NOTE TO FILE:
Garvin H Boyle
Dated: 150201
Revised: Regularly

MppLab Change Diary

1 References

A. OamLab_V1.07.nlogo
B. 150113 NTF Atwood's Machine Revisited R4.docx
C. 150117 NTF Preliminary Design - OAM Lab R2.docx
D. 150201 NTF Preliminary Design - MPP Lab R1.docx
E. MppLab_V1.xx.nlogo

2 Background

OamLab (Ref A) is an agent-based laboratory for exploring the operations of Atwood's Machine (AM) when set up as chains of linked half open AMs (linked HOAMs). For more explanation see Refs B and C. Ref B is the NTF in which the concept of a 'linked half open AM' is developed. Ref C is a brief NTF in which I roughed out the design of the OamLab. MppLab (Refs D and E) is a follow-on project in which I develop a trophic network of generalized Half Open Atwood's Machines (HOAMs).

Legend:

- AM – Atwood's Machine
- OAM – Open AM – open, that is, with respect to energy that flows in when the heavier mass is raised, and flows out when the heavier mass is lowered again.
- HOAM – Half of an OAM – containing either the lighter mass (left half) or the heavier mass (right half). Two HOAMs, each having a different mass, can be linked together to form an OAM, if placed in the correct order.
- RH-HOAM – a temporary assignment of role as a right hand HOAM.
- LH-HOAM – a temporary assignment of role as a left hand HOAM.

In OamLab HOAMs are permanently connected into chains. In MppLab the concept of chains of OAMs in which the permanent association of one HOAM with another is here discarded, and the HOAMs are free to roam and seek to couple energetically (i.e. eat) with whatever prey they can find.

MppLab is an application in which two types of HOAMs are formed. Autotrophs can only be primed with sunlight, and can only act as RH-HOAMs. Heterotrophs can only be primed by eating other suitable HOAMs, notably both autotrophs and heterotrophs. So heterotrophs can act as either RH-HOAMs (as prey) or as LH-HOAMs (as predators). In such a free-wheeling network of linkable halves of Atwood's Machines, what can happen?

3 Purpose

The purpose of this document is to record changes to MppLab application (Ref E) once a working prototype is built.

4 Discussion

4.1 Version 1.09 (170324)

- changed version number from 1.08 to 1.09
- dropped the “I” from the application name to make it just MppLab, and no longer MppLab I. The production of TpLab (Teleological Pruning Laboratory) has removed the need for the intended MppLab II model. The file name is now “MppLab_V1.09”.
- Inserted mention of TpLab into the “info” tab.
- Deleted the code for dpx, dpt and dpg, and refashioned the data dump facility.
- Selected six line graphs for CSV dumps.
- For people who listen to my explanations, it has proven difficult for them to understand that I am using “Mass” as a proxy for some sort of factor that determines the efficiency and effectiveness of energy transfers during digestion of prey. I borrow the role of “mass” from the AM, but I am NOT using it as mass in any other way. Mass happens to determine energy transfers in the AM but plays no similar role in digestion. So, to eliminate that seemingly insurmountable source of confusion, I am going to replace it with a moniker that does not carry with it some persistent baggage of meanings that are not relevant. To that end, I think I will call it the “genetic energy transfer factor”, or “getf”. That says what it is, and does not carry the irrelevant baggage of implied meanings. Nevertheless, the mathematics around its usage in MppLab MUST adhere very closely to the mathematics of the AM, from whence it is derived.
-

4.2 Version 1.08 (150311)

- I uploaded it to my website.
- I made a few edits in the Info tab.
- I saved two copies. One in the MppLab I folder, and the other in the MppLab II folder.
- In the version in the MppLab I folder:
  - I stripped all of the ability of the heterotrophs to mutate or expend energy. They are now just standing flowers fully energized waiting to be eaten, and immediately replaced after being eaten.
  - I wrote the Info tab.
  - I fixed a bug in the hunger sink (x2) and radically altered the way the UB and LB genes function.
  - I changed a number of names of variables.
  - I rearranged the user interface quite a bit.
- It was a major re-work to make it presentable.

4.3 Version 1.07 (150307)

- I gutted a version to make the video better.
- I stored a copy of the version used to make a video.

