PowerSurge: A Serious Game on Power Transmission Networks

Sebastian von Mammen, Fabian Hertwig, Patrick Lehner, and
Florian Obermayer

Organic Computing, University of Augsburg, Germany
{s.vonmammen,fabian.hertwig,florian.obermayer}@gmail.com
lehner.patrick@gmx.de

Abstract. In this paper, we present an interactive serious game about power transmission systems. The system familiarizes novices with the basic design and behavior of such systems. Using simple drag and drop interactions, power plants and consumers are placed and connected in a virtual landscape that is presented from an isometric perspective. A series of tutorials fosters the user's mastery in building and controlling a complex system. The advanced user is challenged by tasks such as the redesign of an established power infrastructure to integrate a large percentage of regenerative power plants. Next to the interface, we detail the model that drives the simulation. The methodologies presented in this paper can be applied to a wide range of serious games about complex network designs.

Keywords: interactive simulation, serious games, power grid, power transmission, complex systems

1 Introduction

Energy management is an important challenge that governments have to struggle with. The general trend to turn away from climate straining and unsafe technology such as nuclear power and fossil fuels to smaller but renewable energy sources requires a more decentralized energy distribution infrastructure [1]. The turn from all-time available power to regenerative but unsteady resources poses non-trivial challenges: Energy has to be efficiently stored locally to counteract the sporadic absence of wind and solar radiation, see for instance [2] and [3]. On the other hand, transportation of electricity from regions with high yield has to happen efficiently. Adapting the infrastructure is a difficult task, and it is useful to explore the existing capabilities and virtualize any changes before their implementation. A broad overview addressing the challenges of current Power Transmission Systems (PTS) is provided by [4].

In this paper, we present PowerSurge, an interactive simulator for PTS. We detail its user interface, gamification elements and the underlying model that drives the simulation. PowerSurge introduces users who are new to the field of PTS to gain a high-level understanding of the challenges faced in their design

and maintenance, and a feeling for complex behaviors in such networks. Due to the similarity to other complex science themes such as social or economic systems, see for instance [5], the methodologies presented in this paper can be transferred to a wide range of serious games.

In Section 2, we first survey existing power simulation systems, emphasizing their distinctive features. Next, in Section 3, we present our software, including its design principles, its user interface, and a detailed discussion of our domain model. We conclude with a short summary of our results and an outlook on future work and future use of the presented concept.

2   Related Work

Existing simulation and analysis software for power transmission and distribution systems is mainly aimed at industry professionals. Requirements for extensive knowledge in the field and access to data for a transmission network's components pose high barriers to entry for these programs.

PowerWorld Simulator [6], for instance, is a commercial product to interactively simulate large-scale power transmission systems in great detail. While there are educational and research licenses available to make it accessible for non-industry users, the project's source code and development are not open so its extensibility is rather limited.

Other projects like *OpenDSS* [7] and *GridLab-D* [8] are open source projects aimed at research and planning purposes. Both of these tools have extensive simulation and analysis capabilities and support a wide array of grid types and distribution elements, but they are not interactive simulators.

Additionally, all three of these systems require very detailed data about the elements of the transmission system to simulate, as they calculate all data of the network. While properties like reactive current and line frequency – both specific to AC power – are important in real systems, their meaning and impact is rather cryptic to the novice user. Beginners would have trouble finding, for example, the resistance and thermal properties of transmission lines, the efficiency of power plants and transformers, and many more data required to set up simulations in these existing systems.

3   PowerSurge Design

In this section, we detail the design concept of the PowerSurge software, including its user interface, gamification elements [9], and its underlying domain model.

3.1   Visualization of Simulated Units

To simplify the visual representation of the power transmission network, the system is based on the look and feel of a board game. Game pieces which can be placed on the playing field (a map of Germany) are *nodes* that represent power

plants, consumers, distribution nodes and *transmission lines* which connect the nodes to form a network. The nodes are composed of a 3D model to represent the type of the node and a base plate on which additional data can be represented. The models are simplified but the optical characteristics of each node type allow for easy visual distinction by the user. The available node types are shown in Figure 1.
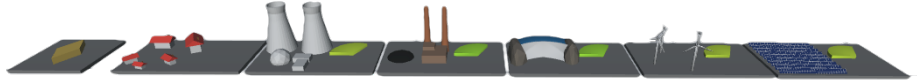

Fig. 1. All node types available in our simulation. From left to right, these are: a distribution node, a consumer node, a nuclear power plant, a coal power plant, a hydro power plant, a wind park and a solar power plant.

