NOTE TO FILE:
Garvin H Boyle
Dated: 150120
Revised: Regularly to 170302

OamLab Change Diary

1 References

A. OamLabV1.xx.nlogo
B. 150113 NTF Atwoods Machine Revisited R4.docx
C. 150117 NTF Preliminary Design - OAM Lab R2.docx
D.

2 Background

OamLab is an agent-based laboratory for exploring the operations of Atwood's Machine (AM)when set up as chains of linked half open AMs. For more explanation see Refs B and C. Ref B is the NTF in which the concept of 'linked half open AM' is developed. Ref C is a brief NTF in which I roughed out the design of the OamLab.

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.
- Chain of HOAMs – a set of HOAMs lined up in order that are serially, linked and allowed to operate one after the other.
- Head – the first HOAM in a chain, the only one which cannot be in the role of LH-HOAM.
- Tail – the last HOAM in a chain, the only one which cannot be in the role of RH-OAM.
- Body – any HOAM that is not a head or a tail.

OamLab is an application in which chains of HOAMs are formed, primed with energy, and allowed to run to conclusion, with OAMs being charged, linked, run, and discharged, and delinked in serial order, right to left.

3 Purpose

To record changes to OamLab application (Ref A) once a working prototype is built.

4 Discussion

4.1 Version 1.10 (170328)

While preparing TpLab and MppLab for upload to the NetLogo commons, I went back and reviewed and updated OamLab to be consistent with what I have discovered, and to dig a little deeper into the insights that this model can provide me. I have been away from it for over two years, and expect to find a few things I can adjust and/or improve upon.

- Changed the version number from 1.08 to 1.10 in two places
- Set default of g-no-of-chains-at-start to 30 from 15.
- Deleted the slider g-no-of-chains-max and made is standard at 30. Two sliders for the one parameter was too much, and too confusing.
- Set default g-length-of-chains to 14 from 8.
- Set default gs-fitness-measure chooser to “Joint”.
- Left the scenario as persistent across saves and loads.
- Set default switches to on. Plots, mutate heads, mutate tails, horse race.
- “Setup” always reset the operational switches to default values. Removed that. Now those defaults are only restored with the “Reset Defaults” button.
- Added a monitor beside the “Active Fitness Measure” plot to tell me what the “active fitness measure” actually is. It is set in the gs-fitness-measure chooser, but that was not obvious. Now it is, though slightly redundant once you make the connection.
- Re-arranged the location of operational controls in Panel 01 to make them fit into the user interface.
- Added vertical panel dividers between the various panels in the UI 38 chars deep.
- Adjusted all panels to be the same width and depth.
- Removed the optional T junction for debug data. Sent now to both command center and to file.
- Removed the reference to Dpg (data per generation) from panel 03 as that is vestigial. There is no dpg capability in this model.
- After checking the various averages, I have decided to remove some of the more experimental ones.
    - I can average fitness measures across all OAMs per chain, or across all chains per OAM, and I can use arithmetic or geometric averages.
    - Fitness of chains is used for determination of fitness. That is stored in the heads. But it is fleeting, as chains come and go quickly. HOAMs are more persistent, and tracking across OAMs and HOAMs is more easily tracked and interpreted. I have added two graphs that track fitness across OAMs.
    - I had four fitness measures where one is redundant. I removed the redundant fitness measure Mu. Eu and Mu were the same number in all cases.
    - At the chain level I used geometric means, but across the chains I used arithmetic means. I converted all to geometric means, making a system-level calculation a little easier, and less confusing to think about.
    - The interpretation of such system-wide fitness measures might be a little tricky. I need to think about that.

- I added a panel and regrouped the graphs and resized them to be more intuitive.
- I added four standard lines in all graphs respecting fitness criteria: 1.000, 0.618, 0.500 and 0.300. For some reason that I don't understand the fitness criterion Mj often evolves to 0.300. I need to research the mathematics of that.
- I tidied up the debug scripts.
- I debugged the dpt and dpx csv output. Found a couple of minor errors, mostly due to the changes to the aggregate averages calculated.
- I revised the "Info" tab to be consistent the info tabs of MppLab and TpLab.
- I revised the User doc and High Level Design, making minor changes to spelling and layout within the model as I wrote to remove inconsistencies and address ambiguities.

