SimCo Handbook, v 0.2 (openABM)

The handbook refers to a user-version of SimCo (SimCo-simple.nlogo), which (in comparison to the full version of SimCo) does not include all options of visualization to keep things simple. It should be usable without any further knowledge concerning the method of agent based modelling and simulation or programming at all. Although this handbook is intended to be read by users, in some places we will give additional information for experts on options to extend the software.

If you are interested in extending SimCo, or want to use all output and debug options available, use the standard version (SimCo.nlogo). Both versions use the same internal algorithms and scenario files and will thus show the same simulation behaviour. In case of any questions regarding this manual or the usage and extension of SimCo, please contact the author at fabian.adelt@tu-dortmund.de.

Contents

1   Introduction

This handbook deals with an agent-based simulation framework called SimCo ("Simulation of the Governance of Complex Systems") that aims to improve our understanding of governance issues – especially regarding the control of network-like infrastructure systems with high complexity (macro-level), which results from the interaction of a large number of strategic decision makers (micro-level).

Although designed as a general-purpose framework, we use the case of urban road transportation as a sample application. In our model individual actors choose from various modes of transport, depending on their individual preferences when following daily mobility routines. They act and interact in a network-like transportation structure with nodes and edges, which is an important boundary condition constraining their choices. These infrastructural constraints can also be used as a starting point for governance measures, e.g. by means of road pricing.

Taking SimCo as a training tool leads to a focus on the governance issue of operational risk management. In the case of traffic control, this issue can be translated to reducing undesirable external effects (e.g. pollution) or avoiding a system breakdown (e.g. congestion). The second issue of governance, system transformation, on the contrary is a long term issue and could more easily be examined by using SimCo in automated, long running experiments. Nonetheless, depending on your choice concerning the mode of governance applied, you might discover changes in population even in short run experiments.

2   Preliminary steps and notes

SimCo (simple and full version) as well as the supplementary network-generator model are implemented in NetLogo. To run these models, you have to install NetLogo on your computer. It is available for all mayor operating systems for free and can be downloaded at https://ccl.northwestern.edu/netlogo. Version 5.3.1 is necessary to run SimCo (due to some syntax changes introduced in version 6.0, SimCo will not run with this newer version). All NetLogo extensions used by SimCo are part of the main folder, so you can directly start using SimCo.

In case you are not familiar to NetLogo, you might want to read some introductory comments on http://ccl.northwestern.edu/netlogo/docs/, where you can find a set of tutorials, as well.

This manual uses *italics* to indicate buttons to be pushed in the software interface. Some of the main buttons can be used via keyboard shortcuts. These are printed to the right upper corner of the respective buttons.

Advanced user's note: In order to evaluate experiment results, you could use any spreadsheet or statistics software you are familiar with. Output will be written in csv (comma separated values) files, so the only precondition is that the software of your choice is able to import csv-files.
SimCo's software code is organized in a set of files (some of them are referenced in the following part, file-type is "nls"); these can be opened using the button *includes* within the code tab of NetLogo. You will have to use them only if you want to inspect implementation details or extend SimCo. The latter case will roughly be introduced in Section 3.3.

3   Using the Simulation[1]

Start SimCo by either opening SimCo-simple.nlogo directly from your file explorer or by starting NetLogo and afterwards opening the model SimCo-simple.nlogo via *File/Open.* This will show you the main interface of SimCo which is depicted in the following figure. We will refer to the marked areas in the following sections one by one.
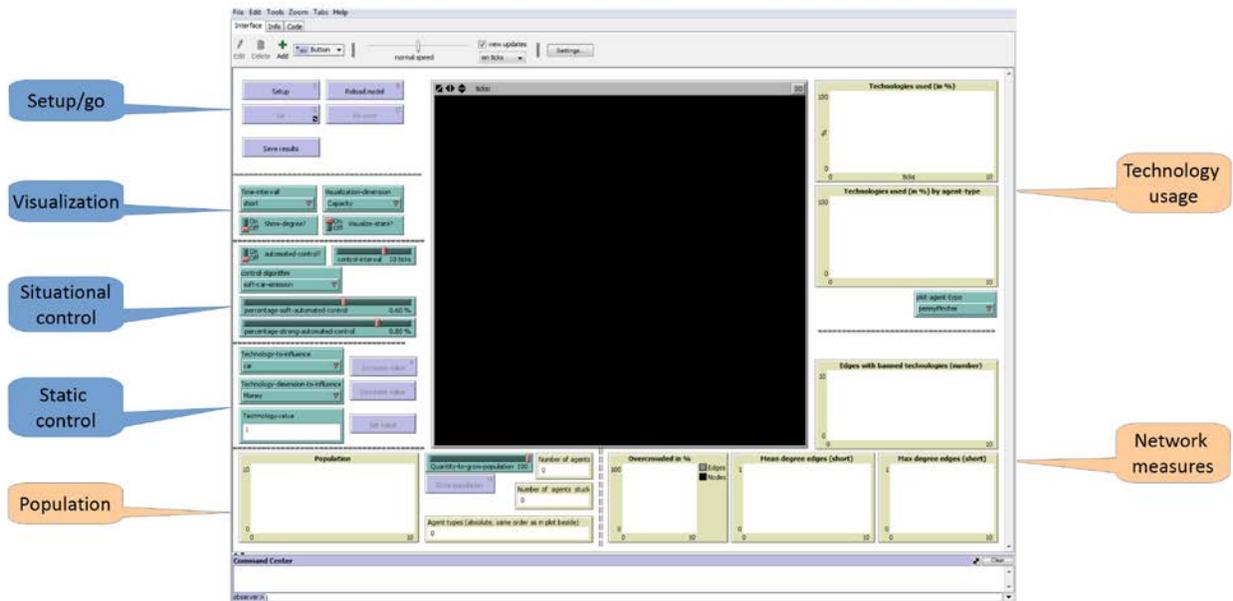
SimCo is just a simulation model and does not contain any scenario data. To use the simulator, you have to load a scenario file in advance. SimCo is shipped with a set of basic, abstracted scenarios from the area of urban transportation. They are stored in SimCo's subfolder "networks" and can be loaded, started, and used as follows. To generate scenarios on your own, please refer to Section 3.2.

