2015-08-01

A Method for Using Pre-Computed Scenarios of Physically-Based Spatially-Distributed Hydrologic Models in Flood Forecasting Systems

Herman Guillermo Dolder
Brigham Young University - Provo

A Method for Using Pre-Computed Scenarios of Physically-Based Spatially-Distributed Hydrologic Models in Flood Forecasting Systems

Herman Guillermo Dolder

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Norman L. Jones, Chair
E. James Nelson
Daniel P. Ames
Rollin H. Hotchkiss
Gustavious P. Williams

Department of Civil and Environmental Engineering

Brigham Young University

August 2015

ABSTRACT

A Method for Using Pre-Computed Scenarios of Physically-Based Spatially-Distributed Hydrologic Models in Flood Forecasting Systems

Herman Guillermo Dolder
Department of Civil and Environmental Engineering, BYU
Doctor of Philosophy

Every year floods are responsible of a significant number of human losses, many of which could be avoided with a broader implementation of flood forecasting systems. Nevertheless, there are still some technological and economic limitations that impede the creation of these systems in many parts of the world. At the core of many flood forecasting systems is a hydrologic model that transforms the weather forecast into a flow forecast. Using real-time modeling for potential floods poses a series of problems: if the model is complex, the computational power required can be significant, and consequently expensive, and if the model is simple enough to run on regular computers in the time allotted, it is likely that the results will not be accurate enough to be useful. I propose the development of a standardized method for using pre-computed scenarios as an alternative to real-time flood modeling. I explain how pre-computing has been used on other realms in the past, and how it is beginning to be implemented in different branches of hydrology, the prediction coastal flooding due to storms or tsunamis being one of the most developed. My research has focused on answering the questions that arise during the design stage of a flood forecasting system not only for rain or snow driven floods, but also by anthropogenic-produced floods. I analyze the number of parameters and their granularity to be used to create the scenarios, the accuracy of the results, different strategies to implement the systems, etc. Finally, I present some test-cases of the application of the method, and assess their results.

Keywords: pre-computing, scenarios, flood forecasting systems, hydrologic models

ACKNOWLEDGEMENTS

CI-WATER is a cooperative research grant funded by the National Science Foundation involving researchers from two states and four universities: Brigham Young University, the University of Utah, Utah State University, and the University of Wyoming. CI-WATER is being used to establish a robust and distributed cyberinfrastructure (CI) consisting of data services, visualization tools, and a comprehensive education and outreach program that will support this integration and improve the manner in which computer models are used to support long-term planning and water resource management in the Intermountain West.

This material is based upon work supported by the National Science Foundation under Grant No. 1135483.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Since 1900 rainfall- and snowmelt-driven floods were responsible of 4.5 million deaths worldwide, which accounted for 21% of the total fatalities produced by all the natural disasters. Modernization and technologic developments have had a great impact on reducing the risks and damages related to flooding. Nevertheless, in the last 10 years 1,671 flood events were registered, producing a death toll of 63,026 people, and injuring 57,467 more. Most of these casualties occurred in Asia (71.5%), followed by the Americas (15.5%), and Africa (11.3%). Much effort has been spent in prevention, both by construction of civil works, such as dams, channels, and levees, and by developing ever-more-accurate weather forecasts and hydrologic models.

Regardless of the advances achieved, crisis managers around the world still face the problem of assessing and addressing potential flood situations on a regular basis. The first problem they face is having access to accurate weather forecasts. The second problem is correctly interpreting those forecasts. The third problem is having a reliable hydrologic model at their disposal. The fourth problem is having the knowledge to adjust the hydrologic model to the current conditions prevailing at the watershed, and then using it to transform the weather forecast into a flood forecast. The fifth problem is having the computational resources to run the model and obtain the results in the short time span that potential flooding scenarios concede.

But even having solved all those problems, deciding what actions to implement is still a difficult job. Issuing warnings for events that later do not occur not only produces monetary losses, but increases the “boy-who-cried-wolf” effect, in which the public’s loss of confidence in the warnings increases their level of exposure to future risks. At the same time, not issuing warnings for events that later do occur can cost human lives. To make things even more difficult, each minute of delay in decision making could result in terrible consequences.

1.1 Types of Flood Forecasting Systems

For these reasons, the implementation of flood forecasting systems is crucial. In these systems, weather forecasts are fed into hydrologic models that produce hydrographs and flood maps, and trigger automatic or human-assisted warnings and preventive actions. There are different approaches to creating these systems, and one way to categorize them is on how they generate their real-time flood forecasts (Henonin, Russo, Mark, & Gourbesville, 2013):

I. Systems based only on rainfall information and empirical scenarios

II. Systems based on rainfall information and pre-simulated scenarios

III. Systems based on real-time data assimilation

IV. Systems with active feedback to the drainage system operation

Type I systems rely on experts’ knowledge to transform the weather forecast to a probable flood scenario. Provided a weather forecast, an expert decides between several possible scenarios, which could be based on historical records or empirical knowledge. Therefore, these kinds of systems are not “technologically intensive”, but given that their accuracy depends principally on a human factor they are less reliable and definitely not portable to different regions.

Type II systems have many of the advantages of the Type I systems, without their limitations. In these systems the “human expert” is replaced by a set of pre-computed scenarios, usually based on scientifically sound hydrologic models. One example of this type of systems is the ESPADA system (Raymond, Peyron, & Martin, 2006), developed in France, which is based on 44 pre-computed scenarios. A scenario consists of a combination of values for a pre-defined set of parameters in the model (Figure 1).

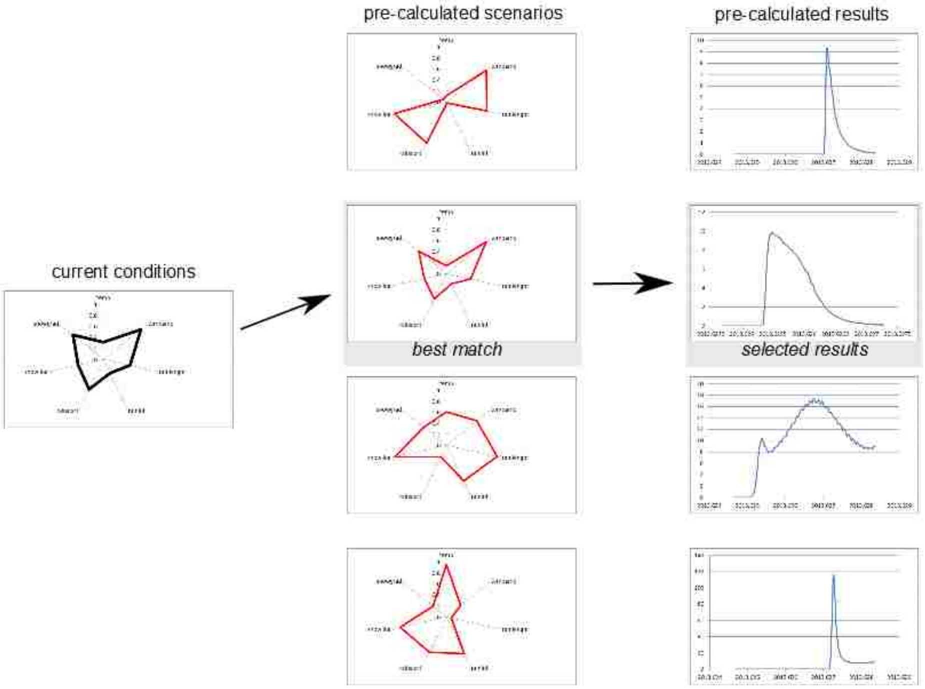


Figure 1. Diagram Showing the Type II Systems Method

Since these scenarios are computed days, weeks, months, or even years in advance of the possible flood, they don’t require intensive computational power, and can take advantage of distributed and cloud computing. In distributed computing (Peleg, 2000), several model runs can be performed in parallel on different computers, which could be physically present, belonging for instance to a local network (Figure 2).

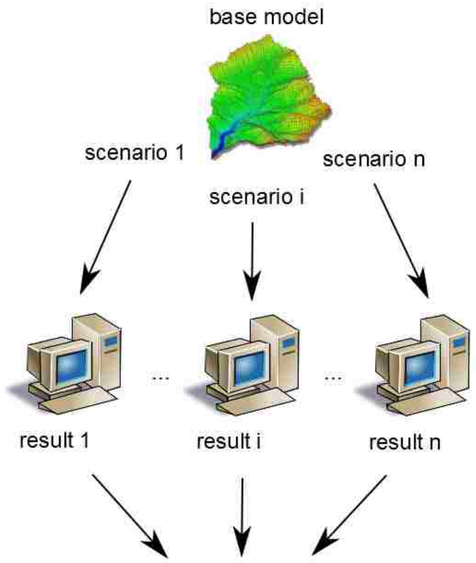


Figure 2. Model Runs Using Distributed Computing

In cloud computing (Armbrust et al., 2010), the model runs are also performed on several computers, but these are located “somewhere” on the Internet (i.e., "the Cloud") (Figure 3). This approach has the advantage of requiring the user to pay only for the computation time required to run the model, which makes this a good alternative for systems strongly constrained by economic limitations and where local computational resources are insufficient.

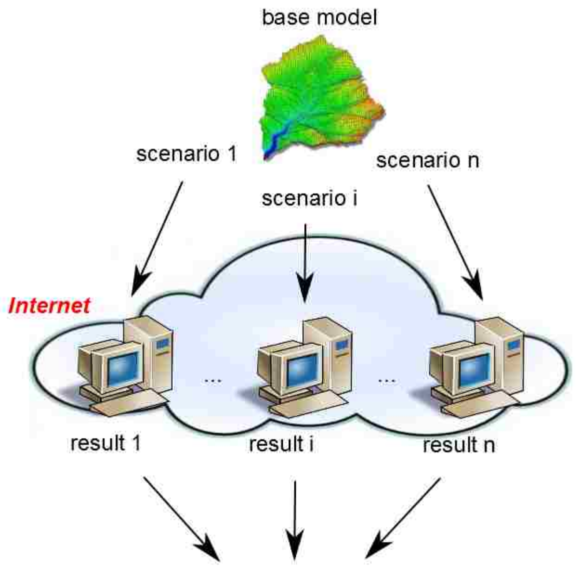


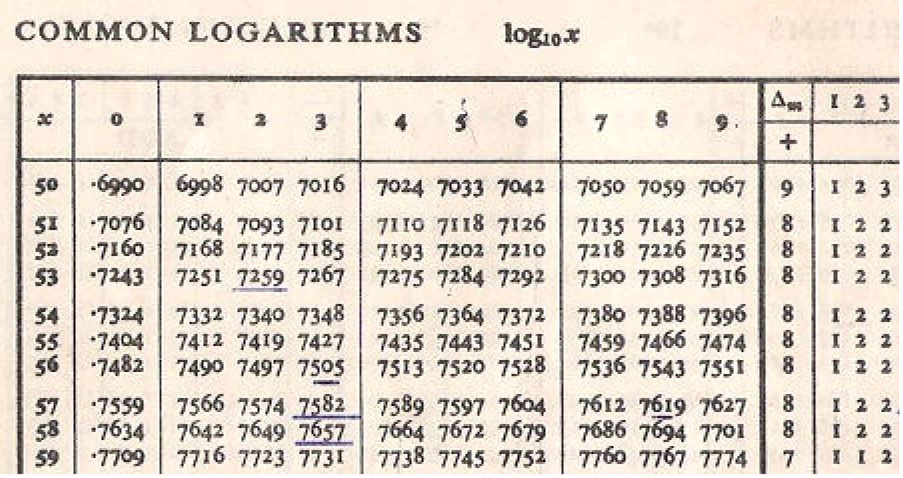
Figure 3. Model Runs Using Cloud Computing

Type III systems usually perform the model runs in real-time—when the event is occurring—using both forecasted and telemetric data. In those cases where physically-based spatially-distributed hydrologic models are better suited to represent the behavior of the watershed the resulting models can be computationally intensive, especially if the time to get a useful result is a strong constraint. Therefore, this type of system would require the use of High Performance Computing (HPC) for fairly complex models, or it should be restricted to very simple models, which in some cases could not be accurate enough for effective decision support and emergency response. HPC, or more commonly known as "Supercomputing", requires access to highly concentrated and optimized clusters of processors, usually built and maintained by large governmental or educational institutions. The use of HPC generally presents issues of security, clearance, and scheduling, and furthermore, usually requires specialized software programs.

Type IV systems are the most technologically complex. The system not only evaluates the conditions, but it acts upon those conditions (e.g. opening or closing valves and gates) in order to minimize the negative effects of the flood. This approach incorporates the complexity of Supervisory Control and Data Acquisition (SCADA) systems, which is usually only feasible on highly urbanized environments.

In conclusion, Type II systems are well suited for a broad range of applications, and can be adapted to different conditions. For instance, in a developed country the hydrologic models could be created locally, and the scenarios can run using existing computing resources. The complexity of the scenarios could be higher, and data from *ad hoc* installed telemetry could be used. In contrast, in a developing country the development of the hydrologic models could be contracted to an expert modeler, and the model runs could be performed incrementally. The models could be simpler, and use forecast and telemetry data available for free through the Internet.

The key to this type of systems is the concept of pre-computation. This concept is not new, for it has been widely used in different realms to produce faster results: Pythagorean Tables, Logarithmic Tables, all kinds of Look-Up Tables in diverse software applications, Index Tables in databases, and many others have taken the advantage of “storing” calculation in a “ready to use” format. There are some disadvantages in using this method; the main one being the discretization of the continuum. For example, when using the logarithmic tables of old (Figure 4) it was not common to find the logarithm value for the very exact number in question, and one usually had to use the closest one, or perform some kind of interpolation between the closest ones.



COMMON LOGARITHMS $\log_{10}x$

x	0	1	2	3	4	5	6	7	8	9	Δ_m	1	2	3
											+			
50	·6990	6998	7007	7016	7024	7033	7042	7050	7059	7067	9	1	2	3
51	·7076	7084	7093	7101	7110	7118	7126	7135	7143	7152	8	1	2	2
52	·7160	7168	7177	7185	7193	7202	7210	7218	7226	7235	8	1	2	2
53	·7243	7251	7259	7267	7275	7284	7292	7300	7308	7316	8	1	2	2
54	·7324	7332	7340	7348	7356	7364	7372	7380	7388	7396	8	1	2	2
55	·7404	7412	7419	7427	7435	7443	7451	7459	7466	7474	8	1	2	2
56	·7482	7490	7497	7505	7513	7520	7528	7536	7543	7551	8	1	2	2
57	·7559	7566	7574	7582	7589	7597	7604	7612	7619	7627	8	1	2	2
58	·7634	7642	7649	7657	7664	7672	7679	7686	7694	7701	8	1	2	2
59	·7709	7716	7723	7731	7738	7745	7752	7760	7767	7774	7	1	1	2

Figure 4. Detail of a Logarithms Table for Numbers With One Decimal Place

These tables had to find a balance between the "usability" of the table, which would ask for fewer and easier to find values, and the "accuracy", which would call for a larger number of values. The same problem arises in this case: We cannot pre-calculate every possible scenario, since there is an infinite number, but we must pick a set that will be sufficiently representative. It goes without saying that similar to the logarithmic table example, the larger the number of scenarios pre-computed, the greater the accuracy of the system.

Pre-computation is being applied to some hydrology-related problems. For example, pre-computation is at the core of systems for coastal floods arising from storm-surges (Taflanidis et al., 2012) or tsunamis (Wächter et al., 2012). For these kind of floods there are several parameters that could strongly influence the outcome. In the case of a storm surge flood, the wind intensity and direction, the atmospheric pressure distribution, and the landfall point of the storm or hurricane are the most sensitive (Kennedy et al., 2012). For tsunamis the sensitive parameters include the location and intensity of the earthquake producing the wave (Titov et al., 2005). The systems developed to assess both of these problems are based on complex 3D finite element models, which require the use of high-performance computing for pre-computing the different scenarios, or combination of parameters values. Since a run of each scenario is computationally demanding, these scenarios are carefully selected, and much research has been done on how to identify the most sensitive parameters to include in the model. Also, since it is only feasible to run a limited number of scenarios it becomes necessary to bridge the gaps between them using interpolation techniques or machine learning methods.

In flood forecasting systems there are also a few examples of this approach, in addition to the ESPADA system already mentioned, we find the SWF (Surface Water Flood) Warning Systems in England (Ochoa-Rodríguez, Thraves, & Johnston), and the FLIWAS (Flood Information and Warning System) in Germany. Nevertheless, in the published bibliography there are neither references about how the scenarios for these systems where generated, nor specifics about the method used to retrieve the results of the pre-computed scenarios from their respective storage systems.

Notwithstanding the fact that a surface flood system based on pre-computed models still has limitations on accuracy due to the inherent discretization that the method requires, the ability to compute hydrologic models without a strong time constraint enables the possibility of running complex models, like physically-based, spatially-distributed models, using regular computational resources, and that sole advantage could largely overcome the loss of accuracy due to discretization. Also, as is the case with the storm surge systems, some interpolation techniques could be implemented if deemed necessary.

1.2 Research Objective

Despite the stated advantages of Type II systems, they still pose a series of difficulties to address, such as:

1) The development of a method to generate and retrieve scenarios

2) The development of a reliable base model of the watershed

3) The identification of the parameters that will define the scenarios

4) The availability of forecast and telemetry data

5) The number and type of scenarios to model

6) The plan for the pre-computation of the scenarios

7) The strategy to increase the accuracy of the system through time

All these issues are inter-related, their resolution is not trivial, and therefore they should be faced as an incremental and iterative process. In this process concepts such as spatial and temporal resolution, sensitivity, non-linearity, granularity, discretization, and interpolation frequently appear. The objective of my research was to develop a complete method to implement a flood forecasting system based on pre-computed scenarios using a physically based, spatially distributed hydrologic model engine.

1.3 Brief Summary of the Chapters that Compose this Dissertation

In the following chapters of this dissertation I will analyze in detail the seven identified difficulties outlined above, and I will propose methods and techniques to address each of them. Each analysis will be illustrated by test cases and real-world implementations. Here is a brief summary of the content of each of the following chapters:

Chapter 2: Defining and Retrieving Scenarios. In this chapter I introduce a method to generate, compute, and store the results of different variations of a "base" model in advance of a flooding crisis. These variations, or "scenarios", consist of a unique combination of parameters. I analyze a way to create the scenarios so to maximize the uniformity of their distribution on the parameters space. Then, I present a simple technique to retrieve the stored result that best matches the current conditions. I illustrate this method in a test case, for which seven model parameters were selected and more than 2,000 scenarios were generated and pre-computed.

Chapter 3: Implementing a Basic Flood Forecasting System. In this chapter I explain in detail how the concepts presented in the previous chapter can be implemented in a simple implementation. I describe the steps I took to generate a base model for the Hobble Creek watershed, how I defined the parameters and their quantiles using sensitivity analysis and assessing the availability and accuracy of the real-time observed and forecasted data, how the 57,600 resulting scenarios were pre-computed, how the system was mounted, and what results it provided.

Chapter 4: Assessing the Relationship Between the Number of Scenarios and the Accuracy of the System. In this chapter I present a technique to assess the relationship between the number of scenarios and the expected accuracy of the results. This technique is intended to test the intuitive assumption that similar conditions produce similar results, which lays at the foundation of the proposed method. To test the technique, I use the results from the system described in the previous chapter and evaluate the development of the performance of the system as the number of scenarios increases.

Chapter 5: Prioritizing the Scenarios to Compute. So far, "all scenarios are born equal". This means that once the scenarios are defined they are sequentially included in the system without any particular order. In this chapter I propose a technique to use information from the previously computed scenarios to determine which of the remaining scenarios should be calculated next.

The objective is to compute and include first the most significant scenarios, which are the ones that define the boundary between the "no flood" and "flood" regions of the parameters space.

Chapter 6: Generating Scenarios for Time-series. Sometimes the scenarios can include or consist of time-series. In this case the "Latin Hypercube" approach is not appropriate because, generally, time-series can't be just a combination of random values. In this chapter I present a technique to generate "meaningful" time-series using historical observed data and clustering algorithms. The method is illustrated in a real-world problem in which this technique was applied.

Chapter 7: Conclusions. Finally, in this chapter I discuss the results of the method and its accessory techniques I presented in the previous chapters.

2 DEFINING AND RETRIEVING SCENARIOS

The main purpose of a hydrologic model used in flood prediction is to transform a weather or meteorological forecast into a flood or hydrological forecast (Nash & Sutcliffe, 1970). The inputs of that model are the weather forecast (rainfall depth and duration, temperature, etc.) and other current conditions on the watershed (soil moisture, the snow depth, etc.) (Refsgaard, 1997). The most likely outputs of the model are a hydrograph and a flood map, or flood animation. Physically-based, spatially-distributed hydrologic models are particularly well suited for this purpose: they have a large set of potential input parameters to modify, and since they model the interaction of multiple hydrologic processes, they are usually considered more reliable than lumped parameters models in performing calculations outside of the ranges of the data that were used to calibrate the models.

Hydrologic models have become increasingly accurate in the prediction of the behavior of a watershed under different meteorological and hydrological conditions (Beven, 2011). Nevertheless, models that consider physical processes over relatively small spatial and temporal scales can require a significant amount of computation time which makes them difficult if not impossible for use during emergencies—when time becomes critical. The adage "when the time to perform arrives the time to prepare has passed" is particularly true in crisis management. As we stated before, one way to overcome the time availability constraint is to pre-compute scenarios generated using hydrologic model engines.

The objective of this chapter is to present a simple method to (1) define evenly distributed scenarios, or unique combinations of pre-selected parameters, and to (2) retrieve the pre-computed results using a basic Euclidian distance calculation. The performance of the method will be assessed by comparing the results they obtain using this method to the ones that would be obtained by actually running the models in real-time.

To illustrate the proposed method I developed a test case using a fully-distributed, physically-based model engine: Gridded Surface/Subsurface Hydrologic Analysis (GSSHA). GSSHA is a hydrologic model engine with the capacity to couple modules of different hydrologic processes, such as infiltration, surface flow, groundwater flow, snow melt and accumulation, and 1D stream-routing. It implements mass-conserving solutions that guarantee overall mass balance (Downer & Ogden, 2004). The modularization of GSSHA provides more control over the individual hydrologic processes through a great number of input parameters; and given its physically-based characteristic, also provides the confidence to use it in an "extrapolation mode"—outside of the range of the variables used for calibration. This characteristic is particularly important when predicting extreme events. Using GSSHA, we can create thousands of different scenarios by modifying and combining a wide variety of hydrologic parameters.

The principle underneath the proposed method is that similar conditions will produce similar results. Using this method we can generate a large set of scenarios representing different weather and watershed conditions, and then compute and store the results. The main advantage is that the pre-computation can be done in the time between flood events, and that it can also be done in parallel. Thus, the resulting archive of results—and possibly the accuracy of the system—becomes better over time. When the next severe storm forecast arrives, the users select the result corresponding to the scenario that most closely matches the current conditions, and they assume that that result will be similar to the results they would obtain by modeling the actual current conditions. With this method a model result can be retrieved in seconds rather than the hours it might take to modify and execute an existing model. Furthermore, as a storm draws nearer and forecasts change, new "best match" simulation results can be pulled from the archive.

2.1 Method

The proposed method has two main components, the generation of scenarios, and the selection of model results. All the scenarios for this work were created as variations of a simple GSSHA model, referred to hereafter as the "base model". Figure 5 shows the grid of terrain elevations of the base model. This model corresponds to the 2.5-km^2 watershed of the Eau Galle River, in Wisconsin. Since the model would only be used to provide a functioning model in GSSHA, and not a real flood forecast, it was not thoroughly calibrated. The model grid has 74 rows and 57 columns, with a cell size of 30 meters. I used a time step of 10 seconds for the calculations as this value produces a balance between the total computation time of each scenario and the stability of the model. I kept the model simple enough to run in a few minutes, thus allowing them to run a big number of scenarios to test the proposed method.

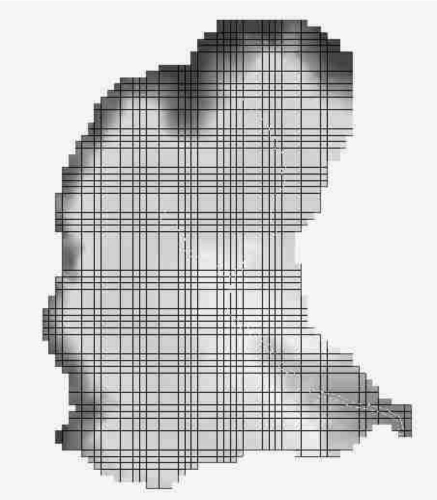


Figure 5. Elevations on the GSSHA Base Model

2.1.1 Parameter Selection

Regarding the selection of the parameters, we must note that for my current purpose I chose a set of parameters that would drive the interaction of GSSHA's runoff and snowmelt modules. No in-depth analysis was made about all the potential parameters in the watershed. In the next chapter a more comprehensive analysis of potential parameters will be presented. For now, it may suffice to say that in a real-world implementation of this method, the selection of the parameters is critical, and many others could be considered besides the ones I am using in this test case. In this regard we must note that each parameter has to be evaluated from two points of view: the first is the sensitivity of the model to the parameter, especially in the flooding regions of the possibilities space; and the second is the real-time availability and accuracy of the values of the parameter, because those values will define the "current conditions" that will be used to identify the best pre-computed scenario to retrieve. In all cases, the decision to include or not include a certain parameter, and at which resolution (or number of quantiles, as will be explained later), should be addressed on a case-by-case basis. Therefore, for this model I assumed that in a previous analysis a set of significant parameters was identified, based on the unique characteristics of the watershed, and on the accuracy of the real-time and forecasted data available in the region.

2.1.2 Scenarios Generation

The method requires a set of different scenarios that populate a “possibilities space”. Each of these scenarios is created by modifying the parameters associated with the base model. For the present case, I elected to create these scenarios by modifying three of the GSSHA model input files: HMET (meteorological data), GAG (precipitation data), and SWE (snow cover data).

