  {372316.49`, 6.29824388`*^6}};
```

This function puts the data into a form recognisable by *Mathematica* as sets of coordinates.

```
propertyBoundariesMetres :=
  Select[Split[someRawGISData, Length[#1] == Length[#2] &],
    Length[#] > 1.5 &]
```

The function `metresToGrid` converts the coordinates from metres to grid intervals shifted to the appropriate location on the figure.

```
metresToGrid[point_] := Module[
  {startC = 363 255.25, startR = 6 304 528.34 - 344 50,
   cellSize = 50},
  {(point[[1]] - startC) / cellSize,
   (point[[2]] - startR) / cellSize}
]
```



This function produces a list of graphics primitives suitable for use as an Epilog in ListDensityPlot.

```
propertyBoundaryLines :=
  Flatten[{RGBColor[0, 0, 1], Thickness[0.01],
    Map[Line, Map[metresToGrid, propertyBoundariesMetres,
      {2}]]}]
```

This function produces a not very realistic table of population densities.

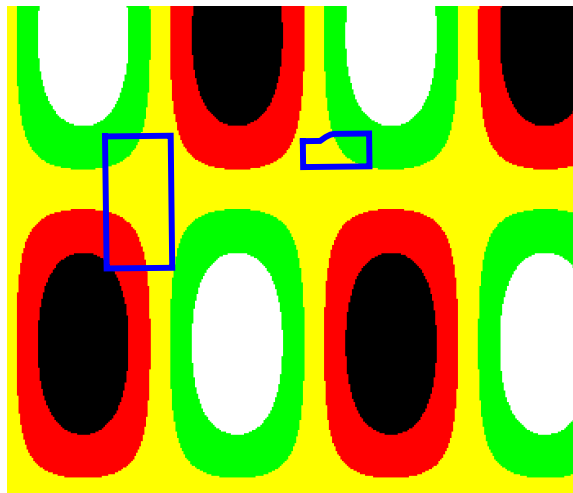
```
populationGrid := Table[2500 (1 + Sin[r / 60] Sin[c / 30]),
  {r, 300}, {c, 350}]
```

The function colourFn is the ColorFunction that will be used in ListDensityPlot.

```
colourFn[f_] := Which[f < 1000, RGBColor[1, 1, 1],
  f < 2000, RGBColor[0, 1, 0], f < 3000, RGBColor[1, 1, 0],
  f < 4000, RGBColor[1, 0, 0], True, RGBColor[0, 0, 0]]
```

The option AspectRatio is set to the ratio of the number of rows of the grid to the number of columns of the grid. The options PlotRange → All and ColorFunctionScaling → False are necessary so the data is not clipped or scaled.

```
ListDensityPlot[populationGrid,
  ColorFunction → (colourFn[#] &), PlotRange → All,
  AspectRatio →  $\frac{300}{350}$ , Mesh → False, Frame → False,
  ColorFunctionScaling → False,
  Epilog → propertyBoundaryLines]
```



## ■ Spread of Disease

In blackspot disease of field peas primary infection occurs by spread of fungal ascospores from infected residue of past pea crops. Maturation of the ascospores occurs at a rate that depends on weather conditions before and during the time that the pea crop is growing. Release of ascospores is triggered by rainfall, and the direction and median distance of spread are determined by the wind conditions following release. The number of ascospores produced during any triggering event depends on the stage of maturation of the ascospores and the age of the residue.

