Adjustment of a simulator of a complex dynamic system with emphasis on the reduction of computational resources

Mariano Trigila[1], Adriana Gaudiani [2] and Emilio Luque[3]

[1] *Facultad de Ingeniería y Ciencias Agrarias, Pontificia Universidad Católica Argentina, Ciudad Autónoma de Buenos Aires, Argentina*

[2] *Instituto de Ciencias, Universidad Nacional de General Sarmiento, Buenos Aires, Argentina*

[3] *Depto. de Arquitectura de Computadores y Sistemas Operativos, Universidad Autónoma de Barcelona, 08193 Bellaterra (Barcelona) España*

Abstract

Scientists and engineers continuously build models to interpret axiomatic theories or explain the reality of the universe of interest to reduce the gap between formal theory and observation in practice. We focus our work on dealing with the uncertainty of the input data of the model to improve the quality of the simulation. To perform this type of process large volumes of data and a lot of computer processing must be handled. This article proposes a methodology for adjusting a simulator of a complex dynamic system that models the wave translation along rivers channels, with emphasis on the reduction of computation resources. We propose a simulator calibration by using a methodology based on successive adjustment steps of the model. We based our process in a parametric simulation. The input scenarios used to run the simulator at every step were obtained in an agile way, achieving a model improvement up to 50% in the reduction of the simulated data error. These results encouraged us to extend the adjustment process over a larger domain region.

1. Introduction

Scientists and engineers make use of computer simulations, as an established tool in many branches of science, to study the behavior of the system modeled. They study the model in function of the different responses of the model when different scenarios are used to run the simulation. As a preliminary act, the simulator will need an adjustment process in which the best set of input values to the simulator is sought which provides the smallest difference between the output data and the reference data set [7]. Adjustment processes are usually computationally expensive since they require running the simulator with each of the possible combinations of inputs in search of the best output. In other words, When the search space of an optimal set of parameters is very large then the computational cost of the optimal search process is very expensive. This paper presents a methodology that proposes to lower computational cost.

The proposed approach exploits a local behavior of the system: The values of the critical parameters, selected for the calibration process, differ very little in sites located spatially close to each other over the domain of the system. This assumption allows reducing the search space of the input parameter to the simulator that minimizes the error between the simulated and the real data. This artifice allows in a direct way to reduce the computational cost of the search process. Therefore, the parameters values to be optimized for each section on the river domain, are calibrated taking advantage of the optimal values which were calculated for the previous section located at an adjacent place.

Using our methodology, we could find input scenarios to run the model, which provided a substantial improvement in the quality of the prediction in relation to the results obtained when the simulation is launched with the initial scenario (currently used for simulation and forecasting). The best results obtained provided a gain of up to 50%. To determine this value, we detected one input parameters set used to launch a simulation which is the one that best fits for a predetermined sampling site located on the riverbed. We calculated an index to quantify the difference between the simulated and the observed series of data. The search process ends when it finds the best parameters set, by which we mean the scenario that gets the lowest index. Therefore, the scenario obtained is the best simulation scenario in a reduced search space. We take advantage of the research and the results of previous works [2, 3].

2. The simulator and the simulation domain

The simulator implements a one-dimensional hydrodynamic model of the Paraná River for hydrological forecast [4, 5]. This computer model calculates the translation of the waves through a channel calculated by the Saint Venant equations. It was developed in the Laboratory of computational hydraulics of the National Institute of water (INA).

The hydrodynamic model simulates a physical system whose domain is set by its parameters values. In summary, the simulator could be described as an "input - process - output" system [6], where: The input is a complete simulation scenario, including the set of parameters and the input data needed by the model to simulate the behavior of the river, which is the simulated physical system. The process is defined by the algorithms of the computational model which relate the system variables and its evolution. This model is based on numerical methods that solve equations in partial derivatives. The simulator output is the set of simulated data returned by the model, when it was executed with a given input scenario. Some of the parameters and inputs that define the system and the simulation domain are the following: Input - height of levees, channel Manning, plain Manning, flow, border and start conditions, among others. The input parameters and the input variables required for system initialization are stored in text files. Process - Arithmetic calculations, algorithms, procedures and functions to resolve the river wave's displacement, are implemented in a Fortran program. Output - the river height, flow, among others, calculated at the monitoring stations. The output data are stored in text files.

