

Source sink dynamics and emergent plant allocation

Liam Langan

January 26, 2018

1 Introduction

In my previous report I highlighted deficiencies with the Thornley allocation model [7]. To briefly revisit these they were:

- that the response of plant allocation to removal of above ground mass was not consistent with observed responses [1, 3, 4, 2, 5].
- Allocation to above ground growth is reduced via decreases in the relative mass of the the above ground compartment.

My solution to this is as follows, Thornley's 1998 paper "Modelling Shoot Root Relations. The Only Way Forward?" [6] separates the substrate concentrations and masses of substrates for roots and shoots. This model also includes substrate transport between roots and shoots. Shoots transport carbon to roots with a rate dependent of the difference in concentrations and the resistances of these compartments. Resistance is a function of the mass of each compartment e.g. $(1/\text{mass-root})$. In this paper transport is unidirectional i.e. shoots transport C to roots and roots transport N to shoots. I have implemented and extended this model and made substrate transport bi-directional so that transport of any substrate between any two compartments is dependent on the difference in concentrations and the transport resistance between them. This allows transfer both C and N (in this example) from roots to shoots. Thornley models shoot and root growth separately in this paper. This means that the change in the relative fractions of plant mass wouldn't change allocation, in fact there is no allocation, relative growth rates of compartments emerges. The effect of allowing bi-directional transfer is to reduce root growth and promote shoot growth since the root C and N concentrations would be reduced (via transport to above ground sinks) and shoot C and N concentrations increased (from zero potentially). The resistance component for shoots will go to infinity if shoots are completely removed and growth will be zero if shoots are completely removed. This is similar to the allocation problem in TAM but if we simply never let any compartment become zero then this implementation appears to solve our problem. To summarise, bi-directional transport of all substrates allows any compartment to become a sink. This transport of substrates reduces the growth of what are then source compartments while increasing the growth of what has become a sink compartment.

In my implementation, plants consist of four compartments, a carbon organ (CO), nitrogen organ (NO), phosphorus organ (PO) and a water organ (WO). Each compartment assimilates an extrinsic substrate, the CO assimilates carbon, the NO assimilates nitrogen, the PO assimilates phosphorus and the WO assimilates water. Each plant compartment has its own concentrations of substrates. Each compartment grows as a function of its mass and the concentration of substrates. Transport of substrates between compartments happens when there is a difference in the substrate concentrations between compartments. The rate of transport of substrates between compartment is a function of the difference between them and the transport resistance. The transport resistance is a function of the masses of the target compartments. With this implementation allocation is an emergent property of source sink dynamics. Below, in section 2, I provide the code for this implementation. The code for the .Rnw file, the original .Rnw file, the associated bibfile and the R-script I used to develop this model version can be found at: https://github.com/liamlangan/Thornley_source_sink.git.

2 Thornley source sink

The following code is an extended implimentation of Thornley's (1998) paper [6] but where substrate transport is bi-directional.

2.1 (

Source sink model code)

```
#####  
# R version of Thornley allcoation model (TAM) as in Thornley and Parsons 2014. -  
# J. Theor. Biol. 342: 1-14  
# Author: Steven Higgins  
# Rewritten to include elements of Thornley 1998 -  
# Modelling shoot-root relations. The only way forward?  
# (https://doi.org/10.1006/anbo.1997.0529)  
# Author: Liam Langan 20.11.2017 re-coded orgiginal SH code to add source sink  
# dynamics and remove predefined allocation.  
# This version with carbon organ, nitrogen organ, phosphorus organ and water organ.  
# This version includes separate substrate pools for each organ and allows  
# bi-directional transport of substrates between organs.  
  
# The code for the .Rnw file, the original .Rnw file, the associated bibfile and  
# and the R-script I used to develop this model version can be found at:  
# https://github.com/liamlangan/Thornley_source_sink.git  
#####  
  
## initial values for mass of structural dry matter in carbon organ (co),  
## nitrogen organ (no), phosphorus organ (po) and water organ (wo)  
size.factor<-0.1;  
  
thorn_Mxco_t0 = 0.25/size.factor;  
thorn_Mxno_t0 = 0.25/size.factor;  
thorn_Mxpo_t0 = 0.25/size.factor;  
thorn_Mxwo_t0 = 0.25/size.factor;  
  
## intitial values for mass of substrates C,N,P for carbon organ  
thorn_Mcs_co_t0 = 0.025/size.factor*0.25;  
thorn_Mns_co_t0 = 0.01/size.factor*0.25;  
thorn_Mps_co_t0 = 0.005/size.factor*0.25;  
thorn_Mws_co_t0 = 0.25/size.factor*0.25;  
  
## intitial values for mass of substrates C,N,P for nitrogen organ  
thorn_Mcs_no_t0 = 0.025/size.factor*0.25;  
thorn_Mns_no_t0 = 0.01/size.factor*0.25;  
thorn_Mps_no_t0 = 0.005/size.factor*0.25;  
thorn_Mws_no_t0 = 0.25/size.factor*0.25;  
  
## intitial values for mass of substrates C,N,P for phosphorus organ  
thorn_Mcs_po_t0 = 0.025/size.factor*0.25;  
thorn_Mns_po_t0 = 0.01/size.factor*0.25;  
thorn_Mps_po_t0 = 0.005/size.factor*0.25;  
thorn_Mws_po_t0 = 0.25/size.factor*0.25;  
  
## intitial values for mass of substrates C,N,P for water organ  
thorn_Mcs_wo_t0 = 0.025/size.factor*0.25;
```

```

thorn_Mns_wo_t0 = 0.01/size.factor*0.25;
thorn_Mps_wo_t0 = 0.005/size.factor*0.25;
thorn_Mws_wo_t0 = 0.25/size.factor*0.25;

## uptake rates of C,N,P,W
base = 0.0002*100;
thorn_Kc = base*1.0;
thorn_Kw = 0.0015;
thorn_Kn = 0.25;
thorn_Kp = 0.1;

## product inhibition
thorn_Jcs = 0.03; # Thornley ('98) 0.1
thorn_Jns = 0.01;
thorn_Jps = 0.005;
thorn_Jws = 0.003;
## mass inhibition via self shading or competition Thornley ('98) eq.3
thorn_Kmc = thorn_Kmn = thorn_Kmp = thorn_Kmc = 1;

## Fractions of C, N, P in structural dry matter
# the first three are as in Thornley, Fwx is calibrated, higher values do not work
# so Fwx is artificial. One should probably think of this as water used in
# metabolism and for cellular growth. These fractions could be easily
# redefined for the different plant compartments now i.e. Fnx_co, Fnx_no, Fnx_po
# could have different values.
thorn_Fcx = 0.4;
thorn_Fnx = 0.02;
thorn_Fpx = 0.005;
thorn_Fwx = 0.0001;

## other constants
#thorn_Kg = 200 ; #structural growth rate constant; Thornley uses 200000
thorn_Kg = 0.1 ; # Thornley used 0.1 in his 1998 paper but this is a tuning param.

#####
#---start model---#
# set initial states
## STATE VARIABLE initial values for mass of structural dry matter in co, no, po, wo.
thorn_Mxco = thorn_Mxco_t0;
thorn_Mxwo = thorn_Mxwo_t0;
thorn_Mxno = thorn_Mxno_t0;
thorn_Mxpo = thorn_Mxpo_t0;

## STATE VARIABLE initial values for mass of substrate C,N,P,W in co, no, po, wo.
thorn_Mcs_co = thorn_Mcs_co_t0;
thorn_Mns_co = thorn_Mns_co_t0;
thorn_Mps_co = thorn_Mps_co_t0;
thorn_Mws_co = thorn_Mws_co_t0;

thorn_Mcs_no = thorn_Mcs_no_t0;
thorn_Mns_no = thorn_Mns_no_t0;
thorn_Mps_no = thorn_Mps_no_t0;
thorn_Mws_no = thorn_Mws_no_t0;

thorn_Mcs_po = thorn_Mcs_po_t0;