The HMET file was modified by two parameters: *temp* and *tempamp*. The *temp* parameter specified the average daily temperature, while the *tempamp* parameter defined the amplitude of the temperature variation throughout the day. The hourly variation of the temperature was defined by a sinusoidal function, setting the maximum at 3:00 pm every day. The remaining factors in the HMET file were not scenario-defining parameters, and therefore they were assigned constant values:

pressure = 29.8 mmHg

relative humidity = 50 %

cloud cover = 50 %

wind speed = 10 knots

The GAG file was modified by three parameters: *rainlength*, *rainint*, and *rainstart*. The *rainlength* parameter defined the total length of the rain event. The *rainint* parameter defined the total depth of the rain event. The *rainstart* parameter indicates when the rain event starts during the first day of the simulation; this parameter is important because I want to consider the coupled effect of both precipitation and snowmelt, which can overlap if the rain event starts near the time of the maximum temperature. These three parameters defined a uniform hyetograph with an interval of 15 minutes.

Finally, the SWE file was modified by two parameters: *snowline* and *snowgrad*. The *snowline* parameter defined the elevation at which the presence of snow starts. The *snowgrad* parameter specified the rate at which the snow depth increases with elevation from the *snowline*. Using these two parameters a map containing the snow depth at each cell was created for a scenario.

In summary, seven input parameters from three GSSHA files were altered and combined to create the set of scenarios. The range of values used for these parameters can be seen in the Table 1. These ranges were defined to produce extreme events, and not necessarily correspond to observed values in the region.

Table 1. Range and Units of Selected Parameters

Parameter	Min value	Max value	units
temp	20	100	°F
tempamp	5	20	°F
rainlength	1	10	hs
rainint	10	100	mm
rainstart	0	24	hs
snowgrad	0	0.002	m/m
snowline	-100	400	m

2.1.3 Statistical Distribution

If the objective had been to generate equally probable scenarios, we would be required to use the best statistical distribution for each parameter —normal, lognormal, Pearson, etc. We also should take into account the correlation of the parameters, such as *temp* and *snowline*. In this case, however, our objective is to create low probability scenarios that produced extreme flows since it is only under extreme conditions that a flood forecast is utilized. It is logical to expect that these scenarios will be produced by some combination of the extreme values of the parameters (e.g. high *temp* and low *snowline*). Therefore, to bias the values to the extremes of the ranges I used the simplest alternative: a uniform distribution. Further, to prioritize the combination of these extremes I considered all of the parameters uncorrelated.

2.1.4 The Importance of Uniformity

One of the keys of the proposed method is that the parameters' multi-dimensional space should be evenly populated. Both gaps and clustering of scenarios will reduce the potential accuracy of the method. In other words, the method should ensure that, given any combination of parameters, a pre-computed scenario should always exist close enough so its results could be considered a valid "substitution" of the searched combination. In reality, the ultimate objective would be to obtain an evenly-distributed set of results, in such a way that if, due to non-linearities, two different combinations lead to one similar result it wouldn't be necessary to store it twice. Nevertheless, it is impossible to know a priori what the results will be, so the second best option is to evenly distribute the parameter combinations.

To illustrate this concept we can think of a simple mathematical function $y = f(x)$. We want to have the dependent variable space (in this case the y axis) evenly populated, so all the possible outcomes can be represented accurately. If *f(x)* is a linear function, it would suffice to pick evenly spaced values of the independent variable, *x*, and apply the function to obtain evenly distributed values of *y* (Figure 6).

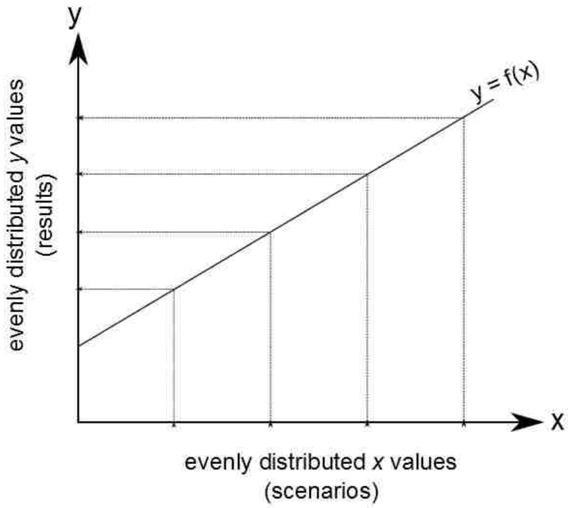


Figure 6. Example of Distribution of Scenarios and Results in a Linear Function

On the contrary, if this function is strongly non-linear, the evenly spaced *x* would not necessarily generate evenly spaced values of *y* (Figure 7).

If we knew the inverse function *x* = *g(y),* we could select the evenly distributed *y* values and then find the corresponding *x* values (Figure 8).

In our models the situation is not that simple: we have several independent variables (the parameters) and the results are not unique values, but hydrographs and flood maps. Also, there is no “inverse model” that could be used to predict combination of parameters based on a target hydrograph or flood map, so we are compelled to select evenly-distributed sets of parameters, anticipating to get quasi-evenly-distributed results.

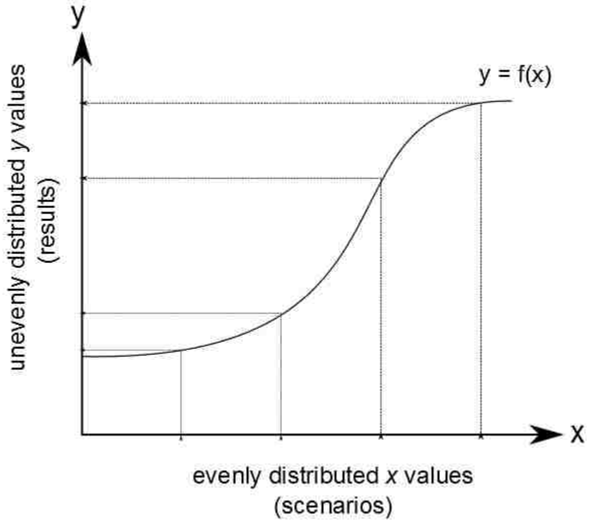

y
x
y = f(x)
unevenly distributed *y* values
(results)
evenly distributed *x* values
(scenarios)

Figure 7. Example of the Attainable Distribution of Scenarios and Results in a Non-linear Function

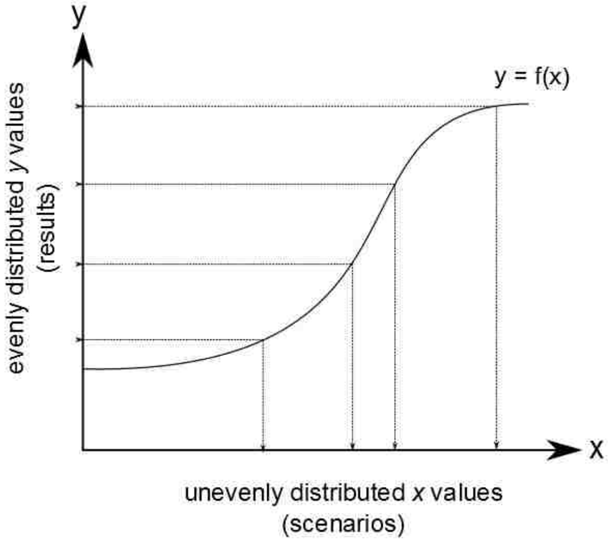

y
x
y = f(x)
evenly distributed *y* values
(results)
unevenly distributed *x* values
(scenarios)

Figure 8. Example of the Desired Distribution of Scenarios and Results in a Non-linear Function

2.1.5 Monte Carlo

The question then is how to create scenarios that are evenly distributed in their multi-dimensional space. Some methods, like the Monte Carlo (Metropolis & Ulam, 1949), are commonly used to generate random combination of parameters. They have the advantage of not requiring a pre-established number of scenarios to complete, since a trigger condition, such as some sort of statistical convergence, can decide when enough scenarios were created.

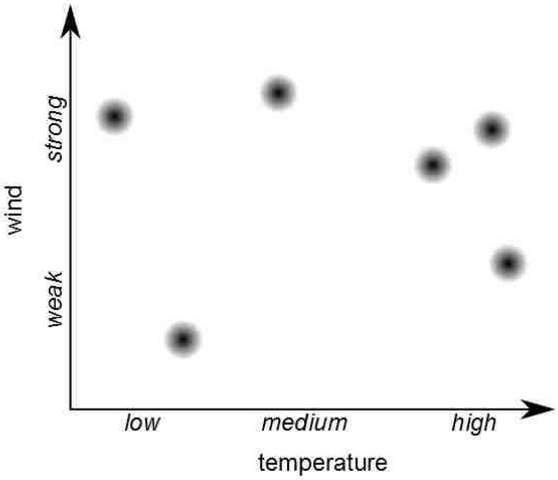


Figure 9. Scenarios Generated Using the Monte Carlo Method

Notwithstanding, since the parameters are randomly selected there is no guarantee that there could exist some empty or low-density regions in the parameters' space. For example, if we had two parameters, "wind" and "temperature", the method could create two scenarios with strong wind and high temperature, but none with weak wind and medium temperature (Figure 9).

2.1.6 Latin Hypercube

A good way to evenly populate the parameters space is to use a combinatorial sampling method, based on the Latin Hypercube sampling method (McKay, Beckman, & Conover, 1979). Succinctly explained, the Latin Hypercube method consists on dividing each set of parameter values range into a series of bins, (e.g. "quantiles", if the bins are equiprobable), and then generating all the possible combinations among the bins from each parameter. For instance, if we had the same two parameters from the previous example: "wind" and "temperature", we could divide "wind" into two bins: "strong", "weak", while we could divide "temperature" into three bins: "high", "medium", "low", then, using the combinatorial sampling method we can create model instances with samples derived from all possible combinations of these parameter values, constituting each combination with a different "scenario": "strong/high", "strong/medium", "strong/low", "weak/high", "weak/medium", "weak/low" (Figure 10).

Combinatorial sampling, therefore, assures an even distribution of scenarios, based on the number of bins or quantiles established for each parameter. The problem with this method, however, is that the generation of scenarios usually follows a systematic procedure (e.g. all the "low temperature" combinations are generated first) and so, while the scenarios are computed, the "parameters space" presents regions that are "empty" (e.g. no "high temperature" scenarios will exist until two thirds of the total number of scenarios are computed). In summary, only after the whole set of scenarios has been computed can the "scenarios space" be considered properly populated, and hence useful. This could be a serious limitation of the method; nevertheless, I will briefly address a possible way to overcome this problem in the next chapter.

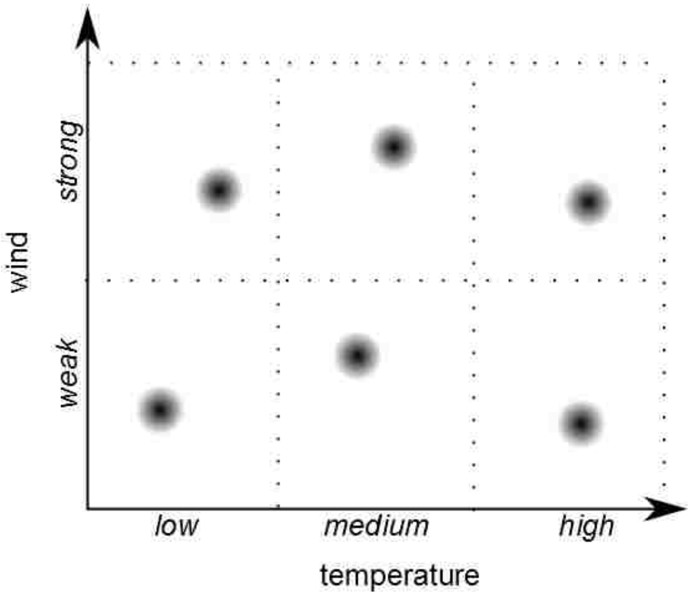


Figure 10. Scenarios Generated Using the Combinatorial Sampling Method

Therefore, for this test case I used a combinatorial approach based on the Latin Hypercube method. I divided the probability distribution of each of the seven parameters into three quantiles. The Latin Hypercube Method ensured that I had scenarios with all possible combinations of low, middle, and high values of each parameter. Given seven parameters and three quantiles, the total number of combinations was:

$$Quantiles^{parameters} = 3^{7} = 2{,}187$$

Since I used uniform distributions, these quantiles were defined by dividing the range of the parameter into three equal bins; and the actual value of the parameter to be used in the scenario generation was then chosen randomly inside of the corresponding bin.

Given that I would use Euclidian distances in parameter space to select the best-matching pre-computed scenario, it was necessary that the distances in each dimension were comparable.

To achieve this, each parameter *i* was normalized (transformed into a number between 0 and 1) for each scenario *j* using the following formula:

$$P_{ji} = \frac{p_{ji} - \min_i}{\max_i - \min_i} \qquad \textbf{(1)}$$

Where:

p_{ji} = parameter *i* of the scenario *j*

min_i = minimum value of the parameter *i* in the range

max_i = maximum value of the parameter *i* in the range

P_{ji} = normalized parameter *i* of the scenario *j*

For instance, a *temp* value of 90.1°F corresponded to a normalized value of 0.87625:

0.87625 = (90.1°F - 20°F)/ (100°F – 20°F)

2.1.7 Resulting Scenarios

Using the Latin Hypercube based method, I created the 2,187 combinations of input parameters, or scenarios that could be pre-computed with in GSSHA and then “stored” for comparison against future forecasts. Radar plots corresponding to three of these scenarios are shown in Figure 11.

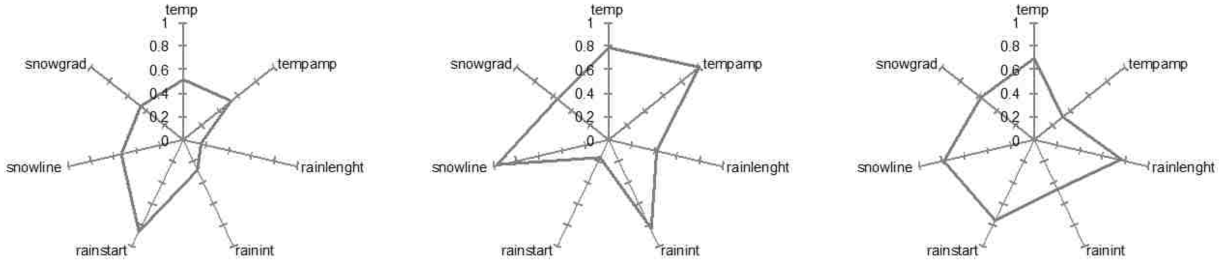


Figure 11. Three Examples of Scenarios

2.1.8 Model Runs

Using the generated parameter combinations I then created the GSSHA model input files corresponding to each scenario, and ran the resulting models. Each of the models returned a hydrograph, which I analyzed and extracted two key values: the peak flow (PF) and the volume (V) of water that crossed the outlet point of the watershed. All of the combinations of PF and V are plotted in Figure 12. (The PF and V values have been normalized by dividing them by the Maximum Peak Flow (MPF) and the Maximum Volume (MV) respectively.)

The resulting plot shows that even though I used a uniform distribution and considered the variables uncorrelated, I still obtained a high concentration of scenarios in the region corresponding to lower values of both V/MV and PF/MPF. Unfortunately, these scenarios are not useful for forecasting extreme events associated with flooding conditions; and because it is not possible to know *a priori* which combination of parameters will produce extreme results, this problem is unavoidable. Nevertheless, in the chapter 4 we will discuss about a technique to determine which regions of the multidimensional parameters space should be populated first to increase the accuracy of the system.

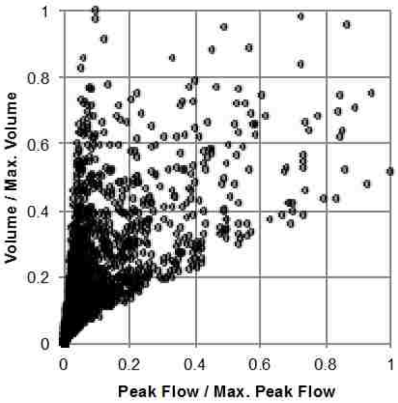


Figure 12. Results of the Scenarios Runs

2.1.9 Scenarios Selection

After the scenarios were pre-computed and placed into the scenario archive, the next step in the process was to a take a set of parameter values corresponding to “current” conditions at the time of a forecasted extreme storm event and select the pre-computed scenario from the archive that best matches those values. Given that each scenario consists of a set of seven parameters, each test scenario can be considered as a "point" in the seven-dimensional normalized space. The seven parameters corresponding to the current condition also defines a "point" in the same space. Therefore, we can find which of the pre-computed points is closest to the forecasted conditions point by calculating the distance between the forecasted point and each of the pre-computed ones.

2.1.10 Multidimensional Euclidian Distance

To determine the closest scenario to the forecasted conditions we can use the Euclidian distance defined by:

$$D_j = \sqrt{\sum_{i=1}^{n} \left(P_i^C - P_{ji} \right)^2} \qquad \textbf{(2)}$$

Where:

n = number of parameters

D_j = distance between the scenario j and the current conditions

P^{c}_{i} = normalized parameter i of the current conditions

P_{ji} = normalized parameter i of the scenario j

Using this formula we can calculate the distance between the forecasted conditions and each of the pre-computed scenarios, and identify which scenario is the best match. Then, assuming that the results of that scenario would be similar to the results we would obtain by modeling the forecasted conditions, we would have model results on which to base flood predictions and implement an emergency response.

To illustrate how this works, I generated three different hypothetical forecasted conditions and determined which of the pre-computed scenarios was closest to it. The results are shown in Figure 13.

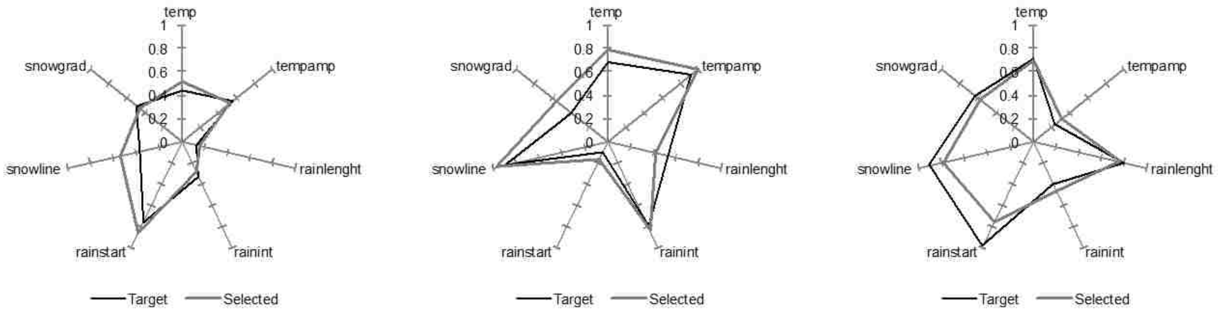


Figure 13. Three Examples of Scenario Selection

For example, in the first scenario in Figure 13, we observe that the method found a scenario with almost identical values for temperature amplitude, rain starting time, snow line, and snow gradient. The temperature of the selected scenario is a little higher, rain length is slightly lower, and rain intensity is somewhat less. In the other two cases, we can observe a similar closeness of fit between forecasts and pre-computed scenarios.

2.1.11 Weighted Distance

Another question might be whether all the parameters should have the same weight in the calculation of the distance. To address this question I created a set of test scenarios or synthetic "current conditions" and ran the models using these parameters. Instead of selecting the scenarios with the closest input parameters, I "inversed" the method by selecting the pre-computed scenario based on the similarity between resulting hydrographs. From those matches I tried different methods to derive weights for each of the parameters. These weights would later be applied in the "direct" method using the following modified distance formula:

$$D_j = \sqrt{\sum_{i=1}^{n} W_i \left(P_i^C - P_{ji}\right)^2} \qquad \textbf{(3)}$$

Where:

W_i = weight corresponding to the parameter i

Through this exercise I discovered that in some of the test cases the weights indeed produced better matches; but in other cases the matches were worse. In consequence, the present system does not weight the parameters. Nevertheless, further study should be made in this regard.

2.1.12 Catalog Method

Finding a best match can be performed by an alternative method that doesn't require computing a distance for each stored scenario. Rather than randomly selecting a parameter value in the selected quantile during the scenarios generation, a central value for each quantile can be used, and then a catalog of the 2,187 scenarios can be created. Then to find the best match, it can be determined in which of the quantiles each parameter of the current conditions would fall, the corresponding scenario in the catalog can be looked for, and the results can be retrieved. In other words, the quantiles combination becomes an indexing system that can be used to instantly locate the best match in the archive.

Notwithstanding, there is a major drawback in this "catalog method". In the proposed method, we may want to increase the precision of the system through time, by the creation of additional scenarios. There are several potential techniques to do that, but one of the most straightforward ways is to redefine the number of quantiles from any—or all—of the parameters.

For instance, in the present test case we could decide to increase the number of the quantiles of all the parameters from three to four. To have this set of scenarios completed, it would require the pre-computation of 4^7—16,384—scenarios. With the "catalog method", it would be necessary to compute all of them, because the values at the centers of a 3-quantile would not correspond to the values at the center of a 4-quantile. Nevertheless, it would be useful for us to be able to "recycle" the previous 2,187 scenarios, from the 3-quantile schema.

Conversely, with the "Euclidian distance method" this can be easily done. It would only require reviewing the existing set of scenarios, and reassigning each of them to their corresponding place in the new schema, so we don't need to define and compute a new scenario there. As an example, in Table 2 we can see how scenario #1,875 in the 3-quantile schema becomes scenario #14,167 in a new 4-quantile schema.

Table 2. Reuse of the #1875 3-quantile Scenario

Parameters	3Quantile	4Quantile	Value
temp	3rd	4th	0.87665
tempamp	2nd	2nd	0.36473
rainlenght	3rd	4th	0.80741
rainint	1st	2nd	0.2967
rainstart	2nd	2nd	0.35116
snowline	1st	2nd	0.31854
snowgrad	3rd	3rd	0.68824

2.2 Results

To test the method I generated a set of 100 test scenarios. Those scenarios would represent different (synthetic) current conditions. Each scenario was defined by seven randomly generated values of the parameters. Since these scenarios were generated to assess the validity of the method it was necessary to run these 100 test scenarios and obtain a hydrograph for each, which could later be compared to the hydrograph corresponding to the closest pre-computed scenario. After the simulations were completed, I calculated both the peak flow (PF) and the volume (V) from the hydrographs of each of the 100 test scenarios and plotted them as shown in Figure 14.

Since our target application is flood forecast, we are only interested in those cases where we have a high values for PF and/or V. Therefore, I only present the results of the scenarios in which any of those values are more than 40% of the maximum—either PF, V, or both. In this case 13 of the 100 test cases met that criterion. In Figure 15 we observe a comparison between the modeled hydrographs of these 13 test cases and their matching hydrographs selected from the pre-computed results.

