%THRESHERLL finds the vector of threshhold patch fitnesses that are

%acceptance criteria during natal dispersal and runs simulations. It then

%alters one or more parameters in response to HIREC and runs simulations

%using the pre-HIREC thresholds. Finally, it finds new thresholds for the

%new post-HIREC parameters and runs more simulations.

tic

Fx = 100; %100; Top end of the patch fitness beta distribution, fitness

a = 4; %4; Alpha parameter of the beta distribution, dimensionless

b = 4; %4; Beta parameter of the beta distribution, dimensionless

n = 30; %30; Number of time steps in a season, dimensionless

h = 0.5; %0.5; Fraction of time steps in which a patch is found, d'less

s = 0.9;%0.98; Chance of surviving a time step, dimensionless

del = 3; %1; Deferred cost of searching per time step, fitness

c = 1; %1; Deferred cost exponent to express non-linearity

FxH = 75; %Fx; Top end of the patch fitness beta distribution, fitness

aH = a; %a; Alpha parameter of the beta distribution, dimensionless

bH = b; %b; Beta parameter of the beta distribution, dimensionless

nH = n; %n; Number of time steps in a season, dimensionless

hH = 0.25; %h; Fraction of time steps when a patch is found, d'less

sH = s; %s; Chance of surviving a time step, dimensionless

delH= del; %del; Deferred cost of searching per time step, fitness

cH = c; %c; Deferred cost exponent to express non-linearity

res = 1000; %1000; Resolution of the continuous graphs and functions

m = 1000; %1000; Number of individuals simulated

seePre = 1; %1; Show simulated individual results pre-HIREC, 0=no, 1=yes

seeMix = 1; %1; Show simulated individual results pre/post mix, 0=no, 1=yes

seePost= 1; %1; Show simulated individual results post-HIREC, 0=no, 1=yes

ETS = zeros(1,3); %Expected time to settle or die for each case

%Calculate functions for pre-HIREC parameters

[p q beta] = BetaVec(Fx,a,b,res); %p is res res'n p vector for Feta

%q is res res'n psi vec for FetaP

I = zeros(1,n); %Vector of threshold psi values

F = zeros(1,n); %Vector of threshold patch fitness values

P = zeros(1,n); %Vector of p\* values

for i = n:-1:1 %Track backwards through the time steps

prod = 1; %This accumulates a multiplier term

for j = i:n-1 %This sums terms for each i

if j > i %The multiplier constraint

prod = prod\*(1-h\*P(j));

end

step = s^(j-i+1)\*h\*P(j+1)\*(I(j+1)-((j+1)^c - i^c)\*del)\*prod;

if step < 0

step = 0;

end

F(i) = F(i) + step;

end

frac = F(i)/Fx; %F is fraction frac of interval

if frac > 1

frac = 1;

end

ii = ceil(res\*frac); %Index of 1/res piece containing F(j)

if ii == 0 %Avoid having the index go to zero

ii = 1;

end

jj = res\*frac - floor(res\*frac); %Piece to subtract off

P(i) = p(ii) - (beta(ii)/res)\*jj; %P is p(ii) minus the piece

I(i) = q(ii) - (beta(ii)/res)\*((ii-0.5)/res)\*Fx\*jj; %Same

end

prod = 1; %This block calculates fitness for the entire 1 to n sequence

F00 = 0; %This variable sums fitness

for j = 0:n-1

if j > 0

prod = prod\*(1-h\*P(j));

end

step = s^(j+1)\*h\*P(j+1)\*(I(j+1)-((j+1)^c)\*del)\*prod;

if step < 0

step = 0;

end

F00 = F00 + step;

end

%ETS (expec tm to settle) & ESN (expec surv tm if not settled): F & HIREC

ETS(1) = 1;