```

```

thorn_Mns_po = thorn_Mns_po_t0;
thorn_Mps_po = thorn_Mps_po_t0;
thorn_Mws_po = thorn_Mws_po_t0;

thorn_Mcs_wo = thorn_Mcs_po_t0;
thorn_Mns_wo = thorn_Mns_po_t0;
thorn_Mps_wo = thorn_Mps_po_t0;
thorn_Mws_wo = thorn_Mws_po_t0;

# initial total mass of plant
# this is now redundant but useful to examine how mass is apportioned.
thorn_Mx_t0 = thorn_Mxco_t0 + thorn_Mxno_t0 + thorn_Mxpo_t0 + thorn_Mxwo_t0;

## AUX VARIABLE initial values for concentrations of substrates C,N,P,W
## for the co, no, po and wo.
thorn_Cs_co = thorn_Mcs_co_t0/thorn_Mxco;
thorn_Ns_co = thorn_Mns_co_t0/thorn_Mxco;
thorn_Ps_co = thorn_Mps_co_t0/thorn_Mxco;
thorn_Ws_co = thorn_Mws_co_t0/thorn_Mxco;

thorn_Cs_no = thorn_Mcs_no_t0/thorn_Mxno;
thorn_Ns_no = thorn_Mns_no_t0/thorn_Mxno;
thorn_Ps_no = thorn_Mps_no_t0/thorn_Mxno;
thorn_Ws_no = thorn_Mws_no_t0/thorn_Mxno;

thorn_Cs_po = thorn_Mcs_po_t0/thorn_Mxpo;
thorn_Ns_po = thorn_Mns_po_t0/thorn_Mxpo;
thorn_Ps_po = thorn_Mps_po_t0/thorn_Mxpo;
thorn_Ws_po = thorn_Mws_po_t0/thorn_Mxpo;

thorn_Cs_wo = thorn_Mcs_wo_t0/thorn_Mxwo;
thorn_Ns_wo = thorn_Mns_wo_t0/thorn_Mxwo;
thorn_Ps_wo = thorn_Mps_wo_t0/thorn_Mxwo;
thorn_Ws_wo = thorn_Mws_wo_t0/thorn_Mxwo;

## AUX VARIABLE initial values for fractions of structural dry matter in
# the carbon organ, nitrogen organ, phosphorus organ and water organ
# these are now redundant but useful to examine how mass is apportioned.
thorn_Fxco = thorn_Mxco_t0/thorn_Mx_t0;
thorn_Fxwo = thorn_Mxwo_t0/thorn_Mx_t0;
thorn_Fxno = thorn_Mxno_t0/thorn_Mx_t0;
thorn_Fxpo = thorn_Mxpo_t0/thorn_Mx_t0;

#R only make a matrix to store the variables for plotting
#n.steps<-360*10
n.steps<-3*1 # number of time steps to run model for
M<-matrix(0,nrow=n.steps,ncol=71)

# Wrapper to allow the model to be called and run.
# variables can be input which define how the model is run.
# e.g. "uptake" changes the rate of nitrogen uptake.
wrapper_fn_mod <- function(uptake, fire)
{
  ## the for loop

```

```

for(i in 1:n.steps)
{

uptake_change <- uptake
fire_on <- fire

# aDGVM produces numbers around 0.00001 to 0.001 for C_net/plant_mass
base = 0.0002*100;
thorn_Kc = base*1.0;

# this code sets carbon substrate uptake to zero between day X & Y if required
# if( i >= 180 && i <= 360 )
# {
#   thorn_Kc = 0.0;
# }

if(uptake_change == 1) # low N uptake rate
{
  thorn_Kn = 0.025;
  # thorn_Kn = 0.05 - pmin(0.001, (i*0.1)); # decreasing N-uptake through time
}

if(uptake_change == 0) # high N uptake rate
{
  thorn_Kn = 0.25;
}

thorn_Kw = 0.015;
thorn_Kp = 0.1;

thorn_Mx = thorn_Mxco + thorn_Mxwo + thorn_Mxno + thorn_Mxpo ; #3

#####
# Here I am assuming a leaf of mass less than 0.1 grammes cannot do photosynthesis.
# This is to test the difference between uni and bi directional transport, i.e. bi-
# directional transport would allow such a leaf to "grow" until is bigger than
# this artificial threshold I have set.
if(thorn_Mxco <= 0.0001)
{
  mass_place_holder = 0.0; # this artificially sets C uptake to zero.
}
else
{
  mass_place_holder = thorn_Mxco;
}

thorn_Uc = thorn_Kc * mass_place_holder; # carbon substrate uptake
thorn_Un = thorn_Kn * thorn_Mxno;      # nitrogen substrate uptake
thorn_Up = thorn_Kp * thorn_Mxpo;      # phosphorus substrate uptake
thorn_Uw = thorn_Kw * thorn_Mxwo;      # water substrate uptake

# Add uptake to existing substrate masses
# carbon added to carbon organ pool mass
thorn_Mcs_co = max(0.000, thorn_Mcs_co + thorn_Uc);
# nitrogen added to nitrogen organ pool mass