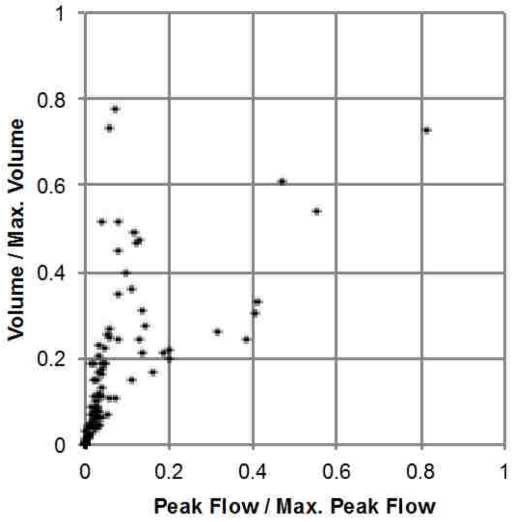


Figure 14. Results of the Test Scenarios Runs

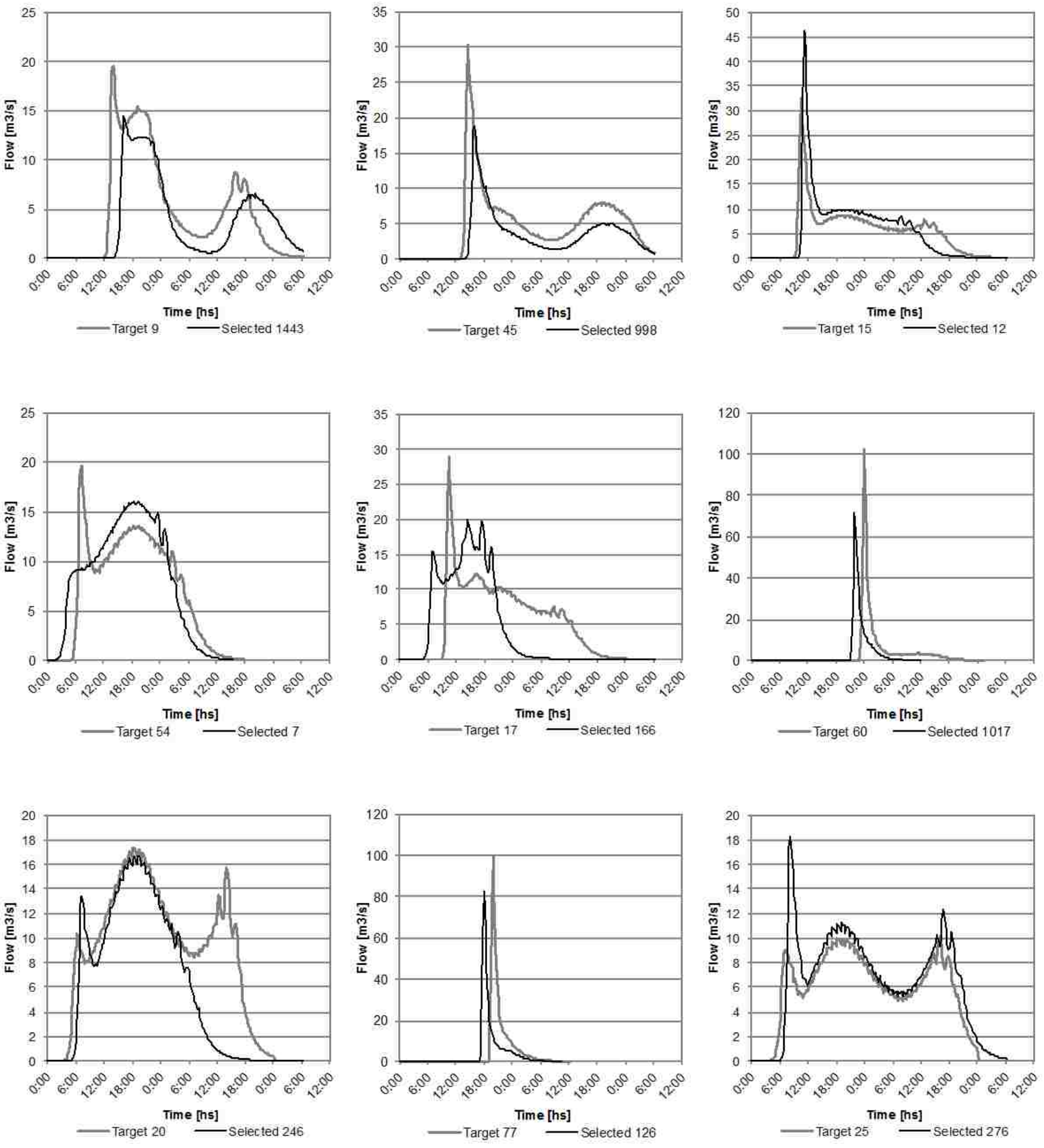


Figure 15. Comparison Between "Tested" (Current Conditions) and "Selected" (Pre-computed) Hydrographs

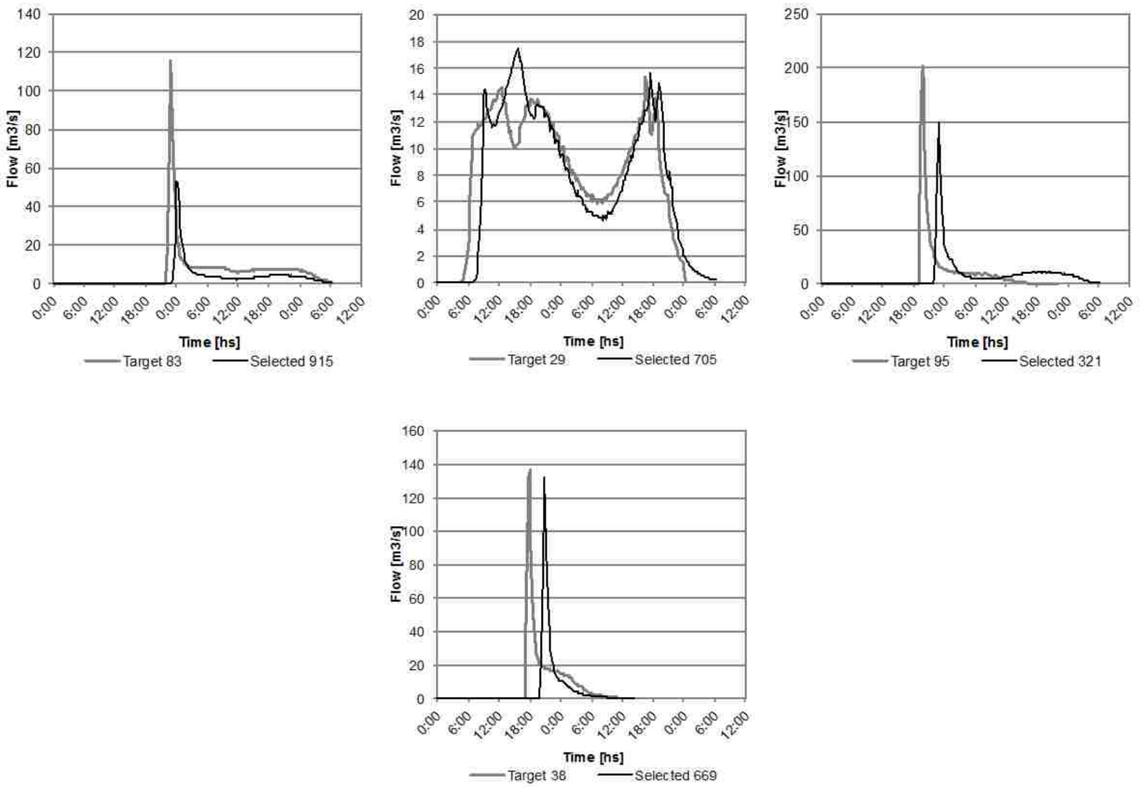


Figure 15 (continued). Comparison Between "Tested" (Current Conditions) and "Selected" (Pre-computed) Hydrographs

We can see that in general the paired hydrographs share at least one of the two most significant characteristics (and, in most cases, both): the total volume and the peak flow. Considering the relatively small number of quantiles for each parameter, the results are promising. Although it is beyond the scope of this work, a further analysis of the sensitivity of the model to each of the parameters, coupled with some objective method to assess the similitude of the results, would provide information about where to increase the resolution—number of quantiles—and consequently the precision of the system.

2.3 Conclusions

The purpose of this chapter was to prototype a systematic method to define evenly distributed scenarios, pre-compute them and retrieve the results to perform flood forecasting. The presented test case included only a few parameters to define the scenarios. In the following chapters this method will be expanded to include other parameters, such as soil moisture, temporal distribution of the precipitation, a more realistic snow model, albedo, etc.

We found that the differences between the tested and the selected hydrographs are in the same order of magnitude as the differences usually observed between a measured hydrograph and one obtained using a well-calibrated model. Consequently, any extra effort made to produce a better match to a pre-computed hydrograph would not necessarily translate into increased accuracy of the system.

In summary, I believe that the presented method provides a simple and scalable approach to develop almost instantaneous model results, and it is well suited for implementation in the core of flood forecasting systems.

3 IMPLEMENTING A BASIC FLOOD FORECASTING SYSTEM

An important part of the proposed methodology is the ability of the system to retrieve real-time observed and forecasted data, process them, generate the current target scenario, and retrieve the pre-calculated results from the database. For this purpose I implemented a test case flood forecast system for the Hobble Creek watershed. To determine the current conditions, this system automatically retrieves data from a set of existing web services. In this chapter we will analyze the development of such system, and its initial results.

3.1 Description of the Watershed

The Hobble Creek is a stream that rises in the west slope of the Wasatch Range, just east of the city of Springville, and after 36 km it outflows in the Utah Lake. The highest elevation in the watershed corresponds to Provo Peak, at 3,359 m. The lowest elevation, in the outlet at the Utah Lake is 1,368 m.

The watershed of the river covers an area of 320 km^2 with two very distinct zones: the mountains and the valley (Figure 16). In the mountains (Figure 17) the river has slopes in the range of 3.5% - 1.5%, while in the valley, which constitutes the lakebed of the pre-historic Lake Bonneville, the slopes are reduced to a range of 1%-0.3%. Given that in the mountains the riverbed is strongly confined by the topography, only in the valley is the stream prone to produce floods (Figure 18). In this potential flooding zone are located two cities: Springville (pop. 30,000) and Mapleton (pop. 9,000). Both snowmelt and rainfall can produce flow in the Hobble Creek, which makes it an interesting test case.

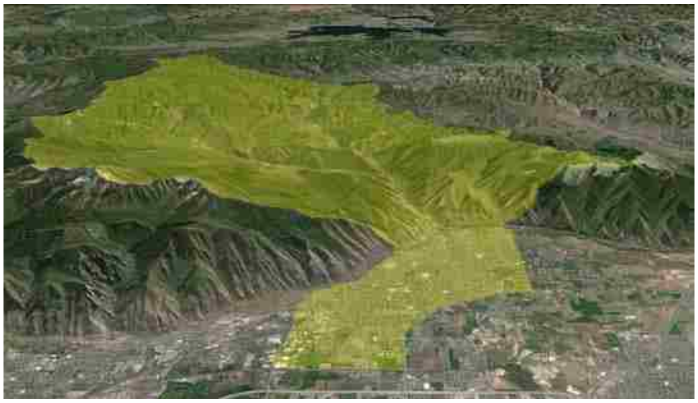


Figure 16. Hobble Creek Watershed

Figure 17. Right Fork of the Hobble Creek in the Mountains Region (Source: Google Maps)

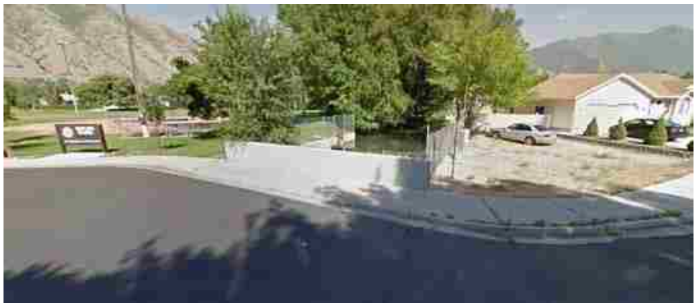


Figure 18. Channeled Section of Hobble Creek in Springville (Source: Google Maps)

3.2 Description of the Model

To create a model for this watershed I selected an outlet located at a bridge on the 1500 W street (Figure 19). The coordinates of the outlet are:

Latitude: 40.178960°

Longitude: -111.637438°

The main reason for the selection of this outlet point was the presence of the USGS 10153100 station located a few meters downstream from the bridge. Data from that station was used in the calibration of the model, and also it is used to generate the current target scenario for the system.

The watershed natural limits that resulted from that outlet were extended at the valley between 1 km and 2 km, both to the north and to the south margins of the river to include the most populated zones of Springville and Mapleton. To the north, the area is limited by the 1400 N street, and to the south, by Maple street, 1600 W street, State street, the railroad, and 400 S street. This resulted on a total area of the modeled watershed of 339.32 km^2

Figure 19. Location of the Outlet (Source: Google Maps)

Once the watershed was delineated, I created a GSSHA base model, which would be modified to generate all the scenarios to pre-compute. This model has a grid of 50 m cells organized in 469 rows and 586 columns. Once the cells outside of the boundaries are masked, this leaves a total of 135,728 active cells.

Terrain data were created using data provided by NASA's Shuttle Radar Topographic Mission (SRTM) (Hounam & Werner, 1999), soil data was obtained from the NRCS's State Soil Geographic (STATSGO) database (Schwarz & Alexander, 1995), and the land use was processed from the USGS's National Land Cover Data 2006 (NLCD) (Homer et al., 2007; Xian, Homer, & Fry, 2009).

GSSHA has the capability of calculating the base flow and the interaction between the groundwater and the stream flow, but that is only possible in long-term simulations. Our purpose, however, is not to analyze the fluctuations of the base-blow, but to simply “add” the peak flow to the existing base-flow to assess the probability of overbank flows. After considering several alternatives, I decided that the simplest way to implement the base-flow in the model was to inject a flow equivalent to that base-flow in a reach just upstream of the valley. This requires a "warm-up" of about three hours to allow the "base-flow" to reach the outlet, and the model to stabilize. The base-flow was defined to be the lowest observed flow in the last 24-hr period at the USGS station.

The model was calibrated using the precipitation event of 2-Dec-2012, where gathered records for weather data, precipitation data, soil moisture data, and snowpack data overlapped. The observed hydrograph can be seen in Figure 20:

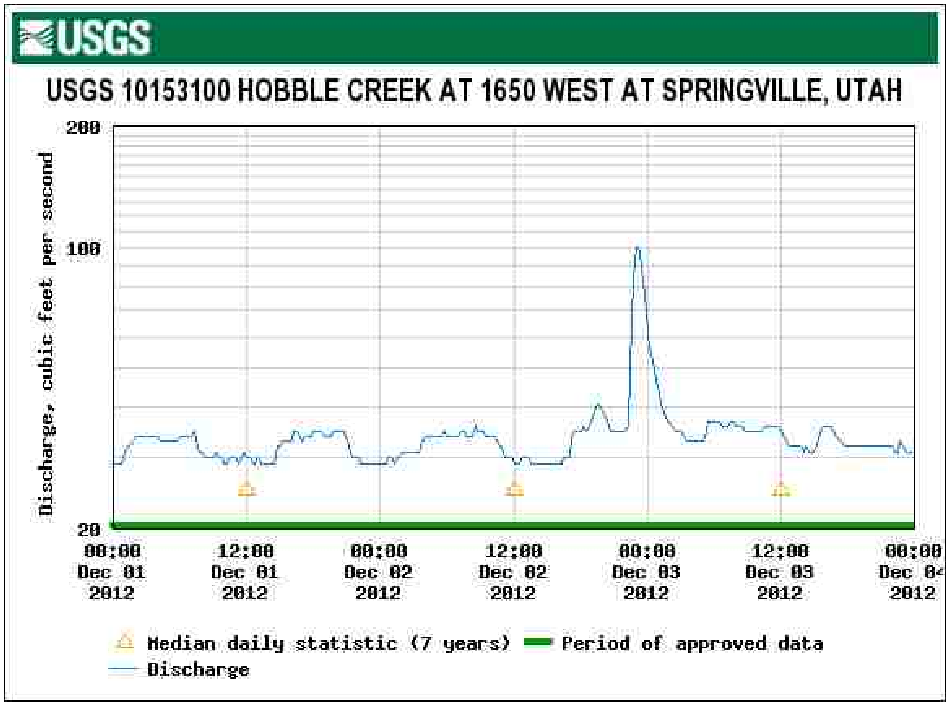


Figure 20. Observed Hydrograph for the 2 Dec. 2012 Event (Source: http://nwis.waterdata.usgs.gov/nwis/uv/?dd_cd=02_00060&format=img_default&site_no=10153100&begin_date=20121201&end_date=20121203)

Due to the lack of more data, the calibration was not exhaustive, and it was considered finished when the peak flow modeled matched the peak flow observed. This was considered sufficient for the scope of this project.

3.3 Selection of Parameters

In the first chapter of this dissertation we identified a set of issues that should be addressed in order to implement a flood forecasting system. In particular, three of these issues are tightly interwoven:

3) The identification of the parameters that will define the scenarios

4) The availability of forecast and telemetry data

5) The number and type of scenarios to model

Given that the parameters will be used both to generate scenarios and to later select the most probable scenario based on current values of those parameters, it is very important that (1) the model results are sensitive to the parameters selected, and that (2) observed – “current” – values will be available from telemetry or as output from other models.

An example will illustrate this: it is a known fact that the magnitude of a flood depends strongly on the content of moisture present on the soil, and in consequence, almost any model would be highly sensitive to this particular parameter. Real-time soil moisture values can be obtained and spatially interpolated from measuring stations like SNOTEL stations, or can be retrieved from satellites (Njoku, Jackson, Lakshmi, Chan, & Nghiem, 2003), or can be indirectly estimated from weather models (Mahfouf, 1991), or even from simple regression models based on the rainfall of the previous days. If any of these sources is available, the number of quantiles to be considered for the model should be related to the known or estimated accuracy of the source to be used. Conversely, if no source is available, the parameter should be excluded from the parameters set altogether: No matter how sensitive the model is to the parameter under consideration, it will not be able to be used to select a specific scenario from the pre-computed set.

For the present model, after performing this type of analysis I identified eight parameters representing current and forecasted conditions that can be retrieved from on-line sources to define the scenarios. These parameters were named: *month*, *snow*, *rain*, *peaktime*, *maxtemp*, *amplitude*, *soilmoist*, *baseflow*.

The first parameter, *month* is, obviously, the easiest to retrieve, and the most certain to be accurate, since we know for sure the month for which we are predicting the flood. We expect this parameter to produce an effect when there is snow present due to the incident angle of sunshine radiation.

The second parameter, *snow,* can also be easily retrieved as snow water equivalent from an existing SNOTEL station in the watershed (Serreze, Clark, Armstrong, McGinnis, & Pulwarty, 1999). Nevertheless, this value represents the measured snow at a fixed point, and has to be spatially extrapolated to the whole watershed. In our model this is done exclusively by a linear function of the elevation. This parameter could be important in the presence of high temperatures, and can greatly affect the total volume of runoff.

The third parameter, *rain*, is the most important, because we expect the model to be highly sensitive to it, except for cases where the soil moisture or the temperatures are low. Unfortunately, the accuracy of this value depends on the accuracy of the weather forecast from which it will be retrieved. In our model, to be conservative, the rain event is always modeled as a 6-hr Type II rain that concentrates the whole forecasted precipitation for the day, regardless of the forecasted temporal distribution (SCS, 1975).

The fourth parameter, *peaktime*, can also retrieved from a weather forecast. The peak of the 6-hr Type II event is calculated so that it matches the mass-center of the precipitation forecast.

The parameters *maxtemp* and *amplitude* can also retrieved and calculated from a weather forecast. The parameter *maxtemp* indicates the maximum temperature forecasted, while *amplitude* corresponds to the difference between the maximum and minimum forecasted temperature. To transform them into an input for the model, which requires hourly data, these values are temporally distributed following a sinusoidal function, with the maximum temperature corresponding to 3:00 pm. These parameters can trigger snowmelt flow, thus increasing the total volume, or transform a precipitation event into snow accumulation. In consequence, the model could be sensitive to them in certain regions of the modeling space.

The seventh parameter, *soilmoist*, can also be retrieved from the SNOTEL station. The model is most likely very sensitive to this parameter, which is by nature spatially variable. Lacking better data, in our model the value obtained is applied uniformly in the watershed.

Finally, the *baseflow* parameter represents the current flow in the stream. Its value can be retrieved from the existing USGS station (Falcone, Carlisle, Wolock, & Meador, 2010).

Each scenario, therefore, will consist of a unique combination of these eight parameters. The range of values for each parameter is shown in Table 3.

Table 3. Range of Values for the Selected Parameters

Parameter	Units	Min	Max
month	[month]	1	13
snow	[m]	-0.2	0.5
rain	[mm]	0	50
peaktime	[hours]	0	24
maxtemp	[°C]	60	120
amplitude	[°C]	20	50
soilmoist	[m^3/m^3]	0.2	0.4
baseflow	[m^3/s]	1	15

3.3.1 Sensitivity of the Model to the Selected Parameters

To determine the number of quantiles to use for each parameter we must analyze the sensitivity of the model to that parameter. Sensitivity to a certain parameter can be understood as the "partial derivatives" of the "results surface" (seen as a dependent variable) in that parameter direction (seen as an independent variable). Given that the results surface is not flat, the sensitivity to a certain parameter will be different depending on the region under analysis. For example, as we noted before, the sensitivity of the model to the parameter *rain* will be smaller in a region of the "results surface" where the soil moisture is low, compared to where the soil moisture is high.

Performing a sensitivity analysis in all the potential regions of the eight-dimensional results surface is an overwhelming task that would require computing *a priori* more scenarios than the ones we are implementing, so for this analysis we will analyze solely the sensitivity at the center of the parameters space, where the value for each parameter is the center value of its range.

But first, we must define what constitutes the "results", or in other words, what are the values that we will measure as "independent" variables to assess the sensitivity to each parameter. For our current purpose we will define two: the peak flow and the total volume of the flow. To properly use these results we must first adjust them by subtracting from the hydrographs the base-flows we injected. These adjusted peak flows and total volumes will better represent the effect of changing the parameters.

We will evaluate the sensitivity of the model by modifying each parameter 20% on each side of its middle ("mid") value. In the Table 4 we can observe how these values are calculated.

Table 4. Definition of the Values to be Used in the Sensitivity Analysis

Parameter	**Min**	**Mid-20%**	**Mid**	**Mid+20%**	**Max**	**Range**	**20% of range**
month	1	5	7	9	13	12	2*
snow	-0.2	0.01	0.15	0.29	0.5	0.7	0.14
rain	0	15	25	35	50	50	10
peaktime	0	7	12	17	24	24	5*
maxtemp	60	78	90	102	120	60	12
amplitude	20	29	34	41	50	30	6
soilmoist	0.2	0.26	0.3	0.34	0.4	0.2	0.04
baseflow	1	5.2	8	10.8	15	14	2.8

* must be integers

Now, we can run the model for the "mid" of all the parameters. We will call it the "base" scenario. In Figure 21 can see the resulting hydrograph. We observe that the peak flow is about 15 m^3/s over the base-flow of 8 m^3/s, and we can calculate the total volume of 65,000 m^3 over the base-flow total volume of 921,600 m^3. This illustrates the importance of adjusting for base-flow.

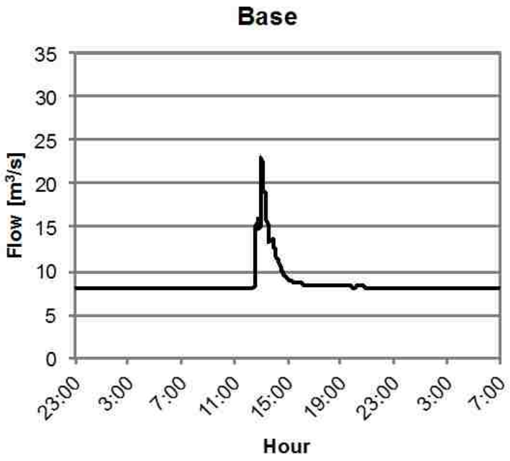


Figure 21. Hydrograph of the Base ("mid") Scenario

At this point we can run all the required scenarios modifying one parameter in one direction at a time. The results of the 16 scenarios are synthesized in the Figure 22.

