Vision, Modeling, and Visualization (2014)
J. Bender, A. Kuijper, T. von Landesberger, H. Theisel and P. Urban (Eds.)

Exploring simulation in sensor network models

M. Steiger[1,2], M. Hutter[1], P. Schader[1], J. Kohlhammer[1,2] and A. Kuijper[1,2]

[1]Fraunhofer IGD, Germany
[2]TU Darmstadt, Germany

Abstract

Simulation is an important measure to estimate different properties of a planned network such as throughput and cost. However, many parameters need to be adjusted to approximate real-world conditions properly. In this paper we present a visualization system that visually supports and guides the analysis of (physical) network simulation problems. Automatic optimizers run as a black box giving an (locally) optimal result in terms of the underlying simulation model and parameter configuration. This is often not ideal for practical usage. Our system assists the user in the process of comparing different simulations to quickly achieve the optimal configuration in terms of user preference. It highlights differences between simulation runs and indicates which parameter modification leads to the best improvement. We expect that this results in large time savings for the domain expert while configuring the simulation system.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques Visual Analytics Computational Steering Simulation Optimization

1. Introduction

Over the past years, urbanization has increased significantly, leading to continuous construction and extension of densely populated areas. Every time a new developing area is set up, not only houses and supermarkets are built, but also supply networks that provide gas, electricity and drinking water. These networks need to be laid out so that all customers are supplied at a minimum of cost, time and effort. This is an optimization problem that needs to be solved. Several software-based solutions exist that simulate flow, cost and other quantities in such networks.

1.1. Goal Description

We have identified two tasks for urban planners and other, related domains: the first one is about defining a *new* plan for an optimal network layout based on a set of given constraints. The second one is about improving an existing network configuration to identify solutions that bring the largest benefit at the lowest cost.

Taking existing structures into account is more complex, but comes with the advantage that simulation results can be compared to physical conditions and real measurements. In either case, the simulation system must be configured to match real-world conditions as closely as possible. Only if this is given, the results of an optimization process are reliable and can therefore be trusted.

In general, the default configuration of the simulation system does not match the real circumstances of the user's task domain. She therefore needs to iteratively tweak the parameters until a desired target result is achieved. In this process, it is important to compare different simulation runs to find out which effect a parameter change has on which target variables.

1.2. Visual Support for Simulations

Based on a formal description of the simulation parameters, optimizers can find a configuration that is “best” for one or more variables in a mathematical sense. However, the mathematical optimal result is not always the optimal result from the user's perspective. The user might have a target state in mind, but it is not always clear how to converge from the current state towards the target state: The optimizer may provide an optimal solution, but it will usually not be able to determine whether the *transition* from the current state to the optimal state is feasible. This feasibility can refer to *explicit* constraints that are imposed by the simulation model, for example, the conservation of momentum and energy, or box constraints that are taken into account during the opti-

© The Eurographics Association 2014.

DOI: 10.2312/vmv.20141286

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

mization. But it may also refer to *implicit* constraints that are not part of the simulation model. For example, whether the modification or restructuring of the supply network that is required in order to achieve the optimal state is possible within a certain time or with a certain budget.

Another problem with this is that these simulators run as a black-box and the user is confronted with a large number of input- and result variables, often in tabular form. An appropriate visualization of the network with interactive analysis properties is key to an in-depth understanding of the optimization problem. In the scientific community, this is sometimes referred to as *Computational Steering*: an iterative, interactive process that intertwines user and machine to achieve better results in less time [WBD00]. Closely related is the field of *Visual Analytics* where iterative, alternating human and automatic analysis are combined.

1.3. Overview

In this paper we present an analysis system to overcome the aforementioned limitations: It consists of several, closely coupled views with interaction support, which can help the user to gain a more intuitive understanding about the interdependencies between the input- and output parameters and the behavior of the system as a whole:

- A network visualization that shows the topological and geographic layout
- An simulation browser that provides the configuration history tree
- Drill-down support that adjust the level of detail depending on the zoom factor
- Visual-interactive configuration of simulation parameters

As a result the user can interactively explore different network simulation parameter settings. The *global* effect of changing individual design parameters is visualized in the parameter configuration view. The *local* effects for the individual network nodes are shown in the network view, and changes can be compared on a per-node basis.