3. Domain modeling features

Simulator represents a hydrodynamic model consisting of two sections or filaments. Each filament represents the path of a river. See data in Table 1 and a graphical representation in Fig. 1. To simulate the transport of water in a filament channel, its route is subdivided into sections. Each section (Sc) that divides the domain results from a discretization process and represents a specific position over river path, as it is shown in Fig. 1.

Table 1 Calculation Network: River model [3].

#	Path	*Long. (km)*	*Sections*
5	Paraná	1083	76
	Paraguay	376	77

The simulator requires setting a set of input parameters values at every subsection in each section. Each set determines a simulation scenario. At the same time, each section has a subdivision called subsection (Su). Each subsection is a cross section to the channel, and it describes the geometry of the river in a section. The set of input parameters of a section is composed of the subsections parameters that it has. For every subsection in each section a set of parameters is specified, of which we consider for this work the roughness coefficient of Manning (m), which varies according to the resistance offered by the main channel and the floodplain, being necessary to distinguish them at a value of Manning of plain (mp), and Manning of channel (mc).

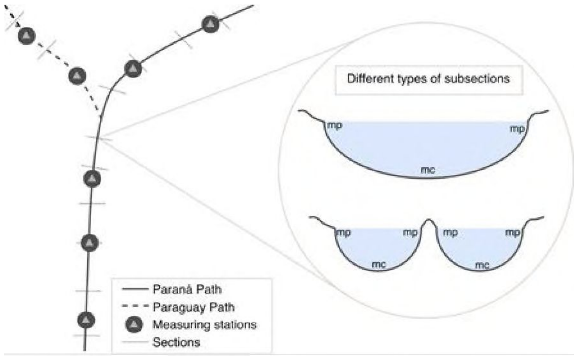

Fig. 1 Discretization of the simulated domain river system.

Depending on the channel geometry in each section, a greater or lesser amount is needed to mp and mc. The different sections can be seen in Fig. 1.

3.1. Observed data measured at monitoring stations

A monitoring or measuring station ($\boldsymbol{St}$) is the "physical and real" place where the river heights are surveyed and recorded. A measuring station is in a city on the banks of the river channel. The data collected and recorded from the height of the river are known as observed data ($\boldsymbol{OD}$) and are measured daily. The period from 1994 to 2011 is available for all monitoring stations and these data were used to implement the experiences carried out in this work.

3.2. Observed data vs. Simulated data

At the beginning of the problem analysis, we concentrate on finding the difference, or simulation error, between the observed data series and the simulated data series. To show these differences we used data visualization techniques, among which is the "Stream Graph" Fig. 2 [9].

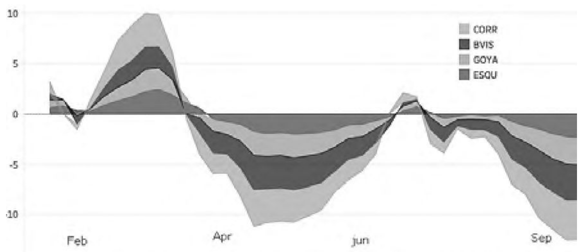

Fig. 2 Difference Observed - Simulated 1997, four stations.

With it we were able to observe the pronounced differences over time, comparing four stations. We could also observe the relationship of similarity between series of stations. The graphed stations are spatially contiguous to one another.

4. Methodology

We propose a calibration process of successive tuning steps to obtain an adjusted input parameters values from a preselected set of successive sections. The process varies the input parameters values in a preset way as we explained in section 4.2. An entire simulation scenario must be used to feed the simulator for each of the possible combinations of parameters values. Each combination determines a simulation scenario and we detail the input scenario structure later. The quality of the simulated data (SD) is measured through calculating a divergence index (DI), as we explain in section 4.4. We propose a search methodology for finding the best set of parameters, to optimize the simulation for a reduced search space ω such that ω ⊂ Ω, therefore, minimizing the use of computing resources to achieve the objective, min (DI); where Ω is the whole search space with all possible combinations of the selected adjustment parameters and ω is the resulting reduced space [8]. We show in Fig. 3 the implemented process to search the adjusted parameters set $\hat{X}$, for the station k, which determines the best simulation scenario $\widehat{S_k}$. We start the method by choosing a monitoring station St_k located in an arbitrary place k on the riverbed and selecting three contiguous sections, which are adjacent to that station. After obtaining the best scenario for a station in k, the tuning method is successively extended to its adjacent stations in k+1, repeating the search and successive adjustment process for the n stations, as we show in Fig. 4.