Version 1.06 (150305)

- I made some aspects of the interface less obtuse, I think.
- I made two scenarios: (1) Herbivores; and (2) Omnivores. Renamed the enumerations variables to start with ge-. I removed the cannibalism switch, as this is now a choice of scenarios.
- I removed all reference to fitness tests, as those are now implicit in the mechanism of the OAMs as they are formed.
- I added some variables to track trophic level. As have a trophic-level-floated = 1. Hs have four variables: (1) no-of-prey-eaten; (2) sum-of-t-plus-one; (3) trophic-level-floated; (4) trophic-level-rounded. Each time an H attacks a prey, it adds 1 + trophic-level-floated to the sum, and adds 1 to the count, then does a division and a round to get the other two.

4.4 Version 1.06 (150303)

- I think my take-away here is that the system can exhibit evolution to many possible weighted average efficiencies, depending on the additional constraints on the system and the biasing effects they have that may mask the underlying MPP efficiency target, but when those constraints are removed, and the MPP is the dominant constraint, the dominant phenotypic character of the system, then the 50% MPP efficiency is expressed by the evolved system. BUT, why is the distribution a bimodal distribution? Is there one more unintended biasing constraint that causes the bimodal distribution.
- So, clearly I had a target efficiency in mind, and I was able to modify the model until I hit the target. Does this mean that this demonstration of the MPP is invalid? I don't think so because most of the modifications have been with the goal of allowing the mechanics of the AM to be exposed to Darwinian selection with as little external bias as possible. But, I do get the curious result that, in herbivorous mode, one weighted average efficiency emerges, and then, in omnivorous mode, another weighted average efficiency emerges. I should run the herbivorous version out to 40,000 ticks as well and see if any structure appears in the distribution of efficiencies. I would hope the efficiencies would have a normal distribution around the mean, or possibly a right-skewed gamma-like distribution.
- At about 4000 ticks, I debugged the g-h-gt-gene and g-h-lt-gene, then turned on the ability of the heterotrophs to eat other heterotrophs, as well as autotrophs, with the same operation that, when finished feeding, the prey dies and is removed from the model. And Voila!!! Within a few hundred ticks the average efficiency of all OAMs jumped to about 50%. A check of the distribution of Eu (efficiency) across all active OAMs showed it to be a uniform distribution, which was a disappointment, as it belied a 'preferred' efficiency. I ran it overnight. It is still very slow. In the morning, at about 40,000 ticks, the distribution of efficiencies has settled to a two-mode, but smooth, distribution, with one tall peak around 0.40 and another smaller peak at 0.90 with a weighted average of 0.50. Wow!
- When cannibalism is off, the gt-gene and lt-gene still mutate, even when not used. They should exhibit a normal distribution around the starting value, due to a 'random walk' like a

Brownian motion effect. But the SD of the smaller gene is proportionately smaller, inserting a kind of bias. So, I changed the basis of the delta in each mutation to 2.5 (vice 1 and 3 respectively), so each gene, as it does its random walk, will have the same SD. As these two genes will have phenotypic effects pitted against each other, I want them to be equally agile in their ability to bring about adaptation. So, they should have the same SD, meaning they have the same ability to migrate their average value.

- With g-h-mutation-factor at 0.40 evolution was too fast and too affected by large random fluctuations, so I altered the mutation rate to 0.1.
- Still a bit slow. I set the target population size of plants (autotrophs) to 1000 vice 2000 and the population of heterotrophs dropped proportionately. Ran much faster.
- As I add more and more graphs and monitors, the speed of the model is slowing. I added a lot of aggregator variables, compute all counts and averages once only, and access those variables in the graphs and monitor. It runs quite a bit more quickly now.
- Often when a heterotroph is finished feeding the plant (the autotroph) is still alive but with very low energy. When the heterotrophs become numerous enough, there are only low-energy prey available. I changed it so plants die and are removed from the model as soon as the heterotroph finishes feeding. Interestingly, this also seemed to remove a source of obfuscation of the evolution. Now heterotrophs only have access to well-energized prey, and the evolution of the system is not only fast, but smooth.
- RATS!!! Again the average efficiency cruises down past 0.5. But, this time quite smoothly and very quickly. Instead of taking 40-50 thousand ticks, the system evolves about ten times as quickly. I suppose because they are reproducing at ten times the speed, or about every 80 ticks vice 800. I suppose the issue here is this: If the phenotypic character is not related to speed of reproduction, but to something else (such as effective search pattern, as it is in PSoup) then I need to allow 800 ticks for the character to be tested sufficiently by the “law of large numbers”. BUT, IF THE PHENOTYPIC CHARACTER IS SPEED TO REPRODUCE, then the RAT=800 constraint simply obfuscates the character being tested.
- To test the idea I removed the effects of the g-h-RAT, essentially commenting it out of action wherever it was in the code. A heterotroph can reproduce as soon its energy level rises to RET (Reproductive Energy Threshold). Then, I made g-h-RET-factor as high as possible, and the g-h-satiation-factor a little higher: say 0.95 and 0.98 respectively. This means a heterotroph will capture prey and eat insatiably until its energy level exceeds reproductive requirements, at which time it will fission, and the daughters drop to 50% of potential, with a mutation of mass, of the mass-gt-gene, and of the mass-gt-gene.
- In version 1.05 the efficiency still migrates towards zero, cruising right past 0.5 towards 0.1. Here’s a thought. Perhaps the fact that I prevent the heterotrophs from reproducing until they are old enough (age > g-h-RAT) is interfering with the phenotypic effects of being a fast reproducer. I.e. the AM has a tradeoff between volume of energy flow and speed of reproduction. Those that get the most energy the fastest should out-compete those that get less more slowly. But, preventing them from reproducing until they are “mature” is an artificial constraint on speed to reproduce, biasing the results. Then, perhaps somewhat counter-intuitively, forcing them to live longer (up to the reproductive age threshold, or