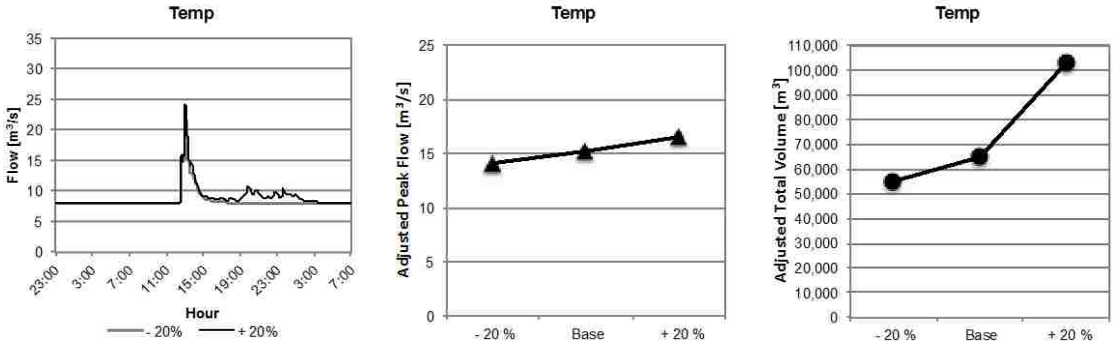


Figure 22. Sensitivity of the Model to Different Parameters

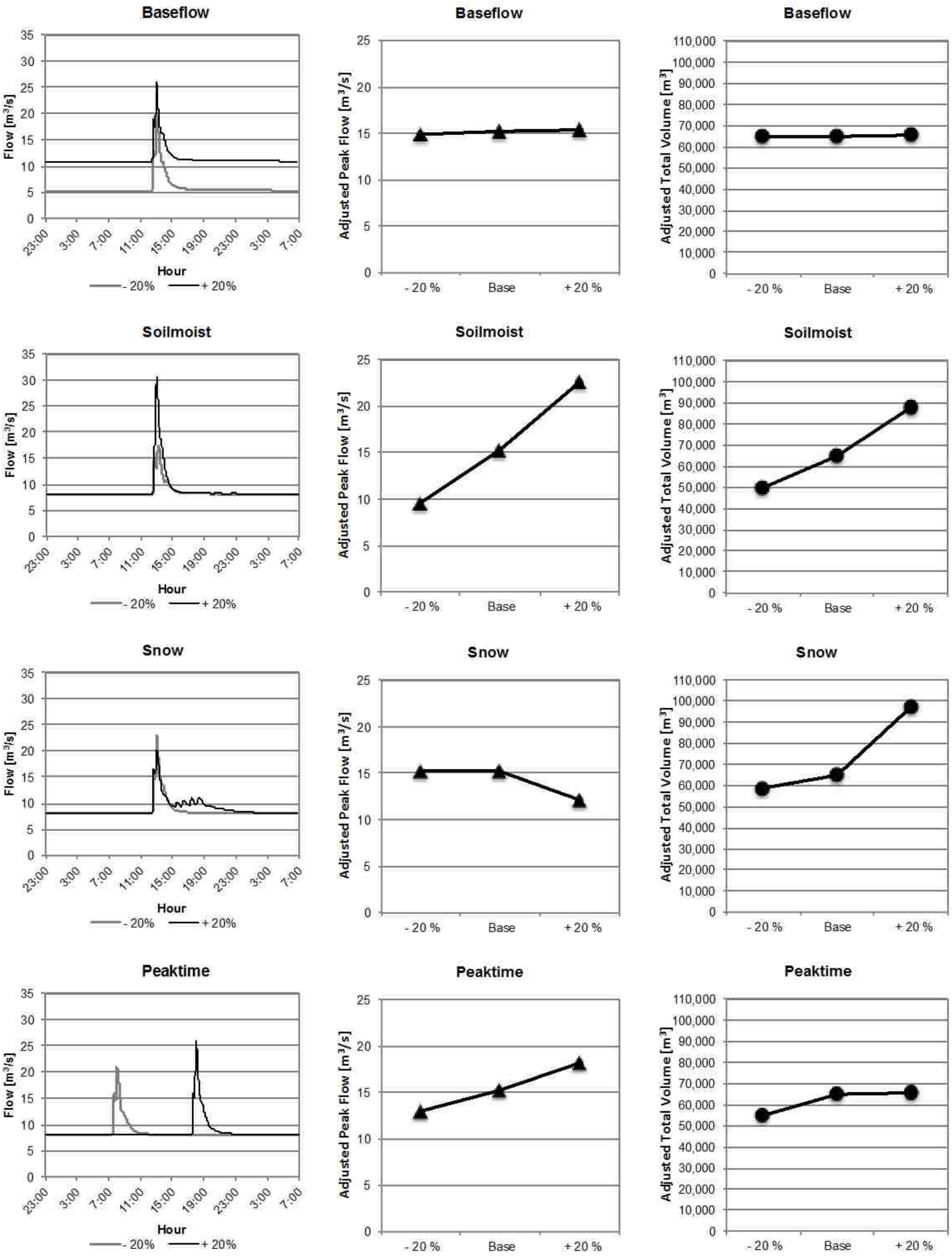


Figure 22 (continued). Sensitivity of the Model to Different Parameters

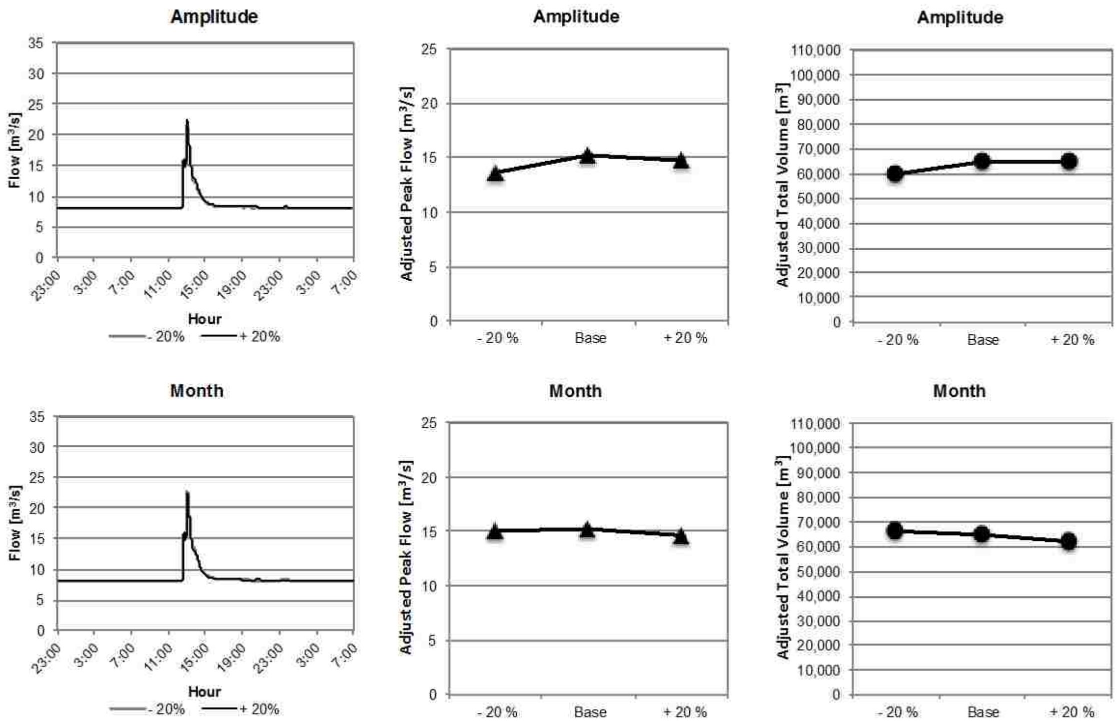


Figure 22 (continued). Sensitivity of the Model to Different Parameters

As we expected, the model displays a different sensitivity to each of the parameters. The parameter *rain* produces the highest effect, both in peak flow and in total volume. Its influence is so big, that I had to use a different vertical axis to be able to plot it. Next, we have *temp* and *soilmoist*. The parameter *soilmoist* has a greater effect on the peak flow, while the parameter *temp* has a greater effect on the total volume. We can anticipate that *soilmoist* will have more influence in rain-driven events, and temp in snowmelt-driven events. Following those, we have *snow*. In this region of the results surfaces the parameter *snow* seems to have opposite effects on the peak flow, and in the total volume. Looking at the hydrographs we can observe that the presence of snow tends to attenuate the effect of precipitation, while exacerbates the effect of temperature. The parameter *peaktime*, has a moderate effect on the peak flow, and in a lesser measure in the total volume. This probably is related to the cyclical effect of the temperature throughout the day. Both *amplitude* and *month* appear to have minimum effect on the results. Nevertheless, experience shows that both parameters become more significant with higher levels of snow.

Finally, a word should be said about *baseflow*. The current sensitivity analysis adjusted the results by subtracting the values of the baseflow, so what we are observing in the *baseflow* graphs is "the effect of modifying *baseflow* on the peak flow and total volume, after subtracting the baseflow". The effect is minimum. Nevertheless, for the purposes of the system, the value of *baseflow* can be quite significant: having an event flow of 20 m^3/s on top of a 70 m^3/s baseflow will definitely produce floods, but having it on top of a baseflow of 30 m^3/s will not. In consequence, the effect of the parameter *baseflow* can be considered more significant than what the graphs show.

Table 5. Sensitivity Ranking

Parameter	Sensitivity Rank
rain	1
temp	2
baseflow	3
soilmoist	4
snow	5
peaktime	6
amplitude	7
month	8

In summary, we will synthesize our findings of this analysis in a "Sensitivity Rank". We must acknowledge that this rank is somewhat arbitrary, mainly because is based on the results obtained for a very specific and particular region of the results space. Nevertheless, it will be useful later to determine the number of quantiles to use for each parameter in the definition of the scenarios.

3.4 Real-time and Forecast Data

The second aspect to consider in order to determine the number of quantiles for each parameter is the availability and accuracy of the real-time observed and forecasted data. In other words, there should be a balance between the number of scenarios, which implies a certain precision, and the quality of the target current conditions, which would have a certain accuracy. These questions were kept in mind during the process of selection of parameters. In Table 6 I summarize how the parameters are treated ("observed" or "forecasted"), and what source is used to retrieve their current values.

Table 6. Parameters Types and Sources

Parameter	Type	Source
snow	observed	SNOTEL
soilmoist	observed	SNOTEL
baseflow	observed	USGS
rain	forecasted	7Timer!
peaktime	forecasted	7Timer!
temp	forecasted	7Timer!
amplitude	forecasted	7Timer!
month	observed	System

3.4.1 SNOTEL

As it was indicated in the Table 6, the observed values of *snow* and *soilmoist* are retrieved from a SNOTEL station (Schaefer & Paetzold, 2001). This is the station 1232, located at:

Latitude: 40.183333

Longitude: -111.366666

Elevation: 2257 m

The data is retrieved hourly using the Web API of the website, after adjusting the dates in the link:

http://www.wcc.nrcs.usda.gov/reportGenerator/view_csv/customSingleStationReport,metric/hourly/1223:UT:SNTL%7Cid=%22%22%7Cname/2014-01-01,2014-08-28/WTEQ::value,SMS:-2:value

This query returns the following ASCII report (only the first rows are shown):

```
# Hobble Creek (1223)
# Utah  SNOTEL Site - 7399 ft
# Hourly Data for 2014-01-01 00:00 to 2014-08-28 23:00
#
# (As of: Thu May 21 12:26:02 PDT 2015)
#  **Provisional data, subject to revision**
# Date,Snow Water Equivalent (mm),Soil Moisture Percent -2in (pct)
2014-01-01 00:00,94,15.1
2014-01-01 01:00,97,15.2
2014-01-01 02:00,94,15.3
2014-01-01 03:00,94,15.0
2014-01-01 04:00,94,15.3
2014-01-01 05:00,94,15.2
2014-01-01 06:00,94,15.0
2014-01-01 07:00,94,15.1
2014-01-01 08:00,94,15.0
2014-01-01 09:00,94,15.1
2014-01-01 10:00,94,14.8
2014-01-01 11:00,97,15.1
...
```

Figure 23. SNOTEL Data

To create a snow water equivalent map suited for GSSHA, the system calculates the value of SWE for each cell adjusting the retrieved value by altitude. This approach has several limitations. For example, when the station reports a SWE value of 0 mm, the system has no way to determine the presence of snow at higher altitudes in the watershed. Also, some other important factors, such as slope, aspect, topographic shadow, terrain concavity, and so on, are not taken into account in the creation of the map. These problems could be addressed by the use of an array of several stations in the vicinity of the watershed.

The station measures the soil moisture at three different depths: 2 inches, 8 inches, and 20 inches. They are reported as "water volume fraction" in %. Given that the main function of this parameter in the model is to affect the surface runoff, I decided to use the most shallow of the three values: 2 inches. The greatest limitation of the model, regarding soil moisture, is that the value obtained from the station, up in the mountains, is assumed to be uniform over the whole watershed in the model. This is obviously not the case in the real world. With the arrival of data from new sources, like NASA's new SMAP satellite (Chen, Crow, Starks, & Moriasi, 2011), or the ones resulting from the efforts of organizations such as the International Soil Moisture Network (Dorigo et al., 2013), spatially-distributed data in real-time will become available, and further research can be performed on temporal-spatial patterns, and how to synthesize them in the generation of scenarios. In the meanwhile, we have to deal with one high-influencing, low-accuracy parameter.

3.4.2 USGS

The value of the parameter *baseflow* is retrieved from the USGS station "10153100 Hobble Creek at 1650 West at Springville, Utah", located at:

Latitude: 40.178738

Longitude: -111.639207

The data is accessed through the Web API with the following link, after modifying the begin date (one day before) and the end date (the current day):

http://nwis.waterdata.usgs.gov/nwis/uv?cb_00060=on&format=rdb&site_no=10153100&period=&begin_date=2014-01-01&end_date=2014-01-02

This query returns the following ASCII report (only the first rows are shown):

```
# ----------------------------------- WARNING ----------------------------------------
# The data you have obtained from this automated U.S. Geological Survey database
# have not received Director's approval and as such are provisional and subject to
# revision.  The data are released on the condition that neither the USGS nor the
# United States Government may be held liable for any damages resulting from its use.
# Additional info: http://help.waterdata.usgs.gov/policies/provisional-data-statement
#
# File-format description:  http://help.waterdata.usgs.gov/faq/about-tab-delimited-
output
# Automated-retrieval info: http://help.waterdata.usgs.gov/faq/automated-retrievals
#
# Contact:   gs-w_support_nwisweb@usgs.gov
# retrieved: 2015-05-21 16:17:46 EDT       (nadww02)
#
# Data for the following 1 site(s) are contained in this file
#    USGS 10153100 HOBBLE CREEK AT 1650 WEST AT SPRINGVILLE, UTAH
# -----------------------------------------------------------------------------------
#
# Data provided for site 10153100
#    DD parameter   Description
#    02   00060     Discharge, cubic feet per second
#
# Data-value qualification codes included in this output:
#     A  Approved for publication -- Processing and review completed.
#
agency_cd       site_no datetime        tz_cd   02_00060        02_00060_cd
5s      15s     20d     6s      14n     10s
USGS    10153100        2014-01-01 00:00        MST     26      A
USGS    10153100        2014-01-01 00:15        MST     26      A
USGS    10153100        2014-01-01 00:30        MST     27      A
USGS    10153100        2014-01-01 00:45        MST     27      A
USGS    10153100        2014-01-01 01:00        MST     27      A
USGS    10153100        2014-01-01 01:15        MST     27      A
USGS    10153100        2014-01-01 01:30        MST     27      A
USGS    10153100        2014-01-01 01:45        MST     27      A
USGS    10153100        2014-01-01 02:00        MST     26      A
USGS    10153100        2014-01-01 02:15        MST     26      A
USGS    10153100        2014-01-01 02:30        MST     26      A
USGS    10153100        2014-01-01 02:45        MST     27      A
...
```

Figure 24. USGS Data

As we can see, the site reports the flow, in cfs, every 15 minutes. The system reads all the values on the report, and defines *baseflow* as the lowest observed flow. The value is then converted from cfs to m^3/s to be used in the retrieval of the scenario.

3.4.3 7Timer!

There are two sets of forecasted data: Precipitation and Temperature. These two variables drive the two distinct mechanisms that produce flooding. The forecasted values of precipitation are used to define two parameters: *rain* and *peaktime*. The first one gives an indication of the magnitude of the precipitation, and the second one addresses its temporal distribution. On its part, the forecasted values of Temperature are also used to define two different parameters: *temp* and *amplitude*.

Several web services exist that report weather forecasts, though not many of them are free. After evaluating many of them I decided to use "7Timer!". According to their website, "7Timer!" is a web service supported by the Shanghai Astronomical Observatory of the Chinese Academy of Sciences. Its original purpose was to provide weather forecasts for astronomical purposes (Ye, 2011), but since then it has been expanded and renovated to include many weather variables. Its products are mainly derived from the NOAA/NCEP's Global Forecast System (GFS).

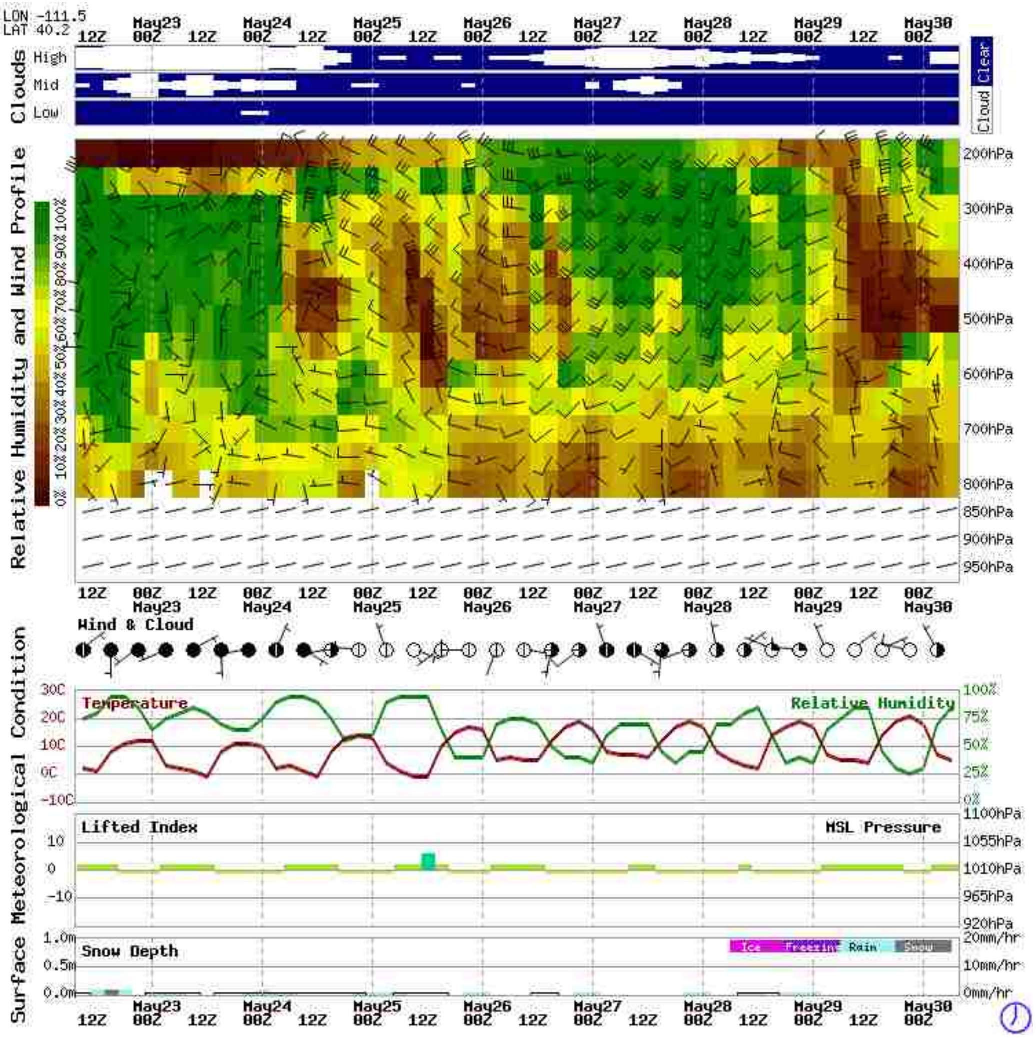


Figure 25. Example of a Report from 7Timer!

The forecast data is accessed through their Web API with the following link:

http://202.127.24.18/bin/meteo.php?lon=-111.5&lat=40.2&ac=0&unit=metric&output=json&tzshift=0

The query returns a JSON file, a sample of which looks as follows (many intermediate rows have been replaced by "..."):

```
{
  "product": "meteo",
  "init": "2015052112",
  "dataseries": [
    {
      "timepoint": 3,
      "cloudcover": 3,
      "highcloud": 3,
      "midcloud": 1,
      "lowcloud": 1,
      "rh_profile": [
        {
          "layer": "950mb",
          "rh": -9999
        },
        …
        {
          "layer": "200mb",
          "rh": 3
        }
      ],
      "wind_profile": [
        {
          "layer": "950mb",
          "direction": -9999,
          "speed": -9999
        },
        …
        {
          "layer": "200mb",
          "direction": 225,
          "speed": 225
        }
      ],
      "temp2m": 13,
      "lifted_index": -1,
      "rh2m": 8,
      "msl_pressure": -9999,
      "wind10m": {
        "direction": "90",
        "speed": 2
      },
      "prec_type": "none",
      "prec_amount": 0,
      "snow_depth": 4
    },
    {
      "timepoint": 6,
      …
    }
    …
    {
      "timepoint": 192,
      …
    }
  ]
}
```

Figure 26. 7Timer! Data

As we can see, the web service reports the forecasted values of a large set of parameters, every 3 hours throughout 8 days. For the whole series of data the system only uses the data of the following day (8 points). In particular we are interested in the parameters "temp2m" (temperature at 2 meters) and "prec_amount" (only when "prec_type" is "rain"). It is interesting to note that they provide a whole gamut of other weather data that could be useful in other models.

With the precipitation data the system calculates the parameter *rain* as the total amount precipitation during the day, in mm. To calculate the *peaktime*, the system calculates the baricenter of the eight 3-hour precipitation periods in the report, and it rounds it to the closest hour. In the generation of the scenarios, the precipitation event is synthesized as a 6-hr Type II event, with a magnitude of *rain*, and its peak at *peaktime*. This approach has the limitation of concentrating all the forecasted precipitation of the day into a short period of time, which could lead to overestimating the precipitation, and producing potential "false positives". If deemed necessary, some more complex methods could be used to better address the temporal distribution of the rain. In the Chapter 6 I will briefly present one of those methods.

Regarding the temperature data, the parameter *temp* is calculated as the average temperature of the 8 periods. The parameter *amplitude* is the difference between the values of the maximum and the minimum of those values. When the scenarios are generated, the values of *temp* and *amplitude* are used to generate a sinusoidal variation of the temperature throughout the day, with the maximum (*temp* + 1/2 *amplitude*) located at 3:00 pm, and the minimum (*temp* - 1/2 *amplitude*) at 3:00 am.

3.4.4 System

The values of *month* and *time* are retrieved directly from the computer where the system is running. The person in charge of the implementation should be particularly careful to check the time-zones in which the web sources report their values, as well as any time-zone shift between the location of the computer and the location of the watershed, and perform all the necessary conversions and adjustments to ensure the consistency of the data.

3.4.5 Accuracy Ranking

As we did with the sensitivity, we will attempt to create a rank for the accuracy of the data. This is not a simple task, because several aspects have to be considered. For instance, in observed data, we can assess the accuracy of the instruments used to measure the parameters. Nevertheless, that accuracy can be easily dissipated by the adjustments we need to implement in order to use that data. For example, the measurement of the soil moisture at the point where the SNOTEL station is located is made quite accurately with a "Stevens Hydra Probe" (with a reported instrument accuracy of ± 0.01 WFV for most soils), but later applied uniformly to the rest of the watershed, as far as 25 km apart from that point.

Forecasted data suffers from an analogous problem. Though it is possible to evaluate the accuracy of a forecast, that accuracy will be largely affected by the way the forecasted value is used in the model.

Table 7. Accuracy Ranking

Parameter	Accuracy Rank
month	1
baseflow	2
temp	3
snow	4
rain	5
soilmoist	6
amplitude	7
peaktime	8

Though further research in this regard is necessary, and encouraged, for our current purpose we will subjectively assign an accuracy rank to each of the parameters, based mainly on our analysis in the previous points, keeping in mind that this assessment of the accuracy is just an intermediate step to determine the number of quantiles that each parameter "deserves".

3.5 Definition of the Quantiles

As we explained before, the number of quantiles for each parameter – its “granularity” – will depend on: (1) the precision and accuracy of that data, (2) the sensitivity of the model to that parameter (which could be non-linear), and (3) the total number of scenarios that are feasible to generate with the resources available. In our present implementation we will determine the number of quantiles for each parameter by combining the Sensitivity Rank and the Accuracy Rank into a Quantiles Rank. We will do so by multiplying the values of both ranks, as shown in the Table 8:

Table 8. Definition of the Quantiles for Each Parameter

Parameter	Sensitivity Rank	Accuracy Rank	S x A	Quantiles Rank	Number of quantiles
rain	1	5	5	1	6
baseflow	3	2	6	2	5
temp	2	3	6	2	5
month	8	1	8	4	4
snow	5	4	20	5	4
soilmoist	4	6	24	6	3
peaktime	6	8	48	7	2 (4)*
amplitude	7	7	49	8	2
				Number of scenarios	**57600**

Now, using the Quantiles Rank we can assign the number of quantiles for each parameter. This numbers will depend on the total number of scenarios that are feasible to pre-compute. Nevertheless, as we observed in Chapter 2, the number of quantiles can be extended a posteriori, and new scenarios can be pre-calculated as new resources become available.

In this present case, my goal was to pre-compute about 50,000 scenarios. A first assignment resulted on a number of 28,800 scenarios, and so I decided to increase the number of scenarios for *peaktime*, given that it describes an important aspect of the flood forecast, not completely represented by the sensitivity or the accuracy analysis. The selected number of quantiles resulted in a total of a 57,600 scenarios

3.6 Defining, Shuffling, and Running the Scenarios

To generate the scenarios for this model I followed the same methodology explained in the first chapter, i.e., a Latin Hypercube approach (Park, 1994), and defined all possible parameter combinations, each of which resulted in a different scenario. In this case, however, I intended to analyze the method of implementing the system with a limited set of scenarios, and then increase the accuracy of the system by sequentially adding new scenarios. The results of such analysis will be presented in the next chapter. One problem with this approach is that the algorithm I used to generate the combinations works one parameter at the time, and that would populate the 8-dimensional scenarios-space unevenly (i.e., the first scenarios in the set would all have low month values). Therefore, it became necessary to randomly reassign the order of the defined scenarios, or "shuffle" them. This shuffling is the simplest way to guarantee uniformity in the population, while sequentially increasing the density of scenarios.

The computation of one scenario could take anything between half an hour and twelve hours, depending on the particular sub-processes involved. This made the computation of all the scenarios infeasible on a single computer. Therefore, we used HTCondor (Thain, Tannenbaum, & Livny, 2005) via CondorPy to schedule runs both in a local cloud of about 200 cores part-time, and in an Amazon cloud of 600 cores full-time. The complete set was computed, with some interruptions, in about two weeks.

The amount of information generated by each run posed another problem. Saving the depth at each cell every 10 minutes ended up generating a file more than 1Gb in size. Notwithstanding, most of that data is useless for the system. The way I addressed this problem was by post-processing the file and using the information it contained to generate three maps for each simulation. The first map contains the maximum depth at each cell during the simulation (Figure 27).

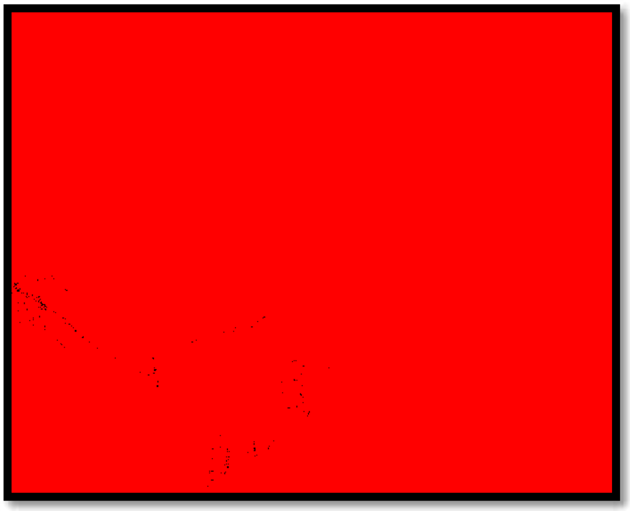


Figure 27. Map Showing the Maximum Depths in the Scenario 8461

The second map contains the time-step at which that maximum depth was attained (Figure 28).

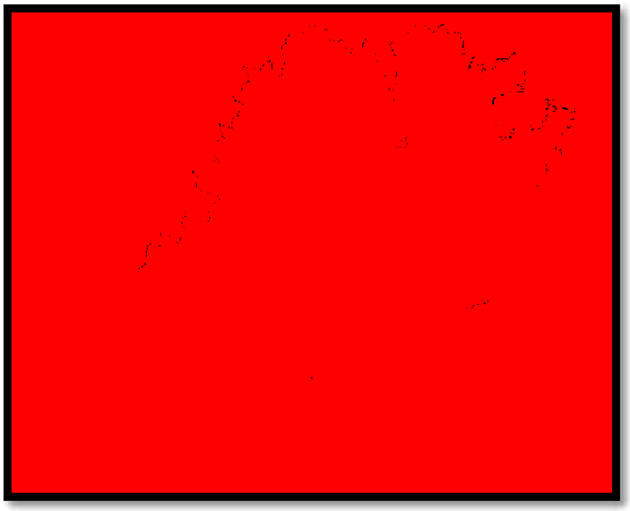


Figure 28. Map Showing the Time-step for the Maximum Depths in Scenario 8461

These two maps can provide an idea of the overall behavior of the flood, but sometimes in flood situations to know when the maximum depth is attained is not as important as to know when the depth exceeds a certain threshold. Therefore, the third map records the time step in which the depth exceeds a threshold of 30 cm (Figure 29).