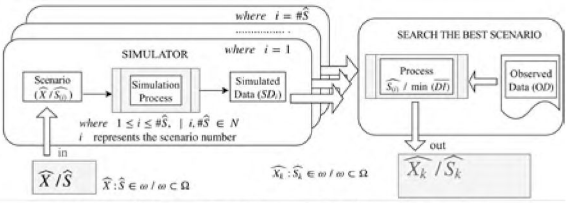

Fig. 3 Search process of the best scenario

4.1. Selecting stations and sections

We chose the first monitoring station St_k which will be the first station located upstream on the river. This is convenient that the chosen place has a simple geometry and that there are measured observed data. The following station to be adjusted, and the next ones, will be chosen by the adjacency to a previously adjusted station. Once selected the first station St_k was selected, we choose a group of sections (Sc_i) located in an adjacent way to the first one station St_k, chosen in the previous step. Three sections surrounding the station k were selected, to carry on for the experiences. They are, the m section, that matches the location of St_k, a second section Sc_{m+1} located adjacent and upstream to St_k and a third section Sc_{m+2} located adjacent and downstream to the section St_k. For this work, we selected simple geometry sections, with three or five subsections, Su.

4.2. Structure of the input scenario

The friction parameter, Manning coefficient, is set for each subsection (Su_m), as we explain in next section. In this case, three subsections determine the section Su_m chosen. Therefore, a section is defined by an j-tuple of Su_m. For this case, the subsections that describe the section Sc_m is defined by 3-tuple:

$$Sc_m = (Su_{J(1)}, Su_{J(2)}, Su_{J(3)}) \qquad \text{(Eq. 1)}$$

where each $Su_{J(x)}$ is defined by a Manning coefficient. $Su_{J(1)}$ and $Su_{J(3)}$ are defined by the same Manning of plain mp_m, and $Su_{J(2)}$ by a Manning of channel mc_m. Thus, Sc_m can be represented by 3-tuple based on Manning coefficients.

$$Sc_m = (mp_m, mc_m, mp_m) \qquad \text{(Eq. 2)}$$

We remark that equation (Eq. 2) has two independent variables, mp_m and mc_m. For k station, three contiguous and adjacent sections were chosen, as we explained previously. The scenario $\widehat{S_k}$ for station k will be defined by:

$$\widehat{S_k} = \begin{bmatrix} Sc_m \\ Sc_{m+1} \\ Sc_{m+2} \end{bmatrix} = \begin{bmatrix} mp_m & mc_m & mp_m \\ mp_{m+1} & mc_{m+1} & mp_{m+1} \\ mp_{m+2} & mc_{m+2} & mp_{m+2} \end{bmatrix} \text{(Eq. 3)}$$

Being a physical system, and because the sections are close together, it is assumed that the three sections have the same values of mp y mc for St_k. Summarizing, equation (Eq. 3) results in:

$$\widehat{S_k} = \begin{bmatrix} Sc_m \\ Sc_{m+1} \\ Sc_{m+2} \end{bmatrix} = \begin{bmatrix} mp_k & mc_k & mp_k \\ mp_k & mc_k & mp_k \\ mp_k & mc_k & mp_k \end{bmatrix} \text{(Eq. 5)}$$

We remark in equation (Eq. 4) that, mp_k and mc_k are independent variables. Therefore, the input scenario used to start the tuning process $\hat{X}$ is determined by the scenarios $\widehat{S_k}$ corresponding to the sections Sc_m, and for the intermediate scenarios $\widehat{S_k}^+$ corresponding to the intermediate sections $Sc_m{}^+$ located between the stations k y $k+1$. Equation (Eq. 5) represents $\hat{X}$ structure for n stations:

$$\widehat{X} = \{\widehat{S_k}, \widehat{S_k}^+, \widehat{S_{k+1}}, \widehat{S_{k+1}}^+, \dots, \widehat{S_n}\},$$
$$with\ k = 1 \text{ (Eq. 6)}$$

4.3. Manning variation range.

We had to set the values of the variation range of plain and channel Manning coefficient, imp and imc, and their corresponding increment value, smp and smc. Both determine the discretization process when determining the parameters values.