prod = 1; %ETS accumlates time surviving but not settling

for kk = 2:nH

prod = prod\*(1-h\*P(kk-1)); %Chance of not settling steps 1 -> kk-1

ETS(1) = ETS(1) + s^kk\*prod; %Chance surv kk steps+didn't settle kk-1

end

%Simulate with pre-HIREC parameters

Sim = zeros(m,2); %Matrix to record simulation results

Cumbeta = zeros(1,res); %The cumulative beta function

Cumbeta(1) = beta(1)/res; %Each slice has height beta(i) and width 1/res

for i = 2:res

Cumbeta(i) = Cumbeta(i-1) + beta(i)/res; %Accumulate them

end

for mm = 1:m %m is the total number of individuals simulated

flag = 1; %Signals that threshold not yet surpassed

i = 1; %Initialize the time step

while flag&&(i<=n)

mort = rand;

if mort > s %Individual dies

Sim(mm,1) = 0; %Fitness set to 0

Sim(mm,2) = i; %Time set to the time step number

flag = 0; %Signal that individual has finished

else %Individual lives and seeks patch

patch = rand; %Random check for habitat

if patch < h %patch is found

FF = FDraw(Cumbeta,Fx,res); %Draw patch fitness

if FF >= F(i) %Fitness >= threshhold

Sim(mm,1) = FF-(i^c)\*del; %Patch fitness - delay cost

Sim(mm,2) = i; %Completion time is step #

flag = 0; %Signal completion

end

end

end

i = i+1; %Increment steps

end

if (i > n)&&flag %If individual doesn't settle within n steps

Sim(mm,1) = 0; %Its fitness is zero

Sim(mm,2) = n; %Its time is set to n

end

end

FitmeanAndStepmean = mean(Sim,1); %MATLAB takes means of fitness & time

EmFitMean = FitmeanAndStepmean(1,1) %Simulation mean fitness

EmStopMean = FitmeanAndStepmean(1,2) %Simulation mean time to settle or die

%Calculate functions for the adapted post-HIREC world\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[pH qH betaH] = BetaVec(FxH,aH,bH,res);%p=p vector @ res res'n for Feta

%q= psi vec @ same res for FetaP

FH = zeros(1,nH); %Vector of threshold patch fitness values

PH = zeros(1,nH); %Vector of p\* values

IH = zeros(1,nH); %Vector of psi values

for i = nH:-1:1 %Track backwards through the time steps

prod = 1; %This accumulates a multiplier term

for j = i:nH-1 %This sums terms for each i

if j > i %The multiplier constraint

prod = prod\*(1-hH\*PH(j));

end

step = sH^(j-i+1)\*hH\*PH(j+1)\*(IH(j+1)-((j+1)^cH - i^cH)\*delH)\*prod;

if step < 0

step = 0;

end

FH(i) = FH(i) + step;

end

frac = FH(i)/FxH; %F is fraction frac of interval

if frac > 1

frac = 1;

end

ii = ceil(res\*frac); %Index of 1/res piece containing F(j)

if ii == 0 %Avoid having the index go to zero

ii = 1;

end

jj = res\*frac - floor(res\*frac); %Piece to subtract off

PH(i) = pH(ii) - (betaH(ii)/res)\*jj; %P is p(ii) minus the piece

IH(i) = qH(ii) - (betaH(ii)/res)\*((ii-0.5)/res)\*FxH\*jj; %Same

end

prod = 1; %This block calculates fitness for the entire 1 to n sequence

F00H = 0; %This variable sums fitness

for j = 0:nH-1

if j > 0

prod = prod\*(1-hH\*PH(j));

end

step = sH^(j+1)\*hH\*PH(j+1)\*(IH(j+1)-((j+1)^cH)\*delH)\*prod;

if step < 0

step = 0;

end

F00H = F00H + step;

end

%ETS (expec tm to settle) & ESN (expec surv tm if not settled): FH & HIREC

ETS(3) = 1;

prod = 1; %ETS accumlates time surviving but not settling

for kk = 2:nH

prod = prod\*(1-hH\*PH(kk-1)); %Chance of not settling steps 1 -> kk-1

ETS(3) = ETS(3) + sH^kk\*prod; %Chance surv kk steps+didn't settle kk-1

end

%The pre-HIREC and post-HIREC fitness distributions

bet = zeros(1,res); %Intermediate step in the beta distribution calc'n

betH = zeros(1,res);