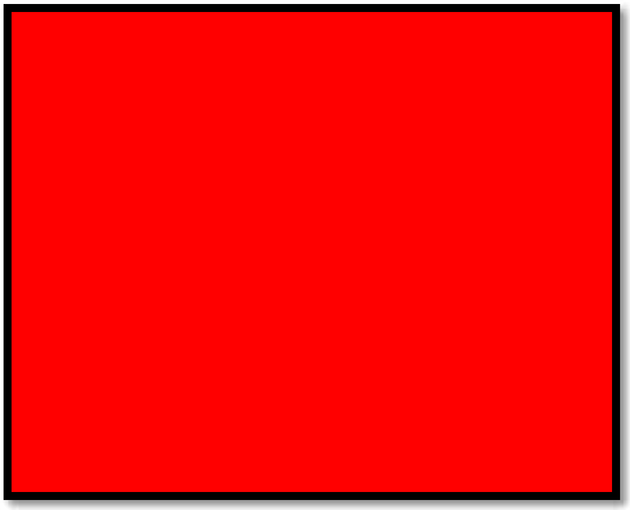


Figure 29. Map Showing the Time-step for the Threshold Depths in the Scenario 8461

With the data from these three maps, a rough but useful limnograph could be reconstructed for each cell, without the necessity of storing all the depth data.

3.7 Implementation of the System

The system was finally put together on 4th October 2014. Since then, it has been retrieving data from the web services and selecting the "most likely" scenario for the following day. In Figure 30 we can observe a summary of these matches.

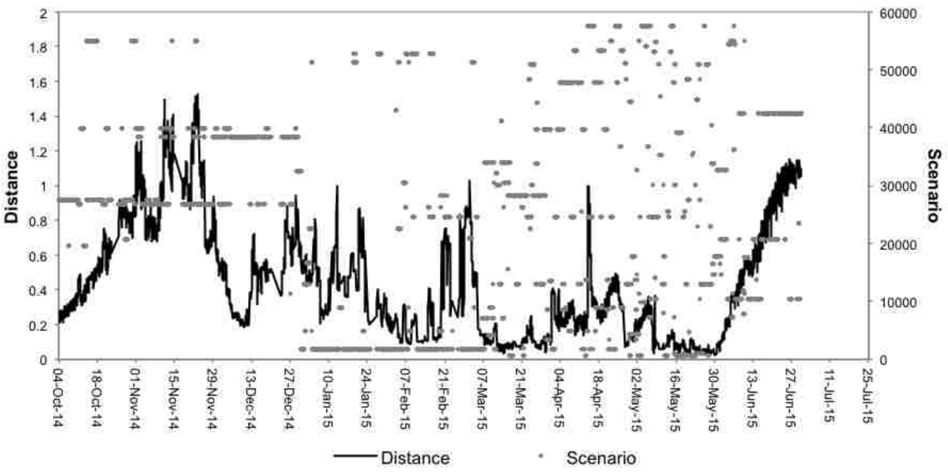


Figure 30. Scenarios Selected and Distances

The black line shows the distance between the current conditions and the selected closest scenario. As we can see, during the fall that distance was high, then it decreases a little during the winter, and finally at spring it goes even lower. The gray dots represent the number of the scenario selected. We can observe that during the fall and the winter only a few scenarios were selected repeatedly. The situation changes at spring, when we can see the appearance of many different scenarios. Finally, at the summer the situation changes again, and few scenarios are selected. This has to do with the way the ranges of parameters were selected. As we explained before, the goal is to prioritize "flooding" scenarios, therefore, the ranges were trimmed at values that wouldn't likely produce floods. During the fall, the winter, and the summer, the observed parameters are more likely to fall outside of the ranges, and consequently the closest scenarios found will be farther than in spring.

In Figure 31 we can see the observed hydrographs and the ones predicted by the system. We can see that during the fall, the winter, and the summer months the predicted hydrographs are consistently greater than the observed ones. This happens because there are no pre-computed scenarios for such low flows. By mid-spring the situation changed, and the system started to work more like intended: the distances are smaller, implying that the retrieved scenarios most closely represent the current conditions.

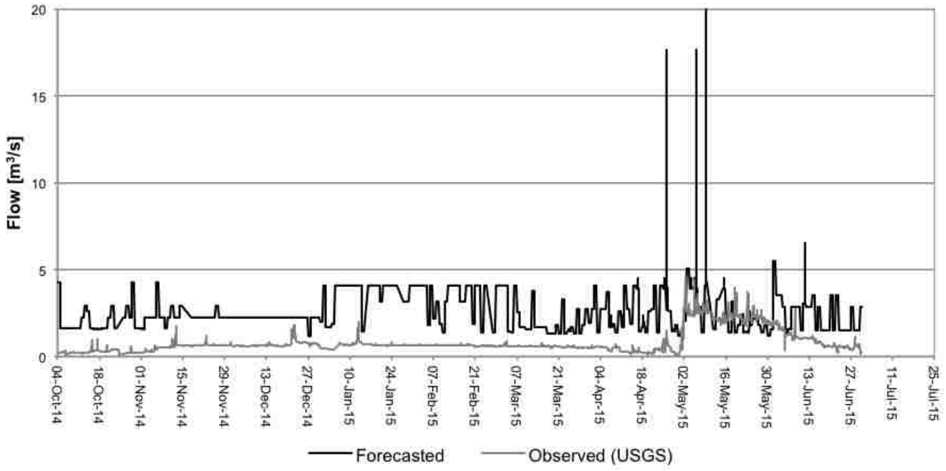
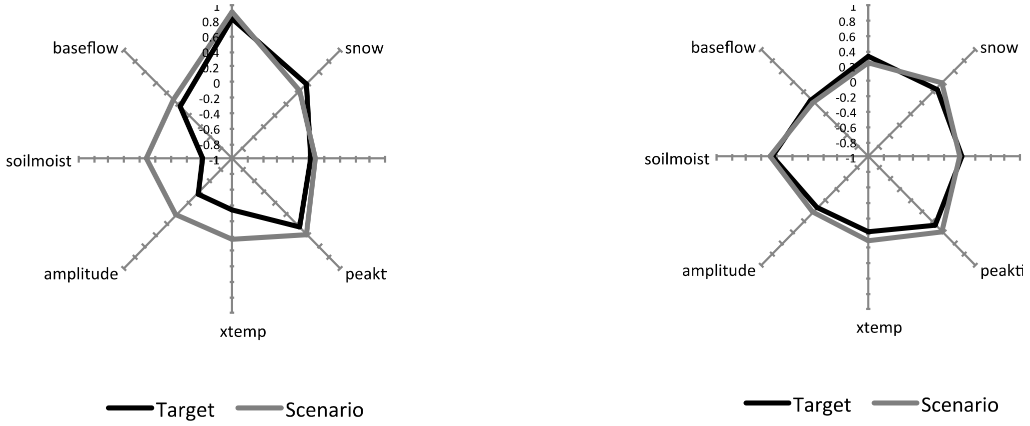


Figure 31. Comparison Between the Observed and the Forecasted Flows



11/29/2014

5/16//2015

Figure 32. Closest Matches for Nov-2014 and May-2015

Finally, in Figure 32, we can observe two typical matches, one in the fall and one in the spring. We can see that during the spring, the matches are tighter than during the fall. Note, however that, in order to be able to display some of the observed values that fell outside of the ranges defined for the parameters in the generation of scenarios, the scale in the graphs goes from -1 to 1.

3.8 Discussion of the Results

The implementation of this system proved successful relative to its goals. It demonstrated that it is possible to create a model, define a set of scenarios, pre-compute them, build a program to automatically retrieve data from web services, and based on those data predict the most likely scenario. Once the system was set in motion, it required no human intervention whatsoever.

Unfortunately, the winter during which the system was operational was a particular dry one. In the Figure 33 we can see the evolution of the precipitation and the snowpack in the Hobble Creek and the nearby Snowbird SNOTEL stations. In the graph from the Snowbird station (top) we can see that both the precipitation and the snow accumulated during the winter were significantly below their historic average. In the graph from the Hobble Creek station (bottom) we can observe that by the end of March the snowpack had melted, so there are no expectations in observing a flood event in the rest of the year.

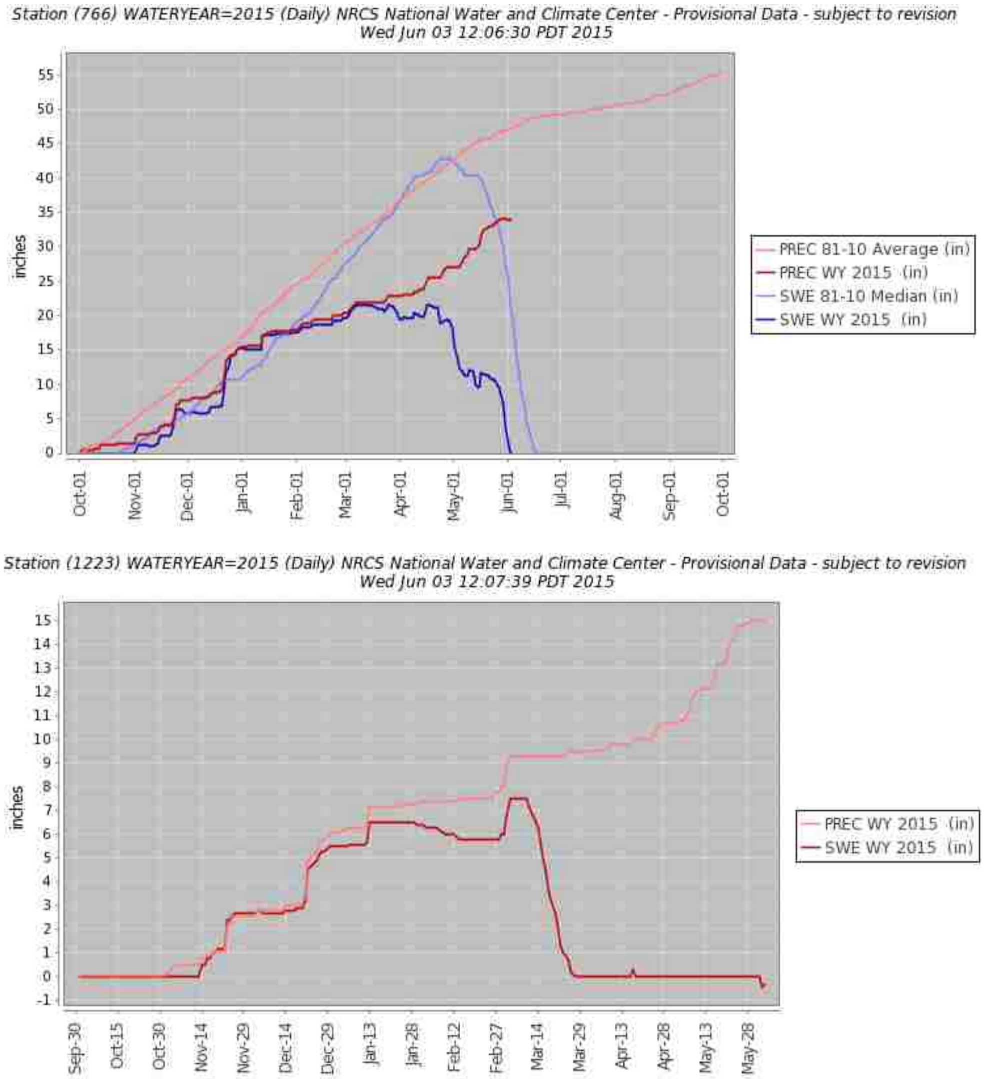


Figure 33. Precipitation and Snow Water Equivalent for Snowbird (Top) and Hobble Creek (Bottom) SNOTEL Stations.

Furthermore, several intrinsic factors limit the usefulness of this particular system: first, the model was not thoroughly calibrated; second, some of the parameters could be better defined (for example, *month* didn't produce the expected effects on the model, and *soilmoist* could have been spatially distributed to better represent the reality); third, the watershed is not unaffected as assumed in the model. As we can see in Figure 34, some control structures exist in the river that affect its the natural flow. Our assumption in building the system is that these structures wouldn't have much effect in a large flooding effect, but they certainly affect the results for the low flows observed during this test.

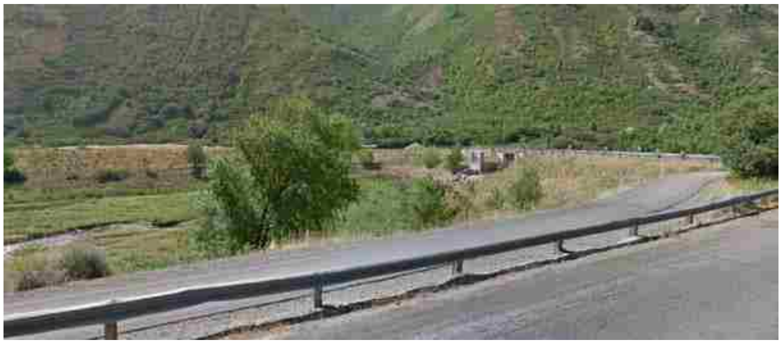


Figure 34. Control Structure in Hobble Creek

The limitations related to the calibration and the use of the parameters can be addressed by the use of "post-calibration". This consists in generating an intermediate function for each parameter to generate a "transformed" current condition that would be used to better select the appropriate scenario. For example, if after several months it is concluded that the observed soil moisture at the SNOTEL station overestimates the average soil moisture at the watershed, a transform function for that value could be implemented to decrease that value before used to retrieve the pre-computed scenario. Another method to overcome these limitations will be presented in a following chapter.

4 ASSESSING THE RELATIONSHIP BETWEEN THE NUMBER OF SCENARIOS AND THE ACCURACY OF THE SYSTEM

The method I am proposing is based on the intuitive assumption that similar conditions produce similar results. This assumption, however, should be tested. A corollary of that assumption is that the larger the number of scenarios, the higher the probability of finding a simulation in the archive that is a close match to the current and forecasted parameters, and therefore, the greater the expected accuracy of the system. The question, however, is how many pre-computed scenarios are necessary to attain a certain level of accuracy? Or the inverse, what accuracy can be expected given a certain number of scenarios?

Determining the similarity between two scenarios is not difficult. As we explained before, a scenario is defined by a unique combination of values of *n* input parameters, and we could consider each scenario as a point in an *n*-dimensional space. A set of current conditions also consists of *n* values, so it can also be considered a point in the same space. Therefore, the similarity between current conditions and a specific scenario can be likened to the Euclidian distance between the points they define. Therefore, we will refer to the measure of similarity as “distance”.

On the output side of the model, the problem is not that simple. The output of the model is not a unique number, but rather a long series of them, in the form of a hydrograph, and eventually a temporally- and spatially-distributed flood map. Comparing two hydrographs or two flood maps and assessing their similarity is not a trivial task; there are different aspects of the hydrographs or the flood maps we would be interested in, and each to a different degree (Green & Stephenson, 1986). In this chapter we will focus on the hydrographs and therefore we identify three characteristics that can be easily quantified: (1) the peak flow, (2) the time to peak, and (3) the total volume. All three characteristics are important in flood forecasting, but not to the same degree. In our model, the most important characteristic is the peak flow, since it determines if a certain flood would go overbanks and what area it would affect, thus giving an indication of the overall severity of the event. Of secondary importance would be the time to peak, since that information would help plan preventive actions. Finally, the least important of the three characteristics is the total volume of the flood, which would indicate the time span that the affected areas would remain flooded, and other potential effects on downstream reservoirs or water bodies. Based on these considerations, for this analysis I assigned a different weight to each of these values: 0.7 for the peak flow, 0.2 for the time to peak, and 0.1 for the total volume. In summary, I quantified the difference between a target hydrograph and a scenario hydrograph as a "compound error", which is calculated as follows:

$$CE = 0.7 * |pf_t - pf_s|/pf_t + 0.2 * |pt_t - pt_s|/pf_t + 0.1 * |tv_t - tv_s|/tv_t \qquad \textbf{(4)}$$

Where:

CE = Compound Error

pf_t = peak flow of the target

pf_s = peak flow of the scenario

pt_t = time to peak of the target

pt_s = time to peak of the scenario

tv_t = total volume of the target

tv_s = total volume of the scenario

4.1 Method

The primary objective of this exercise is to assess the accuracy of a flood forecast relative to the number of scenarios it contains. Theoretically, our system could work with only one scenario, though that system would not be useful, since for any combination of current parameters it would always return the same result. Nevertheless, as the number of pre-computed scenarios increase, the probability of finding a better match, and therefore a more accurate prediction, increases.

To illustrate this process we will define a set of current conditions (target) and see which is the best match as we add new scenarios into the system. The radar plot of the target can be seen in Figure 35.

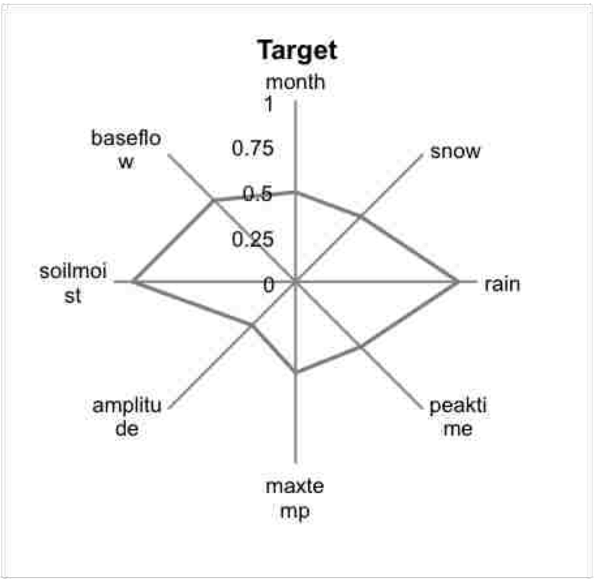


Figure 35. Target Scenario

We will start with a set of only one scenario. When there is only one scenario —scenario 0— that scenario will obviously be the closest one (at a distance of 1.109). In the radar plot at the left of Figure 36 we can see that the scenario doesn't match well any of the target parameters: the scenario has a little rain, while our target has an intense one, etc. On the right side we can see a comparison of the resulting hydrographs. It goes without saying that the accuracy of the system at this point is not acceptable.

Now we will start adding new scenarios to the set. As these new scenarios become available, eventually one of them will be closer to our target than 1.109. In our example, that happens with scenario 2, which is at a distance of 0.968. Then, scenario 3 (distance = 0.952) and scenario 4 (0.594) are even closer. From scenario 5 to scenario 7 the distances are larger than 0.594, but scenario 8 is smaller with a distance of 0.594. No closer scenarios appear until scenario 34 (0.459) is added, followed by scenario 246 (0.458), scenario 250 (0.447), scenario

256 (0.348), and scenario 849 (0.346). In Figure 37 we can see how each of the parameters are closer, and how the hydrographs become more similar.

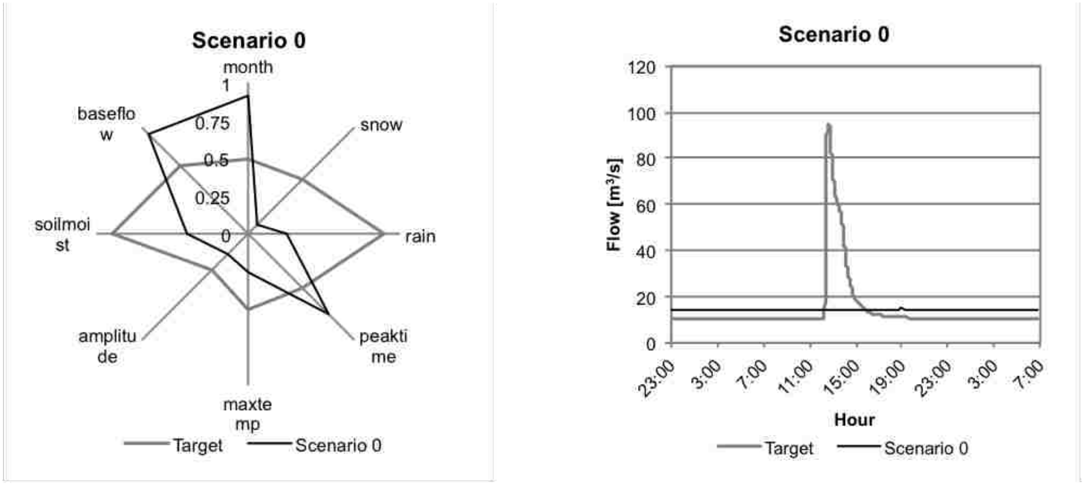


Figure 36. Trivial Closest Match to the Target Scenario When Only One Scenario is Available

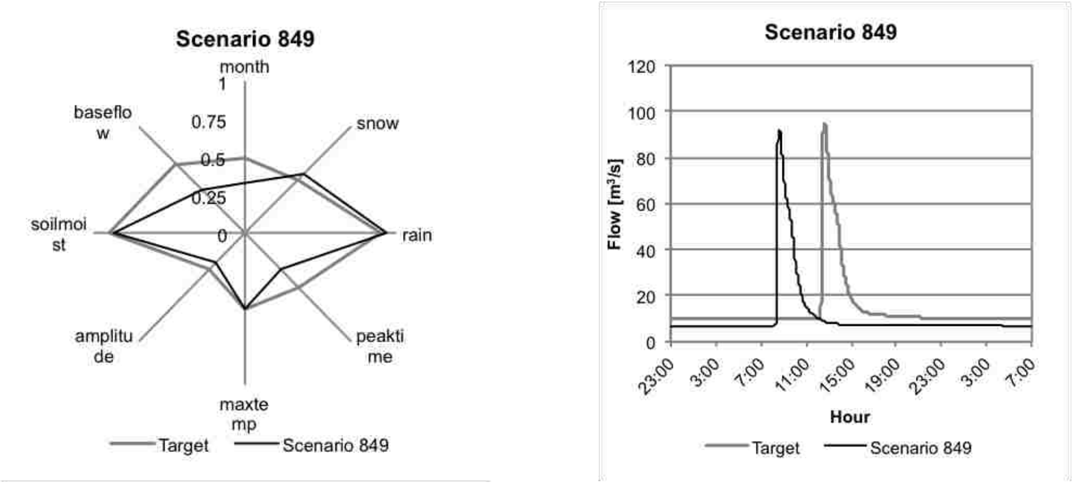


Figure 37. Closest Match to the Target Scenario When 850 Scenarios are Available

All the scenarios from 850 to 6,760 are farther than 0.346, but scenario 6,761 replaces the current smallest distance with a value with a distance of 0.341. This scenario is successively replaced by scenarios 7,513 (0.337), 8,461 (0.264), 31,592 (0.217), and 34,756 (0.132). All scenarios from 34,757 to 57,599, the last one, are no closer than .132.

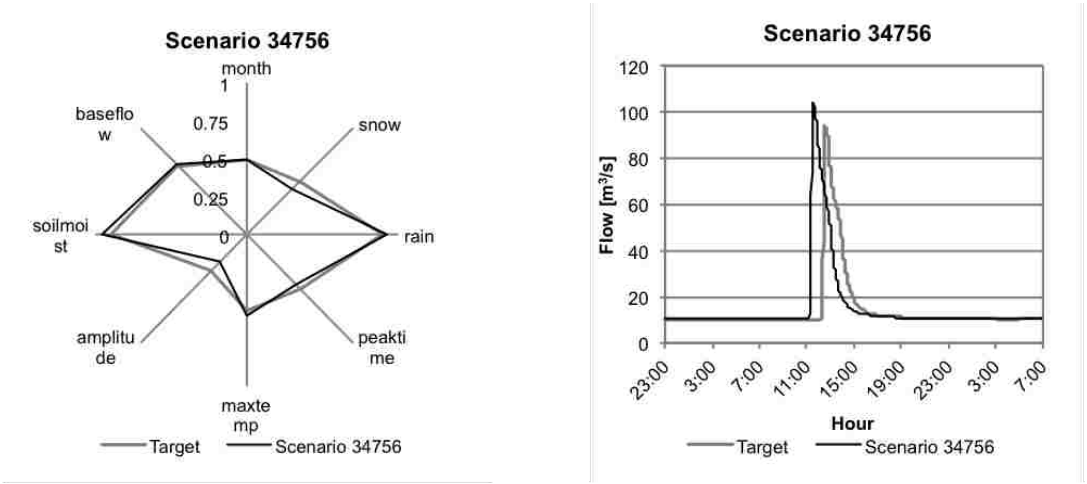


Figure 38. Closest Match to the Target Scenario When 34,757 Scenarios are Available

In Figure 38 we can observe the best match our system can provide us, when all the computed scenarios were included in the set. In Figure 39 we can observe the similitude in the flood maps. It is apparent that the flooded areas are similar enough to use the selected pre-computed one (on the right) as an acceptable surrogate of the targeted result (on the left).

The scenarios that improve the fit are plotted In Figure 40 as distance vs. number of scenarios (i.e., scenario index). This plot illustrates how the similarity increases —the distance decreases— as more scenarios become available. Also, it appears that in the early stages, while there are only a few scenarios, a new smallest distance is discovered with greater frequency and the decrease in the distance or improvement in accuracy is greater. Then, at a certain number of scenarios (around 1,000 in the graph), the relationship apparently “flattens”. In other words, many more new scenarios are required to find a smaller distance and the improvement in accuracy is smaller.