```

```

thorn_Mns_no = max(0.000, thorn_Mns_no + thorn_Un);
# phosphorus added to phosphorus organ pool mass
thorn_Mps_po = max(0.000, thorn_Mps_po + thorn_Up);
# water added to water organ pool mass
thorn_Mws_wo = max(0.000, thorn_Mws_wo + thorn_Uw);

# Update concentrations of substrates C,N,P,W for co, no, po and wo.
thorn_Cs_co = thorn_Mcs_co/thorn_Mxco; # only for substrate specific to organ
thorn_Ns_no = thorn_Mns_no/thorn_Mxno; # only for substrate specific to organ
thorn_Ps_po = thorn_Mps_po/thorn_Mxpo; # only for substrate specific to organ
thorn_Ws_wo = thorn_Mws_wo/thorn_Mxwo; # only for substrate specific to organ

# use this code to simulate a loss of shoots e.g in fire
if(i == 100 && fire_on ==1)
{
    ## 1 gram of co mass left - growth can only happen if structural mass > 0
    thorn_Mxco = 0.0001;

    ## mass of substrates in co set to zero
    thorn_Mcs_co = thorn_Mcs_co*0.0;
    thorn_Mns_co = thorn_Mns_co*0.0;
    thorn_Mps_co = thorn_Mps_co*0.0;
    thorn_Mws_co = thorn_Mws_co*0.0;

    ## recalculate substrate concentrations for carbon organ
    thorn_Cs_co = thorn_Mcs_co/thorn_Mxco; #2
    thorn_Ns_co = thorn_Mns_co/thorn_Mxco; #2
    thorn_Ps_co = thorn_Mps_co/thorn_Mxco; #2
    thorn_Ws_co = thorn_Mws_co/thorn_Mxco; #2

    # print("Cs_Co")
    # print(thorn_Cs_co)
    # print("-----")
}
#####

## thorn_Kg is a calibration parameter. Substrate cocentrations affect growth and are
## modified by uptake inhibition and trasnport
thorn_G_co = thorn_Kg*thorn_Mxco*thorn_Cs_co*thorn_Ns_co*thorn_Ps_co*thorn_Ws_co ;
thorn_G_no = thorn_Kg*thorn_Mxno*thorn_Cs_no*thorn_Ns_no*thorn_Ps_no*thorn_Ws_no ;
thorn_G_po = thorn_Kg*thorn_Mxpo*thorn_Cs_po*thorn_Ns_po*thorn_Ps_po*thorn_Ws_po ;
thorn_G_wo = thorn_Kg*thorn_Mxwo*thorn_Cs_wo*thorn_Ns_wo*thorn_Ps_wo*thorn_Ws_wo ;

# print growth and carbon substrate concentration for carbon organ following fire
# if(i == 100 | i==101)
# {
#     print(i)
#     print("growth")
#     print(c("CO", thorn_G_co))
#     print(c("NO", thorn_G_no))
#     print(c("PO", thorn_G_po))
#     print(c("WO", thorn_G_wo))
#     print("thorn_Cs_co")
#     print(c("thorn_Cs_co", thorn_Cs_co))

```

```

#   print(c("res_co",res_co))
#   print("-----")
# }

# take growth away from substrate mass for each pool in each organ
thorn_Mcs_co = max(0.000, thorn_Mcs_co - thorn_Fcx * thorn_G_co );
thorn_Mns_co = max(0.000, thorn_Mns_co - thorn_Fnx * thorn_G_co );
thorn_Mps_co = max(0.000, thorn_Mps_co - thorn_Fpx * thorn_G_co );
thorn_Mws_co = max(0.000, thorn_Mws_co - thorn_Fwx * thorn_G_co );

thorn_Mcs_no = max(0.000, thorn_Mcs_no - thorn_Fcx * thorn_G_no );
thorn_Mns_no = max(0.000, thorn_Mns_no - thorn_Fnx * thorn_G_no );
thorn_Mps_no = max(0.000, thorn_Mps_no - thorn_Fpx * thorn_G_no );
thorn_Mws_no = max(0.000, thorn_Mws_no - thorn_Fwx * thorn_G_no );

thorn_Mcs_po = max(0.000, thorn_Mcs_po - thorn_Fcx * thorn_G_po );
thorn_Mns_po = max(0.000, thorn_Mns_po - thorn_Fnx * thorn_G_po );
thorn_Mps_po = max(0.000, thorn_Mps_po - thorn_Fpx * thorn_G_po );
thorn_Mws_po = max(0.000, thorn_Mws_po - thorn_Fwx * thorn_G_po );

thorn_Mcs_wo = max(0.000, thorn_Mcs_wo - thorn_Fcx * thorn_G_wo );
thorn_Mns_wo = max(0.000, thorn_Mns_wo - thorn_Fnx * thorn_G_wo );
thorn_Mps_wo = max(0.000, thorn_Mps_wo - thorn_Fpx * thorn_G_wo );
thorn_Mws_wo = max(0.000, thorn_Mws_wo - thorn_Fwx * thorn_G_wo );

# update the structural mass of each organ
thorn_Mxco = max(0.0001, thorn_Mxco + thorn_G_co) ;# eq.1 and 2 in Thronley ('98)
thorn_Mxno = max(0.0001, thorn_Mxno + thorn_G_no) ;# eq.1 and 2 in Thronley ('98)
thorn_Mxpo = max(0.0001, thorn_Mxpo + thorn_G_po) ;# eq.1 and 2 in Thronley ('98)
thorn_Mxwo = max(0.0001, thorn_Mxwo + thorn_G_wo) ;# eq.1 and 2 in Thronley ('98)

# this is part of the substrate inhibition code --- will comment in later
# thorn_Uc = (thorn_Kc * thorn_Mxco) / (1 + thorn_Cs/thorn_Jcs)
# thorn_Uw = (thorn_Kw * thorn_Mxwo) / (1 + thorn_Ws/thorn_Jws)
# thorn_Un = (thorn_Kn * thorn_Mxno) / (1 + thorn_Ns/thorn_Jns)
# thorn_Up = (thorn_Kp * thorn_Mxpo) / (1 + thorn_Ps/thorn_Jps)

# Update concentrations of substrates C,N,P,W for co, no, po and wo.
thorn_Cs_co = thorn_Mcs_co/thorn_Mxco; #2
thorn_Ns_co = thorn_Mns_co/thorn_Mxco; #2
thorn_Ps_co = thorn_Mps_co/thorn_Mxco; #2
thorn_Ws_co = thorn_Mws_co/thorn_Mxco; #2

thorn_Cs_no = thorn_Mcs_no/thorn_Mxno; #2
thorn_Ns_no = thorn_Mns_no/thorn_Mxno; #2
thorn_Ps_no = thorn_Mps_no/thorn_Mxno; #2
thorn_Ws_no = thorn_Mws_no/thorn_Mxno; #2

thorn_Cs_po = thorn_Mcs_po/thorn_Mxpo; #2
thorn_Ns_po = thorn_Mns_po/thorn_Mxpo; #2
thorn_Ps_po = thorn_Mps_po/thorn_Mxpo; #2
thorn_Ws_po = thorn_Mws_po/thorn_Mxpo; #2

thorn_Cs_wo = thorn_Mcs_wo/thorn_Mxwo; #2
thorn_Ns_wo = thorn_Mns_wo/thorn_Mxwo; #2