3.1   Loading of scenarios

Area *setup/go* contains the button *setup* (S) which gives you the choice to open a standard scenario[2] or to open a file dialogue via answering the question in the popup window with "no".

If you decide not to open the standard-scenario-file, you can choose one of the other scenarios provided with SimCo. They are saved in the subfolder "networks" with file ending snf.

---

[1] We focus on individual use of SimCo on a local, personal computer within this manual. Usage of SimCo on a cluster computer is possible, but not in the scope of this document. In case of any questions, please contact the author at Fabian.adelt@tu-dortmund.de. We have several scripts available for deploying experiments to cluster computers using torque/pbs queuing system.

[2] The standard-scenario suggested in the setup dialog can be altered in setup.nls within function *initialize-setup*. The respective variable is named *standard-scenario-file.* See Section *3.3* for details on working with SimCo's code-section.

All the scenarios shipped with the present version of SimCo follow the same naming pattern, indicating the scaling factor of the scenario by their last part

Scaling factor

In order to enable users of less powerful computers to run the simulation in an adequate time, we generated some down-scaled versions of the complete scenarios. A "half" scenario thus contains 50% of the number of nodes, edges and agents of a "complete" scenario. Furthermore, the lower number of nodes and edges makes it easier to evaluate the networks status on a visual basis.

Loading a scenario should end with a popup window, indicating that everything went well. You are now ready to run your first experiment with SimCo.
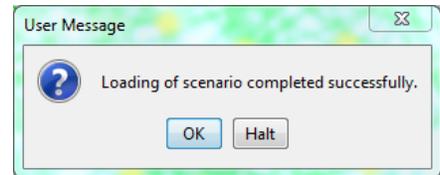

3.2 Running scenarios

To start the simulation, just use the *go* button in area *setup/go.* If you want to run the model stepwise, you can use the button *go once*. Both buttons trigger the same actions, but *go* runs as long as you press it again to stop the simulation. For obvious reasons you cannot press the go-buttons before loading a scenario.

Have a look at the network visualization in the middle of the window and the plots arranged around it. They show different views on the situation which will change in course of time based on the agents actions. You can use the interface elements in area visualization to alter the network's visual representation.

Just wait a few moments for the simulation to reach its standards behaviour: All agents start on nodes in parallel and the first technology they use is chosen randomly. But after just a few ticks (the "time unit" used by NetLogo, printed just above the visual network representation), agents won't go synchronous anymore, as they start to use individual technologies with different velocities and move on various edges.
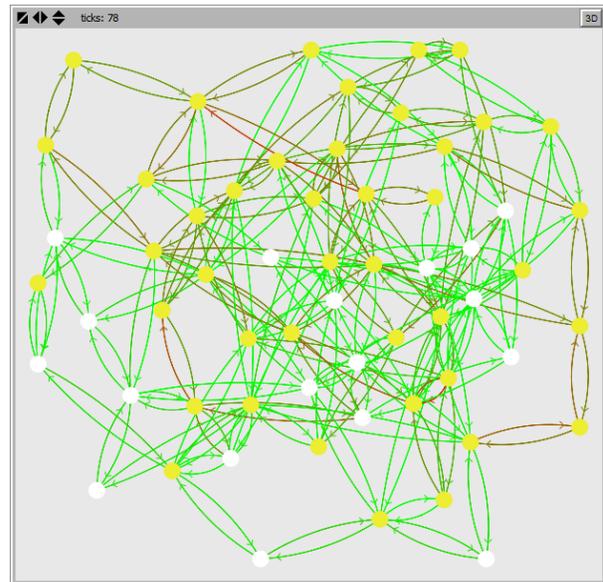
Network-visualization
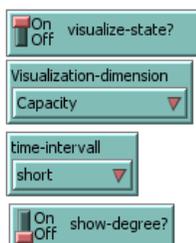
The switch *visualize-state* lets you choose whether to show the standard visualization based on the current status of the simulation – as described below– or based on the nodes' and edges' types. The latter option might be interesting if you want to explore the structure of the graph instead of its state.

Colouring of the nodes simply indicates whether any agent is actually located on that node (yellow) or not (white).

Colouring of the edges is semantically more powerful and can be changed based on your needs and interests. The chooser *visualization-dimension* gives you a choice on different aspects to be visualized. These aspects are, so to say, different levels of the scenario – and of the graph as well. The transportation-scenarios shipped with SimCo entail the dimensions listed below (focus on black ones by now). Their explanation is deliberate short and shall give a hint on the relevance and basic usage of each of them.