The diameters of the transmission lines symbolize their power capacity. The magnitude and direction of the power flow is visualized by the movement speed and direction of the stripes on the connections' surfaces. An example network of the major power plants and cities near Augsburg and Munich is illustrated in Figure 2.
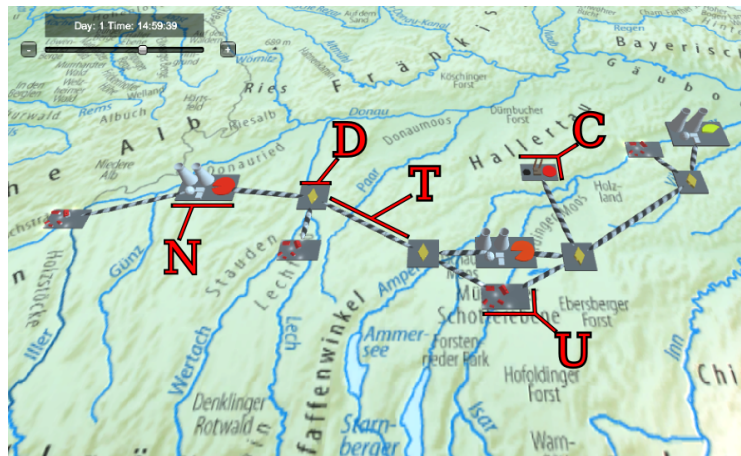

Fig. 2. An example network. The visible game elements are: nuclear power plants (N), a coal plant (C), cities (U) and distribution nodes (D), all of which are connected by transmission lines (T).

We used various techniques to represent the current state of the network. (1) A pie chart on the generator nodes' base plates fills according to the current

load. The indicator also changes its color in a gradient from green to red the closer the load gets to the maximum output. (2) The maximum power output of a node is represented by its size. When a node is dropped onto the field, its size is scaled up or down according to its maximum output relative to the network's overall power generation. Whenever the distribution of power generation among all power plants on the field changes, the relative sizes are updated to reflect these changes. The same technique is used for the transmission lines. The more power it can transport, the thicker the line gets. (3) To show the direction of flow and the amount of electrical power transported by a line, the black lines on each transmission line's surface move in the direction of the power flow. The movement speed depends on the amount of power transferred on this line relative to the maximum power flow of all lines on the network. This way the user can quickly determine which lines transport a lot of power and which lines don't. An example of these relative sizes is shown in Figure 3.
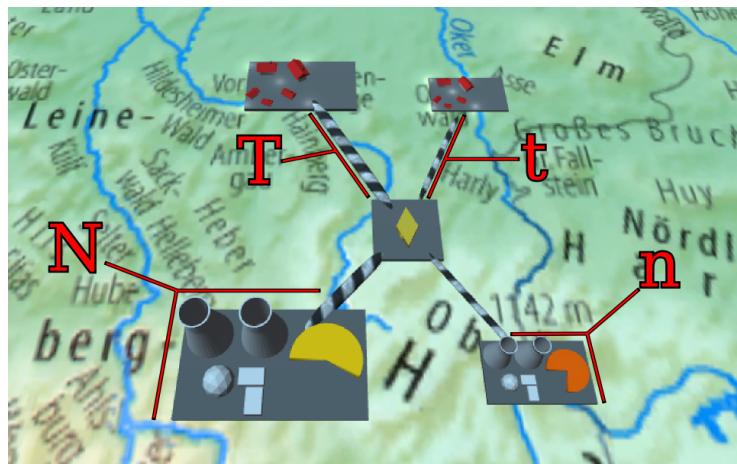

Fig. 3. Nodes and transmission lines are scaled according to their current load or power output. The thicker transmission line (T) currently transports more power than the thinner line (t). Analogously, the larger power plant (N) generates more power than the smaller one (n)

It is possible for a transmission line to be overloaded by a distinct percentage. While this should be avoided and is not a permanent solution for a stable network, it makes the network more resilient toward short bursts of power. In case a line is overloaded, the color of the white stripes on the line's surface turn red to alert the user to the problem.