$$imp = [mp_{min}, mp_{max}] = [0.1,\ 0.71];$$
$$smp = 0.01 \text{ (Eq. 7)}$$
$$imc = [mc_{min}, mc_{max}] = [0.017, 0.078];$$
$$smc = 0.001 \text{ (Eq. 8)}$$
$$\#\hat{S} = 61 \mid \frac{mp_{max} - mp_{min}}{smp} = \#\widehat{S} \wedge \frac{mc_{max} - mc_{min}}{smc} = \#\widehat{S} \text{ (Eq. 9)}$$

The value 61, for the number of scenarios ($\#\widehat{S}$), was obtained empirically after making previous experiences and finding a minimum value of scenarios which allow us to get improved output values when running the simulation. Of course, we can increment $\#\widehat{S}$ if more precision is required but this requirement will result in the need for many more computational resources. Equation (Eq. 9) determines the values of each scenario $\widehat{S_{k(\iota)}}$ depending on the selected step:

$$\widehat{S_{k(\iota)}} = \begin{bmatrix} mp_i & mc_i & mp_i \\ mp_i & mc_i & mp_i \\ mp_i & mc_i & mp_i \end{bmatrix} =$$
$$\begin{bmatrix} (smp \cdot i) + mp_{ini} & (smc \cdot i) + mc_{ini} & (smp \cdot i) + mp_{ini} \\ (smp \cdot i) + mp_{ini} & (smc \cdot i) + mc_{ini} & (smp \cdot i) + mp_{ini} \\ (smp \cdot i) + mp_{ini} & (smc \cdot i) + mc_{ini} & (smp \cdot i) + mp_{ini} \end{bmatrix}$$
(Eq. 10)

Where i is the number of scenario and the range $[i, \#\widehat{S}] \subset \mathbb{N}, where\ 1 \leq i \leq \#\widehat{S}$. mp_{ini} and mc_{ini} are the initial values used to start the search process and running the simulator for each scenario to find the best one, as we describe in next section

4.4. Search of the best scenario

When the simulator is fed with each of the possible experimentation scenarios, the output produces a numerical series of simulated data (SD), which are used to generate hydrographs of the riverbed heights. We select those series corresponding to the chosen station to implement our fitness functions by comparing the SD series with the OD series. A divergence index DI is determined, and is implemented using the root mean square error estimator (RMSE):

$$\boldsymbol{DI_k^y} = RSME_k^y = \sqrt[2]{\frac{\sum_{i=1}^{i=N} (\boldsymbol{H_k^{OD,y}} - \boldsymbol{H_k^{SD,y}})_i^2}{\mathbf{N}}} \text{ (Eq. 11)}$$

The index DI_k^y is calculated based on the RMSE error of the series of river heights simulated $H_k^{SD,y}$ with respect to of the series of river heights observed $H_k^{OD,y}$, for a station k, and for a year y, which is the simulation time, and the number of stations, N. Every time a simulation ends, we evaluate $DI_{k\ i}^y$ for each scenario, which are indicated by the i sub index. The best fit scenario for the k station is denominated $\widehat{S_{k(\iota)}}$ which generates a set of output H_k^{SD} such that DI_y^k is the minimum ($\min(DI_k^y)$) of all the simulations. So, $\widehat{S_k}$ is denoted by the "best fit" scenario for station k where $\propto$ represents the sub index that best fits.

4.5. Successive tuning process

After obtaining the scenario of best fit for a station, the adjustment can be extended to a new station $k + 1$ taking advantage of the locality simulation behavior and the parameters set values of a previously adjusted station k, which is the neighboring station to the $k + 1$ station. For it, the scenario $\widehat{S_k}^+$ is initialized with the values of the best fit scenario $\widehat{S_k}$. This is so because by locality behavior, those sections that are close one to another have similar adjustment scenarios or at least differ very little abrupt jumps (or changes) in parameter values in distinct positions of the selected stretch of river. Therefore, we are already able to run the simulator and look for the best scenario $\widehat{S_{k+1(\propto)}}$ for the station $k + 1$. We can see details in Fig. 4.

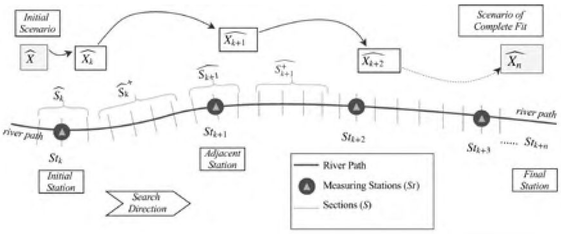

Fig. 4 Methodological Successive tuning process.