beta = zeros(1,res); %Beta distribution

betaH= zeros(1,res);

x = 0.5/res:1/res:(res-0.5)/res; %Increments along (0,1) for beta

xx = x.\*Fx; %x-axis transformed for fitnesses in [0,Fx]

yy = x.\*FxH; %for fitnesses in [0,FxH]

bet = (x.^(a-1)).\*((1-x).^(b-1)); %Unnormalized beta function

betH = (x.^(aH-1)).\*((1-x).^(bH-1));

sum = 0; sumH = 0;

for i = 1:res

sum = sum + bet(i)./res; %Adding all unnormalized values

sumH = sumH + betH(i)./res;

end

beta = bet./sum; %Normalizing so that the area = 1

betaH = betH./sum;

Feta = beta./Fx; %Adjusting the distribution for [0,Fx]

FetaH = betaH./FxH;

maxF = max(Feta);

maxFH = max(FetaH);

if maxF > maxFH

maxx = maxF;

else

maxx = maxFH;

end

figure

hold on

plot(xx,Feta,'k') %Plot the fitness distribution

plot(yy,FetaH,'r')

xlabel('Patch fitness')

ylabel('Frequency density of patch fitnesses')

axis([0 Fx 0 maxx])

hold off

%Calculate functions with pre-HIREC thresholds but post-HIREC parameters

PX = zeros(1,nH); %p values for pre-H thresh but post-H pms

IX = zeros(1,nH); %I values for pre-H thresh but post-H pms

for i = 1:nH; %Begin FbarX calculation

frac = F(i)/FxH; %Get F as frac of FxH-F0H

if frac > 1

frac = 1;

end

ii = ceil(res\*frac); %Calculate fine adjustments

if ii == 0

ii = 1;

end

jj = res\*frac-floor(res\*frac);

PX(i) = pH(ii)-(betaH(ii)/res)\*jj; %Calc PX & IX via F & post-H params

IX(i) = qH(ii)-(betaH(ii)/res)\*((ii-0.5)/res)\*FxH\*jj;

end

prod = 1; %This block calculates fitness for the entire 1 to n sequence

F00X = 0; %This variable sums fitness

for j = 0:nH-1

if j > 0

prod = prod\*(1-hH\*PX(j));

end

step = sH^(j+1)\*hH\*PX(j+1)\*(IX(j+1)-((j+1)^cH)\*delH)\*prod;

if step < 0

step = 0;

end

F00X = F00X + step;

end

%ETS (expec tm to settle) & ESN (expec surv tm if not settled): FX & HIREC

ETS(2) = 1;

prod = 1; %ETS accumlates time surviving but not settling

for kk = 2:nH

prod = prod\*(1-hH\*PX(kk-1)); %Chance of not settling steps 1 -> kk-1

ETS(2) = ETS(2) + sH^kk\*prod; %Chance surv kk steps+didn't settle kk-1

end

%Simulation pre/post follows

SimX = zeros(m,2); %Matrix to record simulation results pre/post

Cumbeta = zeros(1,res); %The cumulative beta function for HIREC

Cumbeta(1) = betaH(1)/res; %Each slice has height beta(i) and width 1/res

for i = 2:res

Cumbeta(i) = Cumbeta(i-1) + betaH(i)/res; %Accumulate them

end

for mm = 1:m %m is the total number of individuals simulated

flag = 1; %Signals that threshold not yet surpassed

i = 1; %Initialize the time step

while flag&&(i<=nH)

mort = rand;