Introspection of the various simulated units allows to access additional data such as the generator nodes' output (Figure 4), the consumer nodes' power intake, maximum load, and daily load patterns, or the power lines' transported power, their maximal capacity, and their possible overload. For an overview of

the date of the whole network, the sidebar presents various information like the total produced and consumed power, the amount of power lost due to resistance, the percentage of fossil or renewable energy production and further more information.
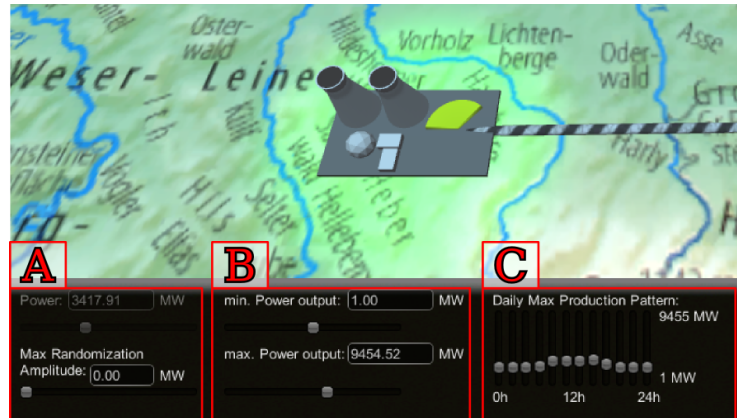

Fig. 4. The information bar for a generator node shows its current power output and the maximum output randomization amplitude (A), the minimum and maximum power output (B), and the daily maximum production pattern (C).

3.2 User Interaction

In our simulation, it is possible to change the parameters of all nodes and transmission lines while the simulation is running, and the user will experience a real time adaptation of the network to the new settings. To add nodes to the system, the user can select the desired type of node from the sidebar and then drop it anywhere on the field, though it is not possible to drop nodes on top of each other. Connections can be drawn between individual nodes, selecting them in *connection* mode. Yellow and red backdrops of the targeted nodes indicate whether or not a connection can be established. By clicking on a node or transmission line, that game piece gets selected, which is indicated by a green highlight around the object. While a game piece is selected, the user may remove it from the game – as long as this action is not forbidden by the game scenario the user is challenged with. This case may also keep the user from inspecting an object and changing all its parameters as seen exemplarily in Figure 4.

3.3 Gamification

In order to encourage the user to explore the simulator the application offers three different stages of the game. We included a set of *tutorial levels* to introduce

the user to the contents of the software, the meanings of the visualizations, the interaction mechanics, and the goals and challenges of the simulated domain. The first tutorial level is exemplarily shown in Figure 5. As soon as the user
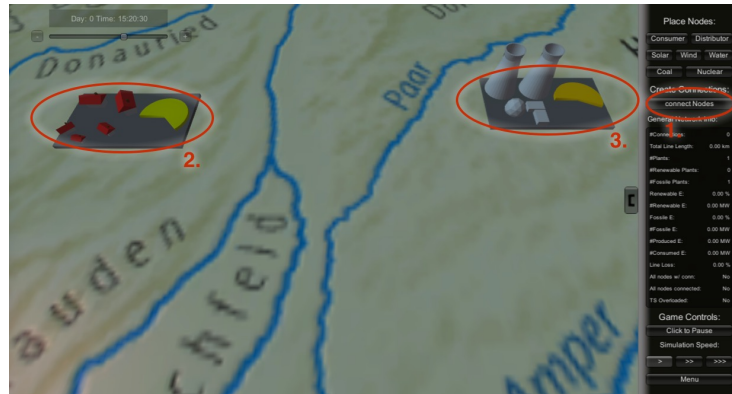

Fig. 5. The first tutorial level to guide the user step-by-step through the interaction mechanics and to familiarize him with the simulated domain. In red, we hint at a sequence of steps that establishes the required connection to move on to the next level.

is familiar with the basic interaction mechanics and the relationships of the simulated units, he can prove himself in scenarios of increasing difficulty. In the according *scenario mode*, the user has to overcome some predefined constraints in order to achieve certain goals. One scenario asks, for instance, to provide at least 20% of consumed power from renewable sources while using no more than 20 power plant and 30 transmission lines supplying a given set of consumers. The user is presented with a list of the subgoals for each scenario before it begins, and he can revisit this list at any time via the game menu. An example of this overview screen is shown in Figure 6. After fulfilling all the subgoals of a scenario, the user is notified and he may advance to the next one. If subgoals are no longer satisfiable, for instance when a time-limit is exceeded, the user fails in that scenario. An appropriate *game over* screen is shown and the user may restart this scenario or switch to a different one.