```

```

thorn_Ps_wo = thorn_Mps_wo/thorn_Mxwo; #2
thorn_Ws_wo = thorn_Mws_wo/thorn_Mxwo; #2

#####
## adding substrate transport between pools following Thornley (1998). Substrates will
## be transported between plants compartments based on concentration differences and
## resistances. This avoids the issue of reduced allocation as this model does not
## allocate growth, growth of each compartment is indepenant of the others but it
## allows every compartment to be both a source and a sink. Should an individual be
## top-killed i.e. loss of the carbon organ and substrates contained within,
## C, N, P and W will be transported from all other organs until there is no
## concentration difference. This increases carbon organ growth and decreases growth
## in all other compartments.

# Specific transport resistances for substrates.
# I'm assuming they are equal for all substrates and transport directions.
# This assumption can be modified in the future (we calculate resistances in aDGVM)
p_co = p_no = p_po = p_wo =1; # eq.4 Thornley 1998

q = 1 # eq.4 scaling parameter (Thornley 1998)

# Transport resistance of each compartment
res_co = p_co / (thorn_Mxco)^q; # eq.4 Thornley 1998
res_no = p_no / (thorn_Mxno)^q; # eq.4 Thornley 1998
res_po = p_po / (thorn_Mxpo)^q; # eq.4 Thornley 1998
res_wo = p_wo / (thorn_Mxwo)^q; # eq.4 Thornley 1998

##-----
# accounting
total_c1 <- (thorn_Cs_co*thorn_Mxco) + (thorn_Fcx*thorn_Mxco) +
            (thorn_Cs_no*thorn_Mxno) + (thorn_Fcx*thorn_Mxno) +
            (thorn_Cs_po*thorn_Mxpo) + (thorn_Fcx*thorn_Mxpo) +
            (thorn_Cs_wo*thorn_Mxwo) + (thorn_Fcx*thorn_Mxwo)

total_n1 <- (thorn_Ns_co*thorn_Mxco) + (thorn_Fnx*thorn_Mxco) +
            (thorn_Ns_no*thorn_Mxno) + (thorn_Fnx*thorn_Mxno) +
            (thorn_Ns_po*thorn_Mxpo) + (thorn_Fnx*thorn_Mxpo) +
            (thorn_Ns_wo*thorn_Mxwo) + (thorn_Fnx*thorn_Mxwo)

total_p1 <- (thorn_Ps_co*thorn_Mxco) + (thorn_Fpx*thorn_Mxco) +
            (thorn_Ps_no*thorn_Mxno) + (thorn_Fpx*thorn_Mxno) +
            (thorn_Ps_po*thorn_Mxpo) + (thorn_Fpx*thorn_Mxpo) +
            (thorn_Ps_wo*thorn_Mxwo) + (thorn_Fpx*thorn_Mxwo)

total_w1 <- (thorn_Ws_co*thorn_Mxco) + (thorn_Fwx*thorn_Mxco) +
            (thorn_Ws_no*thorn_Mxno) + (thorn_Fwx*thorn_Mxno) +
            (thorn_Ws_po*thorn_Mxpo) + (thorn_Fwx*thorn_Mxpo) +
            (thorn_Ws_wo*thorn_Mxwo) + (thorn_Fwx*thorn_Mxwo)

##-----

# Transport of substrates between compartments. Here I extend Thornley uni-directional
# model to make it bi-directional. I will add functionalty to make this
# uni-directional via the wrapper function tomorrow.

# NOTE: The 0.5 in the transport equations is a tuning parameter which dampens

```


*# oscillatoins in substrate concentrations. The model still preforms qualititively the
same without but has faster movement of concentratoins and it makes the plots
difficult to exmaine.*

NOTE: Positive transport values indicate transport away from organ (source).

Negative values indicitate transport to an organ (sink).