RAT) seems to bias them towards a very low efficiency. Perhaps removing the role of RAT will allow the efficiency to settle on a median value of some kind.

4.5 Version 1.05 (150302)

- The satiation factor must be at least as large as the RET factor for heterotrophs, otherwise, most fail to reproduce. I put an automatic adjustment in the pre-tick routine.
- I added two genes to control the feeding of predators when cannibalism is allowed. mass-gt-gene and mass-lt-gene have values between 1 and 4, starting at 1 and 4 respectively. They can only attack prey with a mass between (pred-mass * mass-gt-gene) and (pred-mass * mass-lt-gene). Any organism with gt > lt will starve, as no prey will be allowed. Peak efficiency is when prey-mass = 2 * pred-mass. Will these genes evolve to focus on highest efficiency somehow?
- I tested it with a g-acceleration value (acceleration due to gravity) of 1 instead of 10. This dramatically increased the remaining drop time to about 10 seconds. This means every OAM, on average, continues to exist and function for 10 seconds, instead of the 2-3 seconds I get otherwise.
- I added rem-drop-time (remaining drop time) to the OAMs and compute drop-time-ratio as rem/max. I added a few monitors to see drop time stats better.
- I allowed predators a first bite on the tick that they capture prey. This corrected a bias in the Eu due to the default value of 1 being counted in all just-formed OAMs that actually had some lower other Eu value.
- I set the default for g-h-mutation-factor to 0.40 in place of 0.10 to cause the population to mutate much more quickly. This means that, on mutation of heterotrophs, the mass will go up or down by a random factor of from 0% to 40% with an average amount of 20%. Previously, the average was 5%.
- I may not be giving the OAM sufficient time for both time and energy to each play a phenotypic role. I want the trade-off that is implicit in the AM dynamics to be fully effective in implicitly selecting the most fit of predators. I am not sure this is happening in V1.04. Predator and prey should be locked into a feeding OAM for as long as the energy can flow, but only so long as full BITEs flow. Partial BITES reduce efficiency. Short feeding sessions reduce the effects of the time/energy tradeoff. So, I want long feeds with complete bites. With this in mind, I have:
    - Put the hunger level that triggers hunting activity by predators MUCH lower (i.e. the satiation factor) and made it ride just above EPM factor, but no less than 3 * EPM-factor.
    - Adjusted the hunting technique to select the most energetic (highest potential energy) prey in prey-list.
    - adjusted the ‘release of prey’ mechanism to avoid partial BITEs.
    - adjusted the cur-potential of generated autotrophs to be at the maximum level when autotroph population is stabilized.
    - adjusted move to expend EPM of autotrophs only when population is not stabilized.
- removed some comments that were vestigial from ModEco.

4.6 Version 1.04 (150224)

- I made the mutation of the masses more fuzzy (times random-float 1) to try to address the structure in the Eu.
- There is a problem with Eu. It has structure that I cannot explain.
- Moved the commands to clear per-tick data to before autotrophs are generated. (Bug)
- I added the h-EPM-factor to the Eu line graph. I can't really see it on the graph, as it is so small, but this means I can use the graph for data collection so I can formalize the formula for E(EPM).
- The H-EPM-factor parameter can make it vary between at least 0.44 and 0.53. Higher EPM produced higher efficiency. Now, that is counter-intuitive. Now at 225,000 ticks and still stable near 0.50.
- Good news! I ran it overnight to 80,000 ticks (100 generations) and it stabilized at Eu = 0.50.
- Added many monitors and line graphs.
- Rationalized the operations of birth and death stats collection.
- Rationalized the operations of the energy sinks. I expect that energy in will now equal energy out in the accounting of sources and sinks.