PowerSurge also includes a *discovery mode*, a scenario free of any constraints, where users can try out various model configurations on their own agenda. When the simulation is running in discovery mode, the user may construct arbitrary power transmission networks with no restrictions on size of the network, resources spent or network composition. Even the otherwise imposed fixed time sequence is now softened: here, the user may go back and forward in time as desired. All produced data, i.e. time series of all the graph's variables, can be logged on disk as portable comma-separated lists for further analysis. Through this interface for scientific evaluation, one could, for instance, measure network

LEVEL GOALS

Current Stats:

Current # of Transmission Lines:	0 | 30	(Maximum # of Transmission Lines)
Current # of Power Plants:	0 | 20	(Maximal # of Power Plants)
Act renewable energy:	0.00% | 20.00%	(Minimal renewable energy over time)
All nodes with a connection:	No | Required	(All nodes must have a connection)

Fig. 6. Exemplary display of the goals of a scenario. These goals are presented to the user before diving into the simulation.

properties, such as topology, robustness (in terms of redundancy and minimum supply) etc.

3.4 Scripting Game Contents

Especially in an educational setting it is important to have the flexibility of defining specific problem scenarios. PowerSurge is designed to be extended accordingly. Goals and constraints of new scenarios can be scripted based on a variety of possible subgoals provided by the engine. An overview of these subgoals is shown in Figure 7. In the following, we describe some of these goals.

The first goal, i.e. *Max Time To Achieve Goals In Days*, sets a global time limit for all subgoals to be completed. Exceeding this limit, results in failing the scenario. The second one determines the minimum percentage of renewable energy produced in the Power Network. Additionally, a period can be set, how long the system has to supply that proportion continuously (using *AVGProduction Period Length*). Consider, for Instance, that at night there is less available renewable power available than on a bright sunny day, when solar radiation can also be harnessed. The goal, *Max Network Failure Time* denote the maximum limit of time a network failure is allowed to last. If *Network Failure Time Stacks* is set, that time will not be reset, if the system stabilizes again. A single-component graph is required, if the goal *Network Graph Must Be Complete* is set. This means that starting at any node, every other node on the playing field must be reachable. No disjoint network components are permitted. Each requirement is individually adjustable, being ignored if set to the default value (−1 or *disabled*). During the simulation, all of the subgoals are continuously tested to detect whether the overarching scenario goal is already met, or, in the worst case, cannot be reached any longer.

In PowerSurge, offering challenges to the user is synonymous with restricting his interaction possibilities, i.e. going from an all-flexible editor towards a concrete real-world problem perspective. Otherwise, the user could easily bypass the designed challenge by deleting lines or nodes, or by adjusting their production or throughput values. Therefore, we implemented a simple access rights management system for the simulated objects. As a result, user access can be
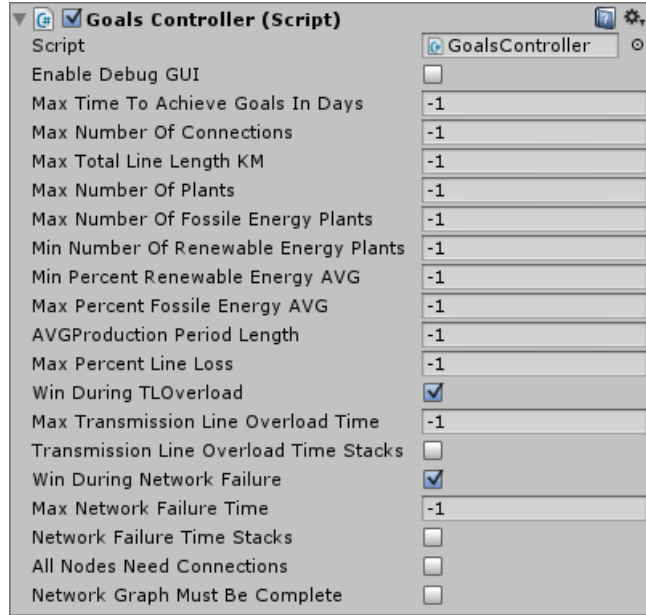
Fig. 7. Scriptable goals for new scenarios (entries with the default parameter −1 are not considered for the evaluation).