carbon substrate transport

Tc_co_to_no = 0.5((thorn_Cs_co - thorn_Cs_no) / (res_co + res_no)) # CO to NO*

Tc_co_to_po = 0.5((thorn_Cs_co - thorn_Cs_po) / (res_co + res_po)) # CO to PO*

Tc_co_to_wo = 0.5((thorn_Cs_co - thorn_Cs_wo) / (res_co + res_wo)) # CO to WO*

Tc_no_to_po = 0.5((thorn_Cs_no - thorn_Cs_po) / (res_no + res_po)) # NO to PO*

Tc_no_to_wo = 0.5((thorn_Cs_no - thorn_Cs_wo) / (res_no + res_wo)) # NO to WO*

Tc_po_to_wo = 0.5((thorn_Cs_po - thorn_Cs_wo) / (res_po + res_wo)) # PO to WO*

nitrogen transport

Tn_co_to_no = 0.5((thorn_Ns_co - thorn_Ns_no) / (res_co + res_no)) # CO to NO*

Tn_co_to_po = 0.5((thorn_Ns_co - thorn_Ns_po) / (res_co + res_po)) # CO to PO*

Tn_co_to_wo = 0.5((thorn_Ns_co - thorn_Ns_wo) / (res_co + res_wo)) # CO to WO*

Tn_no_to_po = 0.5((thorn_Ns_no - thorn_Ns_po) / (res_no + res_po)) # NO to PO*

Tn_no_to_wo = 0.5((thorn_Ns_no - thorn_Ns_wo) / (res_no + res_wo)) # NO to WO*

Tn_po_to_wo = 0.5((thorn_Ns_po - thorn_Ns_wo) / (res_po + res_wo)) # PO to WO*

phosphorus transport

Tp_co_to_no = 0.5((thorn_Ps_co - thorn_Ps_no) / (res_co + res_no)) # CO to NO*

Tp_co_to_po = 0.5((thorn_Ps_co - thorn_Ps_po) / (res_co + res_po)) # CO to PO*

Tp_co_to_wo = 0.5((thorn_Ps_co - thorn_Ps_wo) / (res_co + res_wo)) # CO to WO*

Tp_no_to_po = 0.5((thorn_Ps_no - thorn_Ps_po) / (res_no + res_po)) # NO to PO*

Tp_no_to_wo = 0.5((thorn_Ps_no - thorn_Ps_wo) / (res_no + res_wo)) # NO to WO*

Tp_po_to_wo = 0.5((thorn_Ps_po - thorn_Ps_wo) / (res_po + res_wo)) # PO to WO*

water transport

Tw_co_to_no = 0.5((thorn_Ws_co - thorn_Ws_no) / (res_co + res_no)) # CO to NO*

Tw_co_to_po = 0.5((thorn_Ws_co - thorn_Ws_po) / (res_co + res_po)) # CO to PO*

Tw_co_to_wo = 0.5((thorn_Ws_co - thorn_Ws_wo) / (res_co + res_wo)) # CO to WO*

Tw_no_to_po = 0.5((thorn_Ws_no - thorn_Ws_po) / (res_no + res_po)) # NO to PO*

Tw_no_to_wo = 0.5((thorn_Ws_no - thorn_Ws_wo) / (res_no + res_wo)) # NO to WO*

Tw_po_to_wo = 0.5((thorn_Ws_po - thorn_Ws_wo) / (res_po + res_wo)) # PO to WO*

re-calculate substrate masses for each compartment following transport

thorn_Mcs_co = max(0.000, thorn_Mcs_co - Tc_co_to_no - Tc_co_to_po - Tc_co_to_wo);

thorn_Mns_co = max(0.000, thorn_Mns_co - Tn_co_to_no - Tn_co_to_po - Tn_co_to_wo);

thorn_Mps_co = max(0.000, thorn_Mps_co - Tp_co_to_no - Tp_co_to_po - Tp_co_to_wo);

thorn_Mws_co = max(0.000, thorn_Mws_co - Tw_co_to_no - Tw_co_to_po - Tw_co_to_wo);

thorn_Mcs_no = max(0.000, thorn_Mcs_no - Tc_no_to_po - Tc_no_to_wo + Tc_co_to_no);

thorn_Mns_no = max(0.000, thorn_Mns_no + Tn_co_to_no - Tn_no_to_po - Tn_no_to_wo);

thorn_Mps_no = max(0.000, thorn_Mps_no - Tp_no_to_po - Tp_no_to_wo + Tp_co_to_no);

thorn_Mws_no = max(0.000, thorn_Mws_no - Tw_no_to_po - Tw_no_to_wo + Tw_co_to_no);

thorn_Mcs_po = max(0.000, thorn_Mcs_po - Tc_po_to_wo + Tc_no_to_po + Tc_co_to_po);

thorn_Mns_po = max(0.000, thorn_Mns_po - Tn_po_to_wo + Tn_no_to_po + Tn_co_to_po);

thorn_Mps_po = max(0.000, thorn_Mps_po - Tp_po_to_wo + Tp_no_to_po + Tp_co_to_po);

thorn_Mws_po = max(0.000, thorn_Mws_po - Tw_po_to_wo + Tw_no_to_po + Tw_co_to_po);

```

thorn_Mcs_wo = max(0.000, thorn_Mcs_wo + Tc_co_to_wo + Tc_no_to_wo + Tc_po_to_wo);
thorn_Mns_wo = max(0.000, thorn_Mns_wo + Tn_co_to_wo + Tn_no_to_wo + Tn_po_to_wo);
thorn_Mps_wo = max(0.000, thorn_Mps_wo + Tp_co_to_wo + Tp_no_to_wo + Tp_po_to_wo);
thorn_Mws_wo = max(0.000, thorn_Mws_wo + Tw_co_to_wo + Tw_no_to_wo + Tw_po_to_wo);

# re-calculate substrate concentrations for each compartment following transport
thorn_Cs_co = thorn_Mcs_co/thorn_Mxco; #2
thorn_Ns_co = thorn_Mns_co/thorn_Mxco; #2
thorn_Ps_co = thorn_Mps_co/thorn_Mxco; #2
thorn_Ws_co = thorn_Mws_co/thorn_Mxco; #2

thorn_Cs_no = thorn_Mcs_no/thorn_Mxno; #2
thorn_Ns_no = thorn_Mns_no/thorn_Mxno; #2
thorn_Ps_no = thorn_Mps_no/thorn_Mxno; #2
thorn_Ws_no = thorn_Mws_no/thorn_Mxno; #2

thorn_Cs_po = thorn_Mcs_po/thorn_Mxpo; #2
thorn_Ns_po = thorn_Mns_po/thorn_Mxpo; #2
thorn_Ps_po = thorn_Mps_po/thorn_Mxpo; #2
thorn_Ws_po = thorn_Mws_po/thorn_Mxpo; #2

thorn_Cs_wo = thorn_Mcs_wo/thorn_Mxwo; #2
thorn_Ns_wo = thorn_Mns_wo/thorn_Mxwo; #2
thorn_Ps_wo = thorn_Mps_wo/thorn_Mxwo; #2
thorn_Ws_wo = thorn_Mws_wo/thorn_Mxwo; #2

##-----
# accounting
total_c2 <- (thorn_Cs_co*thorn_Mxco) + (thorn_Fcx*thorn_Mxco) +
            (thorn_Cs_no*thorn_Mxno) + (thorn_Fcx*thorn_Mxno) +
            (thorn_Cs_po*thorn_Mxpo) + (thorn_Fcx*thorn_Mxpo) +
            (thorn_Cs_wo*thorn_Mxwo) + (thorn_Fcx*thorn_Mxwo)

total_n2 <- (thorn_Ns_co*thorn_Mxco) + (thorn_Fnx*thorn_Mxco) +
            (thorn_Ns_no*thorn_Mxno) + (thorn_Fnx*thorn_Mxno) +
            (thorn_Ns_po*thorn_Mxpo) + (thorn_Fnx*thorn_Mxpo) +
            (thorn_Ns_wo*thorn_Mxwo) + (thorn_Fnx*thorn_Mxwo)

total_p2 <- (thorn_Ps_co*thorn_Mxco) + (thorn_Fpx*thorn_Mxco) +
            (thorn_Ps_no*thorn_Mxno) + (thorn_Fpx*thorn_Mxno) +
            (thorn_Ps_po*thorn_Mxpo) + (thorn_Fpx*thorn_Mxpo) +
            (thorn_Ps_wo*thorn_Mxwo) + (thorn_Fpx*thorn_Mxwo)

total_w2 <- (thorn_Ws_co*thorn_Mxco) + (thorn_Fwx*thorn_Mxco) +
            (thorn_Ws_no*thorn_Mxno) + (thorn_Fwx*thorn_Mxno) +
            (thorn_Ws_po*thorn_Mxpo) + (thorn_Fwx*thorn_Mxpo) +
            (thorn_Ws_wo*thorn_Mxwo) + (thorn_Fwx*thorn_Mxwo)

##-----
print("##-----")
print("total_c1")
print(total_c1)
print("total_c2")
print(total_c2)
print("total_n1")
print(total_n1)

