NetLogo BehaviorSpace Tool

Jordan Witte

With additional contributions from:
Melanie Mitchell

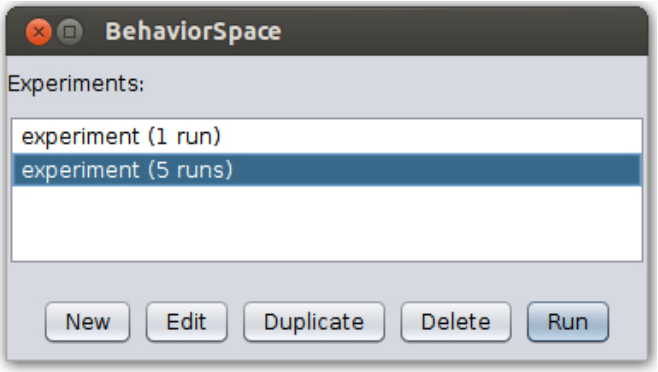
The BehaviorSpace tool is a feature of NetLogo that allows the user to run a model many times across a large spread of parameters. Suppose a model you want to investigate has four settable parameters, each of which can have integer values between 1 and 10. If you wanted to see the results of every combination of parameter values, then there would be 10 * 10 * 10 * 10 = 10,000 combinations to run, which is completely unmanageable to do by hand. Using BehaviorSpace, you can set up this series of tests and record the results of each, and all of the results of the runs will be compiled in a spreadsheet format.

This tutorial is a basic introduction to using the BehaviorSpace tool. For an additional guide that covers some of the more advanced options of BehaviorSpace, see the NetLogo documentation at http://ccl.northwestern.edu/netlogo/docs/behaviorspace.html

This tutorial will demonstrate the operation of BehaviorSpace by way of example, by applying it to the model "Fireflies" from the Models Library.

How it works

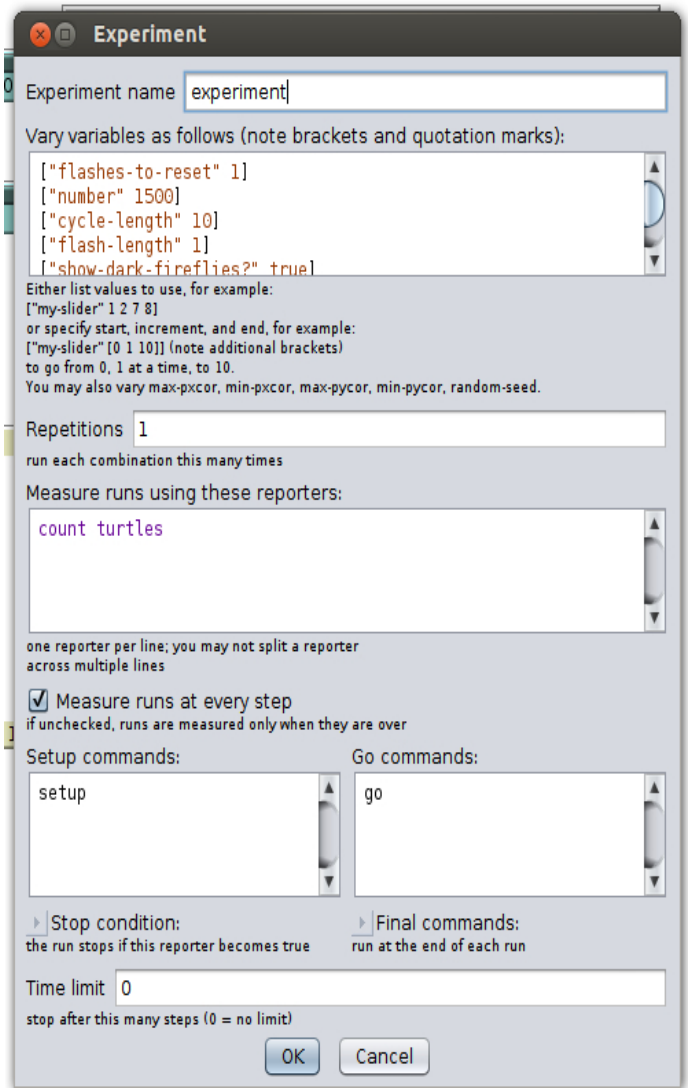
First, open the NetLogo model with which you wish to perform experiments (e.g., Fireflies). Then go to Tools → BehaviorSpace.



