%THRESHERPP addresses a parameter shift from the pre-HIREC condition to

%magnitudes changing along a gradient of possible HIREC parameter shifts.

%It plots the theoretical fitnesses pre-HIREC (black), post-HIREC shift but

%unadapted to the new conditions (red), and post-HIREC with adaptation

%(cyan).

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 when a patch is found, d'less

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

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

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

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

A1 = a/(a+b); %Mean of the beta distribution on [0 1]

FxH = zeros(1,res); %Post-HIREC upper limits of patch quality distribution

aH = zeros(1,res); %Post-HIREC alpha values for the beta distribution

bH = zeros(1,res); %Post-HIREC beta values for the beta distribution

nH = zeros(1,res); %Post-HIREC season length

hH = zeros(1,res); %Post-HIREC chance of finding a patch in a step

sH = zeros(1,res); %Post-HIREC chance of surviving a step

delH= zeros(1,res); %Post-HIREC deferred cost per step

cH = zeros(1,res); %Post-HIREC deferred cost exponent

F1 = zeros(1,res); %Pre-HIREC fitnesses

FX1 = zeros(1,res); %Immediately post-HIREC fitnesses

FH1 = zeros(1,res); %Re-adapted post-HIREC fitnesses

FH2 = zeros(1,res); %This is after-HIREC thresholds

ETS = zeros(3,res); %Expected time to settle, given survival

ESN = zeros(3,res); %Expected survival time, no settlement

%Code that deals with the adapted pre-HIREC world\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[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

prod = 1; %This accumulates a multiplier term

F(i) = 0; %Initialize the fractional fitness x-axis in beta

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

if j > i %The multiplier constraint

prod = prod\*(1-h\*P(j)); %Chance of not settling in step 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 is expected fitness contribution from step j+1

step = 0; %Checks to prevent step from going negative

end

F(i) = F(i) + step; %Add step to accumulate fitness across steps

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 then entire 1 to n seq'ce

F00 = 0;

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

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

for kk = 2:n

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

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

end

ETS(1,1:res) = ETSx;

for k = 1:res %for loop over res parameter magnitudes

FxH(k) = Fx; %Fx; Top end of the patch fitness beta distribution, fit

aH(k) = a; %a; Alpha parameter of the beta distribution, d'less

bH(k) = b; %b; Beta parameter of the beta distribution, d'less

nH(k) = n; %nH; Number of time steps in a season, d'less

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

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

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

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

%\*\*\*\*\*\*\*\*\*\*HERE are the parameters being varied

%All lines for param 2 should be blocked with % unless param 2 is used!

%pmin1 = 0; %This is the minimum value of parameter 1

%pmax1 = 0.98; %This is the maximum value of parameter 1

%delH(k) = pmax4 + (pmin4-pmax4)\*(k-0.5)/res;

%delH(k) = pmin1 + (pmax1-pmin1)\*(k-0.5)/res; %Parameter 1 is left of =

pmin2 = 0; %This is the minimum value of parameter 2

pmax2 = 1; %This is the maximum value of parameter 2

hH(k) = pmin2 + (pmax2-pmin2)\*(k-0.5)/res; %Parameter 1 is left of =

%sH(k) = pmin1 + (pmax1-pmin1)\*(k-0.5)/res; %Parameter 1 is left of =

%cH(k) = pmin1 + (pmax1-pmin1)\*(k-0.5)/res; %Parameter 1 is left of =

%\*\*\*BUT if nH is being varied, need pmax1=n & use the following nH(k) eqn:

%pmin1 = 5;

%pmax1 = 30;

%nH(k) = pmin1 + round((pmax1-pmin1)\*(k-0.5)/res);

%Lines below are when a second parameter also varies

%\*\*\*Varying nH backwards

%pmin2 = 5;

%pmax2 = 25;

%nH(k) = pmax2 + round((pmin2-pmax2)\*(k-0.5)/res);