In the successive input scenarios, we leave fixed the adjusted parameters values found in the previous calibrations, and thus the previous adjustment scenarios of each section are used to find de actual adjusted parameters values, for k station the new k parameter vector is:

$$\widehat{X_k} = \left\{\widehat{S_k}, \widehat{S_k}^+, \widehat{S_{k+1}}, \widehat{S_{k+1}}^+, \dots, \widehat{S_n}\right\}$$

The $k+1$ input scenario, or k+1 parameters vector, is:

$$\widehat{X_{k+1}} = \left\{\widehat{S_k}, \widehat{S_k}^+, \widehat{S_{k+1}}, \widehat{S_{k+1}}^+, \dots, \widehat{S_n}\right\},$$
$$where\ \widehat{S_k}^+ = \widehat{S_k}$$

The $k+2$ input scenario, or k+2 parameters vector is:

$$\widehat{X_{k+2}} = \left\{\widehat{S_k}, \widehat{S_k}^+, \widehat{S_{k+1}}, \widehat{S_{k+1}}^+, \dots, \widehat{S_n}\right\},$$
$$where\ \widehat{S_k}^+ = \widehat{S_k}, \quad \widehat{S_{k+1}}^+ = \widehat{S_{k+1}}$$

For n input scenario to the Simulator (scenario that adjusts the entire domain):

$$\widehat{X_n} = \left\{\widehat{S_k}, \widehat{S_k}^+, \widehat{S_{k+1}}, \widehat{S_{k+1}}^+, \dots, \widehat{S_n}\right\},$$
$$where\ \widehat{S_k}^+ = \widehat{S_k}, \dots, \widehat{S_{n-1}}^+ = \widehat{S_{n-1}} \quad \text{(Eq. 12)}$$

4.6. Experimental Results

After making the experiences, feeding the simulator with the proposed scenarios, and analyzing and comparing the series of outputs delivered by the simulator against the observed series, positive results were obtained in terms of meeting scenarios of better performance than the initial proposed by experts in the domain of the problem.
In search of the best scenario performed on the k station "Esquina" (ESQU), we found scenarios that improved the output of the simulator up to 57% in relation to the initial scenario proposed by the experts in the domain of the problem, determined by ratio of $DI_k^y(Fit)$ to $DI_k^y(Initial)$. We show in Table 2 the synthesis process with the top three scenarios found for processed k station. As also, it shows the second station $k+1$ adjusted. The best scenario is searched at "La Paz" station (LAPA) which is adjacent to ESQU station. As it can be observed in Table 2, a synthesis process with the two best scenarios was found for $k+1$ station.

Table 2 Fit made in k station and k+1 station, several years.

S_i	*Year*	*Station*	*Station ID*	*Improvement* $DI_k^y(F)/DI_k^y(I)$
46	*2008*	*k*	*ESQU*	*57 %*
54	1999	*k*	ESQU	39 %
38	2002	*k*	ESQU	22 %
38	1999	*k+1*	LAPA	45 %
30	2008	*k+1*	LAPA	24 %

Fig. 5 and Fig. 6 show a comparative graph with the observed data series (real measured values), the initial simulated data series (original series loaded in the simulator) and the series of simulated data adjusted for the best fit scenarios in each (k *and* $k+1$) station. We can see that our method achieves the best results since month 4 to 12, when the simulation errors decrease.

The key to the method for the reduction of computational resources lies in:

1. To assign to $\widehat{S_k}^+$ the same value as $\widehat{S_k}$ based on the local behavior of the system.
2. To run a parametric simulation for every parameter value combination in the reduced Search space reduction $\omega \subset \Omega$.

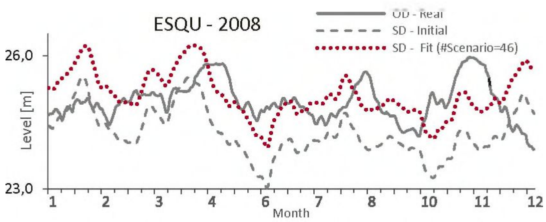

Fig. 5 Comparative OD, SD, Fit (ESQU)

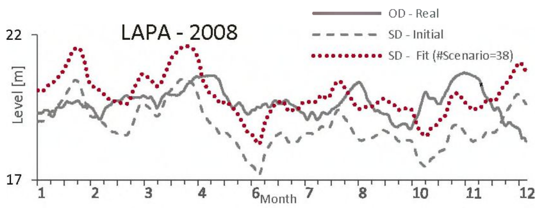

Fig. 6 Comparative OD, SD, Fit (LAPA)

5. Results of experience

In search of the best fit, in one of the stations "Esquina" (ESQU), we found a scenario (*SD – Fit #Scenario=46*) that improves the results up 57% in relation to the initial scenario used by the of INA experts, (*SD - Initial*), as you can see in Fig. 3, where it is also related to OD (*OD - Real*).

