Estimating probabilities and trade-offs for decision analysis in invasive species eradication programs

SUPPLEMENTARY MATERIALS

The Red Imported Fire Ant (RIFA) model described in this paper consists of three components: (1) A spatio-temporal model to simulate the invasion spread and to produce an input dataset; (2) The logistic regression model that uses the input dataset to calculate model coefficients; and (3) The trade-off model that uses the logistic coefficients to generate isoquants. The code used to solve models (2) and (3) is provided below. This code reads the data from a text file (also provided in the online supplementary materials), runs the logistic regression and generates isoquants. The spatio-temporal model used to solve step (1) for the RIFA case study is described briefly and the steps used to generate the datasets are explained.

It is important to note that the input dataset can be generated from field data or from invasion models for other species or locations. As long as the dataset has the proper format it can be processed by the model provided below. In other words, step (1) is specific to the invasion of interest, whereas steps (2) and (3) are generic and can be applied to any dataset. The three models were implemented in the Matlab language (The Mathworks, 2005).

1. Spatio-temporal model (Step 1)

In the RIFA model the landscape is represented by a rectangular square lattice of dimensions \( n_r \) (rows) by \( n_c \) (columns) containing \( n = n_r \times n_c \) cells of one hectare each. The map layers used in the model include:

- habitat suitability, with values ranging between 0 and 1 (Figure S1a);
- restricted area and the region of interest (ROI) (Figure S1b);
- land type: urban=1, rural=0 (Figure S1c);
- treatment type: (air, ATV, foot) (Figure S1d);
- number of and colonies present at any time \( t \) in each cell;
- number of detections (passive, active and follow up) in each cell;
- number of searches in each cell (Figure S1e);
- number of treatments in each cell (Figure S1f).

An invasion is initialised based on previous detections. A set of ant colonies is randomly created based on a probability map generated using the spread model of Schmidt et al. (2010). Each time period (year) \( t \), the landscape is searched and treated according to a protocol defined through an experimental design record. An experimental design consists of a set of experiments, where each experiment represents a Monte Carlo run of the model with 500 iterations. The control variables and parameters (Table 1 in the main text) remain constant within a given experiment. The budget is represented as the amount of surveillance (S) available per year as a proportion of the region of interest (ROI), which consists of 454,135 ha.
In each simulation, the colony map is updated at the start of each period (year) based on the colonies that were eliminated in the previous time step. Local growth and dispersal occurs in the remaining colonies. Dispersal occurs from the remaining colonies and from treated colonies before those colonies were treated. The proportion of treated colonies that produced reproductive offspring prior to treatment is referred to as reproductive escape ($\gamma_E$). The growth-dispersal model (Schmidt et al., 2010) has seven parameters, contained in a row vector $\theta$ (Table S1). The particular realisation of $\theta$ for a simulation run is drawn from a multivariate distribution generated through Bayesian simulation (Schmidt et al., 2010). At the
start of each run two parameter vectors are selected randomly from this distribution. The values for the government agency \((\theta_A)\) in charge of the control program are selected separately from the ‘true’ values \((\theta_T)\). The probability map created by the agency uses parameter set \(\theta_A\) to allocate search and treatment at \(t+1\) based on the location of colonies found at \(t\), whereas the probability map used to generate new infestations at \(t+1\) in the “real” world uses parameter set \(\theta_T\), hence introducing knowledge uncertainty. The probability that a propagule will survive and establish a new colony depends on the habitat suitability of the cell where it lands. Details are given in Schmidt et al. (2010).

Table S1: Details of posterior probability distributions for model parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Interpretation</th>
<th>Mode</th>
<th>Mean</th>
<th>Stand. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta_1)</td>
<td>East-west spread</td>
<td>2.35</td>
<td>2.34</td>
<td>0.32</td>
</tr>
<tr>
<td>(\theta_2)</td>
<td>Reproductive rate (number of new nests produced per nest)</td>
<td>1.91</td>
<td>2.01</td>
<td>0.12</td>
</tr>
<tr>
<td>(\theta_3)</td>
<td>Outliers</td>
<td>0.0000010</td>
<td>0.000009</td>
<td>0.000005</td>
</tr>
<tr>
<td>(\theta_4)</td>
<td>Habitat effect</td>
<td>1.40</td>
<td>1.39</td>
<td>0.015</td>
</tr>
<tr>
<td>(\theta_5)</td>
<td>North-south spread</td>
<td>4.97</td>
<td>4.74</td>
<td>0.67</td>
</tr>
<tr>
<td>(\theta_6)</td>
<td>Urban effect</td>
<td>0.24</td>
<td>0.24</td>
<td>0.03</td>
</tr>
</tbody>
</table>

The Materials and Methods section in the main text describes the steps of a simulation run. The outputs produced by the model include:

- Maps through time of number of colonies, detections, searches and treatments.
- Time series of colonies by urban / rural areas for each iteration.
- Time series of searches by urban / rural areas and by active / follow-up modes for each iteration.
- Time series of treatments by urban / rural areas for each iteration.
- Time series of detections by urban / rural areas and by passive / follow-up / active modes for each iteration.

The results of each Monte Carlo run were summarized as a row in the dataset by indicating the number of runs, out of 500, that resulted in eradication after 10 years for the given management strategy. The full dataset (Table S2) was then used to create the design matrix for the logistic regression (eq. 3). The dataset created in this way (DATA.txt) is available in the Supplementary Online Materials.