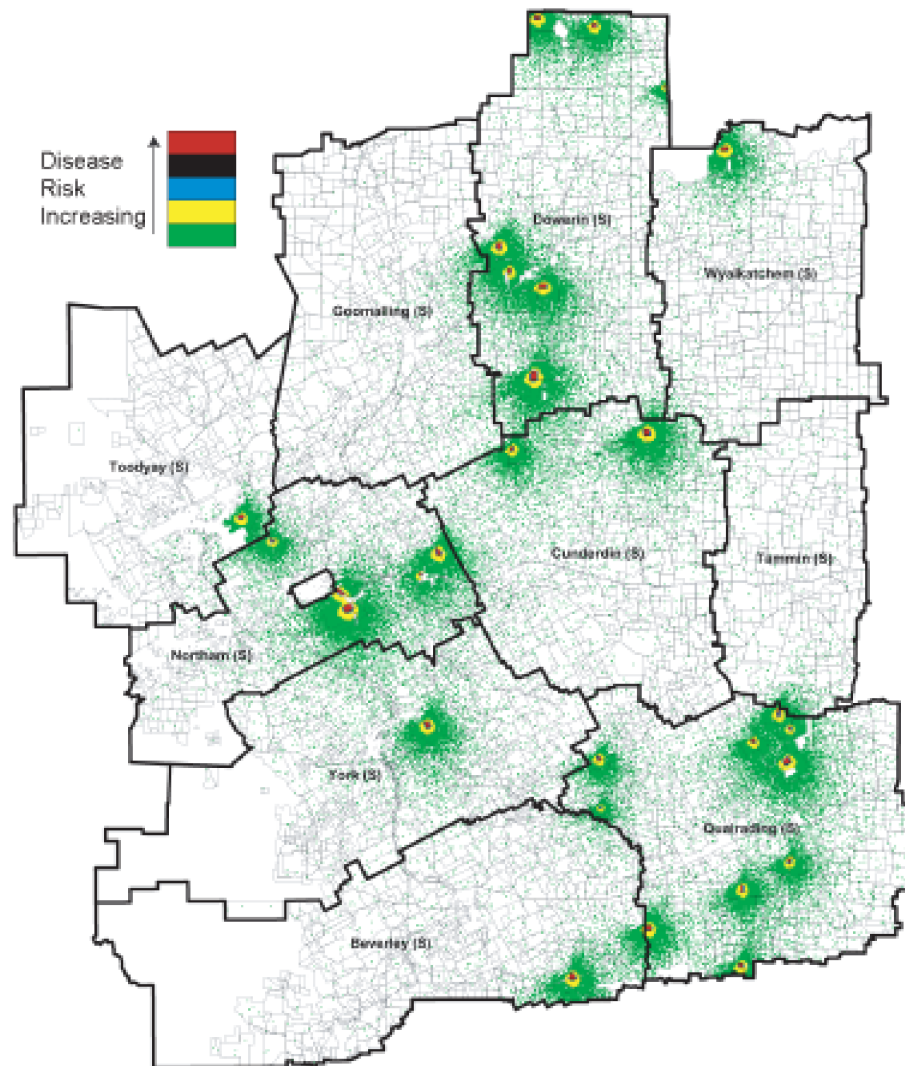
Our model of the spread of blackspot disease is driven by hourly wind and rainfall data, with a spread event occurring in any hour where the amount of rainfall exceeds a threshold. Spread occurs from each cell on the map where peas have been grown. The number of ascospores spread from each cell is the product of the absolute number produced and the probability that each of these ascospores will succeed in producing an infection if it lands on a pea crop. Cauchy distribution along a straight radial path is assumed with the median spread distance being a function of wind speed and the angle of spread being the wind direction.

Figure 1 was produced by the model using `ListDensityPlot`. It represents an approximately 40km square area of farmland near Esperance, Western Australia. The colours indicate the number of ascospores falling on each map cell. White denotes no ascospores, with the number of ascospores increasing in the order  $\text{green} \leq \text{yellow} \leq \text{grey} \leq \text{black}$ . The red and blue lines indicate field and property boundaries. They were produced by a GIS and superimposed on the figure as an `Epilog`.



▲ **Figure 1.** Simulated pattern of ascospore showers near Esperance, Western Australia.

Results from the model can also be exported to a GIS as was done for Figure 2, which represents several Western Australian Shires.



▲ **Figure 2.** Simulated pattern of ascospore in the Northam advisory district of Western Australia.

Models of this form are also suitable for simulating spread of other small wind-borne particles such as pollen.

## ■ Spread of Weeds

A variant of the model has been used to model spread of weeds by harvesting equipment. Numbers of seeds and growing plants for the weed and crop species are maintained for each of the infested cells of the map. This model is driven by a sequence of events which represent stages of the life cycle of the plants and activities of the farmer. These events include:

- germination—translates a proportion of the seeds into growing plants
- spray and tillage—reduces the number of growing plants
- seedset—produces new seed on the growing plants
- harvest—calculates the amount of weed seed that is removed with the crop seed and spreads the weed seed

The map used in this model represents a cropping field. The cell size used is set equal to the width of the harvesting equipment (10m for the example shown here) and the spread path is defined by a list of cell coordinates in the order that the harvester moves over them. Spread distances along the path of the harvester are generated from a Cauchy distribution as before, with an additional constant probability that seeds will move sideways into cells adjacent to the path of the harvester.