individually adjusted for each object when setting up new scenarios. There are three access modes: (1) *Modify and delete* grants the user full rights to modify the object's properties or even remove it from the screen. (2) *Modify only* allows the user to modify all properties but not to remove the object itself. (3) *Sealed* means that the user can only view the placement's current properties to react to its behavior during play.

3.5 Domain Model

Formally, the power transmission network in our simulator is represented by an undirected graph. In this graph, the vertices or nodes are the consumers (e.g. cities), generators (power plants) and distributors (which neither produce nor consume power). All node types share one property, the power production/consumption P_{self}, which is positive/negative if the node generates/consumes power, or zero, if the node solely distributes power. In addition, the constant base consumption of a consumer node is captured by P_{base}. A load pattern defines the node's fluctuating power consumption over the course of a day. A generator node has both a minimum and a maximum power output value P_{min} and P_{max}, respectively. Its output is further modified by a function of time (consider day/night cycle), and a randomized fluctuation with the amplitude of $P_{randAmp}$. The nodes are connected by means of transmission lines. The resulting graph is

irreflexive, i.e. no node can be connected to itself. A transmission line is characterized by the two nodes A and B to which it is connected, its length l, its maximum power load P_{max} and its maximum overload factor $f_{overload}$.

In addition to user-defined or scenario-dependent parameters, the simulation has to solve for specific variables, such as the actual power output of all generators and the power flow on all transmission lines. To derive these values, we set up a system of equality and inequality constraints modelling the behavior of the network. We then minimize an evaluation function within these constraints, using a boundary, linear equality and inequality constraints solver provided by the accessible and established open library ALGLIB [10, 11]. We were able to directly embed it into our development environment, Unity3D[1]. In the following paragraphs we present both the evaluation function and all constraints imposed on its optimization.

As the transmission lines are bidirectional, power on each line can flow in either direction. The power loss due to the line's electrical resistance depends on the amount of power flowing into the line. To properly apply the power conservation to the transmission line, we must therefore know in which direction the power flows. To accommodate this, we split each bidirectional power line into two unidirectional power paths in the context of the optimization. This means that a transmission line whose endpoints we call A and B has four optimization variables, power inflow and outflow for the path from A to B (P_{in}^{AB} and P_{out}^{AB}, respectively) and for the path from B to A (P_{in}^{BA} and P_{out}^{BA}, respectively). The evaluation function h optimized by the solver is composed as follows:

$$h = \sum_{i \in lines} (L(i) + S(i) + O(i)) \tag{1}$$

where for a given transmission line i the power loss $L(i)$, the squared power values $S(i)$ and the overload $O(i)$ are defined as:

$$L(i) = \lambda^{l(i)} * P_{in}^{AB}(i) + \lambda^{l(i)} * P_{in}^{BA}(i) \tag{2}$$

$$S(i) = P_{in}^{AB}(i)^2 + P_{out}^{AB}(i)^2 + P_{in}^{BA}(i)^2 + P_{out}^{AB}(i)^2 \tag{3}$$

$$O(i) = \begin{cases} E(i)^2 & \text{if } E(i) > P_{max}(i) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$E(i) = \max \left(|P_{in}^{AB} + P_{out}^{BA}|, |P_{in}^{BA} + P_{out}^{AB}| \right) \tag{5}$$

where λ is the power retention factor per km of the transmission lines, $l(i)$ is the length of transmission line i, $P_{in}^{AB}(i)$, $P_{out}^{AB}(i)$, $P_{in}^{BA}(i)$ and $P_{out}^{BA}(i)$ are the power inflow and outflow of the two paths of transmission line i as explained above, and $P_{max}(i)$ is the maximum load of transmission line i. Power loss on the transmission lines (2) is the main criterion we want to minimize, so its inclusion is obvious. It should be noted here that the power loss is technically

$$P_{loss} = R \cdot I^2 = R \cdot \left(\frac{P_{flow}}{U} \right)^2 \sim P_{flow}^2$$