- Capacity: Naturally, nodes and edges (representing streets and crossings) are restricted concerning space/capacity. Additionally, technologies differ regarding their size. To sum it up: It depends on agents choices regarding technology, how many agents can use an edge simultaneously. Following, capacity is the one and only aspect equipped with a limit that cannot be exceeded.
- Costs: using technologies to move through the net causes costs. These may differ regarding technologies and may be affected by controlling interventions (see next section).
- Type: helper dimension to distinguish different types of nodes according to agents' tasks.
- Emission: Models the pollution emitted by technologies. A car obviously has higher pollution values than a bike. Limits may be exceeded – and it up to the experimenter (or politics in a more general sense) to decide how to react in such a situation.
- Comfort: depicts different comfort values, mainly used to distinguish technologies regarding their comfort value. A car might be more comfortable than public transport in most scenarios. Values might be affected by controlling interventions.
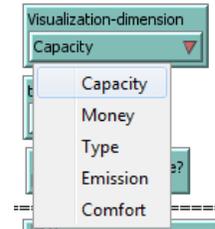
As you can imagine, the dimensions printed in black are those we are interested in when looking at the macro level (i.e. the network status) of our simulation. The other dimensions are nonetheless important, as their values may be influenced by controlling interventions, and their values influence agents when taking their decisions.

Colouring of the edges' visualization is based on limits we can specify for each type of edge and dimension, e.g. concerning capacity or emissions. We use a colour scheme ranging from green (everything is fine, values are far below limits) over yellow (limits could be reached in the future) to red (limits have been reached or even exceeded). If you want to enrich the visualization by numeric values, you can insert the degree-values used for colourization by using the switch *show-degree?*.
Use the chooser *visualization-dimension* to switch to the aspect you want to inspect.

Right below, you will find the chooser *time-interval*. Concerning limits, the model entails two period lengths: A short time period (which could be a day) and a long time period (which could be a month). Having two different period durations allows specifying e.g. day- and month-limits for emission values. The chooser *time-interval* asks the visualization-algorithm to use the correct limit values.
You will presumably use the button to compare short- and long-time emission values. Please note, that colouring based on limits needs to have limits specified. So, if the scenario loaded does not specify e.g. long time limits for emissions, visualization will always stay green – independent of absolute pollution values. But this can only happen if you start building your own scenarios. All the scenarios mentioned above include long time emission limits.

Plots and monitors

In addition to the networks visual representation in the middle of the simulator windows, you can find some plots and monitors in the right and lower area of the graphical user interface, giving additional information on the status of the network as well as some basic agent-based measures. They help to get an aggregated view especially if you experiment with a large scenario, consisting of lots of nodes and edges, where the graph's visualization might look confusing. Let's have a look at the plots and monitors clockwise.

Technology usage (in %)

The first area deals with agents' behaviour concerning their choice of technology. The upper graph shows the whole population's technology usage in course of time, while the bottom graph shows the percent values based on just one agent type's decisions. You might want to change the agent type to be inspected by using the chooser *plot-agent-type*.
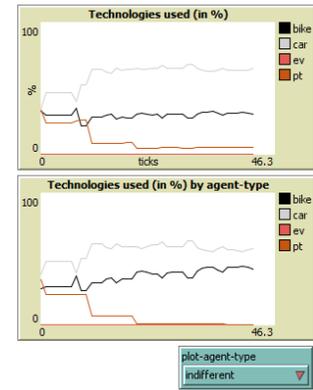
As mentioned above, agents start their journey with a randomly chosen technology, but after just a few moments, they start to decide based on their individual goals and most of them switch to using the car, while just a very few use the public transport.

Network measures

The second area focusses on some network measures in course of time, thus serving as an addition to the main network visualization obviously showing the status in a single moment in time only. Those measures depicted below vary throughout the simulation based on agents' decisions on the one hand, and controlling interventions (depending on the respective mode of governance applied) on the other hand.

As will be shown below in Section 3.3, SimCo provides an algorithm that temporarily bans technologies based on situational parameters like emissions or traffic jams. The plot *edges with banned technologies (number)* shows the count of edges, on which the respective number of technologies is banned, in course of time. Note that the plot does not show *which* technologies are banned, but only, how many of them.
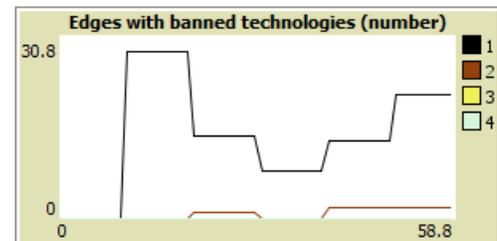
Based on short-time-limits (.i.e. daily limits in the basic scenarios), the plot *Mean degree edges (short)* provides the mean degree values in each dimension specified in the scenario. Note (as mentioned above), that the degree value is 0 if no limits are specified for a dimension, because there is no bad/worst value in this case and we have to treat all values as "good". The example depicted below shows degree values for capacity utilization and emission values. While capacity utilization fluctuates, as agents sometimes are located on nodes (and thus do not count on any edge), the emission degree is continuously rising (until the next short-time-period, i.e. day, starts).

The plot *Max degree edges (short)* right aside functions just the same way and using the same colorization. It allows to easily inspecting values of the worst jammed/polluted edge in the network. Obviously, values are higher than in the *mean degree* plot, and we can see that at least one edge is overcrowded.
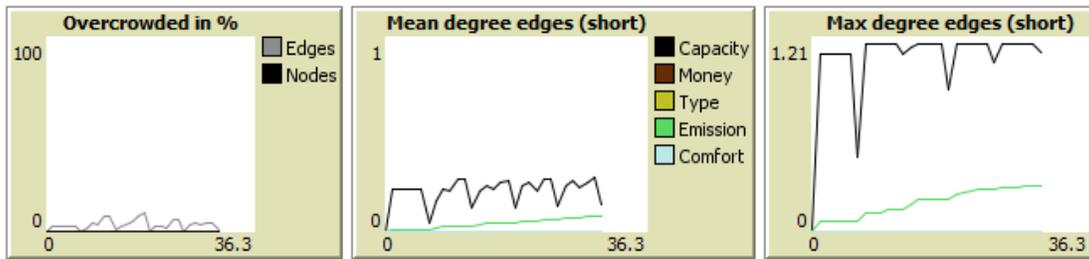
To inspect the proportion of edges and nodes whose capacity is totally used (i.e., they are jammed), just use the leftmost plot *overcrowded in %*, which shows exactly this information. Overcrowding is a situational constraint for agents, because they cannot choose a route using an overcrowded node or edge.