if mort > sH %Individual dies

SimX(mm,1) = 0; %Fitness set to 0

SimX(mm,2) = i; %Time set to the time step number

flag = 0; %Signal that individual has finished

else %Individual lives and seeks patch

patch = rand; %Random check for habitat

if patch < hH %patch is found

FF = FDraw(Cumbeta,FxH,res); %Draw patch fitness

if FF >= F(i) %Fitness >= threshhold

SimX(mm,1) = FF-(i^cH)\*delH; %Patch fitness - delay cst

SimX(mm,2) = i; %Completion time is step #

flag = 0; %Signal completion

end

end

end

i = i+1; %Increment steps

end

if (i > nH)&&flag %If individual doesn't settle within n steps

SimX(mm,1) = 0; %Its fitness is zero

SimX(mm,2) = nH; %Its time is set to n

end

end

FitmeanAndStepmean = mean(SimX,1); %MATLAB takes means of fitness & time

EmFitMeanX = FitmeanAndStepmean(1,1) %Simulation mean fitness

EmStopMeanX = FitmeanAndStepmean(1,2) %Simulat'n mean time to settle or die

%Simulate with post-HIREC parameters and thresholds

SimH = zeros(m,2); %Matrix to record simulation results

for mm = 1:m %m is the total number of individuals simulated

flag = 1; %Signals that threshold not yet surpassed

i = 1; %Initialize the time step

while flag&&(i<=nH)

mort = rand;

if mort > sH %Individual dies

SimH(mm,1) = 0; %Fitness set to 0

SimH(mm,2) = i; %Time set to the time step number

flag = 0; %Signal that individual has finished

else %Individual lives and seeks patch

patch = rand; %Random check for habitat

if patch < hH %patch is found

FFH = FDraw(Cumbeta,FxH,res); %Draw patch fitness

if FFH >= FH(i) %Fitness >= threshhold

SimH(mm,1) = FFH-(i^cH)\*delH; %Patch fit's - delay cost

SimH(mm,2) = i; %Completion time is step #

flag = 0; %Signal completion

end

end

end

i = i+1; %Increment steps

end

if (i > nH)&&flag %If individual doesn't settle within n steps

SimH(mm,1) = 0; %Its fitness is zero

SimH(mm,2) = n; %Its time is set to n

end

end

FitmeanAndStepmean = mean(SimH,1); %MATLAB takes means of fitness & time

EmFitMeanH = FitmeanAndStepmean(1,1) %Simulation mean fitness

EmStopMeanH = FitmeanAndStepmean(1,2) %Simulat'n mean time to settle or die

if nH > n %Determine the x-axis length as max of Fx and FxH

nmx = nH;

else

nmx = n;

end

if FxH > Fx

Fxmx = FxH;

else

Fxmx = Fx;

end

%Output plot

figure %red=adapted pre-HIREC, black=mix, blue=adapted post-HIREC

hold on

xx = 1:n; %x axis for the threshold function F vs time step

xxx = 1:nH; %x axis for the threshold functioins FX and FH

%plot([1 nH],[FxH FxH],'b:')

%plot([1 n],[Fx Fx],'r:') %Upper and lower limits of patch fitnesses

plot([1 n],[F00 F00],'k') %The theoretical fitness mean pre-HIREC

plot([1 nH],[F00X F00X],'r') %Theoretical fitness mean pre/post mix

plot([1 nH],[F00H F00H],'c') %The theoretical fitness mean post-HIREC

plot(xx,F,'kx') %pre-HIREC thresholds

plot(xxx,FH,'cx')%post-HIREC thresholds

%plot([1 n],[EmFitMean EmFitMean],'k--') %Sim fit mean pre-HIREC

%plot([1 nH],[EmFitMeanX EmFitMeanX],'r--') %Simulation mixed

%plot([1 nH],[EmFitMeanH EmFitMeanH],'c--') %Simulation post-HIREC

%plot([1 nH],[EmFitMeanL EmFitMeanL],'b--') %Simulation learning

for mm = 1:m

if seePre

plot(Sim(mm,2),Sim(mm,1),'k.','MarkerSize',7) %Simulation pre-HIREC

end

if seeMix

plot(SimX(mm,2),SimX(mm,1),'r.','MarkerSize',7) %Simulation mix

end

if seePost

plot(SimH(mm,2),SimH(mm,1),'c.','MarkerSize',7) %Simulation post-HIREC

end

end

xlabel('Time step')

ylabel('Fitness')

axis([1 nmx 0 Fxmx])

hold off

toc