The user is supported in the exploration by a visual history of parameter changes. The manual analysis process is documented in the history view, supporting an active, user-steered exploration of the design space.

The rest of the paper is organized as follows: In Section 2 we discuss related work grouped into different categories. Section 3 contains our approach – each subsection discusses one individual view. Before we conclude the paper in Section 5 with discussion and outlook, a case study is given in Section 4.

2. Related Work

This section discusses related scientific publications, split into the different areas of research our work relates to.

2.1. Network Representation

Visual representations of network structures have been studied extensively, with various specializations for different use-cases. We refer to the report of von Landesberger [vLKS*11] for an overview. Our focus is on supply networks with nodes that correspond to physical entities, and their properties that change over time. Similar to our network visualization, Hadlak et al. use embedded line charts to link time series data to graph nodes [HSCW13]. An overview of the design space for temporal graph visualizations can be found in the work by Kerracher et al. [KKC14].

The survey of Cockburn [CKB09] discusses several aspects of the user's focus in graphical interfaces. It also discusses *semantic zooming*, a technique where the level of detail of the visualized entities corresponds to the zooming level. Applied to map navigation tasks, it reduces the task completion time and we therefore employ it in our visualization system.

2.2. Simulation History

In the work of Afzal et al. the impact of user decisions is shown in a simulation of epidemic spreading [AME11]. Similar to our history view, one of the views illustrates the decision history the user made during the simulation and its impacts on chosen target functions. Unger et al. describe a system to visualize different statistical experiments of biochemical reaction networks [US09]. The focus of this work was on comparing different simulation runs. Compared to our approach, it focuses on the results in terms of minimal and maximal extent, not on differences in time or in the parameter space. In the work of Brodlie et al. [BPW*93] a history tree is used, again similar to our approach, to enable the user to backtrack parameter value changes. A recent survey of methods to "visualize alternatives in multiple criteria decision making problems" can be found in the publication by Miettinen et al. [Mie14].

2.3. Parameter Space Exploration

The exploration of parameter spaces for complex simulation models is a task with many practical applications, and a large variety of tools supporting this exploration and analysis process exist.

A visualization for refining a coarsely sampled subspace of the parameter space was realized by Luboschik et al. [LRHS14]. Heterogeneous information between adjacent scales of the parameter and time space of the simulation results is used to enable the user to refine the simulation results at a reasonable level. Their work, however, does not focus on networks. Matkovic et al. [MGJH11] present a visual-interactive system for simulation analysis. They define a fixed set of *control parameters* and *response parameters* for the simulation. Similar to our approach, the control

© The Eurographics Association 2014.