Figure 3 shows the simulated distribution of the weed Bedstraw in a 1km square field 11 years after an initial incursion of 100 randomly placed seeds. In each year the harvester followed a counter-clockwise spiral path from the lower right corner to the centre and then travelled diagonally back to the starting point. Shading is proportional to the density of the seed.



▲ **Figure 3.** Simulated pattern of seeds of the weed Bedstraw spread by harvesting equipment in a 1km square field.

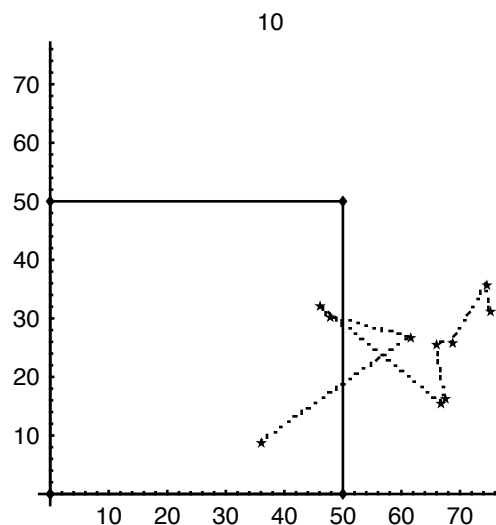
## ■ Spread of Insects

In the examples of seeds and fungal ascospores, the endpoint of each spread event is independent of the suitability of the environment for survival. Movement of insects often differs in that insects are attracted to suitable environments. Hence the endpoint of a movement event is influenced by the suitability of some or all of the cells of the map. European House Borer is a flying beetle that is attracted to pine wood on which it lays eggs and in which the larvae of the insect feed and grow. We have approximated the spread of a beetle as a series of random flights with a probability that the beetle will find a suitable environment at the end of each flight. The probability of finding a suitable environment is a function of the density of pine in the cell where the flight terminates. If unsuccessful, the insect will fly repeatedly up to a maximum number of flights after which it dies. As for the earlier examples the distance travelled in each flight is chosen from a Cauchy distribution.

A function using `NestWhileList` that produces a sequence of flights as a list of `realCoordinatePair` endpoints can be implemented as follows; the function `fly` returns a `realCoordinatePair` endpoint from a starting point and a median spread distance, and the function `habitatNotFound` returns `True` if a habitat is not found and `False` otherwise.

```
flightPath[m_map, start_realCoordinatePair,
  medianSpread_ /; 0 ≤ medianSpread,
  maxFlights_Integer /; 0 < maxFlights] :=
  NestWhileList[fly[#, medianSpread] &,
    start, habitatNotFound[m, #] &, 1, maxFlights]
```

Figure 4 shows a graph of the flight path for a beetle that starts at a random location in a 50m square cell of origin and finds a suitable habitat in the neighbouring cell after 10 flights.

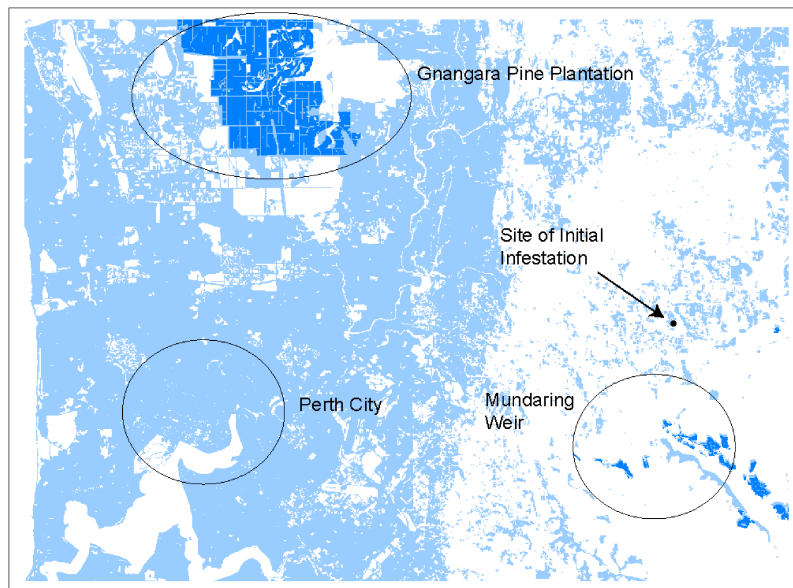


▲ **Figure 4.** Simulated flight path of European House Borer.

