Ensemble simulation overview & other specifications (short)

Madis Ollikainen

June 5, 2015

1 Stefano's comment

I will write my comments in red, so that you can spot them easily.

2 Structure of the joint directory

I now made two directories: *Ensemble* and *Single* to hold the files for the ensemble simulations and the single 'society' simulation. Also I used the word 'society' in the code comments for referencing to one entity of the ensemble. Both of the directories have there own *src*, *include* and *obj* directories, as well as a *makefile* and the other extra files. Thanks, very nice. It's much clearer now.

3 Make, libs and file names

Running *make* in either of the directories should produce a executable named *run_ensemble* or *run_single*. Also the file names now have the prefix referring to either *ensemble* or *single*. Even with the synchronous strategy updating the wealth grow with huge speed, so it is still a good idea to use a multi-precision library for the logit probability sampling. Both of the code use MPFR multi-precision library at the moment. For this it is needed to link both MPFR and GMP multi-precision library. You can also change the multi-precision library to GMP or the only BOOST headers CPP_FLOAT, for that just have a look at *include/mpEXP.h* and uncomment/comment the necessary thing both in the includes and in the *myMP_float* definition sections. Info about the multi-precision libraries and downloads can be gotten from:

- `<http://www.boost.org/doc/libs/1_58_0/libs/multiprecision/doc/html/boost_multiprecision/tut/floats.html>`
- `<https://gmplib.org/>`

- <http://www.mpfr.org/>

Also the configure file has be changed for the ensemble version. It now has an entry NE for the number of 'societies' in the ensemble aka. the ensemble size. And both the ensemble and the single version lack the info about the exponential time sampling (although under the Single directory you can still find the files for the asynchronous strategy updating). Thanks for the explanation, I installed all the libraries, so we can keep using both GMP and MPFR.

4 Ensemble outputting

The ensemble code writes six output files.

1. *parameters.txt* → Re-prints the configure file as before.
2. *talent.txt* → Talent for each player in each society, with the structure:

Player 1 Sc.1	Player 1 Sc.2	Player 1 Sc.3	Player 1 Sc.4	...
Player 2 Sc.1	Player 2 Sc.2	Player 2 Sc.3	Player 2 Sc.4	...
⋮	⋮	⋮	⋮	

3. *time.txt* → Ensemble averages at each time step, with the structure:

Time step	Total wealth	Growth %	Gini coef.	Average co-op/strategy in society
⋮	⋮	⋮	⋮	⋮

4. *wealth.txt* → Total wealth for each society at each time step, with the structure:

Time step	Total wealth Sc. 1	Total wealth Sc. 2	Total wealth Sc. 3	...
⋮	⋮	⋮	⋮	

5. *cooperation.txt* → Average cooperation/strategy for each society at each time step, with the structure:

Time step	Avg Co-op/strategy Sc. 1	Avg Co-op/strategy Sc. 2	...
⋮	⋮	⋮	

6. *GiniCoef.txt* → Gini coefficient for each society at each time step, with the structure:

Time step	Gini coef. Sc. 1	Gini coef. Sc. 2	Gini coef. Sc. 3	...
⋮	⋮	⋮	⋮	

5 Stefano's suggestion for outputting

General comments:

- If you write the main in the way I suggested, you cannot print anymore ensembles as columns and time as a row. You will have to do it the other way round.
- There is no need to print the time steps, because they will be implicit in the number of elements that you write (in this case, the number of elements in a row will indicate how many time steps we had (e.g. for T=35, the length of a row would be 36)

The ensemble code writes six output files.

1. *parameters.txt* → Re-prints the configure file as before.
2. *talent.txt* → Talent for each player in each society, with the structure:

Sc.1 Player 1	Sc.1 Player 2	Sc.1 Player 3	Sc.1 Player 4	...
Sc.2 Player 1	Sc.2 Player 2	Sc.2 Player 3	Sc.2 Player 4	...
⋮	⋮	⋮	⋮	

3. *time.txt* → Ensemble averages at each time step, with the structure: You won't be able to take averages in the main file anymore (due to the lack of saving all the data at the same point). So there is no need for a time file here. The averages will be taken during the analysis phase.
4. *wealth.txt* → Total wealth for each society at each time step, with the structure: again, invert rows and columns and no time

Total wealth Sc. 1 at time 0	Total wealth Sc. 1 at time 1	Total wealth Sc. 1 at time 2	...
Total wealth Sc. 2 at time 0	Total wealth Sc. 2 at time 1	Total wealth Sc. 2 at time 2	...
⋮	⋮	⋮	⋮

5. *cooperation.txt* → Average cooperation/strategy for each society at each time step, with the structure: again, invert rows and columns and no time

Avg-coop Sc. 1 at time 0	Avg-coop Sc. 1 at time 1	Avg-coop Sc. 1 at time 2	...
Avg-coop Sc. 2 at time 0	Avg-coop Sc. 2 at time 1	Avg-coop Sc. 2 at time 2	...
⋮	⋮	⋮	⋮

6. *GiniCoef.txt* → Gini coefficient for each society at each time step, with the structure:

   again, invert rows and columns and no time

Gini-coeff. Sc. 1 at time 0	Gini-coeff. Sc. 1 at time 1	Gini-coeff. Sc. 1 at time 2	…
Gini-coeff. Sc. 2 at time 0	Gini-coeff. Sc. 2 at time 1	Gini-coeff. Sc. 2 at time 2	…
⋮	⋮	⋮	⋮

7. *Efficiency.txt* → Let's compute the efficiency in the c file instead of computing it in Python. We can and it will be faster.

   Remember, efficiency is defined as $\frac{newwealth-oldwealth}{oldwealth}$

   Structure:

Eff. Sc. 1 at time 0	Eff. Sc. 1 at time 1	Eff. Sc. 1 at time 2	…
Eff. Sc. 2 at time 0	Eff. Sc. 2 at time 1	Eff. Sc. 2 at time 2	…
⋮	⋮	⋮	⋮

6 Analysis & plotting

No analysis nor plotting tool is yet available for the ensemble version. Do you have any specific ideas in mind, which kinds of plot would you like to get? Or maybe you think that for the ensemble version the plots will be straight forward from the output data and no special script is necessary? Also, would you like to change anything in the outputting or add anything?