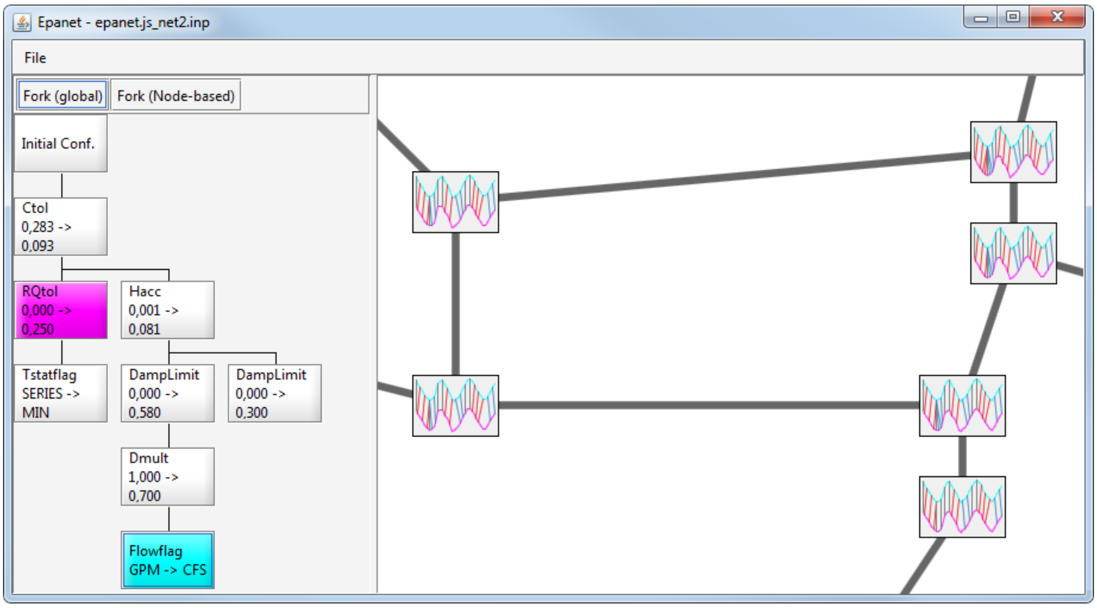


Figure 1: *The main window of the visualization system with its two views. The Simulation History View on the left illustrates the different branches of the simulation parameter changes. The Network View on the right gives an overview on the simulation results distributed in the network topology.*

parameters are sampled with a certain number of variations, but the simulation runs are pre-computed.

The *HyperMoVal* tool, developed by Piringer et al. [PBK10], also aims at the optimization of engine construction, but more focused on the validation of regression models. The work identifies several key tasks, like comparison of results and models, as well as quality estimation and assessment of plausibility. We picked up these ideas by differentiating local and global parameters and their effects in the simulated network.

The concept of *Design Steering*, as a special case for the *Computational Steering* during the design process, was investigated by Wright et al. [WBD00]. Their system allows the user to manually or automatically navigate in a six-dimensional space of design parameters, while building a trajectory that also served as a history of previous design attempts. In our work, all parameters are projected into 2D space for better comparison and also to allow for a higher number of parameters.

A different application case of parameter space exploration was examined by Bruckner et al. [BM10]. In this case, the parameters are shown as circular parallel-coordinate plots, because the main focus of this work was not on the visualization of the *parameter space*, but on the visualization of the *simulation space* and its time characteristics.

An approach for parameter space exploration that is agnostic from the application domain was presented by Berger et al. [BPFG11]. It covers visual guidance to quickly identify interesting parameter regions. This is mainly achieved by sampling parameter values in a certain range and with a certain step size. As a result, the sensitivity and general effects of changing individual parameters can be estimated. This approach is similar to the *Parameter Configuration View* that we describe in Section 3.5.

3. Concept

In this section we will first present the data and the simulation system we are working with. We will then outline the idea behind of our visualization systems and discuss the main views in more detail.

3.1. Data

We start with the simulation data for the visualization system. The underlying data structure is a (geographical) network with one or more time-dependent variables for each graph node. This could be, for example, water pressure sensors in a network of pump stations, or the time-varying demand of water supply at a particular node.

Using a simulation system, we can compute different quantities for the nodes in this network. It can be modeled as a function that takes a set of global parameters and parameters for each node. This input data is transformed into a time-based series of output variables. The data in this case stems from purely artificial data sources, namely from the

© The Eurographics Association 2014.

simulation. Therefore we do not have to take into account measurement imprecision or missing values. It is, however, important to either specify valid parameter ranges or identify illegal configurations. Some parameters might have explicit minimum and maximum values, while some have cross-dependencies to other parameters. In the latter case, this specification is difficult and it is often easier to deal with invalid results. We will describe in Section 3.5 how invalid simulation results may be treated in the visualization.

3.2. Visualization System Overview

The visualization system consists of complementary views and a supporting configuration dialog. The main window consists of the Network View and the Simulation History View (see Figure 1). This window is complemented by the Parameter Configuration Dialog (see Figure 4). All views are discussed in detail in the order they appear in a typical workflow pattern in the following sections.

3.3. Network View

The *Network View* that gives an overview on the simulation network, as shown in Figure 2. In our use cases, the number of nodes is rather limited and usually associated with a geographical position. The natural choice for the visual representation of these networks are *node-link diagrams* with a geo-referenced layout. This makes it easier for the user to identify the nodes and map them to the real entities (pumps, tanks, etc). The user can thus quickly get the 'big picture' of the layout, including domain-specific elements such as tanks, pumps, etc. Nodes are represented by filled rectangles, the links in between are indicated by straight line segments.