New infestations reduce the number of uninfested habitats in the cell and consequently reduce the probability that future infestations will occur in the cell.

In the following example, spread of European House Borer has been modelled in the region surrounding Perth, Western Australia. The median flight distance used was 5m, with a maximum of 10 flights per series. Figures 5 through 9 were generated using `ListDensityPlot`. Dark blue cells are in pine plantations and have a high density of suitable environments. Light blue cells have a low density of suitable environments, and white cells have no suitable environments. Black dots indicate locations of infested habitats and red dots indicate endpoints of a series of flights. The dots were superimposed on the figure as an `Epilog`. Labels, ovals, and arrows were added outside of *Mathematica*.

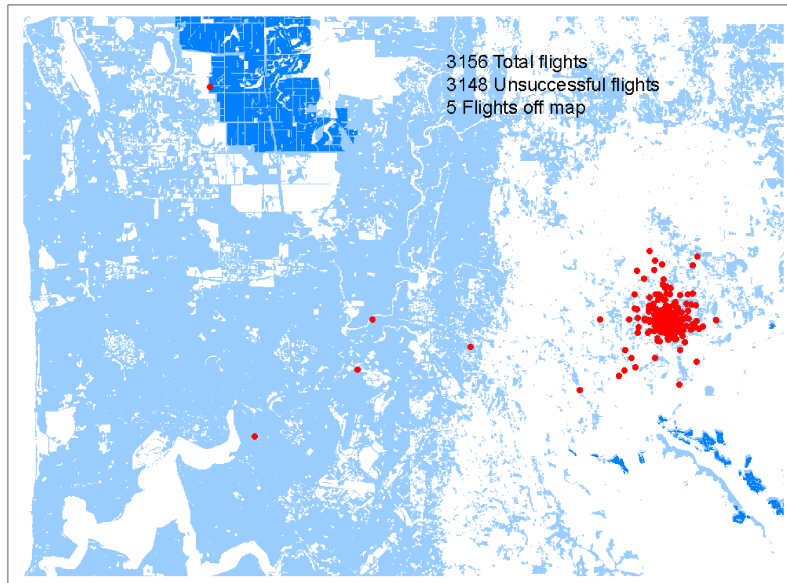
The simulation was initialised with a single beetle laying eggs into a suitable habitat at the indicated location (Figure 5).



▲ **Figure 5.** Site of initial infestation for simulated spread of European House Borer near Perth, Western Australia.

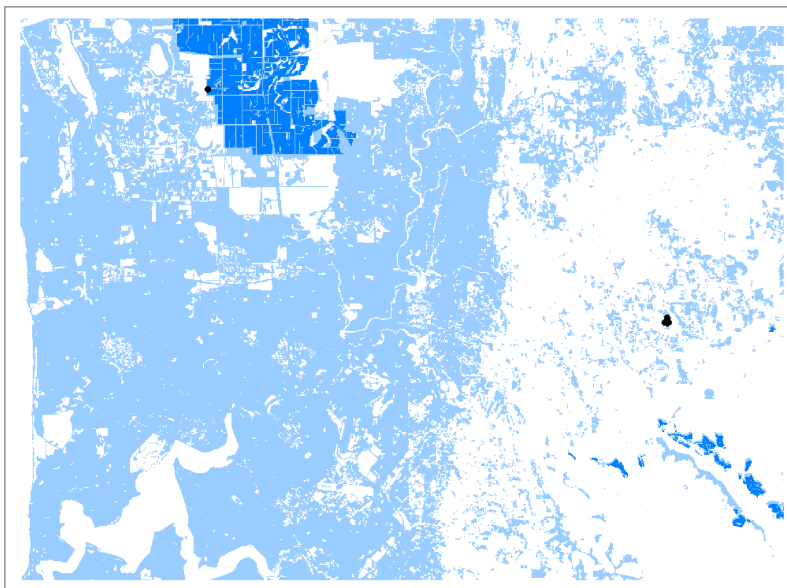
In early generations the insect population built up gradually. Most series of flights ended near the site of the original infestation, but in a few cases beetles travelled many kilometres. In Generation 5 one beetle landed on the edge of the Gnangara pine plantation (Figure 6). Total flights, Unsuccessful flights, and Flights off the map refer to series of flights rather than individual flights within a series.