Population

The section in the lower left area of SimCo's user interface finally gives an overview of the simulated population and a handy tool to grow its size.
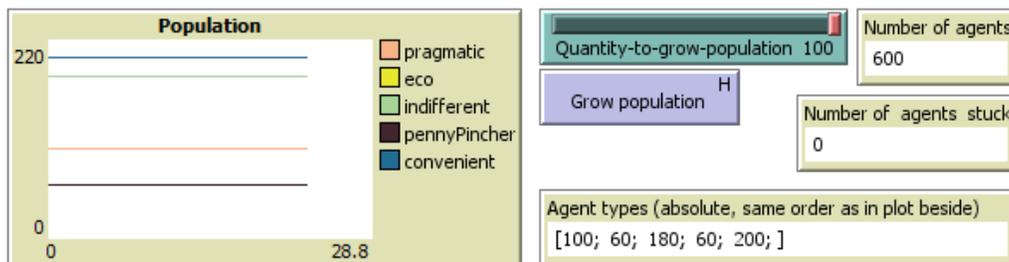
The monitor N*umber of agents stuck* shows the number of agents that cannot move at the moment. This could be caused by infrastructural constraints (like dead ends), the current situation (like a traffic jam in their surroundings, i.e. on outgoing edges) or controlling interventions (like a ban of cars when they can only drive by car). In most situations, this monitor should show the value "0", indicating that all agents may move from starting at their current position.

Monitor *Number of agents* shows the size of the population currently simulated. This value is constant throughout a simulation run, as each agent leaving the simulation will be replaced by a new one. It can change solely of you try to grow the population size, possibly in order to test the carrying capacity of the system. To do so, decide on the number of agents to be added to the running simulation and use the slider *Quantity-to-grow-population* to insert the value to the simulation. Each time you afterwards click the button *Grow population*, the respective number of agents is added. You cannot influence the type of agent using this button, as the type of agent is chosen randomly based on the current population. As an advantage, the proportion of types keeps mostly stable when growing the population.

Mentioning the proportion of agent types in the population finally leads to the plot *Population* and the corresponding monitor *Agent types (absolute, same order as in plot beside)*. The plot gives a fast view on the development of the different agent types in course of time, while the monitor shows their absolute values for the current point in time. Remember, that agents exceeding their limits (e.g. a long time bank account limit) have to leave the simulation and are replaced randomly by one of the agent types still living at that moment. The other cause of changes here is growing of the population as described right above.

The plots and monitors explained above are a good starting point to examine the emerging macro level behaviour of the network, e.g. traffic jams or local and total pollutions. Detecting such unintended, and in a global perspective unwanted, system behaviour is the first step. Now you can try to intervene to the system in order to change agents' behaviour. Can you change the parameters of the simulation such that some – or all – agents act in a way you want them to act? The next section will give a brief overview of possible interventions.

## 3.3   Intervening to the system – how to apply governance measures

In order to conduct experiments on governance and control issues, you may use different instruments of intervention. Do you want to simulate a fixed toll or tax for all cars? Do you want to examine in how far investments to improve the public transport's perceived comfort cause any change in agents' behaviour and thus in e.g. emission values? Or do you want to try out a load dependent congestion charge? All these options are available in SimCo.

As a reminder: You cannot directly influence agents concerning their choice of way or technology (which could be rated as coercion). In fact, like a politician or stakeholder in real life, you can change some system parameters only, which are then perceived by agents and thus possibly influence their decision taking.

You can intervene to the simulation in two different ways: Either manually, applying more or less static measures (depending on your choice to alter values), influencing the whole simulation, or situational based, by parametrizing an algorithm which will act on your behalf. The situational control will affect just those areas (and agents) in the simulation which do not comply with requirements like emission limits. The following sections will introduce the means of governance as well as examples of their usage.

Manual, static control

As a starting point, you may want to increase the car's costs globally, lower the price of using the public transport or raising the bike's comfort values, possibly caused by investments to more bike parking facilities or improved labelling of cycle tracks on roads to be used by cars and bikes in parallel.
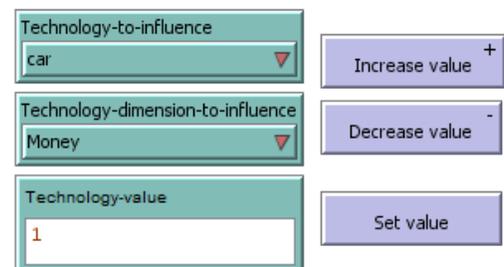
To execute such governance measures (and inspect the changes caused by those actions), you first have to set which technology shall be affected, and what dimension shall be altered. To take the general car toll as an example, set chooser *Technology-to-*influence to "car" and chooser *Technology-dimension-to-*influence to Money as shown in the screenshot above. To finally trigger any change in the simulation, you can either alter the cars' costs by (repeatedly) clicking the buttons *Increase* value or *Decrease value*. The result is shown in the monitor *Technology value*, which simply shows the factor of cost increase (or decrease), as set by you. Each time clicking one of the two buttons mentioned the factor changes by 0.1 upwards or downwards. If you want to give the factor directly instead of clicking many times, just use the button *Set Value*. Of course, you can add another measure and lower the costs of public transport usage afterwards; the settings taken first will remain.