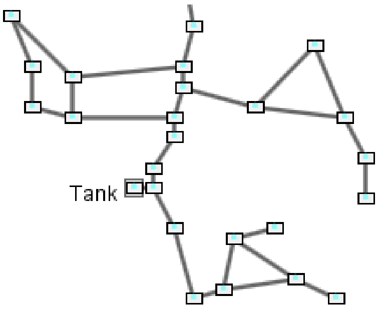


Figure 2: *An initial overview over the simulation network.*

Although the nodes are typically referenced geographically, we do not show a map in the background to keep the visual attention on the network. Geographical layout is not the focus in this application – we therefore chose a plain white background to avoid any kind of distraction.

The user can interact with this view with the mouse, similar to well-known online web map services. Using a virtual camera, the user can pan the view by dragging the mouse cursor and zoom in and out using the mouse wheel.

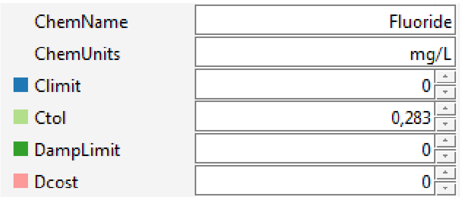
3.4. Forking

In case that the user is not satisfied with this initial simulation parametrization, a new simulation run can be defined. This new parameter configuration is based on the current (in the first iteration the initial) setup, but with slightly different settings. Akin to the natural phenomena and the operation in software revision control systems, we call this operation "fork", because it describes a deviating branch.

There are two different modes: a fork can modify global simulation parameters or based on a single element (i.e. node or edge) in the network. Forking the global configuration is always permitted while the other mode requires the user to first select the element that should be reconfigured. This is done by simply clicking individual nodes or edges.

3.5. Parameter Configuration View

The forking operation is performed using the *Parameter Configuration View*. The parameter set of the underlying simulation is displayed as a series of GUI components based on the data type of the parameters: Categorical values are mapped to combo boxes and Boolean flags to checkboxes. Numerical values are represented as spinning text fields. The mapped parameter model is illustrated in Figure 3.



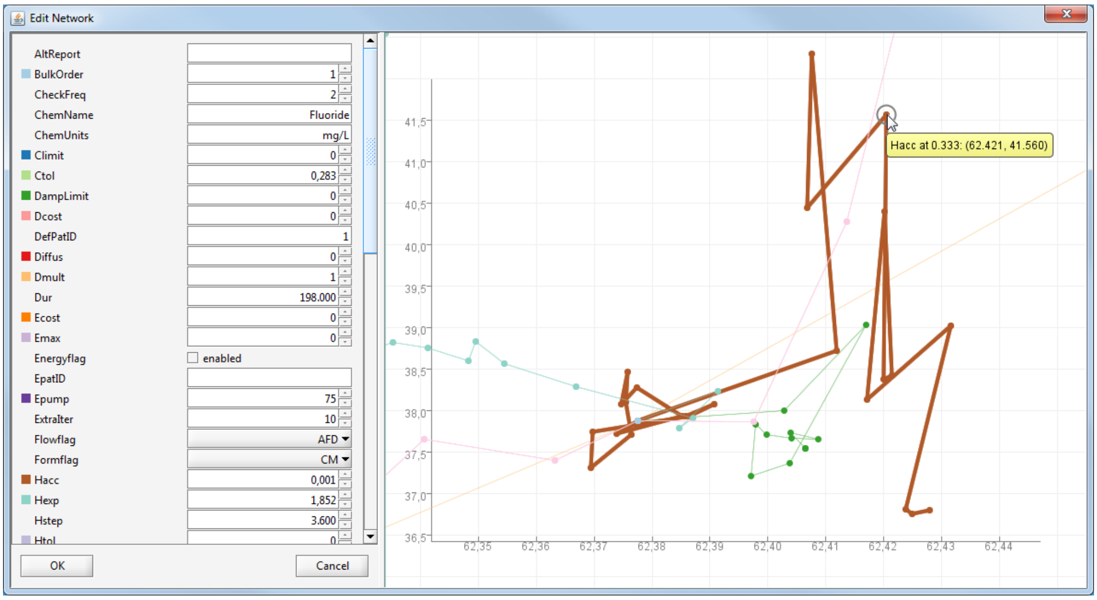
ChemName	Fluoride
ChemUnits	mg/L
🟦 Climit	0
🟩 Ctol	0,283
🟩 DampLimit	0
🟥 Dcost	0

Figure 3: *The parameter configuration model is converted into a set of UI component to enable the user to interactively modify values within valid bounds. Parameters that are continuous are simulated and therefore annotated with a small legend icon.*

The user can then modify the parameter configuration using the UI controls. Our tool assists the user with an interactive preview: The effect of changing parameters with continuous ranges is shown in aggregated form (see Figure 4). Starting from the current configuration, changes in individual parameter values are simulated in a background process and displayed as soon as the value is computed.