The first dialogue box that opens is your list of *experiments*. An experiment is a collection of runs defined by the sets of parameters that will be tested in those runs, and the measurements that will be taken for each.

The box will initially be empty if no experiments have yet been defined for the model.

Create a new experiment with "New". Another dialogue box will appear with default values, looking something like this:



The “Vary variables” window is where you will define the parameter ranges to use during the set of runs. In this window, you can set the parameters of any variables that can by set by the user in the model interface. For example, in the Fireflies model, in this window you could set values for any of the following variables: "number", "cycle-length", "flash-length", "show-dark-fireflies?", "strategy",

For each variable, you are allowed to define a single value, a list of values, or a range with a custom step size. The format used for each is as follows:

	Format	Example
Constant	**[“var-name” value]**	**[“flashes-to-reset” 1]**
List of values	**[“var-name” value1 value2...]**	**[“number” 100 200 300]**
Range	**[“var-name” [min step-size max]]**	**[“cycle-length” [5 1 10]]**

Note that the Range option requires an extra set of brackets in order to be interpreted correctly.

When this dialogue box is first opened, the window will initially be filled with code that sets each variable to a single value, namely, the value that is presently set in the model's interface.

It is possible to leave variables undefined in this window, but not recommended. If any variable name is missing, then that variable will take on the value currently set in the model interface. It is more reliable to simply set some preferred value in the experiment definition.

The "Repetitions" input box defines how many times each parameter combination is run and recorded. Since many NetLogo models contain elements of randomness (including the current Fireflies example) it is often a good idea to record multiple runs and consider the mean and standard deviation of the results.

The "Measure runs..." window lets you define what data gets reported from each run and saved in the exported spreadsheet. In the Fireflies model, we might be interested in the number of flashing fireflies at each timestep. In this case we can copy in the code from that model's Plot object, which is

count turtles with [color = yellow]

This will export to the spreadsheet the same data that is expressed in the model's Plot, namely the number of simultaneously "flashing" fireflies over time.

The boxes for "Setup commands" and "Go commands" are initially filled with the default commands of **setup** and **go**, respectively. Usually you will not need to change these unless you are attempting something advanced with the BehaviorSpace tool.

The "Stop Condition" box defines a condition on which each run of the experiment will halt. The results of an experiment are output only if the experiment halts on its own, so adding a Stop Condition can be very important. If the model you're working with already has a halting condition built into the "go" function, then adding a separate stop condition to the BehaviorSpace experiment may be unnecessary.

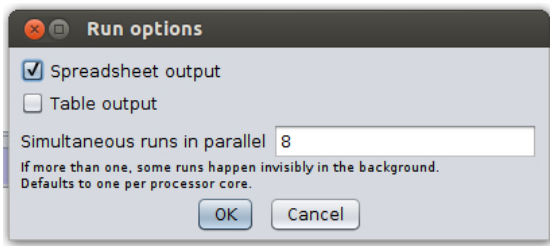
An alternative to having a stop condition in the experiment is to define a time limit (measured in number of ticks). This is set in the final input box of the Experiment dialogue. Depending on the model you're using, you may prefer to set either a time limit or a stop condition, or possibly both.

For example, the Fireflies model will run forever if you hit "go", so in order to effectively export results from our experiment, we'll need to set either a time limit, or a stop condition. Since the interesting question of the model is "how long does it take for all the fireflies to start flashing synchronously?", let's set the stop condition to

count turtles with [color = yellow] = count turtles

We may want to ALSO set a time in this example, since some combinations of parameter values may cause the Stop Condition to not occur in a reasonable time.

Having set all the fields in the dialogue, we can now close it and run the experiment with "Run". Doing so will open a final dialogue box:



BehaviorSpace provides two options for exporting data. The user is encouraged to try both and see which one better suits their task. Both produce files in the .csv format, which can be opened in Excel or another spreadsheet program.

Performing more than one run in parallel may speed up the entire process, since simultaneous runs beyond the first will not graphically update in NetLogo. Again, this should be set as the user sees fit.

After clicking OK, the experiment will run and eventually export everything to a .csv format.

After following this walkthrough, the user should be comfortable with the basics of BehaviorSpace. More information on the tool including advanced features, is available on the NetLogo site at

http://ccl.northwestern.edu/netlogo/docs/behaviorspace.html