4.2 Version 1.09 (16????)

- ???

4.3 Version 1.08 (150131)

- In several comments in the code old ModEco remnants remain, e.g. references to wrkrs and frmrs. Removed.
- In 'Info' tab changed 'step 2' to 'Step 2'.
- In 'Info' tab changed LH_HOAM to LH-HOAM; changed 'like' to 'link'.
- Corrected all instances of rh-oam to rh-hoam.
- Corrected all instances of lh-oam to lh-hoam.
- Corrected comment code in 'to go' routine.
- Corrected comment code on declarations of Eu, Mt, Mu and Mj. They had the comments associated with the 'lower means fitter' definition. Altered to reflect the 'higher means fitter' version in the released software.

4.4 Version 1.07 (150130)

- Packaged for upload to the 'Modelling Commons' website.
- Added possible extensions of the model to the 'Info' tab.
- Added references to the dispute between Odum and Silvert.
- Added a 'no horse race' switch for faster evolution with no possibility of bias due to the time bias of the horse race.

4.5 Version 1.06 (150127)

- Adjusted some monitors to have names consistent with code.
- Tidied up more comment in the code.
- Converted graph of fitness measures from GofA to AofG. Focus on what is actually used.
- Added a 'Special Scenario' in panel 2. This is the special case in which the fitness measure is energy, the setup is mixed, neither heads nor tails mutate, and the overall fitness never changes, but yet, there is evolution within the chains. Curious!
- Attempted to be more precise about discarding 'worse' chains, but it became very complicated. The problem is, once the system has become mature, there may be many chains as equally fit as the daughter that is to displace them. So, the logic of finding the

least fit that is also less fit than the daughter almost always gives an empty set, even when there is clear room for improvement of the system as a whole. There are two options in this case: do not let the second daughter displace an equally fit chain that is already running; or, let it be displaced. The second option required less changes to the code, which is currently stable. The implication is – in 50% of such instances, the daughter may ultimately be less fit, having been mutated after the fitness tests were done. This may cause (continue to cause) some wobble in the fitness when steady state is reached. But it seems to engender faster convergence to steady-state, sort of like annealing techniques.

- Adjusted 'Info' tab to be consistent with the changes below.
- Added three histograms for Mu, Mt and Mj.
- Changed the three fitness measures. Instead of highest measure being the least unfit, now the highest measure is the most fit. I made them all the reciprocal of the previous measure.
- Clarified the use of words "index" and "measure" in code. Index is access to list. Measure is fitness measure. Changed variable names to be consistent. Iu is now Mu. It is now Mt. Ij is now Mj.

4.6 Version 1.05 (150123)

- Removed all references to DPG (data per generation) CSV output.
- Added serial numbers as unique identifiers of chains, since turtles get used over and over and over.
- Made several changes in variable names to be consistent.
- Completed DPX (data per x-action) CSV output. No need now for DPG. Will remove it.
- Added a small line graph in the main panel that tracks the active fitness measure.
- Completed the programming for the DPT (data per tick) Excel Export. Have yet to do the DPX and DPG exports.
- Added a plot for energy transferred/exhausted per tick.
- Changed "Show-World" to "Export-World" in info tab.
- Added four pens to the Per HOAM graph of Ln(Mass).
- Found a bug in the last usable pen of the PER OAM graph of Ave(Eu). Fixed. Added four pens.
- I sent a copy to Dr Hall and Dr Campbell, and put a copy on my website.

4.7 Version 1.04 (150120)

- I cleaned up the user interface.
- I removed the references to ABM entropy and Gini index.
- I removed the references to aif and noted the CSV output is disabled.
- I wrote the Info tab. (150122)
- I added a chooser that allows me to select the fitness criterion to be used in a run. This means, all together, there are four mutation modes (heads only, tails only, both, and neither), and three fitness tests (time, energy and joint), and three initialization modes (high e, low e and mixed e), for a total of 36 different kinds of test to be run.