---

where R is the line's electrical resistance, U is the line voltage, I is the current and $P_{flow} = U \cdot I$ is the total power flowing on the line. While the evaluation function could handle this quadratic function, the constraints for the optimizer must be linear in all variables. We therefore opted to use the linear relation $P_{loss} = R' \cdot P_{flow}$ (where $R' = \lambda^l$ is the power retention factor which acts as a stand-in for the line resistance) in both the constraints and the evaluation function in order to keep our system consistent. Including the sum of the squared endpoint power values (3) in the evaluation function minimizes the total power flow on the network. On the one hand, this serves to prevent power flow over detours – the more line endpoints need to be traversed by power flowing from generators to consumers, the more it factors into this summand. On the other hand, this prevents power from flowing in both directions at once on any transmission line, which would incur more power loss while reducing the net power transported over the respective line. Lastly, the overload summand (4) ensures that line overloading, while allowed, is discouraged. To achieve this, the overload summand is zero if the line is not overloaded. Once the line load enters the overload interval, this summand contributes the squared effective power flow of the line.

When searching for the equilibrium of a given power transmission network, the evaluation function discussed above is minimized with a set of constraints. This set is composed of two constraints for each node and eight constraints for each transmission line present in the network. A node n of any type must meet the power balance constraint:

$$P_{self}(n) + \sum_{i \in cons(n)} (P_{in}(n,i) + P_{out}(n,i)) = 0 \tag{6}$$

where $cons(n)$ is the set of all transmission lines connected to node n, $P_{in}(n,i) \leq 0$ is the power flowing from n into the line i, and $P_{out}(n,i) \geq 0$ the power flowing from i into n. This equality constraint (6) states that the total amount of power flowing into a node n must equal the total amount of power flowing out of the node. In other words, taking the node's own power generation or consumption P_{self} into account, no power may "magically" appear or disappear on the node. As previously mentioned, a distribution node n_d neither generates nor consumes power, so its local power is constrained to 0:

$$P_{self}(n_d) = 0 \tag{7}$$

The local power of a consumer node n_c must exactly match the node's current consumption:

$$P_{self}(n_c) = P_{consume}(n_c, frac(t)) \leq 0 \tag{8}$$

where the power consumption $P_{consume}(n_c, t)$ at simulation time t (in days) is determined by the load pattern of the consumer node over the course of a day. As the simulation time t is given in days, we extract its fractional part $frac(t)$ as the time of day. On a generator node n_g, the local power is constrained by the power output bounds:

$$0 \leq P_{min}(n_g) \leq P_{self}(n_g) \leq P_{max}(n_g, t) \tag{9}$$

where the simulation time dependent maximum power output $P_{max}(n_g, t)$ is determined as:

$$P_{max}(n_g, t) = P_{pattern}(n_g, frac(t)) + uniform(-P_{randAmp}(n_g), P_{randAmp}(n_g))$$

The constant parameter $P_{min}(n_g)$ is the plant's minimum power output boundary. The plant's maximum power output is determined by a pattern $P_{pattern}(n_g, frac(t))$ analogously to the load pattern of consumer nodes. For renewable energy generation like wind and solar power, which are heavily subjected to natural fluctuations, an additional summand samples a uniformly random value within the given randomization amplitude $P_{randAmp}$. This fluctuation can be reduced or completely disabled for other plants, e.g. nuclear power plants, by setting $P_{randAmp} = 0$. For each transmission line i, these eight constraints apply:

$$\lambda^{l(i)} \cdot P_{in}^{AB}(i) + P_{out}^{AB}(i) = 0 \qquad \lambda^{l(i)} \cdot P_{in}^{BA}(i) + P_{out}^{BA}(i) = 0 \qquad (10)$$

$$P_{in}^{AB}(i) \leq 0 \qquad P_{in}^{BA}(i) \leq 0 \qquad P_{out}^{AB}(i) \geq 0 \qquad P_{out}^{BA}(i) \geq 0 \qquad (11)$$

$$|P_{in}^{AB}(i)| \leq f_{overload}(i) \cdot P_{max}(i) \qquad |P_{in}^{BA}(i)| \leq f_{overload}(i) \cdot P_{max}(i) \qquad (12)$$