The gain value (quality) is obtained by dividing the $DI_{inicial}$ respect of DI_{Fit}.
We used the same methodology to tune forecasting for LAPA, which is a contiguous station to ESQU, getting improvements, as you can see in Table 2. Other stations that were adjusted (and not found in table 2) showed substantial improvements in error reduction.
These promising results indicate the great potential of our successive tuning steps method and encourage us to continue our research in this direction.

6. Conclusions

The main result of this work was to reduce the simulation error of the computational model using the local properties of the river channel, in order to reduce the search space of its optimal set of parameters. The method provides promising results by finding higher quality scenarios with improvements close to the 50 %. The method is simple and manages to reduce computational resources by lowering the volume of data processed in each stage by the following fundamental reasons:

1) Each successive adjustment step results in adjusted sections. These adjusted sections will be useful for the configuration of the previous sections to a station to be adjusted (contiguous), in a new adjustment step.
2) We assume that the value of the adjustment scenario of intermediate section ($\widehat{S_k}^{+}$) is equal to the value of the previous immediate contiguous adjustment, by the principle of location of the system.
3) Significate reduction of search space, used to find the adjusted set of input parameters to the simulator.

We observe that the search process of the best scenario, Fig. 3, is about an embarrassingly parallel problem. Consequently, we are currently working on the implementation of the adjustment method in successive steps on HPC cloud computing platform [10, 11].

7. Acknowledgments

The MICINN/MINECO Spain under contracts TIN2014-53172-P and TIN2017-84875-P has supported this research. We are very grateful for the data provided by INA and we appreciate the guidance received from researchers at INA Hydraulic Laboratory.

8. References

[1] E. Bladé, M. Gómez-Valentín, J. Dolz, J. L. Aragón-Hernández, G. Corestein, y M. Sánchez-Juny, "Integration of 1D and 2D finite volume schemes for computations of water flow in natural channels", Advances in Water Resources, vol. 42, pp. 17–29, 2012.

[2] E. Cabrera, E. Luque, M. Taboada, F. Epelde, y M. L. Iglesias, "Optimization of emergency departments by agent-based modeling and simulation", in Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on, 2012, pp. 423–430.

[3] A. Gaudiani, E. Luque, P. García, M. Re, M. Naiouf, y A. De Giusti, "How a Computational Method Can Help to Improve the Quality of River Flood Prediction by Simulation", in Advances and New Trends in Environmental and Energy Informatics, Springer, 2016, pp. 337–351.

[4] A. N. Menéndez, "Three decades of development and application of numerical simulation tools at INA Hydraulics Lab", en First South-American Congress on Computational Mechanics, Santa Fe-Paraná, Argentina, 2002.

[5] T. Krauße and J. Cullmann, "Identification of hydrological model parameters for flood forecasting using data depth measures", Hydrology & Earth System Sciences Discussions, vol. 8, n.o 2, 2011.

[6] D. P. Solomatine and A. Ostfeld, "Data-driven modelling: some past experiences and new approaches," Journal of hydroinformatics, vol. 10, no. 1, pp. 3–22, 2008.

[7] R. G. Sargent, "Verification and validation of simulation models," in Simulation Conference, 2007 Winter, 2007, pp. 124–137.

[8] W. Long-Fei and S. H. I. Le-Yuan, "Simulation optimization: a review on theory and applications," Acta Automatica Sinica, vol. 39, no. 11, pp. 1957–1968, 2013.

[9] J. Heer, M. Bostock, and V. Ogievetsky, "A tour through the visualization zoo," Queue, vol. 8, no. 5, p. 20, 2010.

[10] D. Sikeridis, I. Papapanagiotou, B. P. Rimal, and M. Devetsikiotis, "A Comparative Taxonomy and Survey of Public Cloud Infrastructure Vendors," arXiv preprint arXiv:1710.01476, 2017.

[11] G. Fox, J. Qiu, S. Jha, S. Ekanayake, and S. Kamburugamuve, "Big data, simulations and hpc convergence," in Big Data Benchmarking, Springer, 2015, pp. 3–17.