The process iterates on every parameter individually, starting from the current value towards both the minimum and the maximum value. The parameter range is sampled at equidistant intervals, but more sophisticated strategies are

© The Eurographics Association 2014.



	AltReport	
■	BulkOrder	1
	CheckFreq	2
	ChemName	Fluoride
	ChemUnits	mg/L
■	Climit	0
■	Ctol	0,283
■	DampLimit	0
■	Dcost	0
	DefPatID	1
■	Diffus	0
■	Dmult	1
	Dur	198.000
■	Ecost	0
■	Emax	0
	Energyflag	☐ enabled
	EpatID	
■	Epump	75
	ExtraIter	10
	Flowflag	AFD
	Formflag	CM
■	Hacc	0,001
■	Hexp	1,852
	Hstep	3.600
■	Htol	0

Hacc at 0.333: (62.421, 41.560)

Figure 4: *The parameter view gives a first impression of the effect a single parameter change has. Each parameter is represented with a unique color, points along a line represent different parameter values. The point location is derived from two global target functions.*

also feasible. As the individual simulations are not dependent on each other, they can be run in parallel, which is key to the interactive display of the view. The view is continuously updated as new values are computed in the background. The tasks for the computation of the simulation results are scheduled in an order that corresponds to a breadth-first search in the parameter space: The tasks with small changes in each parameter are scheduled first, and the tasks for the simulation with the largest parameter changes are scheduled last. This has the effect of the result paths “growing“ starting at the initial state. This allows the user to quickly evaluate the most promising parameter changes without having to wait for all simulation runs to complete.

We can define an arbitrary projection of the result space into the screen space. The simplest is to pick two variables from the result space and use them as x and y coordinates to visualize the outcome of the simulation in a scatterplot-like display. Axes and a grid are drawn in the background to provide information about the actual parameter values and the currently displayed area of the projected result space.

Individual parameters are assigned to unique colors, shown as a legend next to the corresponding UI controls. The most important decision criterion for the set of used colors was to be discriminative on white background. This is why we chose the set of qualitative colors as defined by Harrower and Brewer [HB03].

Depending on the zoom level, individual results are depicted as dots or small circles. Results from consecutive adjustments on the same parameter value are connected with a straight line segment in the color of the corresponding parameter to indicate their relatedness. We refer to these connected sets as *result paths*. We prefer straight lines over splines, because they clearly indicate that no information is available in between two consecutive points.

Hovering the mouse over any of the results provides a tooltip with information about the name and value of the parameter that was changed to achieve the result, as well as optional information about the actual simulation output. Additionally, the corresponding path is painted with a thicker stroke, while the others are painted thinner. This allows the user to focus on the analysis of the behavior of a single parameter even when multiple parameter paths are displayed or have a similar shape.

Depending on the underlying simulation system, information about the feasibility of a given parameter configuration can be incorporated in the output. One option would be to simply omit the corresponding result paths. In other cases, it could be preferable to paint the invalid results in a pale color, so that they are not distracting the user from the feasible results. This would still allow her to see the connections between the results for the case that one parameter was only passing though an infeasible region during the interpolation.

Depending on the validity of the simulation results, the stability of the simulation and the chosen projection of the results into 2D, the points representing the results may be at extreme positions. For this reason, we chose to not perform an automatic normalization, zooming or clipping. Instead, a virtual camera – similar to that one in the main view – was introduced to support interactive panning and zooming. This camera additionally allows to independently zoom the horizontal and vertical axis. The codomains of the result space are both abstract and independent from each other, so the user can freely choose the region of the result space that is to be displayed, focusing on the result path that he is currently interested in.

To summarize, the Parameter View enables the user to view the effect of parameter changes in a single and quick overview. She can find out which parameter changes brings the largest improvement based on the current configuration or tweak a mathematically optimal solution to other, user-defined preferences.