%sH(k) = pmin2 + (pmax2-pmin2)\*(k-0.5)/res; %Parameter 2 is left of =

pmin3 = 50;

pmax3 = 150;

FxH(k) = pmin3 + (pmax3-pmin3)\*(k-0.5)/res;

%\*\*\*BUT use the next line instead if parameters 1 and 2 are negatively corr

%sH(k) = pmax2 + (pmin2-pmax2)\*(k-0.5)/res; %Parameter 2 is left of =

%Code that deals with the adapted post-HIREC world\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[pH qH betaH] = BetaVec(FxH(k),aH(k),bH(k),res);

%p is res res'n p vec for Feta; q is res res'n psi vec for Feta

PH = zeros(1,nH(k)); %Vector of threshold p values

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

FH = zeros(1,nH(k)); %Vector of thresh patch fitness values post-HIREC

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

prod = 1; %This accumulates a multiplier term

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

if j > i %The multiplier constraint

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

end

step = sH(k)^(j-i+1)\*hH(k)\*PH(j+1)\*(IH(j+1)...

-((j+1)^cH(k) - i^cH(k))\*delH(k))\*prod;

if step < 0

step = 0;

end

FH(i) = FH(i) + step; %Accumulates fitness from each forward step

end

frac = FH(i)/FxH(k); %F is fraction fraction of patch qual range

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 subt off (interpolation)

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(k)\*jj;

end

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

F00H = 0;

for j = 0:nH(k)-1

if j > 0

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

end

step = sH(k)^(j+1)\*hH(k)\*PH(j+1)\*(IH(j+1)...

-((j+1)^cH(k))\*delH(k))\*prod;

if step < 0

step = 0;

end

F00H = F00H + step;

end

FH2(k) = F00H;

ETS(3,k) = 1; prod = 1;

for kk = 2:nH(k)

prod = prod \* (1-hH(k)\*PH(kk-1));

ETS(3,k) = ETS(3,k) + sH(k)^kk\*prod;

end

%Calculate with pre-HIREC thresholds but post-HIREC parameters & calc FbarX

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

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

for i = 1:nH(k); %Begin FbarX calculation

frac = F(i)/FxH(k); %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(k)\*jj;

end

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

F00X = 0;

for j = 0:nH(k)-1

if j > 0

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

end

step = sH(k)^(j+1)\*hH(k)\*PX(j+1)\*(IX(j+1)...

-((j+1)^cH(k))\*delH(k))\*prod;

if step < 0

step = 0;

end

F00X = F00X + step;

end

ETS(2,k) = 1; prod = 1;

for kk = 2:nH(k)

prod = prod \* (1-hH(k)\*PX(kk-1));

ETS(2,k) = ETS(2,k) + sH(k)^kk\*prod;

end

F1(k) = F00; %Record the four fitnesses for each k

FX1(k) = F00X;

FH1(k) = F00H;

end %for loop over res values of the parameter

FHmax = max(FH);

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

bar = Fx;

else

bar = FHmax;

end

%Output plot

x = 0.5/res:1/res:(res-0.5)/res; %h & s on (0,1)

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

hold on

plot(x,F1,'k') %pre-HIREC theoretical fitness

plot(x,FX1,'r') %mixed-HIREC theoretical fitness

plot(x,FH1,'c') %post-HIREC theoretical fitness

xlabel('Chance of patch-finding/step; max patch quality 50-150')

ylabel('Fitness')

-----------------------------------------------------------------------------

axis([0 1 0 bar])

hold off

figure %Expected time to settle\_ and to survive without settling--

hold on

axis([0 1 0 n])

xlabel('Chance of patch-finding/step; max patch quality 50-150')

ylabel('Time spent choosing')% (-) or or surviving without choosing (--)')

plot(x,ETS(1,1:res),'k')

plot(x,ETS(2,1:res),'r')

plot(x,ETS(3,1:res),'c')

hold off