**DiNuzzo & Griffen 2020 – Technical discussion**

Here we point out several issues regarding the implementation of the model of DiNuzzo and Griffen 2020, Proc. R. Soc. B 287: 20201095) in NetLogo. All issues cause the simulations to deviate from the model description in the paper and can influence the simulation outcomes in undesirable ways. Relevant lines of their code are highlighted in yellow, see notes below:

patches-own [ quality

possible-consumption

expected-consumption

]

breed [ active-animals active-animal ]

breed [ sedentary-animals sedentary-animal ]

turtles-own [ consumption-rate

avg-consumption-rate

time-since-moved ]

globals [ ;marginal-value

max-consumption

countdown

x

stop-countdown]

to setup

clear-all

setup-patches

setup-turtles

set countdown number-of-active + number-of-inactive

set stop-countdown 0

reset-ticks

end

to setup-patches

resize-world (number-of-patches \* -1) number-of-patches (number-of-patches \* -1) number-of-patches **(1)**

ask patches

[set quality (2 + random 8) **(2)**

set pcolor scale-color green quality 1 10 ]

end

to setup-turtles

create-active-animals number-of-active

[ set color white

set size .4

setxy random-xcor random-ycor ]

create-sedentary-animals number-of-inactive

[ set color blue

set size .4

setxy random-xcor random-ycor ]

ask turtles

[ set time-since-moved number-of-active + number-of-inactive ]

end

to Go

if stop-countdown > 50 or number-of-active + number-of-inactive = 0 **(3)**

[ ask patches [ calculate-expected-consumption ]

stop ]

ask turtles

[ set time-since-moved time-since-moved - 1 ]

ask turtles

[ calculate-consumption ]

ask turtles

[ calculate-avg-consumption-rate ]

ask patches

[ calculate-max-consumption ]

ask patches

[ calculate-expected-consumption ]

move-turtles

ifelse countdown > 1

[ set countdown countdown - 1 ]

[ set countdown number-of-active + number-of-inactive ]

set stop-countdown stop-countdown + 1

tick

end

to calculate-consumption

set consumption-rate ( [ quality ] of patch-here ) / ( count turtles-here ) **(4)**

end

to calculate-avg-consumption-rate

set avg-consumption-rate mean [ consumption-rate ] of turtles

end

to calculate-max-consumption

ifelse TypeII-functional-response?

[ifelse

count turtles-here > 0

[ let food-available (quality - ((count turtles-here + 1) \* ( quality / ( quality + count turtles-here + 1 ))))

ifelse food-available > max-feeding-rate

[set possible-consumption max-feeding-rate]

[set possible-consumption food-available]]

[ ifelse quality > max-feeding-rate

[set possible-consumption max-feeding-rate]

[set possible-consumption quality]]]

[ifelse

count turtles-here > 0

[ set possible-consumption ( quality ) / ( count turtles-here + 1 ) ]

[ set possible-consumption quality ]]

set max-consumption max [ possible-consumption ] of patches

end

to calculate-expected-consumption

ifelse TypeII-functional-response?

[ifelse

count turtles-here > 0

[ let food-available (quality - ((count turtles-here + 1) \* ( quality / ( quality + count turtles-here + 1 ))))

ifelse food-available > max-feeding-rate **(5)**

[set possible-consumption max-feeding-rate]

[set possible-consumption food-available]]

[ ifelse quality > max-feeding-rate **(5)**

[set possible-consumption max-feeding-rate]

[set possible-consumption quality]]]

[ifelse

count turtles-here > 0

[ set possible-consumption ( quality ) / ( count turtles-here + 1 ) ]

[ set possible-consumption quality ]]

end

to move-turtles

set x random-float 1

ask one-of turtles with-min [ time-since-moved ] **(6)**

[ ifelse breed = active-animals

[ ifelse consumption-rate > max-consumption **(7)**

[ fd 0 ] **(8)**

[ if x < .8

[ move-to one-of patches with-max [ possible-consumption ]

set time-since-moved number-of-active + number-of-inactive

set stop-countdown 0 ]]]

[ ifelse consumption-rate > max-consumption **(7)**

[ fd 0 ] **(8)**

[ if x < .2

[ move-to one-of patches with-max [ possible-consumption ]

set time-since-moved number-of-active + number-of-inactive

set stop-countdown 0 ]]]

]

end

Notes:

1. *number-of-patches* is here used not as the real number of patches but as the number of rows and columns in the grid. However, the real dimensions of the grid are not equal to the parameter: If the grid went from 0 to *number-of-patches* – 1 as it should, a *number-of-patches* = 7 would yield a 7x7 grid. But because the grid is in the code specified to go from – *number-of-patches* to + *number-of-patches*, a *number-of-patches* = 7 produces a 15x15 grid.

1. This initializes the patches with random quality levels between 2 and 9, instead of from 1 to 9 as claimed in the article.
2. The stop condition for the simulation does not perform a rigorous check whether the ideal free distribution is reached. Even if the IFD has not yet been attained, the simulation stops when individuals cease movements for 50 time steps. This can readily occur when the population harbours a large proportion of very inactive individuals.
3. Individuals calculate their current intake rate using this function, regardless of the functional response specified as a parameter. This is problematic when a functional response type two is specified in the parameters, which is then used to calculate *max-consumption* and *possible-consumption* (lines 76ff and 94ff) for all patches. The comparison in lines 115 and 122 between individual *consumption-rate* and patch *max-consumption* is then drawn between values calculated with different formulas. It would be necessary here, just as in lines 76ff and 94ff, to use the function for a type 2 functional response.
4. Here, in case a type 2 functional response is used, DiNuzzo and Griffen state in their supplement S1.4 that *possible-consumption* is constrained to a maximum of one. There is no reason for such a constraint, and indeed it is quite problematic because a) in the function in line 68 (or note 4), no such constraint is posed (these two values compared in lines 115 and 122), and b) a lot of patches will then have equal intake rates, leading to problems in (7). We suspect that DiNuzzo and Griffen expected to use a functional response that saturates at a value of one, such as in Abrams & Ginsburg 2000. Instead they calculate *food-available*, using a function that does not saturate and goes well beyond 1.
5. Here, individuals are not randomly selected to move, as stated in the paper, but randomly from among the individuals that have not moved for the longest time. This is a deviation from the model description that could accelerate the time until the IFD is reached. This feature also becomes important in (8).

1. The condition to not move only includes cases where *consumption-rate* is greater than *max-consumption*. This implies that individuals move also between patches when *consumption-rate* is equal to *max-consumption*. Since resources are discrete, this leads individuals to move between equal patches and risks an infinite loop, when movements never cease.
2. Here, individuals that have a consumption rate higher than any potential consumption rate are told to do nothing. Importantly, they do not update their *time-since-moved*, and hence remain and eventually fill up the *one-of turtles with-min[ time-since-moved ]* category (turtles that have not moved for the longest time, see 6). This is what ultimately brings the simulation to halt. As a side effect, it gets progressively more difficult to select non-optimal individuals for movement from among the optimal ones in this category, and therefore it becomes very likely that there are still individuals with suboptimal intake rates after 50 time steps have run off and the simulation ends (consider that an individual with activity 0.1 and a suboptimal intake rate, mixed with 9 other already optimal individuals in the selection category, gets to make only one movement every 100 timesteps).