▲ **Figure 6.** Endpoints of flights in Generation 5 of simulated spread of European House Borer near Perth, Western Australia.

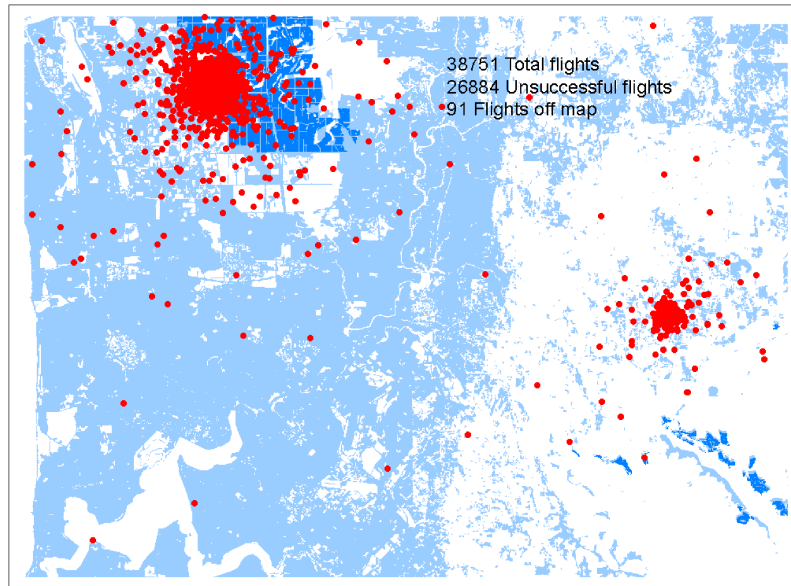
The landing on the edge of the Gnangara pine plantation produced the first successful infestation remote from the initial infestation (Figure 7).



▲ **Figure 7.** Sites of infestation after Generation 5 of simulated spread of European House Borer near Perth, Western Australia.

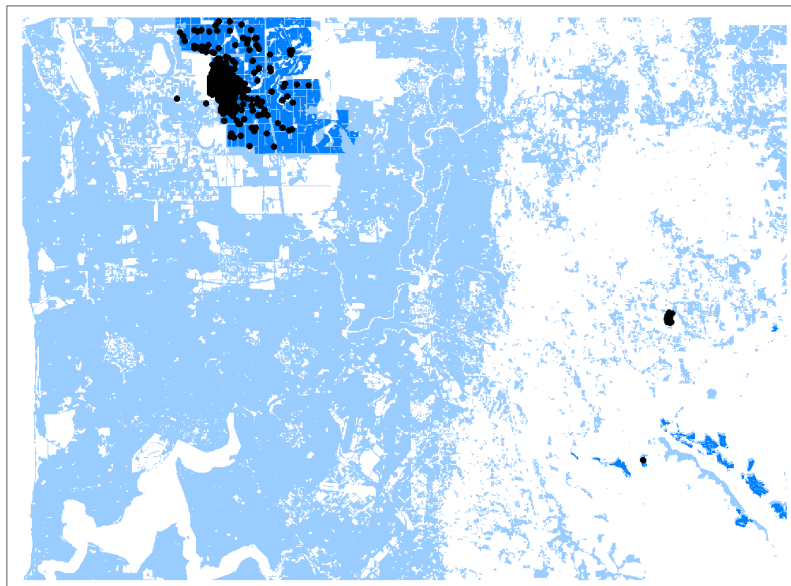
The insect population built up rapidly in the plantation and by Generation 9 most flights originated from the plantation (Figure 8).





▲ **Figure 8.** Endpoints of flights in Generation 9 of simulated spread of European House Borer near Perth, Western Australia.