3.6. Simulation History View

Once the user has decided on which parameter to change, the corresponding values are set and the user returns to the main window. This completes the forking operation and the *Simulation History View* is updated accordingly. In this view, an overview over previously computed simulation runs is given in a tree-based hierarchical layout. Starting at the top-left corner with the initial configuration, directly derived configurations are placed one level (on the vertical axis) below. Configurations that stem from the same parent configuration reside on the same level. The first child element is always put directly below the parent element, its siblings are added to the right. Related configurations are linked by elbow connectors. See Figure 5 for an example.

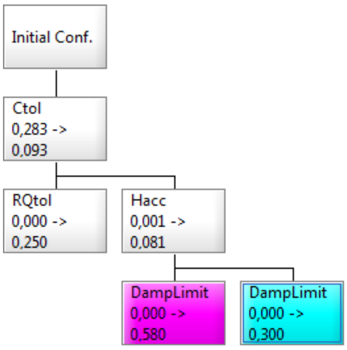


Figure 5: *The History View depicts all simulation parameter configurations that were derived from an initial starting point. This tree-view shows the history in terms of configuration changes. It can be split into different branches and any two configurations can be shown and compared by selecting them (cyan and magenta).*

Two of the configurations can be selected by the user to compare them in detail in the Network View. We chose the colors *cyan* and *magenta*, because this pair fulfills the following criteria: They have similar brightness and saturation (almost 100%), they are discriminative even for color-deficient users and they do not have a intrinsic attribution.

3.7. Network Node Glyph

When the user has selected two configurations, the Network View allows to compare them on a per-node level. The content of the node glyphs depends on the zoom level and the user selection. Figure 6 depicts a node with two time-varying output variables as an embedded line chart. The two line charts have the same color as the selected configurations in the History View. This enables the user to map the line charts to the corresponding simulations. Initially, a simulation based on the default settings is run on startup, and the results are then shown in the node glyphs. When two different simulation runs are selected for comparison, they are drawn in the upper and lower half of the glyph, respectively. Zooming in increases the granularity of the displayed time series.

The third part of the glyph is the display of temporal shifts between the two simulations. We use the approximate Dynamic Time Warping (DTW) approach by Salvador and Chan [SP04] to compute the distance between two time-series data sets. We are, however, not only interested in the total distance, but also in the actual computation of that distance. In contrast to the Euclidean distance, DTW aligns sequences of the data based on a cost function to respect shifts along the time axis. These shifts are displayed in the node glyph by connecting those nodes with the smallest computed distance. If the corresponding indices are identical for both configurations, a thin vertical black line is drawn. Different indices indicate distortions and the line is drawn thicker to raise more awareness. The line color is either *red* or *blue*, depending on the direction of the shift.

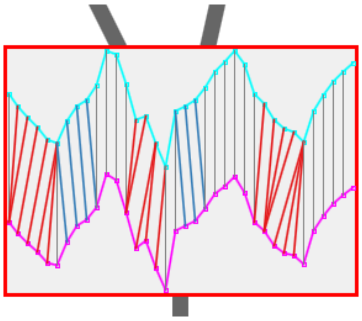


Figure 6: *Comparison of simulation results for a single node. One of the output variables can be displayed in the node glyph (here: pressure over time) The color of the line chart corresponds to the color of the selected configurations. Shifts in the data are highlighted by connecting lines. The line color indicates the direction of the shift.*

In this section, we described the different views and their

© The Eurographics Association 2014.

use in detail. The "Fork" operation creates a new simulation based on the current one. This new branch can be configured using the Parameter View. The Simulation History keeps track of changes while Network View enables the user to analyze differences in the network in detail.

4. Case Study

We performed a case study on the basis of a water distribution piping system in order to evaluate the applicability of our system for optimization tasks, and in order to validate the workflow.

4.1. Setup

The network structure for our test case is taken from an anonymized small real-world data set containing 25 nodes, one tank, one pump and 41 connecting pipes. The public domain simulation software EPANET [US 08] was used to simulate the node pressures, the flow in the pipes and the height of the water in each tank. It was originally developed by the US Environmental Protection Agency (EPA), later continued and ported to Java by the company Addition [Add12]. This software was used to simulate the network for 24 hours in 10 minute sample intervals. Computing such a simulation took less than one second on a 2014 commodity computer.