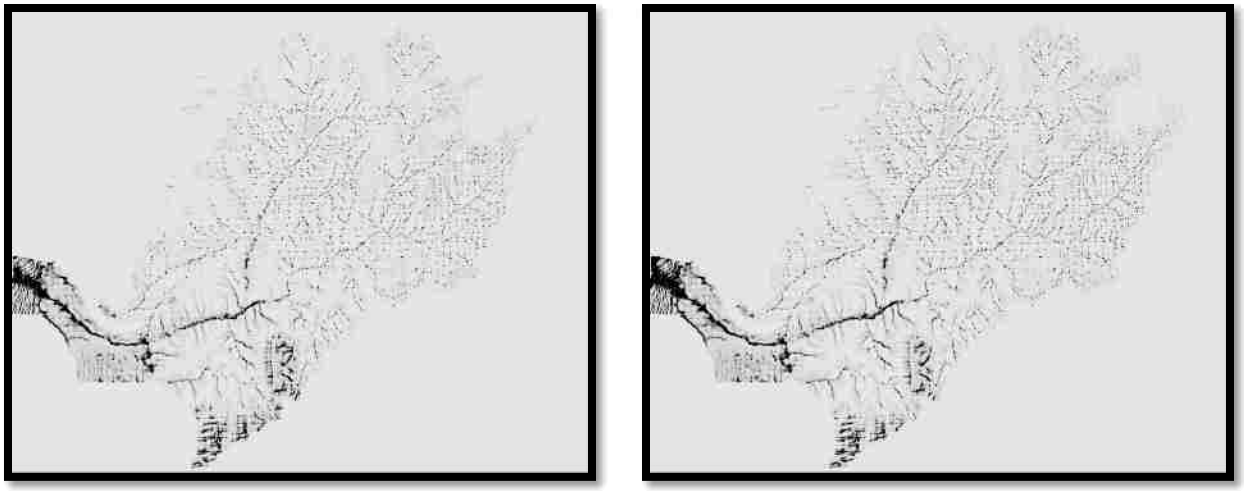


Figure 39. Comparison of the Flood Maps of the Target Scenario and the Closest Match

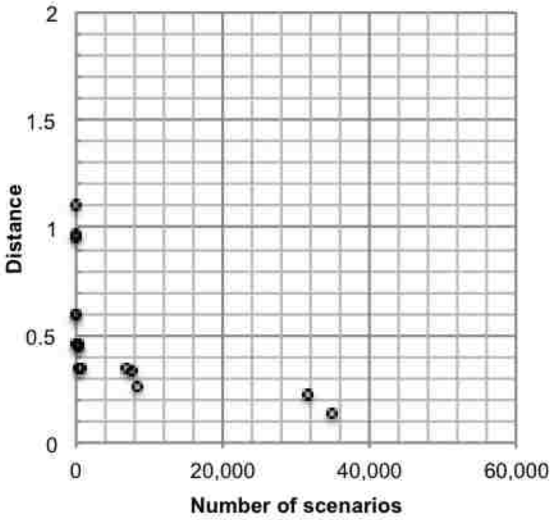


Figure 40. Distance to the Closest Scenario for Different Numbers of Scenarios Available

Notwithstanding, if we plot the log of the distance versus the log of the number of scenarios, as in Figure 41, we observe linearity between them. The reason for this is that every new scenario that becomes the "closest" to the target reduces the chances of the following scenarios to become such, and also constrain the potential reduction in distance for the sequent "closest".

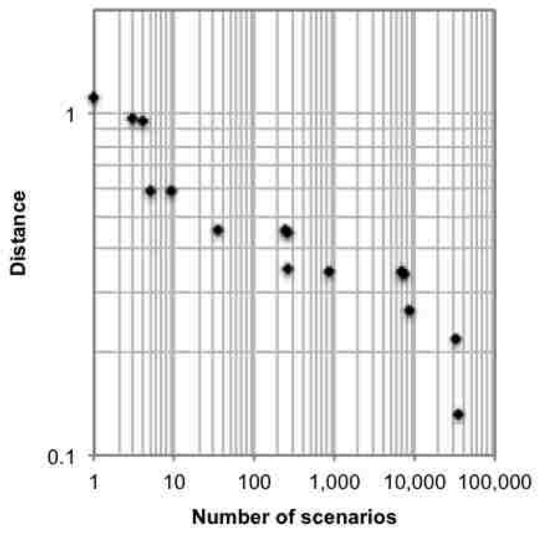


Figure 41. Distance to the Closest Scenario for Different Numbers of Scenarios Available in Log-Log

At this point, it is constructive to consider the distance we are calculating in the multi-dimensional spaces. Given that we experience reality in three dimensions, it can be difficult to understand how things work in higher dimensions. For instance, in our example we started with a distance of 1.109, and after more that 34,000 scenarios we attained a distance of barely one order of magnitude less. Was that reduction worth the effort? The answer lies in what Richard Bellman called the "curse of dimensionality" (Bellman, 1956). Succinctly, this means that the characteristics of a problem become less intuitive as we increase its number of dimensions (Aggarwal, Hinneburg, & Keim, 2001; Köppen, 2000). This effect has been widely analyzed and applied in different fields, such as optimization (Bentley, Weide, & Yao, 1980; Deb & Saxena, 2006), data mining (Verleysen & François, 2005), etc. In the next few paragraphs we will briefly analyze it from the pre-computed hydrologic model scenarios point of view.

If we would have only one parameter, the parameter space would simply be a segment going from 0 to 1. The maximum possible distance in that space would be only 1. If we had two parameters, now our space would be a square of side 1. The maximum possible distance, now, is on one of the two diagonals, and it would be $(1^2+1^2)^{0.5}$ = 1.414. For three parameters, our space would be a cube of side one, and the maximum possible distance would be one of the four diagonals, and equal to $(1^2+1^2+1^2)^{0.5}$ = 1.732. For our eight-parameter space, the maximum possible distance is one of the 49 diagonals of the 8-dimensional hyper-cube, and would measure $(1^2+1^2+1^2+1^2+1^2+1^2+1^2+1^2)^{0.5}$ = 2.828. In other words, the distance from the center of the hyper-cube to the farthest vertex is 1.414. Therefore, having an initial distance between two “random” points of 1.109, shouldn’t surprise us.

If we divide our spaces in sections by increasing the number of scenarios, a counter-intuitive phenomenon materializes. If we go back to one dimension and we divide the space (segment) in sections, we would find that the sections neighbor by pairs. In other words, two sections share one common point. In two dimensions, the division would tile the space in rectangles (if the subdivisions are not equal in each dimension), now we find that four rectangles share one vertex. In three dimensions, the subdivision would create cuboids (rectangular parallelepipeds), and, as we can see in Figure 42, eight cuboids would share one vertex. If we extend this same reasoning to our eight dimensions, we find that one vertex is shared by 256 hyper-cuboids. That means that the total distance from the vertex to the center of each of these 256 hyper-cuboids is the same. Now, if we think of the center of each cuboid as the location of a scenario, we see that there are many scenarios “close” to a certain point. Thus, a slight difference in distance can result in the selection of a different scenario.

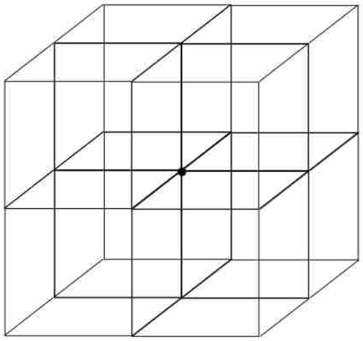


Figure 42. Eight Cuboids that Share a Point in Common in a 3-dimensional Space

In Figure 43 we can observe the graphs that illustrate this geometric property. We can see that for any number of dimensions the line seems to flatten, though in the log-log representation the relationship is linear. If we now compare the graphs in Figure 43 with the ones in Figure 40 and Figure 41, we discover that the manner in which the system finds each new closest scenario is not surprising at all.

In summary, if we augment the number of parameters in our system, thus increasing the dimensionality of the scenarios space, the system becomes increasingly sensitive due to the overpopulation of neighboring scenarios. We can also conclude, by observing the graph in Figure 43 that the number of scenarios required to find one within a certain distance increases exponentially with the number of dimensions of the scenarios space. A couple of corollaries can be derived from this: first, the log of the Euclidian distance is a better indicator of the goodness of a match, and second, the distances we obtain using different numbers of parameters are not comparable.

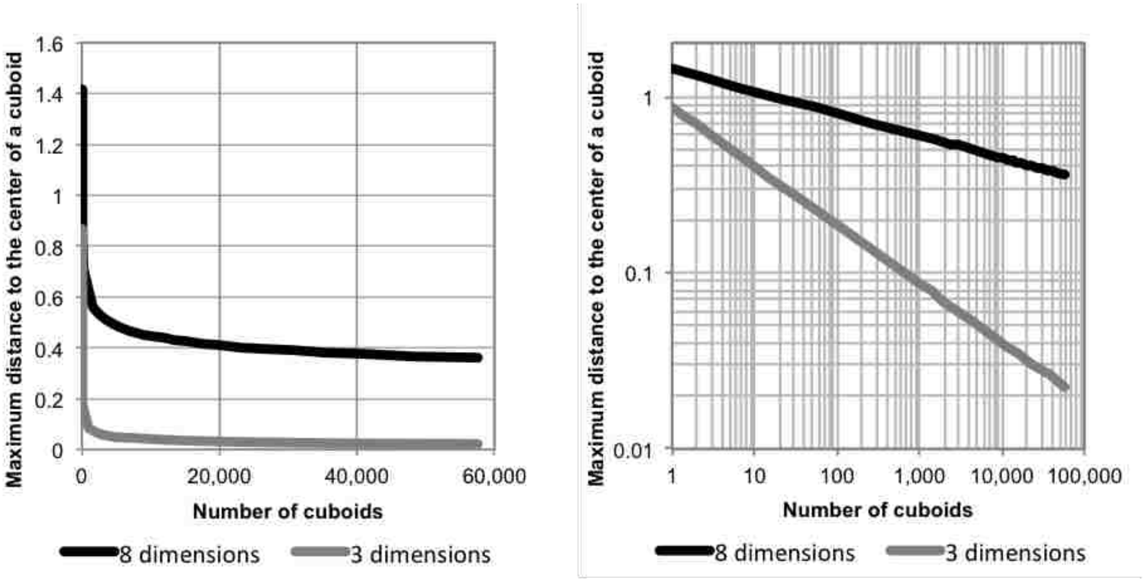


Figure 43. Maximum Geometric Distance to the Center of a Centroid for Two Different Number of Dimensions

4.1.1 Analyzing the Compound Error

As stated above, our primary objective is to determine if there is a relationship between the distance and the compound error of two scenarios. In other words, we want to verify that similar input will produce similar output. In our test example we can calculate both, because we actually did run the model for the target. Therefore, using Equation 4 we calculated the compound error between our target and each of the successive closest scenarios, summarizing the results in Figure 44. As we can see, the log-log behavior is still manifest, but not as clearly as in Figure 40 and Figure 41. We also find that some earlier matches resulted in lesser compound errors. In particular we can see that scenario 849 has a significantly lower compound error (CE = 4.2%) than the final best match, scenario 34756 (CE = 8.9%).

If we compare both scenarios (Figure 37 and Figure 38) we can see that in the former, the peak flow is almost exactly the same as the peak flow of the target (92.7m^3/s versus 95.2m^3/s), while in the latter, the difference in peak flows is larger (104.4 m^3/s versus 95.2 m^3/s), and given that we assign the greater weight to this component of the peak flow error, that similarity reduced the overall error. Notwithstanding, the result is good, but "for the wrong reasons": the rain is higher than the target, but the base flow is lower, which compensates for the difference in peak flows. We can easily identify these "lucky hits": if the compound error obtained for a number of scenarios available is greater than the compound error obtained for a lower number of scenarios, we can conclude that the lower one was a "lucky hit". For example, in the graph at the right on the Figure 44 we can observe that the first compound error was a "lucky hit", because the compound error of the next is higher. After it, we again have two "lucky hits", exposed by the fifth one, and so on. If we would remove the "lucky hits" from the set, we would have a better representation of the behavior of the system. In this example we would consider only the 7 points at the top of the graph. These could be considered a "top envelope" of the points.

Now that we have both the distances (Figure 40) and the compound errors (Figure 44), and we found that both follow log-log relation, we can combine them by plot each on different axes and see what relationship they hold (Figure 45).

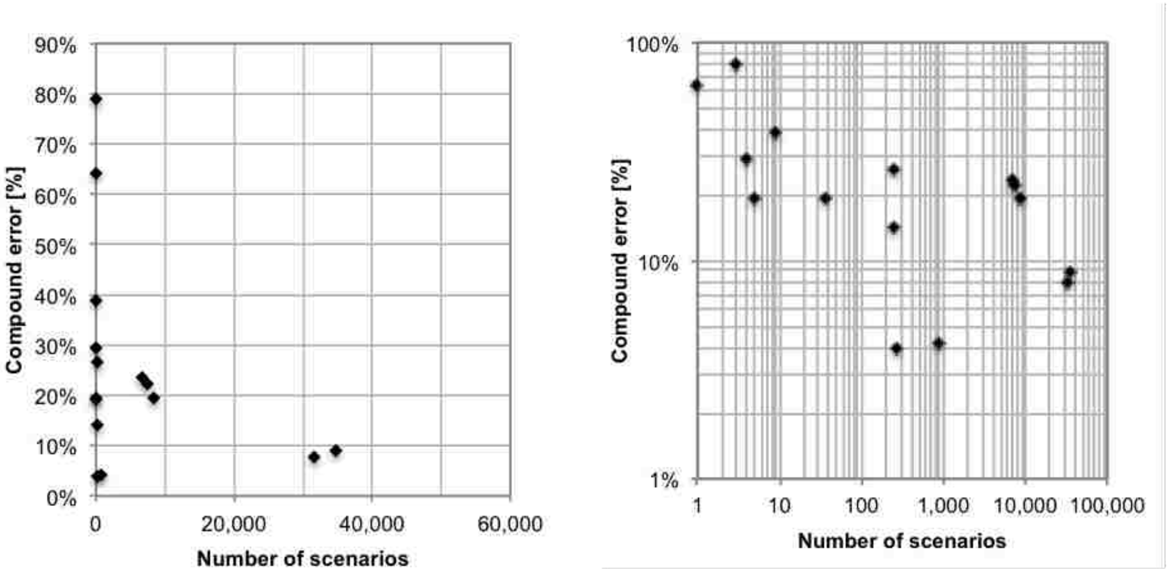


Figure 44. Compound Error with the Closest Match for Different Number of Scenarios Available

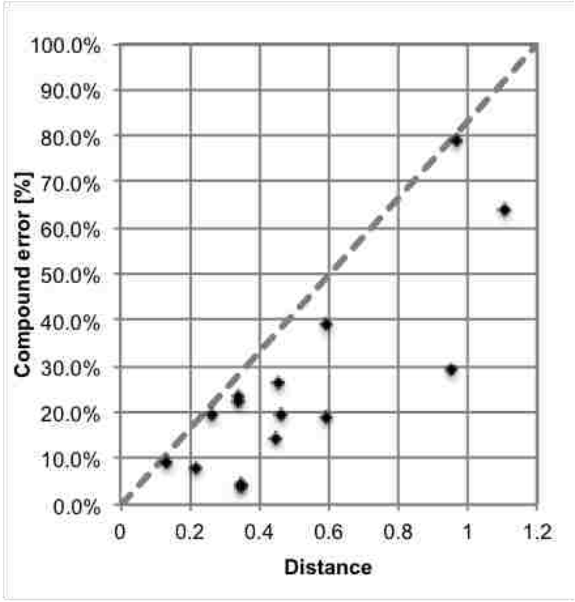


Figure 45. Relationship Between the Compound Error and the Distance Between aT and its Closest Matches. Each Point Corresponds to the Best Match for a Different Number of Scenarios Available

We observe (keeping again the "top envelope" in mind, conceptually represented here with the dashed line) that there is an almost linear relationship between the two values. We would expect compound errors of 80% when the distance is 1, and compound errors around 10% when the distance is 0.1.

Nevertheless, with only one test case it would be inappropriate to draw general conclusions. The obvious path to follow would be to repeat the same analysis for a large set of targets, and see if the same pattern we observe in the Figure 45 holds. The problem, however, is that in order to calculate the compound errors using the Equation 4 we need to run each target to obtain its resulting hydrograph. This would require the calculation of a new large set of scenarios.

To avoid the need of calculate a whole new set of scenarios, I devised a method to use scenarios already computed as "targets". Using this method I randomly pick one of the scenarios in the pre-computed set and promote it as the new "target". Since I already have a hydrograph for this "target" there is no need for a new run of the model. Obviously, since our target now is part of the set of results, while performing the search of the closest one it will eventually "find itself" at a distance of 0 with a compound error of 0%. For our analysis, we will ignore that last "best match", and keep the previous one as the result. I acknowledge that this shortcut would introduce a slight error in the result, but given that our analysis at this point intends to be more qualitative than quantitative I assume that the results will nevertheless be significant.

4.2 Results

For this next step of the analysis I randomly chose about 7,000 scenarios to be used as targets. I performed the same analysis previously outlined to each of them, and I plotted the results in the same graph.

In Figure 46 we can observe the relation between distance and number of scenarios. These graphs could be understood as the superposition of 7,000 graphs like the ones in Figure 40 and Figure 41. It is interesting to see how the top envelopes of these graphs follow the lines presented in Figure 43. The points below those top envelopes correspond to results better than expected, due to chance.

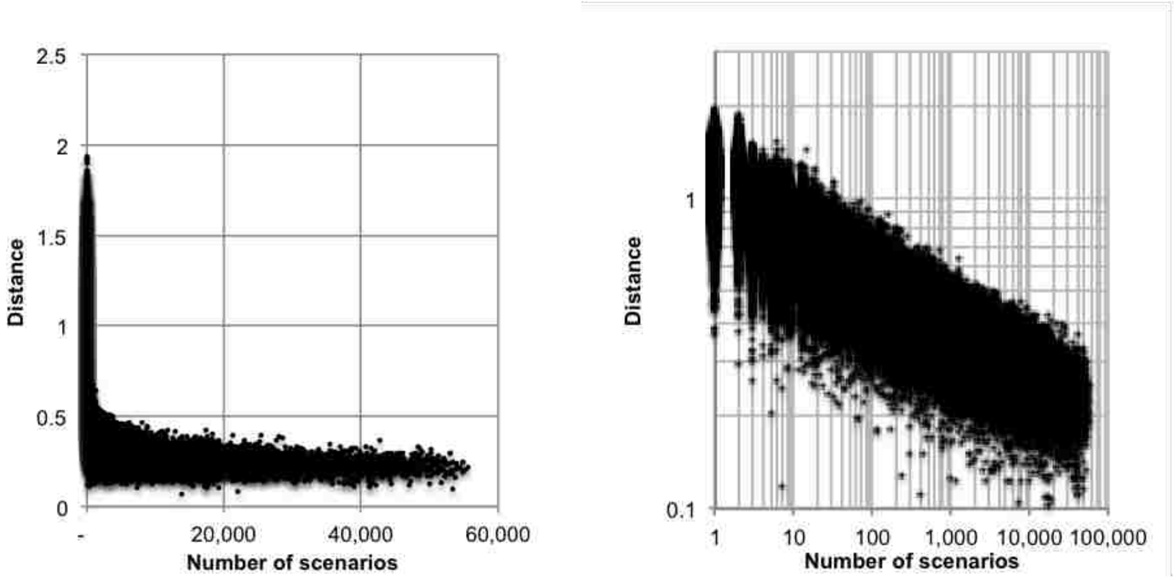


Figure 46. Superposition of the Distance vs Number of Scenarios Graphs for 7,000 Random Targets

Figure 47 shows the relationship between the compound error and the number of scenarios. They can, likewise, be interpreted as the superposition of 7,000 graphs like the ones in Figure 44. We observe that the log-log relation is still present.

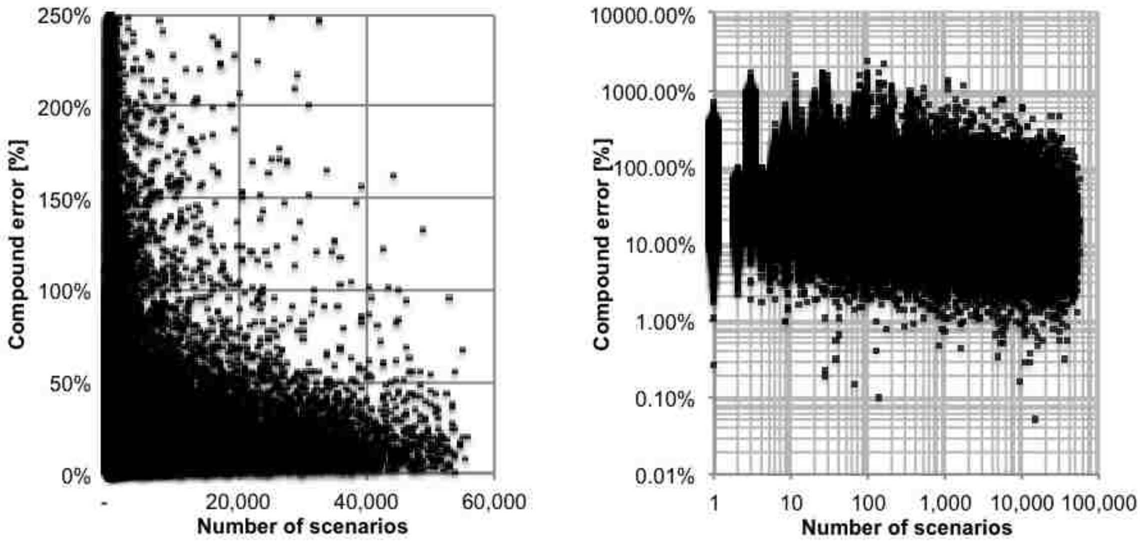


Figure 47. Superposition of the Compound Error vs Number of Scenarios Graphs for 7,000 Random Targets

Finally, we can plot the relation between distances and compound errors (Figure 48), which is analogous to the Figure 45. We observe that the top envelope still follows a linear trend, though the dispersion is greater. Why do we have such great errors? The answer is in the way we chose the targets. Since we chose them randomly, we can expect that many of them are "low peak flow" ones. The way we calculate the compound error creates more variability when the values are smaller —the way we calculate the compound error makes it more sensitive when the values are smaller.

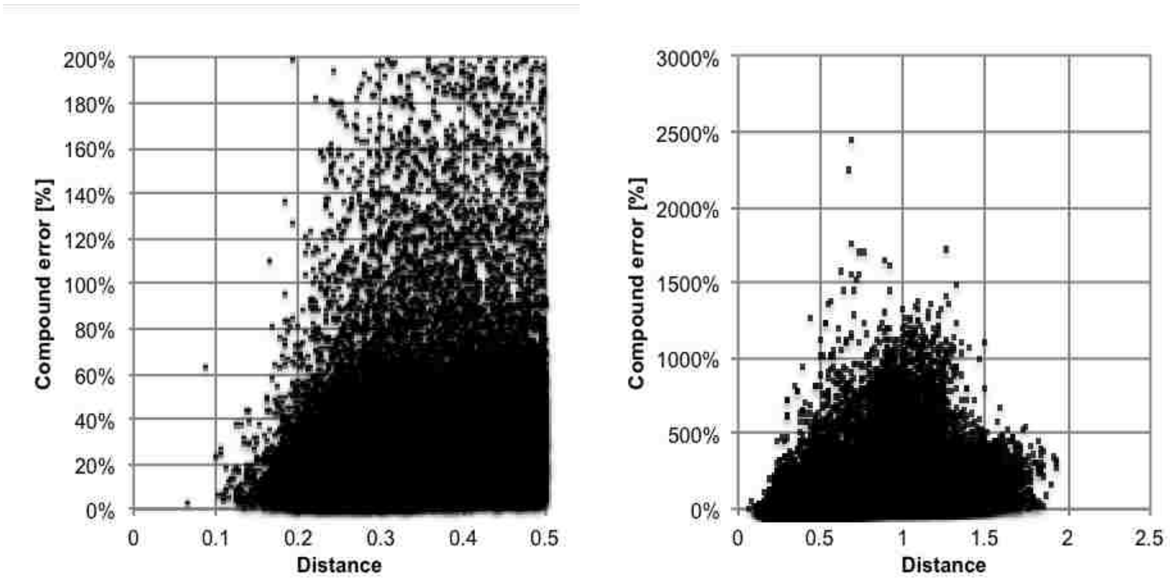


Figure 48. Close-up and General View of the Superposition of the Compound Error vs Distance Graphs for 7,000 Random Targets

4.2.1 Flood Producing Scenarios

At this point we should keep in sight the purpose of our system: to predict flooding events. The accuracy we are trying to assess should be related to the prediction of those scenarios. But what peak flows correspond to flooding events? A hydraulic analysis of the channel in the valley showed that overbank flows start at a peak flow of about 75m^3/s. So, the next logical step is to select the targets among the scenarios that produced peak flows higher than this flooding threshold.

One of the limitations of the pre-calculated scenarios approach is that we can't know in advance the outcome of a particular combination of parameters, so our best option is to generate a set of scenarios "as evenly distributed" in the parameters hyper-space as possible. This approach, obviously, leads to the generation of a large number of scenarios that produce useless results —scenarios in which no flood occurs. The advantage of our present *ex-post* analysis is that we now know the useful scenarios, and we can specifically analyze them.

In consequence, analyzing the data I found that 6,878 scenarios produced these kinds of peak flows, and can be used as targets (as a matter of fact, that number is the one that determined the number of scenarios we used in the previous analysis, so the results —especially the graphs— could be visually comparable).