By Generation 9 the borer was widely established within the plantation (Figure 9). New infestations had occurred in cells with a low density of suitable habitats near the plantation, and an infestation had occurred in one of the small plantations to the south of the site of the original infestation.



▲ **Figure 9.** Sites of infestation after Generation 9 of simulated spread of European House Borer near Perth, Western Australia.

## ■ Conclusion

The focus on the behaviour of individual organisms in these models is well suited to representing the knowledge of scientists that study the organisms. The scientists tend to make observations of individuals or small groups of organisms, and as humans, they tend to be efficient at thinking about mechanisms and generating hypotheses at that scale. These models are able to represent the key processes that affect movement of individuals. Aggregation of the behaviour of individuals to predict the behaviour of entire populations is then a matter of repeated calculation and is well suited to computers.

The models are very computation intensive, and the examples presented here have taken hours or days to calculate on personal computers. Fortunately, understanding the dynamics of movement of organisms tends to be most important when populations are building up from low levels, so it is possible to produce valuable analyses. It is no doubt possible to develop more efficient algorithms to estimate spread. Also these problems are parallel in nature and consequently are adaptable to calculation on clusters of computers using *gridMathematica*. For these reasons much larger analyses will be possible in future.

These models are already proving useful in the management of crop diseases and incursions of serious weeds and insect pests in Western Australia. For example in the case of blackspot disease of peas a strategy whereby the farmers of the district cooperate to sow peas in bands across the prevailing wind and plant the next year's peas upwind of this year's peas markedly reduces the overall disease levels. The model is currently being adapted to predict spread of genes in established weed populations and will be applied to manage the buildup of herbicide resistance in weeds.

## ■ References

- [1] A. J. Diggle, M. U. Salam, G. J. Thomas, H. A. Yang, M. O'Connell, and M. W. Sweetingham, "AnthracnoseTracer: A Spatiotemporal Model for Simulating the Spread of Anthracnose in a Lupin Field," *Phytopathology*, **92**(10), 2002 pp. 1110–1121.
- [2] M. W. Shaw, "Simulating Dispersal of Fungi Spores by Wind, and Resulting Patterns," in *Modelling in Applied Biology: Spatial Aspects (46)* (E. White, G. Russell, J. Benjamin, M. Mugleston, P. Brain, and P. Hamer, eds.), Wellesbourne, Warwick, England: The Association of Applied Biologists, 1996 pp. 165–172.
- [3] X. M. Xu and M. S. Ridout, "Effects of Initial Epidemic Conditions, Sporulation Rate, and Spore Dispersal Gradient on the Spatio-Temporal Dynamics of Plant Disease Epidemics," *Phytopathology*, **88**(10), 1998 pp. 1000–1012.

A Diggle, M. Salam, and M. Monjardino, "Individual-Based Models of the Spread of Disease, Weeds, and Insects," *The Mathematica Journal*, 2012. [dx.doi.org/10.3888/tmj.10.2-8](https://doi.org/10.3888/tmj.10.2-8).

## About the Authors

Art Diggle is a Senior Research Officer with the Department of Agriculture, Western Australia and a member of the Cooperative Research Centre for Australian Weed Management. He uses *Mathematica* primarily for modelling the spread of organisms and the selection of polygenic traits in plant populations.

Moin Salam is a Research Officer with the Department of Agriculture, Western Australia. He develops simulation models and computer-based tools for the management of crop diseases.

Marta Monjardino has until recently been a postdoctoral research fellow with the Cooperative Research Centre for Australian Weed Management based at the Department of Agriculture, Western Australia. She has developed simulation models of the spread of weeds and development of herbicide resistance in farming systems.

### **Art Diggle**

*Department of Agriculture, Western Australia  
3 Baron-Hay Court  
South Perth WA 6151, Australia  
adiggle@agric.wa.gov.au*

### **Moin Salam**

*Department of Agriculture, Western Australia  
P.O. Box 483  
Northam WA 6401, Australia  
msalam@agric.wa.gov.au*

### **Marta Monjardino**

*Consultant, Agricultural Economics  
7 Guilford Street  
Kensington Park SA 5068, Australia  
giesbertz@ozemail.com.au*