The equality constraints (10) represent the power balance on the transmission line i. The amount of inflowing power $P_{in}^{AB}(i)$ or $P_{in}^{BA}(i)$, scaled by the line's power retention factor $\lambda^{l(i)}$, must equal the amount of outflowing power $P_{out}^{AB}(i)$ or $P_{out}^{BA}(i)$, respectively. The boundary constraints (11) state that the variables for power flowing into the line, $P_{in}^{AB}(i)$ and $P_{in}^{BA}(i)$, must have a negative sign, while those for power flowing out of the transmission line, $P_{out}^{AB}(i)$ and $P_{out}^{BA}(i)$, must have a positive one. Notice that it suffices to include only one of the inequality pairs (11) in the actual optimization. Together with the power conservation constraint (10), either of these boundary pairs implies the other one. Finally, the inequality constraints (12) ensures that power flow on no transmission line exceeds that line's effective maximum load, which is the product of the line's regular maximum load $P_{max}(i)$ and its maximum overload factor $f_{overload}(i)$.

4 Results & Future Work

We designed a system that allows the users to interactively learn about and explore the complexities inherent in power transmission systems. Several basic tutorials cover the basic relationships of producer and consumer nodes, introduce the intricacies of patching overloaded networks, and to minimize the utilization of resources such as the overall length of the power lines.

In a competition on interactive simulations, we presented PowerSurge to about 30 people, most of them students. They voted PowerSurge to be the best out of seven projects, including interactive simulations in domains as far apart as biology and traffic systems. Criteria in the competition comprised the complexity of the scientific model, the usability, and the visual appeal.

At this point, important aspects such as local energy storage systems are still missing, however, an understanding of the complex interplay between producers

and consumers can already be gained. Apart from improving the functionalities of the simulator, we deem the following aspects as especially beneficial extensions: A highscore system could lead to a better grasp on the user performance and build social ties between the users, which is an important factor of motivation. Along these lines, multi-player modes could promote the collaborative (re-)design of power grids at moderate scales or increase the fun through one-on-one or team competitions. An extension of the time system could allow the user to go back in time and review the changes that were made to the system. Also it should be possible to create different branches of the timeline to tackle some of the problems with different ideas. New scenarios should allow for the definition of more constraints, for example the number of each type of node that can be planted. In addition to numerous other user-centered and technical improvements, we are planning to test our software as part of an educational curriculum.

References

1. Steghöfer, J.P., Anders, G., Siefert, F., Reif, W.: A system of systems approach to the evolutionary transformation of power management systems. In: GI-Jahrestagung. (2013) 1500–1515
2. Herrmann, U., Kelly, B., Price, H.: Two-tank molten salt storage for parabolic trough solar power plants. Energy **29**(5) (2004) 883–893
3. Paatero, J.V., Lund, P.D.: Effect of energy storage on variations in wind power. Wind Energy **8**(4) (2005) 421–441
4. Masters, G.M.: Renewable and efficient electric power systems. John Wiley & Sons (2013)
5. Colella, V.S., Klopfer, E., Resnick, M.: Adventures in Modeling: Exploring Complex, Dynamic Systems with StarLogo. Teachers College Press, Columbia University, New York (2001)
6. PowerWorld Corporation: PowerWorld Simulator. `http://www.powerworld.com/products/simulator/overview` (October 2014)
7. Dugan, R.C.: Reference guide: The open distribution system simulator (opendss). Electric Power Research Institute, Inc (2012)
8. Chassin, D., Schneider, K., Gerkensmeyer, C.: GridLAB-D: An open-source power systems modeling and simulation environment. In: Transmission and Distribution Conference and Exposition, 2008. T&D. IEEE/PES, IEEE (2008) 1–5 `http://www.gridlabd.org/`.
9. Deterding, S., Sicart, M., Nacke, L., O'Hara, K., Dixon, D.: Gamification. using game-design elements in non-gaming contexts. In: CHI'11 Extended Abstracts on Human Factors in Computing Systems, ACM (2011) 2425–2428
10. Shearer, J., Wolfe, M.: Alglib, a simple symbol-manipulation package. Communications of the ACM **28**(8) (1985) 820–825
11. ALGLIB Project: ALGLIB: a cross-platform open source numerical analysis and data processing library. `http://www.alglib.net/` (January 2014)