Can you see any change in agents' technology choice behaviour? Remember that they may switch to another technology each time they are on a home node, only. So it may take a moment – and eventually, you will recognize no change at all. Maybe, the comfort improvement of the public transport was too

small, or agents stay at using the car although you raised costs to a (low) degree. And finally, as each type of agent tries to pursue individually rated goals, it might occur that just a fraction of agents switches behaviour – remember that you may visualize the technology usage of single agent types in the upper right area of the graphical user interface.

All measures applied here can be characterized as soft control, and they affect the whole network in the same manner. To improve this – and to add an option to apply strong control within the simulation, we will introduce another type of controlling intervention in the next section.

Automated, situational control

Besides manually intervening to the simulation, you have the option to activate one out of several pre-defined configurations of an algorithm that can change parameters based on the present situation. So, these means of governance do only affect some parts of the network, and the amount and strength of intervention may additionally change in course of time. Examples from practice are manifold, just think about a pollution based toll to be paid by car users, or even a restricted ban of the car in high load situations.
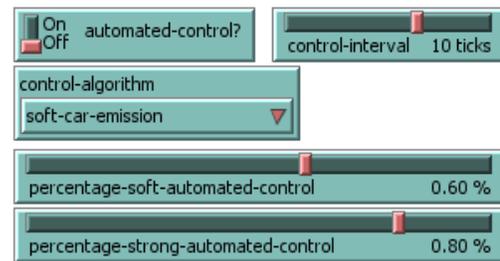
The introductory examples directly lead to the fact that the algorithm for automated, situational control can operate in different modes of governance, namely soft control (e.g. by altering the cost structure), strong control (by banning one or more technologies) or a combined version of both modes, integrating soft and strong interventions, based on the urgency of the situation. In all three cases, agents on the micro level of the simulation might find their own solution to handle the present situation, there are explicitly not totally determined by the situation or controlling interventions, be they caused by human test persons and their manual interventions directly or by the algorithm described in this section.

Besides operating in different modes of governance, the interventions could differ regarding the technology to be influenced. With the present version, the car as main driver of emissions can be addressed. Section 4.3 will explain in short, how easily one can add new configurations for the algorithm without having to implement any new logic to SimCo.

Finally, governance is not an end in itself, but driven by goals mostly on the macro level, be it stable operation or change of the system. Thus, you may decide which dimensions (in our example: capacity/congestion and emissions) shall be used as an input for the algorithm. Of course, you can use both dimensions in parallel, as well, by using the parameter sets whose names end with "all".

To take an example: If you want to implement a congestion based local charge for cars (which is a soft control measure), just set the chooser *control-algorithm* to the value "soft-car-capacity". Or would like to additionally ban cars if raising costs in a high emission situation does not lead to any change in sufficient time (which is a combined control measure)? Just set *control-*
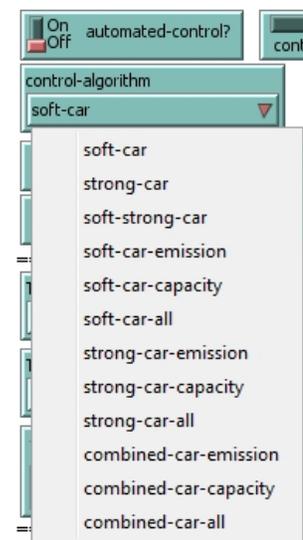
algorithm to "combined-car-emission". The figure beside lists all 12 options available.

The basic experiments presented in the JASSS paper used the first three algorithms "soft-car", "strong-car", and "soft-strong-car".

Finally, you need to activate the algorithm by setting the chooser *automated-control?* to "on".

Now, you can inspect if there are any changes in the system behaviour. And, you can adjust some more parameters in order to influence the functioning of the algorithm: By altering the slider *control-interval*, you easily set how often the algorithm shall examine the systems' state in order to intervene in neuralgic points. A lower value corresponds to more frequent changes of costs and bans – and to a fluctuation of situational constraints perceived by the agents. The two sliders at the bottom of the automated-control section finally are very important. By setting *percentage-soft-automated-control* you can specify, at which level of a daily emission limits (to take "combined-car-emission" as an example here) the algorithm shall start to raise costs. The same applies for *percentage-strong-automated-control* and banning of the car. Hopefully, agents are influenced by soft intervention and change their behaviour. If they do not – and key figures still move towards limits – the algorithm will ban the car once *percentage-strong-automated-control* is reached.

Of course, the bans and tolls will be withdrawn as early as measures fall below the thresholds set with the two sliders as explained above.

It might take some time to show effect when raising costs, as agents to some degree adhere to their standard behavior, irrespective of situational constraints like current costs. Is there enough impact caused by soft control? Or do you want to activate strong interventions additionally or as a single measure? Strong control could be a useful option to avoid emission overload on a street level. Remember, that you can inspect the worst position in the network with the plot *Max degree edges (short)*. But keep in mind, that banning of technologies might restrict agents to such an extent, that they could get stuck on a node. So, maybe, you would like to try a combination of static and situational control.

Combinations
Situational and static interventions are not mutually exclusive – you can combine them, of course. What about an emission based local toll combined with reduced prices for using the public transport and improved comfort for bike users? The simulation enables you to test the effects of such combinations.