4.2. Task Description

The goal of the user is to modify the network in order to minimize the fluctuations in the node pressure and increase the stability in the network. He wants to achieve this by replacing a tank in the network with a different one.

4.3. Workflow

The first step is to load the simulation input file containing the network information and the default simulation settings. The *Network View* gives the analyst an overview on the infrastructure. Zooming and panning and the node labels allow to quickly locate and focus on the tank. Selecting the tank allows for forking the initial simulation configuration based on different parameters for the tank.

Therefore, the user opens the *Parameter Configuration View*. Variations of the simulation settings are computed in the background, and displayed in the view. Based on the shape of the result paths, the user decides to increase the size of the tank by 50% to have additional reserve capacity. After changing this parameter, he returns to the main window, where the new configuration is displayed as a new child node of the initial configuration in the *Simulation History View*.

Zooming into the Network View increases the level of detail of the information displayed in the node glyphs to show the time-dependent variables. Selecting two nodes in the Simulation History View reveals the temporal shifts between these properties in form of a dynamic time warping comparison. This shows that the node pressure changed, but there still are large fluctuations in the node pressure. So he continues his exploration with another parameter. The Parameter Configuration View showed a strong influence of the tank base height on the node pressure, indicated by the result path covering a wide range of possible values of the objective function (see Figure 4). So instead of changing the tank size, the user now decided to change the base height of the tank. Selecting the original and the new node in the Simulation History View again visualizes the effect of this change on the network: The fluctuations in the node pressure of the nodes close to the tank become smaller. With increasing distance to the tank, the effect becomes weaker (see Figure 7).

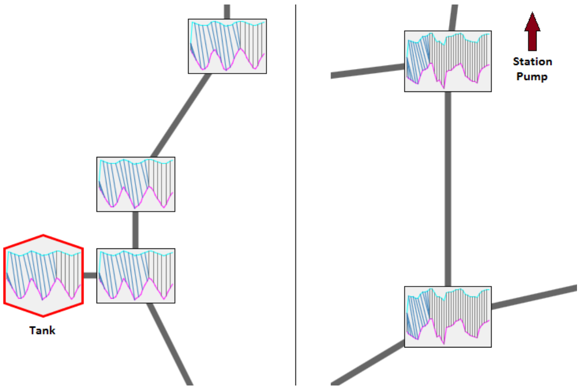


Figure 7: *The nodes that are close to the modified tank are strongly affected by the change while those that are further away and close to the main pump (right image) are almost identical to the previous simulation.*

The actual goal that has been achieved with this modification – namely, reducing the fluctuation of the pressure – could also be achieved with a completely automated optimization. However, there are several advantages when using the visual-interactive approach: First of all, it is not necessary to find a mathematical description of the design goal that can serve as an objective function for the optimizer. The user can gain a more intuitive understanding of the simulation model. Additionally, the process of how the optimization goal was achieved is documented in the history view.

5. Discussion & Outlook

In this paper we demonstrated a visualization system to visually explore different parameter configurations for time-dependent network simulations. Using the combination of Network View, History View and the Simulation Configuration Dialog, the user is enabled to configure the simulation system and keep track of these changes. The influence of individual parameters on the target function and their effects

© The Eurographics Association 2014.

on the time-dependent output variables can be evaluated visually. This may lead to a deeper understanding of the network structure and interdependencies between properties of the individual network nodes, and a more focused and goal-oriented exploration of the parameter space while analyzing and optimizing the network model.

In our current implementation, the edges are represented as simple lines, suggesting an undirected graph. In many real-world applications and tasks, like the water supply networks that we are examining in our use case, the networks are actually *directed graphs*. An appropriate representation of the edge direction might serve as an additional visual aid in order to understand the structure of the network as well as the effects that modifications in the network will have. Holten et al. discuss representations for directed edges in terms of precision and readability [HIvWF11]. For now, we focused on the support of the analysis workflow itself. Further adjustments and improvements of the individual views will be part of our future work.

The most important limitation for interactive exploration of simulation data is the responsiveness of the simulation. In those cases where simulation takes a second or longer, the use of simplified model that is faster to compute could be considered. Similarly, sophisticated sampling strategies could help to reduce the number of runs or increase the quality of the result path.