4.7 Version 1.03 (150218)

- Good news! I finally got an evolving set of OAMs. Bad News! They evolved to have an Eu of 0.08 and then it stabilized at 0.08. I expected the 0.61, 0.50, or even 0.28 that has come up in recent email. OR, I expected it to continue to evolve downwards towards 0.00. Why at 0.08?
- I tried varying the masses of h on startup, but that just allowed them to eat each other immediately. I reversed this, then went further and disallowed them from eating each other at all when prey are selected. I may need to revisit this decision once I figure out how to get a viable steady-state population of heterotrophs and OAMs to exist.
- I moved all release from OAMs to the death routine. Causes of release are death of predator, death of prey, fission of predator or prey, etc.
- I rebuilt the routines to check for death to be more clear in action. This is now the only place where the b-is-ready-to-die flag is set. All other instances of death-causing conditions only set the cause-of-death flag.
- I added a slider so I can tweak the initial number of h added when I drop them in. Default # is 100.
- The heterotrophs eat all of the autotrophs far too quickly and all populations collapse. I added a switch that adds the deficit in autotrophs in every tick. So, if 1000 a is the target population and there are only 400, 600 are added immediately. With this, I managed to grow a population of 2000 a,1400 h, with a continuous supply of about 600 OAMs all functioning at the same time. That's enough for a good Darwinian competition.
- I made it so energy is expended on every tick, rather than when the MH slams to the floor. The math is easier, and it opens the door for more flexible open energy systems. I don't have to restore energy if an hoam dies before a mass drops.

- I moved the ‘expend EPM’ code to separate functions for both autotrophs and heterotrophs to simplify the move routine, and also to put all of the conditions for expenditure in a single place to make it more comprehensible. The new functions are f-autotroph-expends-EPM and f-heterotroph-expends-EPM.
- Added sliders for key mass-derived heterotroph life function control parms (RET and EPM). Since the energy flow through heterotrophs is closely related to the relative masses of prey and predator, the amount expended per tick (EPM) and the threshold for reproduction (RET) must be scaled to mass.
- Added sliders for both autotroph and heterotroph mutation factors (delta-mass = mass * factor), and for heterotroph satiation factor (if xrg < (factor * max-xrg) then feed.
- Fixed bug in which mass was not transmitted from mother to daughter before mutation.
- Added a “corrective-factor” from empirical inspection of target autotroph population based on analysis versus what actually emerges from the model. Seems to work over the range of allowed target population sizes.
- Renamed all enumeration variables from g- to ge- prefixes. I.e. “global enumerative”. This affected sinktype, cause-of-death and is-in-oam enumeration variables.
- Removed DET (death energy threshold) from model to simplify it, and used zero in all cases and energy threshold for hunger leading to death.
- Moved RET and EPM from global controls to internal life function controls for autotrophs to manage the dynamics of the primary energy source for the ecosystem.

4.8 Version 1.02 (150218)

- Built a two breed version in which the heterotrophs eat whomever they can. Needs tuning and a lot of polish.

4.9 Version 1.01 (150201)

- Built an autotroph-only prototype.

5 Test Results

5.1 (Version 1.04 – 150302)

To get a stable population of OAMs I need to stabilize the autotroph population at about 2000. Then I need to set the h-mutation-factor to about 0.40 to make the evolution happen quickly enough. With h-EPM = 0.0125 the masses evolve to be very small, and the efficiency very low. This means (a) it does not stop at 0.5 on the way down, as I had posited earlier. The “fast reproduction” overpowers the need to transmit much energy. So, now, I wonder if upping the h-EPM factor will change the efficiency at which it converges.

Try h-EPM-factor values of 0.0125, 0.025, 0.050, 0.075 and 0.100

- 0.0125 on Dell Precision –
- 0.025 on Red Laptop –
- 0.050 on Annie’s machine –

- 0.075 on Evo Compaq –
- 0.100 on Del Dimension –

The data is stored, but the results were not spectacularly different, and not worth deeper differential analysis at this point. All converged to a weighted average efficiency of approximately 0.20.

6 Yet to do

- Redo all data exports
- Review all TODOs embedded in the code