Note that as long as the input dataset has the structure shown in Table S2, there are no restrictions on the source of the data. The spatio-temporal approach used here is one of many possible ways of generating the dataset, which could also come from field trials.
Table S2. Structure of the Input Dataset used in step (2)

<table>
<thead>
<tr>
<th>obs</th>
<th>S</th>
<th>T</th>
<th>γ_S</th>
<th>γ_T</th>
<th>γ_R</th>
<th>γ_E</th>
<th>n_erad</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.01</td>
<td>0.8</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>0.03</td>
<td>0.8</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.05</td>
<td>0.8</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>0.07</td>
<td>0.8</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>118</td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td>0.09</td>
<td>0.8</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>184</td>
</tr>
<tr>
<td>6</td>
<td>0.03</td>
<td>0.01</td>
<td>0.8</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>0.03</td>
<td>0.03</td>
<td>0.8</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Logistic model (Step 2)

The logistic regression model is explained in the main text in equations 1 to 3. The implementation of this model in Matlab is presented in Section 4 below (lines 17-26 of code). The Matlab Statistics Toolbox (The Mathworks 2013) is required to run this code, but note that the same results can be obtained with any statistical package capable of running logit regressions using the dataset provided. The main output required from this analysis is the coefficient vector β (represented by B in the code). The output of this code is presented in Figure S2.

Figure S2. Output of the logit analysis in Matlab (lines 23-26 of code)
3. Trade-off model (Step 3)

The trade-off model uses search algorithms to derive isoquants between pairs of variables, the main code is presented in lines 35-59 of section 4. The remaining code (lines 63-113) presents supporting functions to estimate isoquants based on the logistic model. Figure S3 shows the output from this code.

![Eradication probability isoquants](image)

Figure S3. Isoquants for surveillance against reproductive escape.

4. Matlab Code

The code presented below can be copied and pasted into the Matlab environment to create the Script (Plot_Isoquant.m) and functions (Get_Isoquant.m and LogisticFunc) that will solve the model and produce figures S2 and S3.
% Plot_Isoquant.m
% Runs logistic regression and produces isoquants
% Read dataset from ASCII file and define variables
load DATA.txt -ascii;
clear
obs = DATA(:,1);
S = DATA(:,2);
T = DATA(:,3);
gS = DATA(:,4);
gT = DATA(:,5);
gR = DATA(:,6);
gE = DATA(:,7);
n_erad = DATA(:,8);
n_tested = ones(length(n_erad),1)*500;
clear DATA
%% Quadratic logit
x = [S, T, gS, gT, gR, gE]; % linear terms
x = [x, x.*x, x(:,1).*x(:,2)]; % add quadratic terms and interactions
[B,dev,stats] = glmfit(x,[n_erad, n_tested], 'binomial', 'link', 'logit');
ypred = glmval(B,x, 'logit');
y = n_erad ./ n_tested;
% Fitness Plot
figure(1); plot(y,y, y,ypred, '.');
xlabel('Observed eradication probability');
ylabel('Predicted eradication probability');
title('Logistic model fitness');
for i = 1:length(ptarget)
[Y,P] = GetIsoquant(xtest, B, ptarget(i), v1, v2);
iso{i} = Y;
end
% Plot Isoquants
figure(2);
hold on
for j = 1 : length(iso)
plot(iso{j}(1,:), iso{j}(2,:), 'k');
text(iso{j}(1,1), iso{j}(2,1), num2str(ptarget(j)), 'FontSize',8, 'Color','blue');
end
grid
xlabel(vnames(v2));
ylabel(vnames(v1));
title('Eradication-probability isoquants');
hold off
%
function [y, p_hat] = GetIsoquant(x, B, ptarget, v1, v2)
% Returns the values y on the isoquant with probability ptarget for
% variables in positions v1 and v2 within the control vector x.
% B is a vector of estimated parameters using logistic regression
% INPUTS:
%   x: vector of initial x values [S, T, gS, gT, gR, gE]
%   B: logit coefficient vector
%   ptarget: vector of probability targets
%   v1, v2: variables to be used in isoquant
% OUTPUTS:
%   y: matrix of isoquant values [Surv RDP]
%   p_hat: predicted eradication probability
%
x0 = x;

npts = 50;
y2 = linspace(0, 0.5, npts)';
y = zeros(npts, 2);
p_hat = zeros(npts, 1);
for i = 1:npts
    u0 = 0.2;
    z = fsolve(@ObjFn, u0);
    y(i, 2) = y2(i);
    y(i, 1) = z;
    x = x0;
    x(v2) = y2(i);
    x(v1) = z;
    p_hat(i) = LogisticFunc(x, B);
end
invalid = find(abs(p_hat - ptarget) > 0.000001 | y(:, 1) < 0 | y(:, 1) > 1);
y(invalid, :) = [];
p_hat(invalid, :) = [];

function g = ObjFn(u)
    x = x0;
    x(v2) = y2(i);
    x(v1) = u;
    pe = LogisticFunc(x, B);
    g = pe - ptarget;
end

function [p] = LogisticFunc(x, b)
% Quadratic logistic function with interaction
x = [x, x.^2, x(:, 1) .* x(:, 2)];
% add quadratic and interaction terms
n = size(x, 1);
x = [ones(n, 1), x];
p = exp(x*b) ./ (1 + exp(x*b));

end
References