```

```

print("total_n2")
print(total_n2)
print("total_p1")
print(total_p1)
print("total_p2")
print(total_p2)
print("total_w1")
print(total_w1)
print("total_w2")
print(total_w2)
print("##-----")

# re-calculate the fractions of mass in plant organs
thorn_Fxco = thorn_Mxco/thorn_Mx; #4
thorn_Fxwo = thorn_Mxwo/thorn_Mx; #4
thorn_Fxno = thorn_Mxno/thorn_Mx; #4
thorn_Fxpo = thorn_Mxpo/thorn_Mx; #4

# the double << make the values global and available outside the wrapper fn.
M[i,1] <<- thorn_Mxco
M[i,2] <<- thorn_Mxwo
M[i,3] <<- thorn_Mxno
M[i,4] <<- thorn_Mxpo

M[i,5] <<- thorn_Fxco
M[i,6] <<- thorn_Fxwo
M[i,7] <<- thorn_Fxno
M[i,8] <<- thorn_Fxpo

M[i,9] <<- thorn_Mcs_co
M[i,10] <<- thorn_Mws_co
M[i,11] <<- thorn_Mns_co
M[i,12] <<- thorn_Mps_co

M[i,13] <<- thorn_Mcs_no
M[i,14] <<- thorn_Mws_no
M[i,15] <<- thorn_Mns_no
M[i,16] <<- thorn_Mps_no

M[i,17] <<- thorn_Mcs_po
M[i,18] <<- thorn_Mws_po
M[i,19] <<- thorn_Mns_po
M[i,20] <<- thorn_Mps_po

M[i,21] <<- thorn_Mcs_wo
M[i,22] <<- thorn_Mws_wo
M[i,23] <<- thorn_Mns_wo
M[i,24] <<- thorn_Mps_wo

M[i,25] <<- thorn_Cs_co
M[i,26] <<- thorn_Ws_co
M[i,27] <<- thorn_Ns_co
M[i,28] <<- thorn_Ps_co

```

```

M[i,29] <- thorn_Cs_no
M[i,30] <- thorn_Ws_no
M[i,31] <- thorn_Ns_no
M[i,32] <- thorn_Ps_no

M[i,33] <- thorn_Cs_po
M[i,34] <- thorn_Ws_po
M[i,35] <- thorn_Ns_po
M[i,36] <- thorn_Ps_po

M[i,37] <- thorn_Cs_wo
M[i,38] <- thorn_Ws_wo
M[i,39] <- thorn_Ns_wo
M[i,40] <- thorn_Ps_wo

M[i,41] <- thorn_Cs_co/thorn_Ns_co
M[i,42] <- thorn_Cs_co/thorn_Ps_co
M[i,43] <- thorn_Ns_co/thorn_Ps_co

M[i,44] <- thorn_G_co
M[i,45] <- thorn_G_no
M[i,46] <- thorn_G_po
M[i,47] <- thorn_G_wo

M[i,48] <- Tc_co_to_no
M[i,49] <- Tc_co_to_po
M[i,50] <- Tc_co_to_wo
M[i,51] <- Tc_no_to_po
M[i,52] <- Tc_no_to_wo
M[i,53] <- Tc_po_to_wo

M[i,54] <- Tn_co_to_no
M[i,55] <- Tn_co_to_po
M[i,56] <- Tn_co_to_wo
M[i,57] <- Tn_no_to_po
M[i,58] <- Tn_no_to_wo
M[i,59] <- Tn_po_to_wo

M[i,60] <- Tp_co_to_no
M[i,61] <- Tp_co_to_po
M[i,62] <- Tp_co_to_wo
M[i,63] <- Tp_no_to_po
M[i,64] <- Tp_no_to_wo
M[i,65] <- Tp_po_to_wo

M[i,66] <- Tw_co_to_no
M[i,67] <- Tw_co_to_po
M[i,68] <- Tw_co_to_wo
M[i,69] <- Tw_no_to_po
M[i,70] <- Tw_no_to_wo
M[i,71] <- Tw_po_to_wo

}
}

```

3 Source sink model behaviour to reductions in pool sizes

Depicted in Figs. 1 & 2 is the modelled behaviour of the new source-sink substrate transport scheme. In an undisturbed system growth is exponential (Fig. 1 A), emergent allocation and plant fractions stabilise (Fig. 1 C,D), growth rates stabilise (Fig. 1 F) and the carbon organ is a source of carbon substrate (Fig. 1 B,E). In contrast, in a disturbed system (Fig 2) where fire has been simulated at time step 100, the system destabilises following fire. The growth rates and fractions of structural mass in plant compartments are altered (Fig. 2 C,D). However, following fire the carbon organ switches from being a carbon source to a carbon sink (Fig. 2 B,E) and draws substrate from the other substrate pools. This substrate induces increased relative growth in the carbon organ while inducing lower growth in the other plant compartments (Fig. 2 F). The rapid regrowth following fire halts transport of carbon substrate to other compartments (Fig. 2 B,E) which halts growth in these compartments owing to lack of carbon substrate (Fig. 2 C, F). Growth in other compartments recommences when the carbon substrate concentration in the carbon is not entirely consumed by its own growth due to lack of other substrates and growth stabilises again (Fig. 2 F).

```
## [1] "##-----"
## [1] "total_c1"
## [1] 4.3
## [1] "total_c2"
## [1] 4.3
## [1] "total_n1"
## [1] 0.925
## [1] "total_n2"
## [1] 0.925
## [1] "total_p1"
## [1] 0.35
## [1] "total_p2"
## [1] 0.35
## [1] "total_w1"
## [1] 2.5385
## [1] "total_w2"
## [1] 2.5385
## [1] "##-----"
## [1] "##-----"
## [1] "total_c1"
## [1] 4.35
## [1] "total_c2"
## [1] 4.35
## [1] "total_n1"
## [1] 1.550001
## [1] "total_n2"
## [1] 1.550001
## [1] "total_p1"
## [1] 0.5999999
## [1] "total_p2"
## [1] 0.5999999
## [1] "total_w1"
## [1] 2.576
## [1] "total_w2"
## [1] 2.576
## [1] "##-----"
## [1] "##-----"
## [1] "total_c1"
## [1] 4.4
## [1] "total_c2"
```