The parameter visualization view shows two global output variables. When more than two dimensions are required, different visualizations such as a scatterplot matrix might be more appropriate. On the other hand, the parameter changes that affect only parts of the network could be of interest. Then, a small parameter visualization could be shown inside the node glyphs.

Acknowledgements

We would like to thank the experts at Addition Lda. for their friendly support.

References

[Add12] Addition, Lda.: EPANET Java, 2012. URL: `http://baseform.org/np4/tools/epanetTool.html`. 7

[AME11] Afzal S., Maciejewski R., Ebert D.: Visual analytics decision support environment for epidemic modeling and response evaluation. In *VAST* (Oct 2011), pp. 191–200. `doi:10.1109/VAST.2011.6102457`. 2

[BM10] Bruckner S., Möller T.: Result-driven exploration of simulation parameter spaces for visual effects design. *Visualization and Computer Graphics, IEEE Transactions on 16*, 6 (2010), 1468–1476. 3

[BPFG11] Berger W., Piringer H., Filzmoser P., Gröller E.: Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. In *Computer Graphics Forum* (2011), vol. 30/3, pp. 911–920. 3

[BPW*93] Brodlie K., Poon A., Wright H., Brankin L., Banecki G., Gay A.: GRASPARC-a problem solving environment integrating computation and visualization. In *IEEE Conference on Visualization* (Oct 1993), pp. 102–109. `doi:10.1109/VISUAL.1993.398857`. 2

[CKB09] Cockburn A., Karlson A., Bederson B. B.: A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv. 41*, 1 (Jan. 2009), 2:1–2:31. `doi:10.1145/1456650.1456652`. 2

[HB03] Harrower M., Brewer C. A.: Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal 40*, 1 (2003), 27–37. 5

[HIvWF11] Holten D., Isenberg P., van Wijk J., Fekete J.: An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE* (March 2011), pp. 195–202. `doi:10.1109/PACIFICVIS.2011.5742390`. 8

[HSCW13] Hadlak S., Schumann H., Cap C., Wollenberg T.: Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *Visualization and Computer Graphics 19*, 12 (Dec 2013), 2267–2276. `doi:10.1109/TVCG.2013.198`. 2

[KKC14] Kerracher N., Kennedy J., Chalmers K.: The design space of temporal graph visualisation. 2

[LRHS14] Luboschik M., Rybacki S., Haack F., Schulz H.-J.: Supporting the integrated visual analysis of input parameters and simulation trajectories. *Computers & Graphics 39*, 0 (2014), 37 – 47. `doi:http://dx.doi.org/10.1016/j.cag.2013.09.004`. 2

[MGJH11] Matkovic K., Gracanin D., Jelovic M., Hauser H.: Interactive visual analysis supporting design, tuning, and optimization of diesel engine injection. In *Proceedings of IEEE Visualization* (2011), vol. 2011, p. 3. 2

[Mie14] Miettinen K.: Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum 36*, 1 (2014), 3–37. `doi:10.1007/s00291-012-0297-0`. 2

[PBK10] Piringer H., Berger W., Krasser J.: HyperMoVal: Interactive Visual Validation of Regression Models for Real-Time Simulation. In *Computer Graphics Forum* (2010), vol. 29/3, Wiley Online Library, pp. 983–992. 3

[SP04] Stan Salvador, Philip Chan: FastDTW: toward accurate dynamic time warping in linear time and space. In *KDD Workshop on Mining Temporal and Sequential Data* (2004), pp. 70–80. 6

[US 08] US Environmental Protection Agency: EPANET, 2008. URL: `http://www.epa.gov/nrmrl/wswrd/dw/epanet.html`. 7

[US09] Unger A., Schumann H.: Visual support for the understanding of simulation processes. In *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific* (April 2009), pp. 57–64. `doi:10.1109/PACIFICVIS.2009.4906838`. 2

[vLKS*11] von Landesberger T., Kuijper A., Schreck T., Kohlhammer J., van Wijk J., Fekete J.-D., Fellner D.: Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum 30*, 6 (2011), 1719–1749. `doi:10.1111/j.1467-8659.2011.01898.x`. 2

[WBD00] Wright H., Brodlie K., David T.: Navigating high-dimensional spaces to support design steering. In *Visualization 2000. Proceedings* (2000), pp. 291–296. `doi:10.1109/VISUAL.2000.885707`. 2, 3

© The Eurographics Association 2014.