3.4   Evaluating
Basic plots for first evaluation of system performance while running the simulation are integrated to the graphical user interface, as already described in Section 3.2. They mainly deal with fractions of technology-usage, measures concerning hard controlling interventions (i.e. banning technologies on distinct edges) and the degree of utilization regarding the different dimensions. Furthermore, the main part of the window contains a visualization of the network. The latter can be altered using the interface elements in area "visualization", changing from state visualization to visualization based on node- and edge types or de/activating visibility of agents.

After running the experiment, use the button *"Save results"* in the setup/go-area. This will trigger function *write-result,* which automatically exports all data gathered to different, mnemonically named files within SimCo's subfolder *"results"*. The file name is based on the scenario file's name you used in your experiment, added by a timestamp to generate individual files each time you end an experiment. These files afterwards can be opened using Excel or any statistical software of your choice[3].

The experimenter should specify in advance which data should be saved, as there are several output methods implemented within *helpers-output.nls*. To reduce the amount of data gained in every experiment, memorizing of such data has to be activated in *setup.nls* at the very beginning of the function *setup-globals,* using the values *"true"* and *"false"*.

4   Expert's usage

The following sections present a rough overview on some additional features of SimCo which might be used by experts when conducting automated experiments or building own scenarios. Further information will be published very soon on the SimCo homepage www.simco.wiwi.tu-dortmund.de, which is currently under development. It will serve as the main point of information for users and developers of SimCo in the future, containing technical and semantical documentations, code updates and discussions. We will use openabm.org to present the next stable versions of the software or new scenarios after validation.

4.1   Setting up and running automated experiments

NetLogo allows conducting automated experiments without manual interventions. This can be used to test a set of different parameter settings easily. To do so, the supplementary tool "Behaviour Space" is included to NetLogo which can be open via the *Tools* menu. An introduction to the tool is provided at https://ccl.northwestern.edu/netlogo/docs/behaviorspace.html.

SimCo is provided with a basic setup for four different experiments, listing all variables needed to perform your first set of experiment. We recommend starting by just using one of these examples or altering it stepwise based on your interests (you can create copies of the basic experiments to do so). Be aware that the base-experiment entails a large number of parameter variations in order to show nearly all options available.

To take an example: Do you want to examine which cost-raising-factor for the car is necessary to show any change in agent behaviour at all? Just ask behaviour space to increase the respective variable *technology-value-helper* in small steps, set variable *technology-to-influence* to "car" and *technology-dimension-to-influence* to "Money" (most variable names are already known from the corresponding instruments of intervention embedded in SimCo's user interface) and analyse the technology usage statistics afterwards.

To examine results, you can use the output options as described in the previous Section 2.4. Just set the respective out-*-variables to "true" in function *setup-globals*, to be found in *setup.nls*, and trigger *write-results* as final command in the experiment's settings. You furthermore might want to specify some variables to be measured using the options integrated to behaviour space.

---

[3] We built some import scripts for such files, as we had to handle thousands of them in automated calibration and governance experiments conducted on TU Dortmund's cluster computer. Contact the author if those scripts could be helpful for you.

4.2 Generating scenarios using SimCo-net-generator

The net-generator-model (simco-net-generator-2.nlogo in the main folder) enables users to simply build individual scenarios based on predefined types of nodes, edges, agents and technologies. Thus, one can easily change the number or type of the basic components of a scenario.

There are mainly two options to create a scenario: starting from scratch or using an existing master-file. A master-file includes all information necessary to rebuild (and possibly change) a scenario.

If you want to start the random number generator of NetLogo with a specific value to get a reproduction of another experiment, you may want to use one of the buttons *set seed-value 1* or *set seed-value 2*. Otherwise, the generator will use an individual seed value each time (which is the standard behaviour of NetLogo).
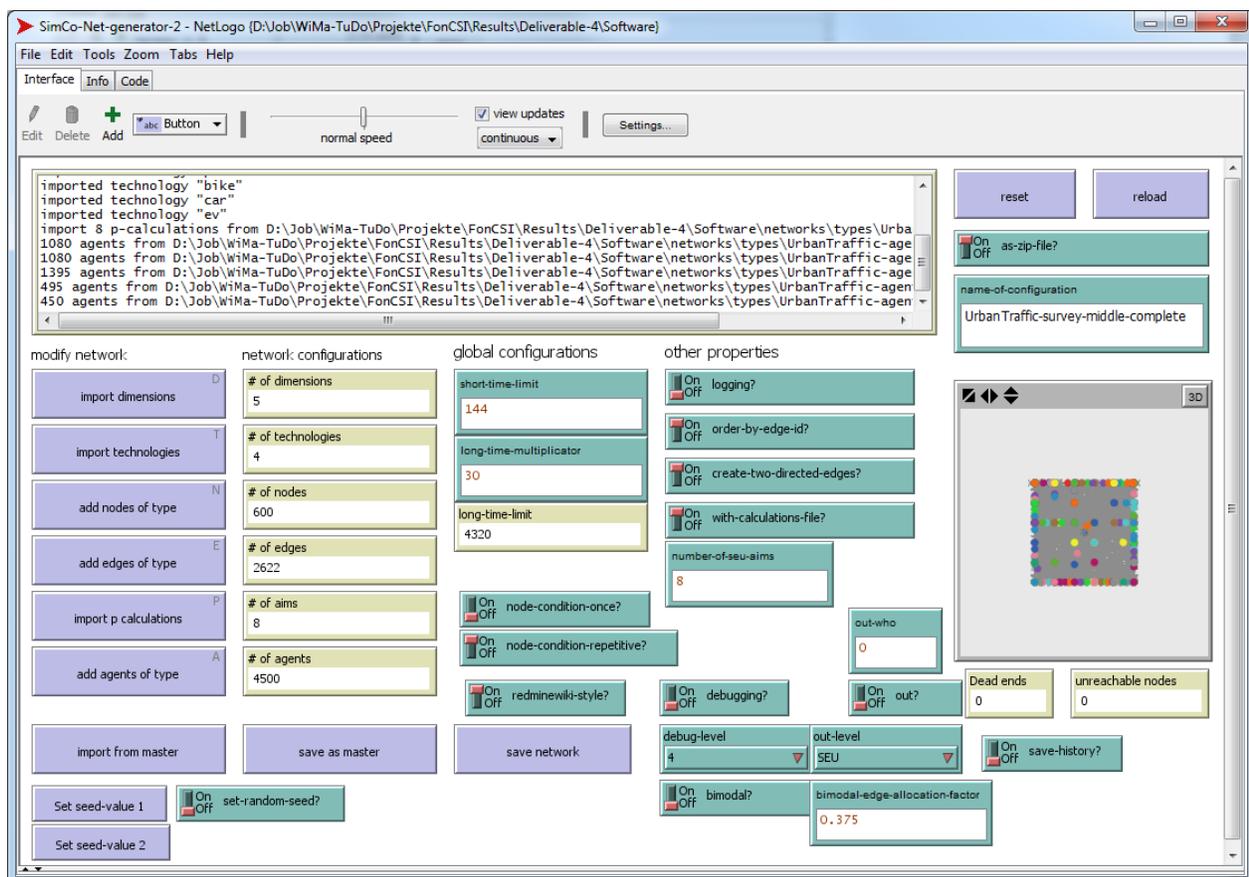
Using a master-file