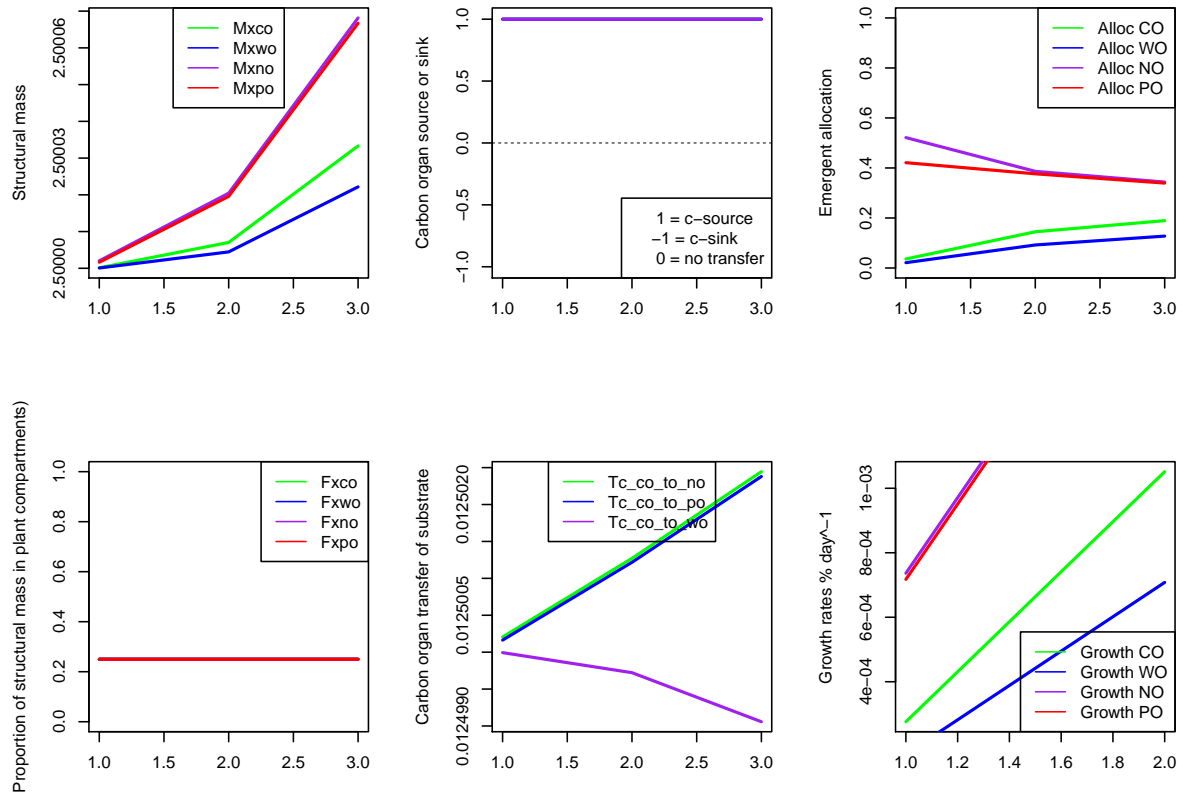


Figure 1: Source sink dynamics with high N uptake and no fire. (A) The structural mass of plant compartments. (B) Indicates whether the carbon organ is a carbon source or carbon sink. 1 indicated a carbon source. -1 indicates a carbon sink. 0 indicates no substrate transport. (C) Emergent allocation depicted as the growth for each compartment divided by the total plant growth. (D) The fraction of plant mass in plant compartments. (E) The transfer amount of carbon substrate between compartments. (F) The relative growth rates of compartments calculated as $((\text{biomass}(\text{day } t+1) - \text{biomass}(\text{day } t))/\text{biomass}(\text{day } t)) \times 100$.

```
## [1] 4.4
## [1] "total_n1"
## [1] 2.175006
## [1] "total_n2"
## [1] 2.175006
## [1] "total_p1"
## [1] 0.8500011
## [1] "total_p2"
## [1] 0.8500011
## [1] "total_w1"
## [1] 2.6135
## [1] "total_w2"
## [1] 2.6135
## [1] "##-----"
```

```

## [1] "##-----"
## [1] "total_c1"
## [1] 4.3
## [1] "total_c2"
## [1] 4.3
## [1] "total_n1"
## [1] 0.925
## [1] "total_n2"
## [1] 0.925
## [1] "total_p1"
## [1] 0.35
## [1] "total_p2"
## [1] 0.35
## [1] "total_w1"
## [1] 2.5385
## [1] "total_w2"
## [1] 2.5385
## [1] "##-----"
## [1] "##-----"
## [1] "total_c1"
## [1] 4.35
## [1] "total_c2"
## [1] 4.35
## [1] "total_n1"
## [1] 1.550001
## [1] "total_n2"
## [1] 1.550001
## [1] "total_p1"
## [1] 0.59999999
## [1] "total_p2"
## [1] 0.59999999
## [1] "total_w1"
## [1] 2.576
## [1] "total_w2"
## [1] 2.576
## [1] "##-----"
## [1] "##-----"
## [1] "total_c1"
## [1] 4.4
## [1] "total_c2"
## [1] 4.4
## [1] "total_n1"
## [1] 2.175006
## [1] "total_n2"
## [1] 2.175006
## [1] "total_p1"
## [1] 0.8500011
## [1] "total_p2"
## [1] 0.8500011
## [1] "total_w1"
## [1] 2.6135
## [1] "total_w2"
## [1] 2.6135
## [1] "##-----"