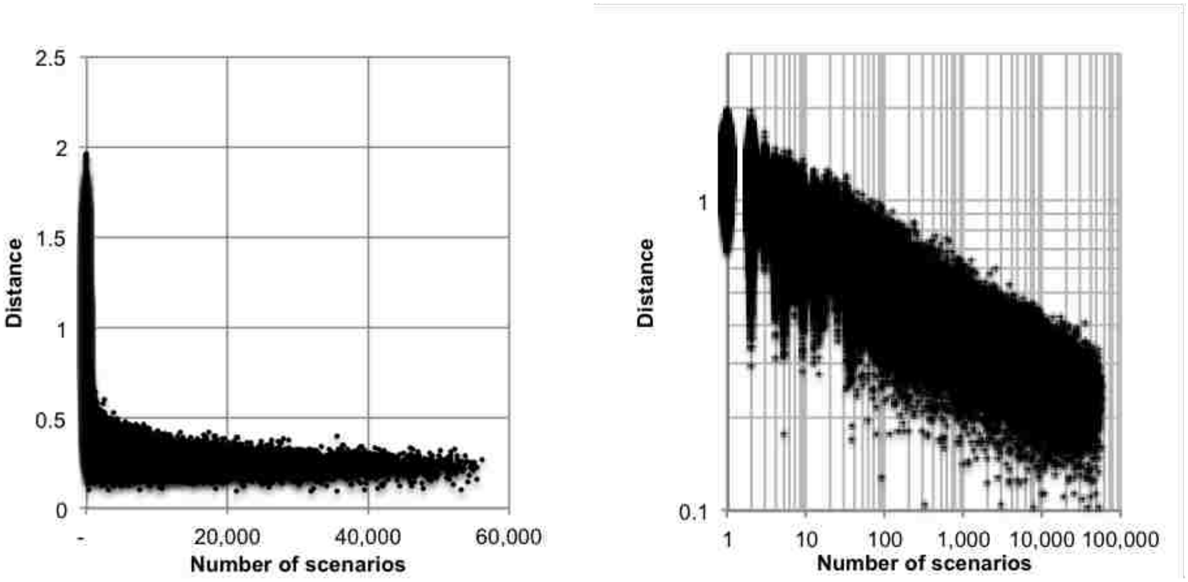


Figure 49. Superposition of the Distance vs Number of Scenarios Graphs for 6,878 Flood Producing Targets

In Figure 49 we can see the resulting graphs of distance versus number of scenarios for the 6,878 target scenarios that produced floods. If we compare them with the graphs in Figure 46 we don't see a significant difference, nor should we expect one. Nevertheless, when we graph the compound error versus the number of scenarios, as we can see in Figure 50, we find that they are more clearly defined than the corresponding graphs generated from the random selection Figure

47. This is even more noticeable when we graph the compound error versus the distances (Figure 51). Comparing these graphs with the ones in Figure 48 we see a sharper top envelope.

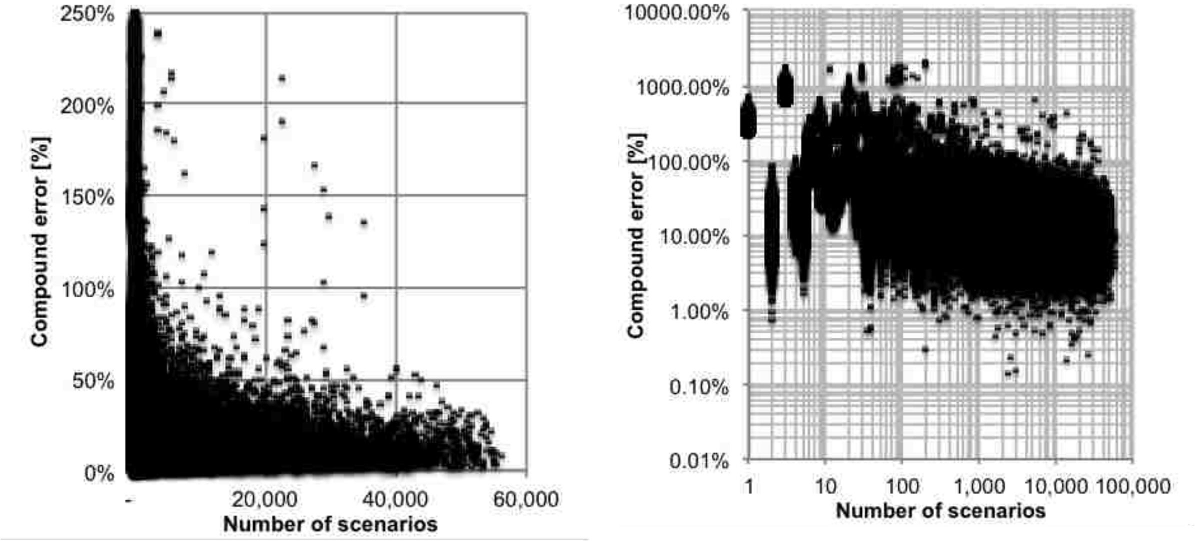


Figure 50. Superposition of the Compound Error vs Number of Scenarios Graphs for 6,878 Flood Producing Targets

It is interesting to observe some differences between the general views in Figure 48 and Figure 51. In particular, we observe a "cloud" that appears on the top in Figure 51 which is not present in Figure 48. Let's keep in mind that for this analysis our targets are not randomly chosen, but they correspond to flooding events. Therefore, we would expect their parameters to be similar to the parameters' other targets—high *rain*, high *soilmoist*, etc.

Let's also remember that the order in which the scenarios are added to the set is the same for every target under consideration. In consequence, we should expect that some early scenario that is close to one of those targets would more likely be close to the other targets. The cloud we observe shows that phenomenon. All the targets were in turn matched to a particular "good" scenario, and therefore their distances and compound errors are similar, constituting then a cloud. This phenomenon did not occur in the randomly selected targets because from the very beginning their paths where different, or in other words, no particular scenario would be "good" for all the targets. In a closer analysis we can see that in the Figure 51 there are actually several similar "clouds", which correspond to clustering around other early scenarios. These clouds, however, are merged together making them difficult to identify in the graph.

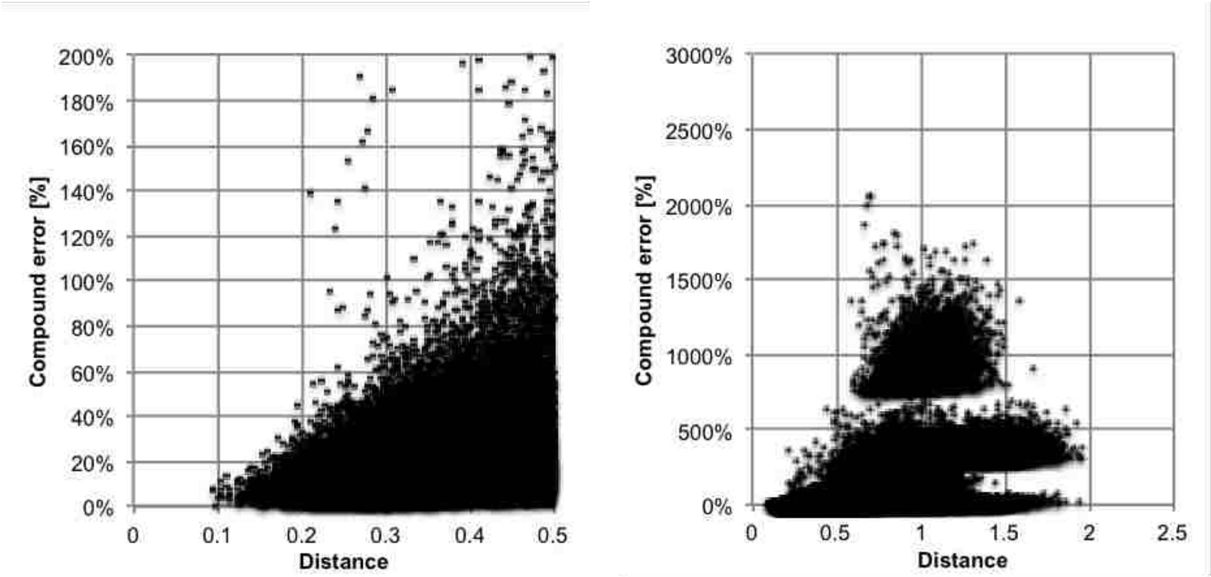


Figure 51. Close-up and General View of the Superposition of the Compound Error vs Distance Graphs 6,878 Flood Producing Targets

In the graph at the right of Figure 51 we can observe that the linearity of the top envelope of most of the scenarios. Nevertheless, we can see some outlying scenarios (about 0.2% of the total) that deserve some special attention. For instance, the scenario located at a distance 0.21 that has a compound error of 139%, when the rest of the scenarios at similar distances have compound errors less than 30%. A thorough analysis of this scenario indicates that is extremely sensitive to the parameter *maxtemp*: a difference of 5.8°F produced a difference of the peak flow from 91 m^3/s in the target scenario to 31 m^3/s in the selected scenario. This analysis could help identify zones of high sensitivity and develop strategies to increase the granularity there. One of such strategies will be presented in the next chapter.

4.3 Conclusions

The objective of this study described in this chapter was to analyze the relationship between the number of scenarios and the accuracy of a scenario-based flood forecasting system. We conclude that there is a clear relationship: on average, the smaller the distance between scenarios, the smaller the difference between their results. In the process of confirming this relationship, we draw some interesting additional conclusions:

- Sometimes the compound error is smaller than expected. This is because the difference in one of the parameters is compensated by the difference in another one. The conservative prediction about the accuracy of the system should be therefore performed using a "top envelope" of the data.
- The logarithm of the distance between two scenarios has a negative linear correlation with the logarithm of the number of scenarios.
- Due to "the curse of dimensionality", the higher the number of parameters used to define the scenarios, the "less negative" the linear correlation is. Therefore, a higher the number of scenarios is required to attain a certain distance between them.
- For short distances, the top envelope of the compound error versus the distance seems to become linear.

- Some scenarios constitute "outliers" over the top envelope. These scenarios should be studied in particular to identify the characteristics that produce increased sensitivity in the model. If the analysis concludes that the sensitivity in the model corresponds to the sensitivity in the real world, these outliers could provide an important clue about non-linear zones, in which small changes can trigger catastrophic flood events.

Using the correlation and the technique presented in this chapter an equation could be derived in the early stages of the scenarios computation to estimate the total number of scenarios required to attain a certain accuracy in the system. This equation could be systematically updated as the number of scenarios available increases.

5 PRIORITIZING SCENARIOS COMPUTATION

In the previous chapters we explained that the use of pre-computed scenarios in flood forecasting systems provides the possibility to increase the accuracy of the system through time by systematically adding new scenarios to the database of results, thus allowing earlier implementations with a limited set of scenarios. One approach to perform this is to define in advance a complete set of scenarios, shuffle them to guarantee the uniformity of their distribution in the possibilities space, and then calculate them sequentially. This method of sorting assumes that "all scenarios are born equal", though it is obvious that some scenarios would be more relevant, or useful than others. The problem, however, is that it is not possible to determine the relevance of a particular scenario before it is computed. In this chapter I present a technique to use the results of the previously computed scenarios to determine the following scenarios to be computed. The basic idea is to identify two neighboring scenarios that produced significantly different results, and identify the scenario closer to the mid-point from both, compute it, and add its results to the system. By repeating this process the regions of the possibilities space with greater gradients are populated earlier than the "flat" regions. These "high gradient" regions, where neighboring scenarios produce significantly different results, are regions where small changes in the parameters could result in big changes in the results obtained from the forecasting system. Having a higher resolution in those areas could help to better predictions.

5.1 Description of the Technique

The first step of the technique presented in this work is the computation of an initial set of scenarios to be included in the system. These will define the rough multi-dimensional results-surface that will be analyzed to define where new scenarios are needed. The new scenarios, once computed, will be included in the system and will gradually refine the results surface.

To define where a new scenario will be relevant, some or all of the scenarios already computed are analyzed. For each scenario, its closest scenario is identified and their results are compared. Since their results are hydrographs the comparison is done on some of the characteristics of them, such as peak flow, total volume, etc. The pair of scenarios with the greatest difference in their results is selected, and a new target scenario is defined at the mid-point between them.

At this point we must note that two different techniques are possible, each having advantages and disadvantages. The first alternative is to define the new scenario exactly where the method proposes (the mid-point between the neighbor scenarios), and the second alternative is to select the nearest pre-defined scenario instead. The first alternative presents the advantage of increasing the density of scenarios in the regions where the system is more sensitive, but on the other hand it will not likely find local maxima that were not caught in the initial rough surface. The second alternative, conversely, will just have the purpose of prioritizing the scenarios of a pre-defined set, populating first the regions more sensitive, but will ultimately populate the whole space uniformly, providing thus a better chance to find local maxima not evident in the initial rough surface. This will prove useful especially in cases of incremental implementation of forecasting systems, where a large number of scenarios are queued to be computed. This approach will prioritize the scenarios to be included first, what will lead to better predictions earlier than with the random selection method. To illustrate the method, for simplicity, in this chapter I will use the second method.

Using all the computed scenarios and finding the closest neighbor pair can become computationally overwhelming as the number of scenarios grows, therefore using a subset of them is usually a better option. In this work I used two different methods to find the better location for a new scenario: first, I used "overthrow", an optimization algorithm to navigate the results surface looking for regions of high gradients; and second, I used a "shotgun approach" to randomly select a subset of scenarios. The first method provided slightly better results, but the computational requirements didn't justify its use, therefore, in the rest of the chapter I will present the results obtained by the use of the "shotgun approach".

To analyze the performance of the proposed technique I used the Hobble Creek model presented before. An initial set of 5,760 scenarios (10% of the total) was computed and added to the system. These defined the first rough surface of results. From this initial set, 2,880 scenarios (50%) were randomly selected. For each of those scenarios the closest one in the whole set was identified. Then, the hydrographs corresponding to each scenario of the pair were compared. This comparison consisted of a compound difference of three characteristics of the hydrograph: 70% of the compound difference corresponded to the difference of the peak flows, 20% to the difference of the total volume, and the remaining 10% to the difference of the peak times.

After calculating the compound differences of the 2,880 pairs of scenarios, it was found that the pair comprised of scenario 4,389 and scenario 5,624 is the one with the largest difference in results (Figure 52).

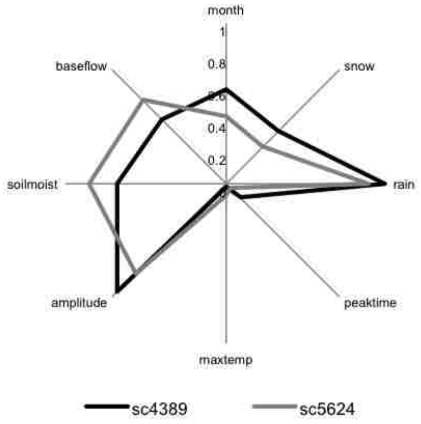


Figure 52. Similar Scenarios with Different Results

In the Figure 52 we observe that both scenarios have similar values of *maxtemp*, *peaktime* and *rain*, but scenario 4,389 has higher values of *amplitude*, *snow*, *month*, and lower values of *baseflow* and *soilmoist*. These differences in the parameters result in an important difference in their hydrographs, as it can be seen in Figure 53.

As we can see, both scenarios produce significantly different results. The three characteristics we are interested in, the peak flow, the total volume, and the peak time are very different. This creates a sort of discontinuity in the results surface, where a small change in the parameters will produce a great difference in the results.

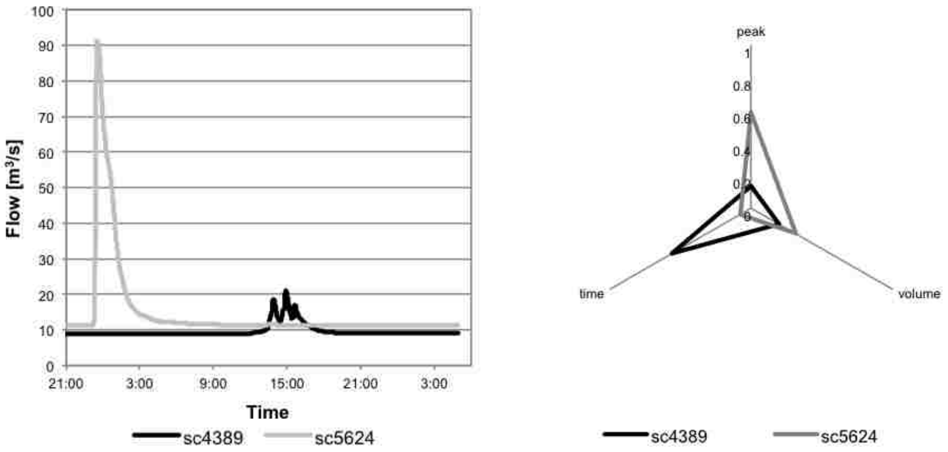


Figure 53. Different Results from Similar Scenarios

A target location for a new scenario was then defined by averaging the values of the parameters of both scenarios (Figure 54).

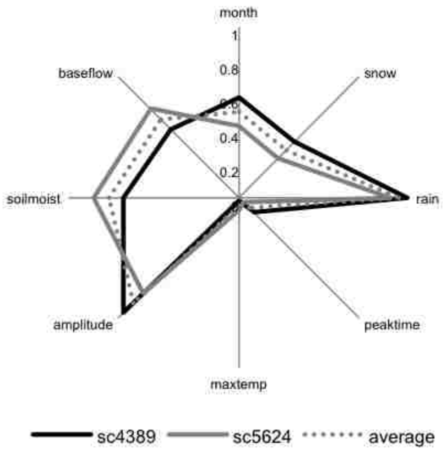


Figure 54. Average Scenario from Similar Scenarios with Different Results

If we were using the first alternative, we would just compute this new location and we would include the results in the set. Since we used the second alternative, we compared this target location to the remaining 51,840 scenarios not yet computed, and we retrieved the closest one —the scenario 56,976 (Figure 55).

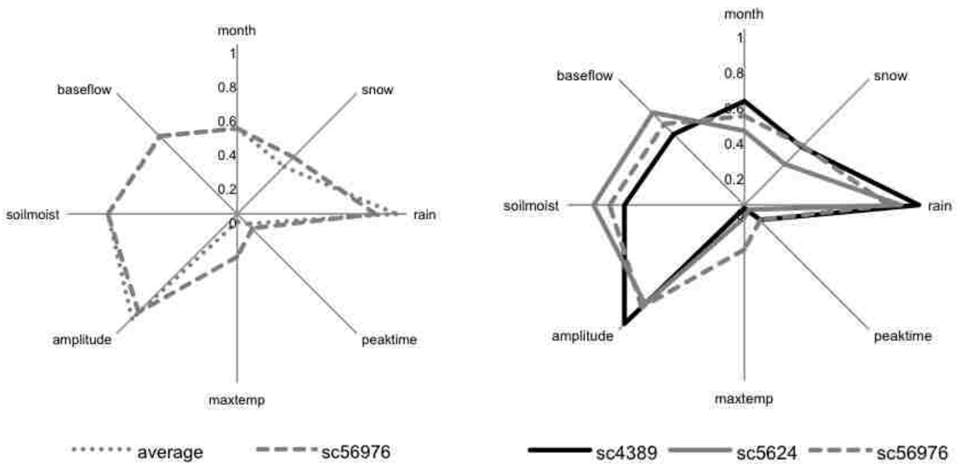


Figure 55. Closest Match to the Average Scenrario

This new scenario was computed and its results were added to the system. As we can see in Figure 56, the new scenario provides a new point that refines the results surface. Since the difference between the scenario 4,389 and the new scenario 56,976 is still significant, it is highly probable that in a future iteration an intermediate scenario between them will be selected and computed.

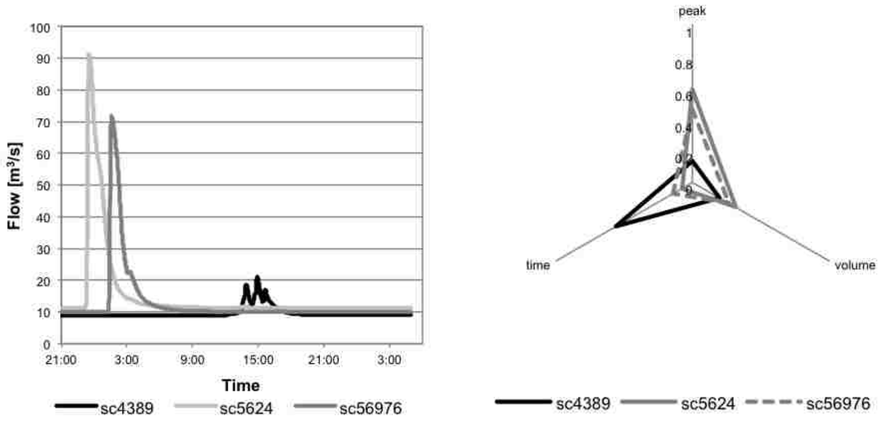


Figure 56. Results of the Closest to the Average Scenario

5.2 Assessing the Performance of the Technique for Different Initial Sets

To assess the performance of this method we need to define the concept of "relevance" of a given scenario. The relevance is a measure of how the system is improved by the adding of a particular scenario, and it is calculated as follows: once the method has been applied and all the pre-defined scenarios have been included in the set, the list of scenarios is navigated, starting from the third scenario in the set. For each scenario *i* the two closest scenarios in the set of scenarios previous to it are selected (that is the reason why the first scenario analyzed must be the third). Next, the compound difference between these two scenarios "$cd12_i$" is then calculated as explained above. This compound difference represents the situation in that region of the results space before the introduction of the scenario under analysis. Then, to evaluate the effect of the addition of the scenario to the set, the compound difference with the two closest scenarios "$cd1n_i$" and "$cd2n_i$" is computed. These two compound differences represent the situation in the region after the introduction of the scenario under analysis. To consider the "worst case", only the highest of both will be considered. The relevance then is calculated with the following equation:

$$\begin{array}{ll} r_i = cd12_i - max(cd1n_i, cd2n_i) & \text{if } r_i > 0 \text{ (relevant)} \\ ri = 0 & \text{if } ri \leq 0 \text{ (not relevant)} \end{array} \qquad (5)$$

Where:

r_i = relevance of the scenario i

$cd12_i$ = compound difference between the two closest scenarios to scenario i

$cd1n_i$ = compound difference between the closest scenario and scenario i

$cd2n_i$ = compound difference between the second closest scenario and scenario i

The accumulated relevance of all the scenarios up to scenario i is then calculated:

$$ar_i = \sum_{j=3}^{i} r_j \qquad (6)$$

To determine the performance of the technique, the series of ar_i are calculated for both the set resulting from applying the method and the set resulting from the random shuffle of scenarios, and then compared by subtracting the latter from the former and plotting the resulting curve. By observing the curves obtained with different initial populations we can compare their relative performance.

Even though the calculations are not complex, the finding of closest scenarios for large sets can take a few seconds for each scenario. For that reason in this analysis I will use a subset consisting of the first 2,000 scenarios instead of the 57,600 available. Given that this 2,000 scenarios where previously shuffled, and therefore are uniformly distributed in the scenarios space, the results obtained in this work should be applicable to the whole set.

I applied this technique using eight different initial populations, each indicated as a percentage of the total: 5%, 10%, 15%, 20%, 25%, 30%, 35%, and 40%. A total of 100 scenarios (5% of the total) are randomly selected at each step, and the one that produced the greatest compound difference with its closest neighbor is selected, computed, and included in the set. Since the selection of scenarios is random, each time the method is applied the results can differ. For that reason the method was applied five times for each initial population, and the results were averaged. The results corresponding to an initial population of 5% are shown in Figure 57.

The graph represents the difference between the accumulated relevance of the set sorted using the proposed method and accumulated relevance of the original shuffled set. Obviously, in the first 5% of the scenarios both accumulated relevances are coincident, and therefore the difference between them is zero. At this point, 5%, the method tries to determine the best scenario to calculate next, based on the data provided by the initial set. As we can see, the first scenarios included are more relevant, but when 10% of the scenarios have been added the results of the technique are not different from the shuffled set. In a following region, between 25% and 75% the technique performs better than the shuffled set, but at the end the shuffled set beats the technique. This can be explained by the small size of the initial population. The results surface defined is too rough to be useful to determine where the better regions to try new scenarios are.

The rest of the graphs in Figure 57 show the influence of the initial set in the development of the relevance.