Import from master opens a file dialogue. Choose a maser file you want to use (they are located in the subfolder networks/masters). The model will load all elements specified in the master-file and build the corresponding scenario. If you want to add elements, you can do so as described above in section "starting from scratch".

Using the button *save as master*, you can save the scenario specification to use it later on again. By altering the master file with a text editor of your choice, you can simply generate different scenario

settings. You can, for example, change the number of actors for each type – which is possibly more convenient than clicking through the whole process described above in "starting from scratch".

SimCo is shipped with master files for all scenario files described in the first section of this manual. Take these as a starting point, of you want to create adaptions of those scenarios.

Starting from scratch

Starting from scratch means using pre-defined types[4] of nodes, edges, agents and technologies in order to create a scenario. The basic approach is to use the six buttons on the very left one after another, starting at *import dimensions* and going up to *add agents of type*. It is common for all of these six options that you have to choose a file afterwards, containing the respective information you want to add to the scenario. These types are saved in subfolder "networks/types" and named mnemonically, so each file containing e.g. a definition of an edge type has "edge" as part of the file name.

- *Import dimensions:* This file mainly contains the names of the dimensions to be used in the scenario. They will be used in the graphical user interface of the simulation. The number of dimension names imported must fit the other elements of the scenario, as the number of dimensions is used in all parts of it. Thus, using the button *import dimensions* several times (loading different files) is only suitable if the total number of dimensions afterwards has the correct value.
- *Import technologies*: Imports characteristics of technologies that should be part of the scenario. Agents refer to the technologies' names later on, so these should fit to each other. This button can be used several times, if technology specifications are split over several files.
- *Add nodes of type***:** Imports characteristics of node types that should be part of the scenario and create as much nodes as specified in the input box presented. This button can be used several times, if several node types should be added to the scenario.
- *Add edges of type*: Imports characteristics of edge types that should be part of the scenario and create as much edges as specified in the input box presented. This button can be used several times, if several edge types should be added to the scenario.
- *Import p calculations*: Imports specifications used to calculate probability-values of actions within agents' seu-based action taking process. Contains mainly function calls to seu-helper functions implemented in SimCo and some parametrization.
- *Add agents of type*: Imports characteristics of agent types that should be part of the scenario and create as much agents as specified in the input box presented. This button can be used several times, if several agent types should be added to the scenario.

Saving a scenario

The field *name-of-configuration* in the upper right corner provides the opportunity to specify a name that will be used for the scenario's file when exporting it. Saving the scenario to disk is done via *save network* in the lower middle of the user interface. Scenarios are saved to subfolder "networks". You should ensure

---

[4] This manual focuses on using pre-defined types. If you want to alter the types available or define your own ones, you have to manipulate the csv-files in subfolder networks/types. Please contact the author if you need further information on this aspect.

that switch *as-zip-file?* right above the field *name-of-configuration* is set to "on" to get just one file, containing all information specifying the scenario you just created.