```

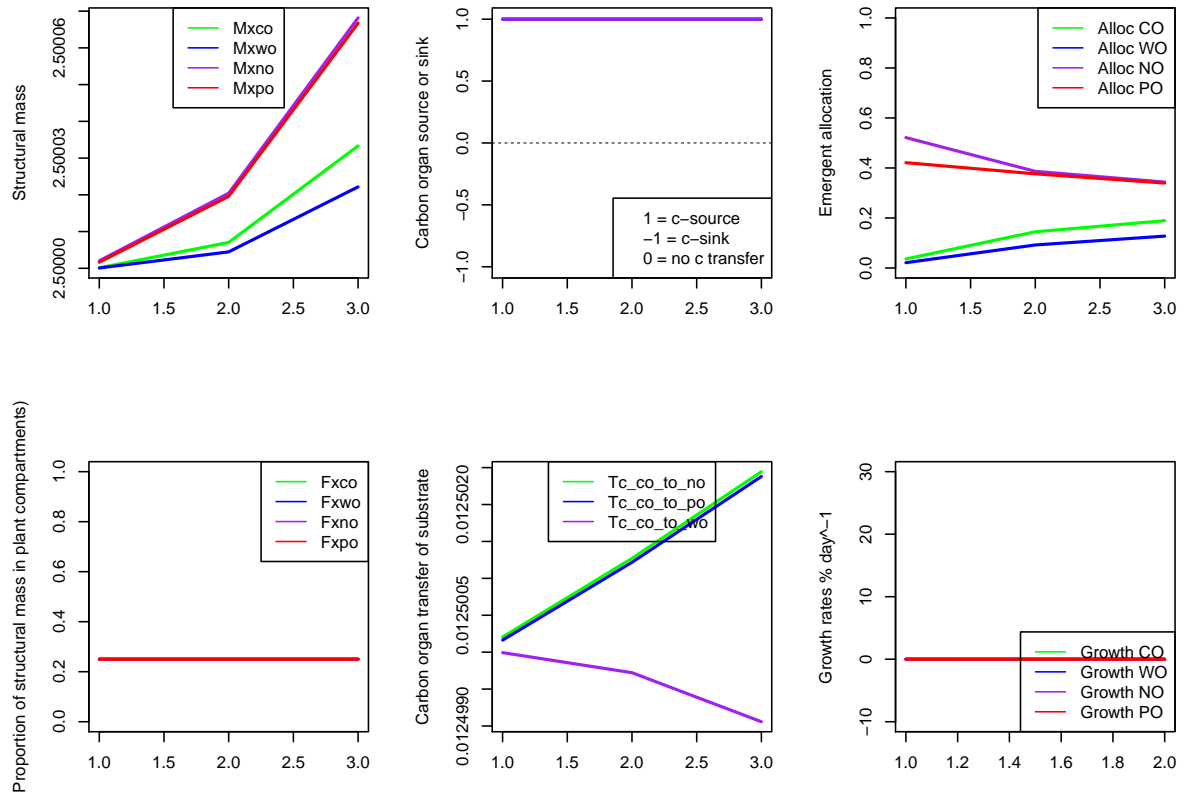


Figure 2: Source sink dynamics with high N uptake and fire. (A) The structural mass of plant compartments. (B) Indicates whether the carbon organ is a carbon source or carbon sink. 1 indicated a carbon source. -1 indicates a carbon sink. 0 indicates no substrate transport. (C) Emergent allocation depicted as the growth for each compartment divided by the total plant growth. (D) The fraction of plant mass in plant compartments. (E) The transfer amount of carbon substrate between compartments. (F) The relative growth rates of compartments calculated as $((\text{biomass}(\text{day } t+1) - \text{biomass}(\text{day } t))/\text{biomass}(\text{day } t)) \times 100$.

References

- [1] M. M. Caldwell et al. “Coping with herbivory: Photosynthetic capacity and resource allocation in two semiarid *Agropyron* bunchgrasses”. en. In: *Oecologia* 50.1 (Aug. 1981), pp. 14–24. ISSN: 0029-8549, 1432-1939. DOI: 10.1007/BF00378790. URL: <https://link.springer.com/article/10.1007/BF00378790> (visited on 10/27/2017).
- [2] E. William Hamilton III et al. “Biomass and Mineral Element Responses of a Serengeti Short-Grass Species to Nitrogen Supply and Defoliation: Compensation Requires a Critical [N]”. In: *Oecologia* 116.3 (1998), pp. 407–418. ISSN: 0029-8549. URL: <http://www.jstor.org/stable/4222101> (visited on 11/02/2017).
- [3] S. J. McNaughton. “Compensatory Plant Growth as a Response to Herbivory”. In: *Oikos* 40.3 (1983), pp. 329–336. ISSN: 0030-1299. DOI: 10.2307/3544305. URL: <http://www.jstor.org/stable/3544305> (visited on 11/02/2017).
- [4] M. Oesterheld and S. J. McNaughton. “Effect of Stress and Time for Recovery on the Amount of Compensatory Growth after Grazing”. In: *Oecologia* 85.3 (1991), pp. 305–313. ISSN: 0029-8549. URL: <http://www.jstor.org/stable/4219515> (visited on 11/02/2017).
- [5] Alexander Schutz, William Bond, and Michael Cramer. “Juggling carbon: allocation patterns of a dominant tree in a fire-prone savanna”. In: *Oecologia* 160.2 (2009), pp. 235–246. ISSN: 0029-8549. DOI: 10.1007/s00442-009-1293-1. URL: <http://www.springerlink.com/content/087637452m011410/abstract/> (visited on 06/14/2012).
- [6] JOHN H. M. THORNLEY. “Modelling Shoot[ratio]Root Relations: the Only Way Forward?” In: *Annals of Botany* 81.2 (Feb. 1998), pp. 165–171. DOI: 10.1006/anbo.1997.0529. URL: <http://aob.oxfordjournals.org/content/81/2/165.abstract> (visited on 11/29/2011).
- [7] John H. M. Thornley and Anthony J. Parsons. “Allocation of new growth between shoot, root and mycorrhiza in relation to carbon, nitrogen and phosphate supply: Teleonomy with maximum growth rate”. In: *Journal of Theoretical Biology* 342 (Feb. 2014), pp. 1–14. ISSN: 0022-5193. DOI: 10.1016/j.jtbi.2013.10.003. URL: <http://www.sciencedirect.com/science/article/pii/S0022519313004955> (visited on 05/15/2017).