- I added monitory for two methods of calculating aggregated fitness measures – eight such monitors in all. Every RH-HOAM can associate with the HOAM on the left to form an OAM. Every RH-HOAM contains four fitness indices for the OAM so formed. I have aggregator variables that hold the sum of a fitness index across all HOAMs, one such sum for each class of HOAMs in the same position In the chain. I also have a geometric average of each such fitness index associate with each chain, stored in the head of each chain. So, I can average across the chains, then up to the top, or along the chains, and then across the heads.
- I learned how to make monitor feed programs, with precision control.
- I designed three fitness tests - time, energy, and joint (both).
- If the only fitness test is for minimum time to discharge, the efficiency always ends up at zero. I need a fitness test that both shortens time and minimizes wasted energy. I can make a time index (discharge time) / (base time) that declines to 1 over time, and a waste index (E-1). The product of these will be the fitness test. If I minimize time efficiency drops. If I minimize waste, efficiency climbs. The product of these two indices should have a trade-off on the interval [0, 1]. I need to add the time index and the fitness test for each chain, and I need to adjust the definition of 'worst' to be based on the new fitness test.

4.8 Version 1.03 (150120)

- I am not happy with the synced behaviour of the chains in a long run. I am not sure this is better, but I changed it to replace the 'worst' instead of the 'oldest'. The total time to discharge is determined when the chain is mutated. I pick the one with the longest time to discharge, and eliminate it and replace it with the mutated daughter of the one that just discharged. This eliminates the worst regardless of their age.
- I added a series of monitors that let me see the efficiency of each class of OAMs in the chain, in real time, as the chains compete. I now have two monitor that show me overall average efficiency. They are calculated two ways. One is the Arith Mean of the Geometric average efficiency of each chain. The other is the Geo Mean of the arithmetic averages of each class of OAMs in the chain. They produce similar answers, but not the same.
- In version 1.02 tails could not mutate their masses, and heads had to. I added two switches that put these under user control. I expect that fixing the size of the tail mass drives the head mass upwards. Fixing the size of the head mass drives the tail mass down. I don't know what to predict if neither or both are fixed in value. When both are fixed the overall geo efficiency of the chain will not change, but will the intermediate efficiencies change? When neither are fixed, ???. Then it floats freely???
- Added a histogram of the natural logarithm of all masses. Curiously, there is a large group of very small masses in a mature population.
- At first, all chains were far too much in sync with each other, so all were of the same age when the first mutated, and it was possible some chains were given extra time, and other minor sources of bias may have happened. I introduced a means to randomize the order of processing chains in the 'do-check-chains' routine.
- Again, to reduce bias-generating sync, the replacement of oldest chain by daughter chains was randomized. Also, the replaced chain was removed from the list of chains to be

checked in this tick so a daughter chain would not be immediately checked in the same tick in which it was born.

- Mass in the head mutated upwards at a very high rate. I had accidentally biased it upwards. The gross bias upwards has been removed. Now, under mutation, the mass in the head randomly mutates upwards or downwards by a scaled amount based on the difference between the head mass and the next hoam's mass. On average, a mutation in either direction will be the same size. However, on average a double 'up' mutation will be compounded, as will an average double 'down' mutation, and the compounding factor will still leave some bias. I may have to revisit this issue. I don't want bias in the mutation size of the head to drive the ultimate efficiency when run to completion.

4.9 Version 1.02 (150119)

- Resolved the last problem to get version 1.02 prototype working as intended.
- In a long run, geometric average efficiency declines to close to zero.
- Not supporting MPP hypothesis. Will look for and eliminate any possible bias.

5 Summary

Nil.

6 Yet-To-Do

- Does the little 'horse race' in the arena cause bias?? I know it isn't needed, but I use it to trigger a fission. And it's cute. This means time-to-drop may be selected against, even if the time-related fitness criterion says a chain is fit. Hmmmm? But, even if that does introduce a bias, it doesn't explain the high Eu when MPP predicts 0.5. If the horse race causes a bias, that bias would push the Eu down, and not up.
- With both heads and tails mutating, and joint fitness test, it should reach a steady state at some point. It doesn't. It appears that the mass per OAM declines very slowly over time. this is the equivalent of genetic drift, but it is not a random walk.