4.3   Extending SimCo, changing program code
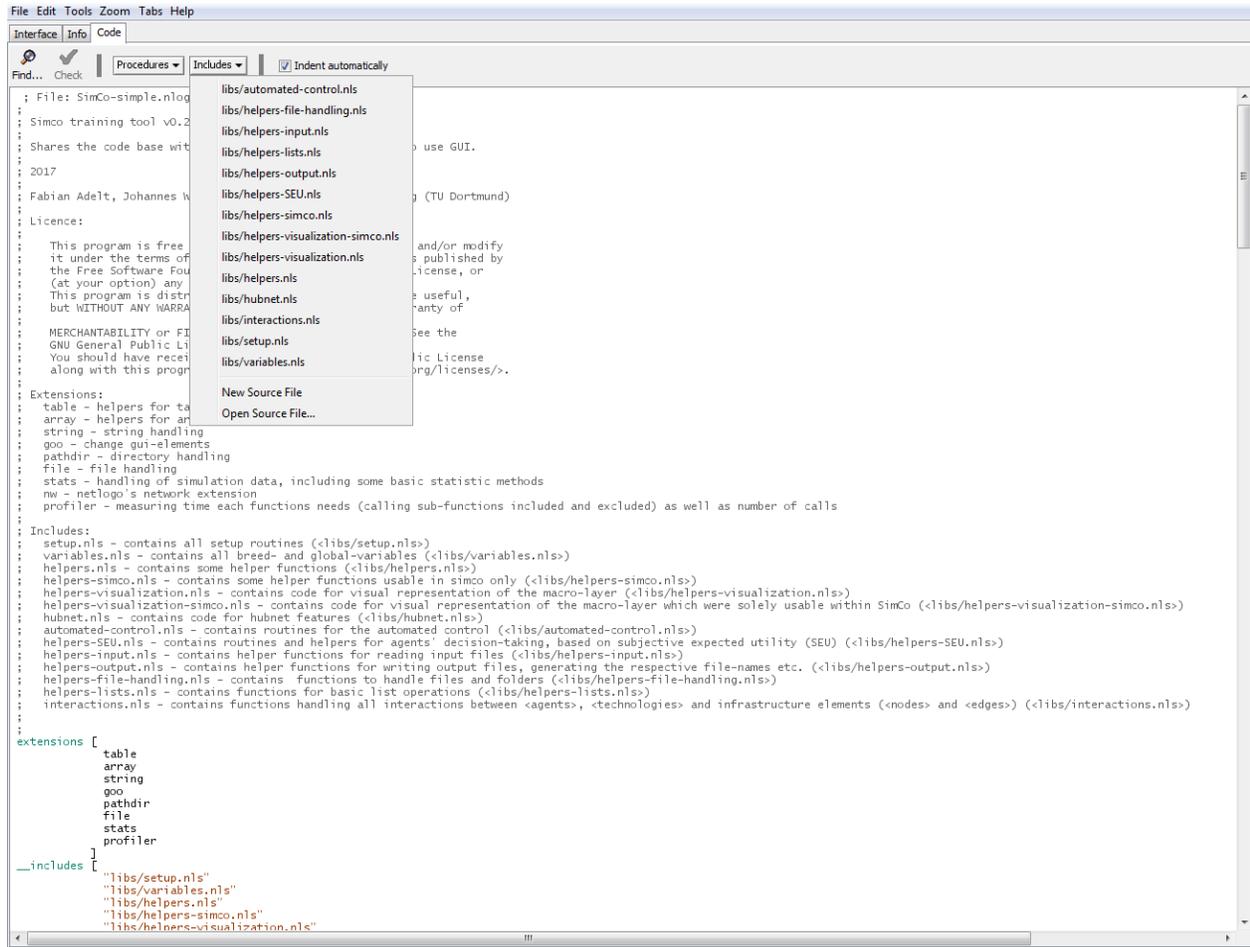
In order to inspect or extend the code of SimCo, switch to the *Code* tab. By clicking the button *Includes*, you can see a list of all files with additional code. We organized code in different files to improve readability and provide a basic order of functions. The very first file *automated-control.nls* for example contains all the functions needed to perform controlling interventions and the algorithm for situational, automated control as described above. We will take this file to show a possible extension of SimCo's governance module.

```
; Function: automated-control-strong-car
; Automated control algorithm, banning technology car from edges and nodes
; if any short time limit is about <percentage-strong-automated-control>,
; re-allowing the car if limit falls below <percentage-strong-automated-control> again.
;
to automated-control-strong-car
  automated-control-edges "strong" "short" "all" "car"
  automated-control-nodes "strong" "short" "all" "car"
end
```

Each function in SimCo is prefixed with a short documentation, stating the main purpose of the function and, if appropriate, the parameters to give as input as well as return values. This helps to understand what happens e.g. in the example above, as we can see the semantics of the four parameters used when calling the *automated-control-nodes*. This information is additionally used to create a complete technical documentation of SimCo which is available in HTML format (including e.g. links between functions and variables as well as a search option, and as PDF export. We used naturaldocs[5] as a tool to extract the information from NetLogo code to HTML.

```
; Function: automated-control-nodes
; Handles automated control (soft and strong) on nodes, based on short-time-limits/degrees
;
; Parameters:
;   my-mode - a string ("strong", "soft" or "both", indicating which type of intervention shall be triggered.)
;   my-time - string, "short" or "long", indicating wether short- or long-time values shall be used to calculate
;             "worst-degree"
;   my-dimension - a string, indicating wether a distinct dimension (given by name) or all ("all") dimensions
;             available shall be used to calculate "worst degree"
;   my-technology-name - a string, indicating wether a distinct technology (given by name) or all ("all")
;             technologies shall be considered for controlling interventions.
;
to automated-control-nodes [my-mode my-time my-dimension my-technology-name]
```
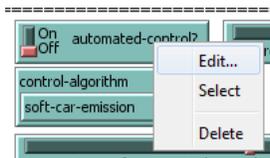
So, if you want to create your own governance algorithm, you can just copy one of the functions yet implemented and change the code, using soft instead of strong control, applying control on edges only or based on long time limits and so on. To enable users to simply use your newly created algorithm, add a fitting block of code to function *automated-control* and add the keyword used to the chooser *control-algorithm* in the graphical user interface. To do so, just click the chooser with the right mouse button, choose *edit* and add a new line with the keyword, surrounded by hyphens.