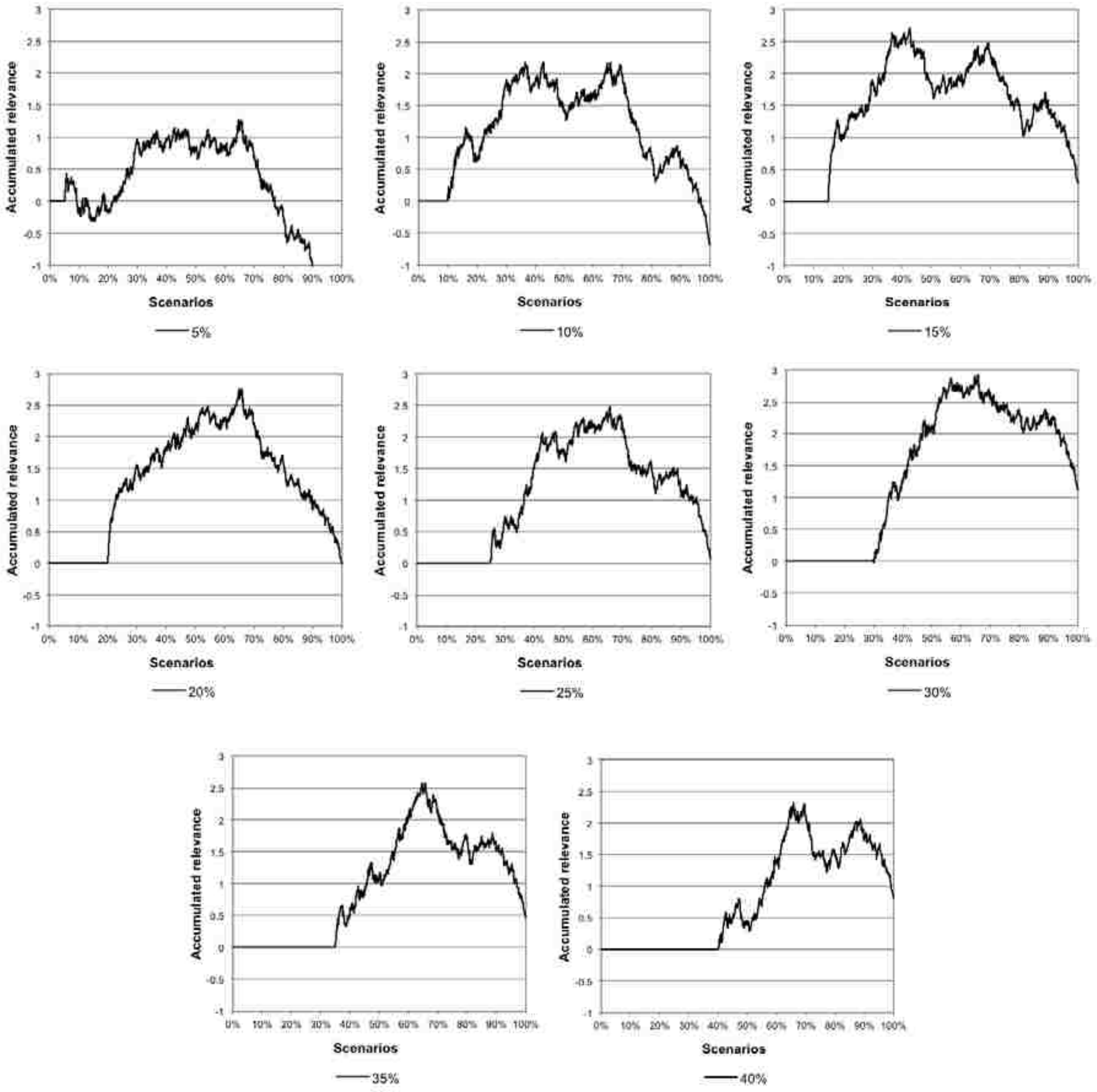


Figure 57. Accumulated Relevance for Different Intial Number of Scenarios

5.3 Conclusions

Given that we used the second alternative of the technique, namely the prioritization of a pre-defined set of scenarios, we could compare the development of the accumulated relevance of this technique compared to the accumulated relevance of the randomly shuffled set used in the previous chapters. We can conclude that in most cases this technique will attain a higher accumulated relevance, which translates in a better prediction capability of the system earlier in its implementation.

Analyzing the results presented in Figure 57 we observe that using a low number of initial scenarios make the initial results surface too rough to be useful. Conversely, using a high number of initial scenarios produces a more refined initial results surface, but provides fewer potential scenarios to select from. For our present model an initial set of 20% of the scenarios seems to provide the best balance.

A better approach may involve a combination of both proposed alternatives: the population of high-gradient regions would increase the resolution of the system where needed ("exploitation", in the Optimization jargon), while the prioritization and inclusion of a pre-defined set of evenly-distributed scenarios would help find new high-gradient regions not well-represented in the initial rough results-space ("exploration"). Further research on this topic is encouraged.

6 GENERATING SCENARIOS FOR TIME-SERIES

In the previous chapters we defined a scenario as a combination of parameters where each parameter was represented as a different dimension in a multi-dimension space. A similar approach can be used to represent time-series as scenarios. In these type kind of scenarios, each value of the time-series will be represented in a different dimension. Therefore, a particular time-series could be represented as a point in a multi-dimensional space, and the similarity between two time-series can be evaluated using the Euclidian distance between those points. Nevertheless, some peculiarities should be considered for this new kind of scenarios. In the previous definition of scenarios I proposed to ignore the statistical correlation between parameters in order to maximize the creation of extreme scenarios. This approach, however, is not appropriate for time-series because time-series of random values will not likely produce greater effects than observed time-series. For example, in a precipitation time-series, an alternating series of high and low values results in the neutralization of the effect of that precipitation on the watershed.

A better approach would consist of somehow synthesizing time-series using patterns present in observed series. In this chapter I will propose a method to generate time-series scenarios using clustering on observed data. Clustering of time-series is not a new concept: as pointed in the rather skeptic paper by Keogh and Lin (Keogh & Lin, 2005), clustering along with other data-mining techniques have been used in time-series analysis in different fields, such as biology (Bar-Joseph, Gerber, Gifford, Jaakkola, & Simon, 2002), finance (Fu, Chung, Ng, & Luk, 2001), geology (Harms, Goddard, Reichenbach, Waltman, & Tadesse, 2002), space exploration (Honda, Wang, Kikuchi, & Konishi, 2002), robotics (Oates, 1999), and human motion analysis (Uehara & Shimada, 2002).

Research on time-series clustering has also been performed in the field of hydrology. That research had two main objectives: (1) to characterize a watershed, and (2) to forecast a flow. Our present objective, though sharing the same mathematical technique, is completely different: to reduce the set of potential time-series to a manageable number. In this sense, our approach is similar to the data compression often used in the field of communications.

6.1 Method

The proposed method consists of extracting segments of a fixed length *n* from a time-series of *nt* observed values. These segments will overlap as follows: the first segment will be composed of the values from 1 to *n*, the second segment from 2 to *n*+1, and so on. At the end, this will result in a set of *nt*-*n*+1 segments. As we explained before, if we consider each of the values in the segment as a different dimension, each segment becomes a point in an *n*-dimensional space, and similar segments will correspond to close points in this space.

To generate the scenarios we can use a clustering algorithm to group similar segments together. The centroid of each of the groups (clusters) is representative of the characteristics of all the segments of the group. Depending on the intended use, these centroids can be "exaggerated" simply by multiplying them by a factor.

6.2 Application of the Method

To test the proposed method I used a time-series of 17,218 observed stream flows of the Mono River (Amoussou, 2010), in Togo, immediately downstream of the Nangbeto Dam (Ago, Petit, & Ozer, 2005) (Figure 58).

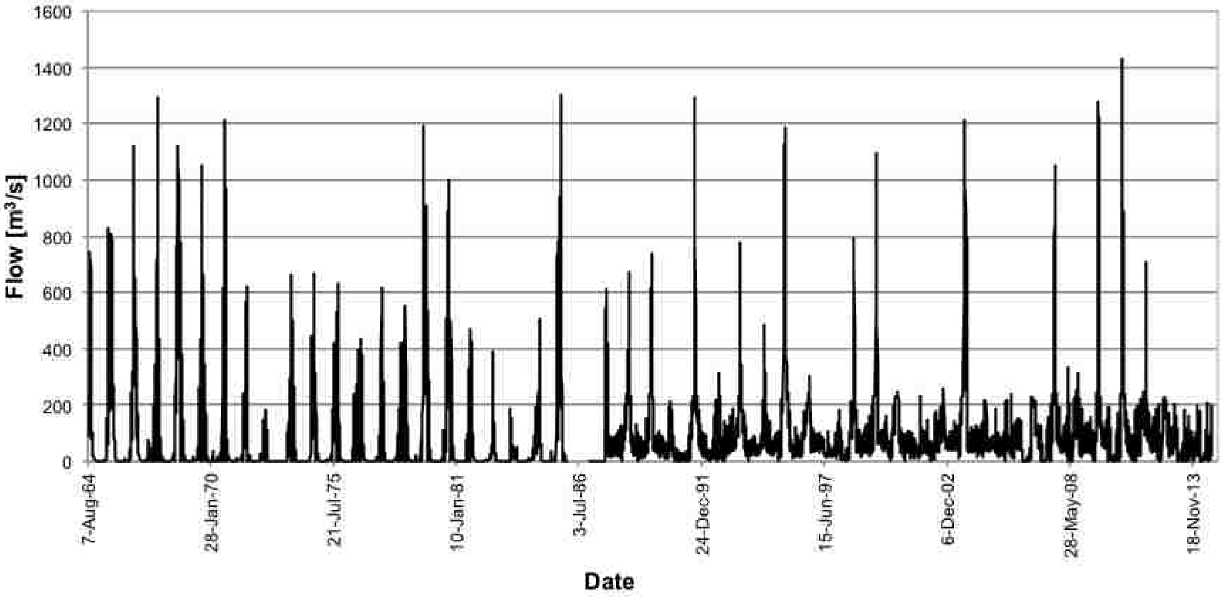


Figure 58. Hydrographs for the Mono River Just Downstream of the Location of the Nangbeto Dam

The initial 7,202 values of the original time-series correspond to observed flows before the construction of the dam, while the following 10,016 values correspond to the outflows of the dam in operation. For the creation of the scenarios I used the complete series, and afterwards we will analyze the usefulness of the results for the flows before and after the construction of the dam.

From this series of 17,218 observed stream flows I extracted 17,214 5-day segments. Each of these segments was represented as points in a 5-dimension space. The objective of the method is to generate a set of potential outflows from the dam, and pre-compute a model for each outflow. In a previous analysis it was determined that flows below 300 m3/s wouldn't produce significant floods, so the segments with all their values lower than this threshold were filtered out from the set. The analysis also showed that the effects of the flow would take up to three days to manifest in the most downstream populations. Also, since the pre-computed models wouldn't have the initial conditions (especially the stream flow in the river), and therefore the river would start "empty" at the beginning of each simulation, the models would require a "warm-up" period of two days before the results could be considered valid. In summary, the length of the segments was determined to be of 5 days: two days of observed flows, and three days of forecasted flows.

A *k*-means clustering algorithm (Hartigan & Wong, 1979) with random initialization was use to reduce the set of segments to a set of 200 centroids. These centroids will constitute the scenarios to be pre-computed. We can observe some of the results in Figure 59.

As we can see, several similar segments resulted in belonging to the same cluster, where a "mean" segment, the centroid, becomes representative of the set. At this point there are several potential transformations that could be applied to these centroids to exaggerate their effects, depending on their intended use. For instance, the centroids could be shifted a constant factor, or a factor such that their maximum value corresponds to the maximum observed value of the cluster, or they could be replaced by the top envelope of the observed values, and so on. Furthermore, the standard deviation of the relative error between the observed values and the centroid of the cluster can be calculated for later use. In our present example we will just use the centroids as they resulted from the clustering analysis without modification.

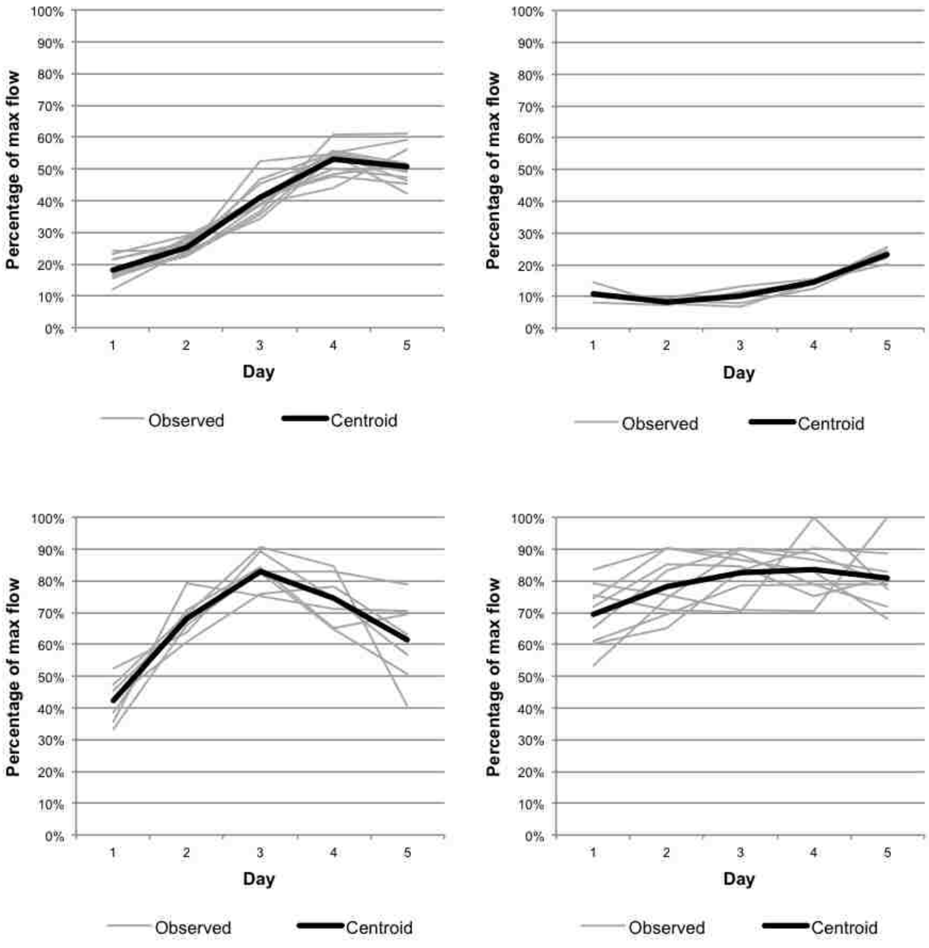


Figure 59. Four Examples of the Clustering Performed

A first assessment of the value of the method is to use these synthetic segments (centroids) to reconstruct a section of the observed time-series. We will do this by extracting a five-day segment of the observed data, retrieve the closest synthetic segment. Since the retrieved centroid will consist of five values, for this assessment we are just comparing the third value of the series with the third value of the segment of observed data. A similar analysis could be performed by comparing any of the other four values, with similar results.

We will first reconstruct two flooding events previous to the build of the dam. In Figure 60 we can observe how the reconstructed time-series match the observed time-series.

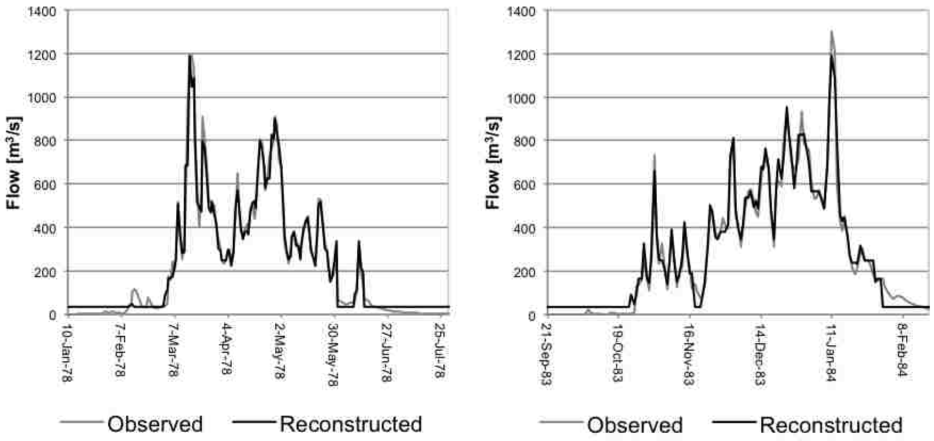


Figure 60. Comparison of the Observed and Recontructed Hydrographs for Two Flooding Events Occurred Before the Construction of the Nangbeto Dam.

In Figure 61 we can observe the correlation between the observed and reconstructed flow for the whole pre-dam series. The layered nature of the graph is a manifestation of the clusters approach: several observed flows are represented by the same centroid.

Next, we will analyze two flooding events after the dam began operating. In these two examples we can see the flood mitigation effect of the dam. The incoming peak flows were absorbed by the dam, which released a non-flooding flow (below 300 m3/s) until the reservoir was filled and the dam operators were forced to increase the releases. Again, as in the previous analysis, we can observe that the reconstructed flows over 300 m3/s matched well with the observed flows (Figure 62).

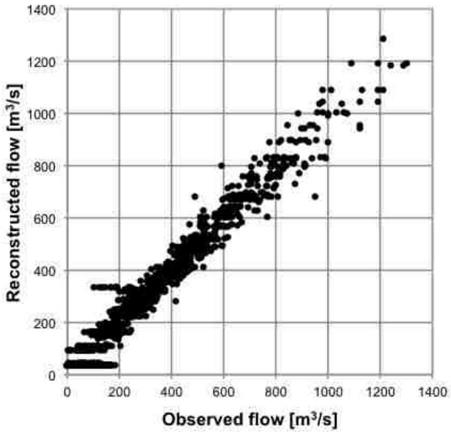


Figure 61. Correlation Between All the Observed and the Reconstructed Flows Before the Construction of the Nangbeto Dam

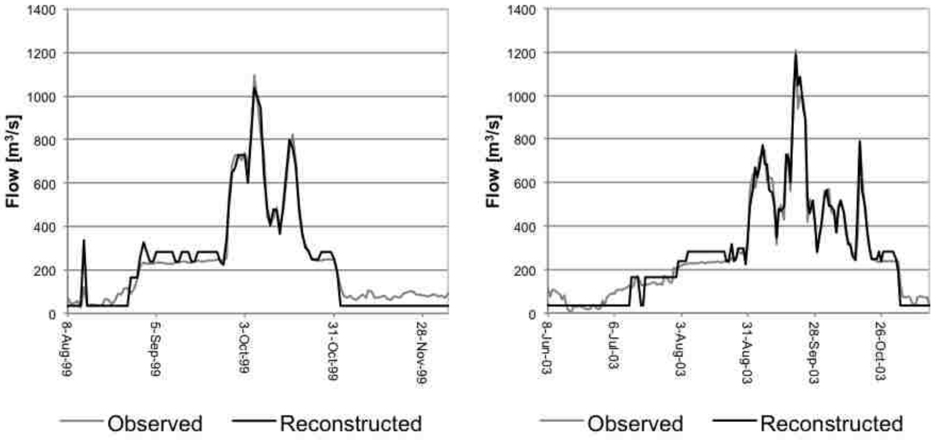


Figure 62. Comparison of the Observed and Recontructed Hydrographs for Two Flooding Events After the Construction of the Nangbeto Dam.

Also, in this case we observe a good correlation between the observed and the reconstructed flows for the whole series (Figure 63).

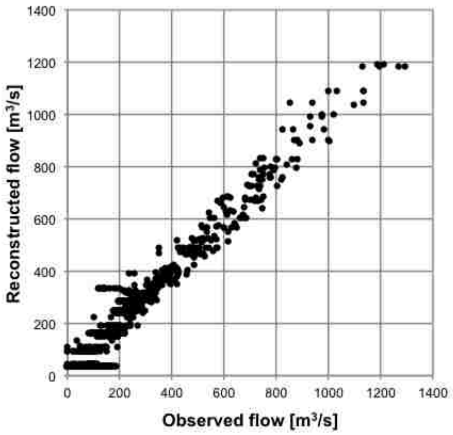


Figure 63. Correlation Between All the Observed and the Reconstructed Flows After the Construction of the Nangbeto Dam

As I explained earlier, the objective of this application of the method is to pre-compute a reduced set of flooding scenarios which could later be used for flood forecasting. To test this capability I created a 5-day time-series that assumes the dam reported a release of 504 m3/s two days ago, and 1119 m3/s yesterday. They estimate, based on an upstream forecast, that they will be releasing 1062 m3/s today, 1003 m3/s tomorrow, and 934 m3/s the day after tomorrow. This 5-day time-series will become our "target". A spatially-distributed physically-based model using these flows as input would take too long to run, but all 200 scenarios (centroids) were already pre-computed, and the results are immediately accessible. The results we are interested in are the water depth at three downstream villages: Mawoussou, Atikpatafo, and Agbanakin.

The closest match for this target corresponds to scenario 3, which consists of the following flows (in m3/s): 596, 964, 1171, 1051, and 867. The resulting water depth time-series for this pre-computed scenario can be seen in the Figure 64:

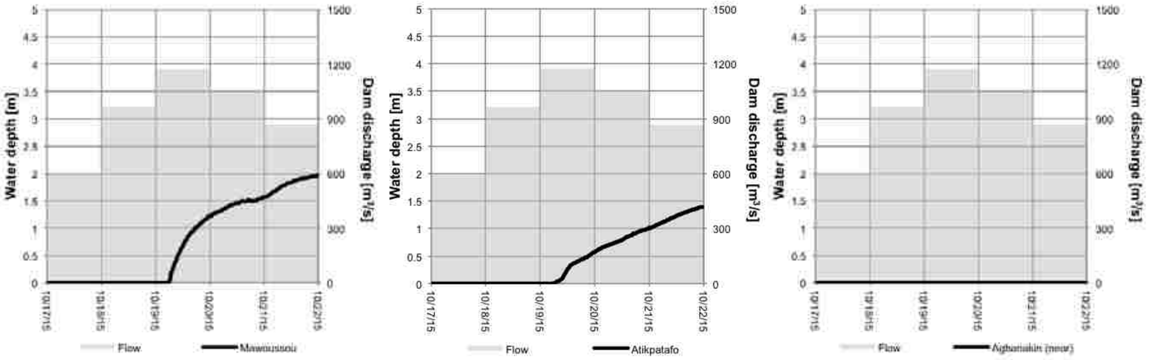


Figure 64. Hydrographs and Water-depth Graphs for Three Locations Downstream of the Nangbeto Dam for the Closest Scenario

In order to be able to assess the goodness of the method we will now run the model for the target, and obtain the "exact" water-depth time-series. These can be seen in the Figure 65:

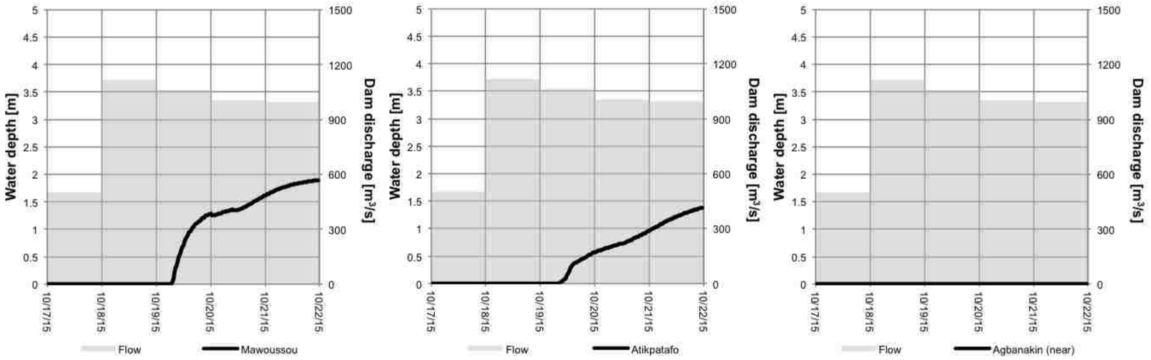


Figure 65. Hydrographs and Water-depth Graphs for Three Locations Downstream of the Nangbeto Dam for the Target Scenario

Now, we can compare the increased water-depth in the two villages that resulted in flooding (Agbanakin was not flooded during this event). We will compare the results of the pre-computed scenario and the post-computed target. In the Figure 66 we can observe that there is no significant difference between the results.

At this point it is interesting to note how two rather different hydrographs produce similar floods. Most of the analyses on models are focused in comparing hydrographs, and as we explained before, we sometimes find that similar conditions can produce significantly different hydrographs. Nevertheless, it is likely that this divergence can be compensated by a convergence such as the one we observed here, from hydrographs to flood maps.

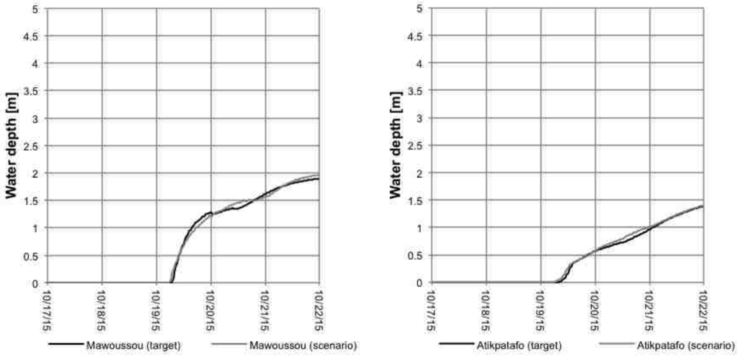


Figure 66. Comparison of the Water-depth Resulting by Computing the Target Scenario and its Closest Match

6.3 Discussion of the Results

The clustering method presented here can be adapted to be used in situations different than the one we just analyzed. One such adaptation could be the connection of global models to local models. For example, at the moment of writing this dissertation, a big effort is being put in generating a Flood Forecast System for the whole continental United States. This system would provide forecasted flows for 2.67 M reaches in the USGS NHD-Plus data set. Nevertheless, transforming the flow into flooded areas for all these reaches in real-time is impossible with the current technology. It would be feasible, however, to create local flood models in zones of particular interest. The input for these models would be the incoming flow predicted by a global system. Since these models could also be too complex to be run in real-time, using a variation of the method proposed in this chapter, several "inflow" scenarios can be defined, and the models for those scenarios can be computed and their results stored. Each time the forecasted flow from the global system is updated, the closest scenario is retrieved and the results can be immediately displayed.

7 CONCLUSIONS

The research I performed provides a method to implement resilient flood forecasting systems in a broad range of circumstances and constraints, that will help to minimize the losses from flooding, especially in human lives, that still threaten vast parts of the globe.

7.1 Contributions

Some of the contributions from my research are:

1. Using the method I defined, a system can be implemented immediately, theoretically with the first scenario computed, and, since the method provides a systematic way to continually increase the number of scenarios through time (both by increasing the number of quantiles for one parameter or the total number of parameters) the predictive accuracy of the system will correspondingly increase.

2. Since the method is "model engine independent" it allows the implementation of systems based on any model engine —or even combination of model engines— available now or in the future. It can also include observed data in the results database.

3. The method provides flexibility in the selection of the number, type, and resolution of parameters.

4. The method allows for separation of the hydrologic modeling, the scenarios generation, and the system implementation. Different specialists, even from different institutions or countries, could be in charge of each of those stages.

5. The computation of the scenarios, the storage of the results, and the hosting of the application to access the selected scenario could all be done either locally or remotely. In addition, the computation of the scenarios can be performed using a single computer or using distributed computing.

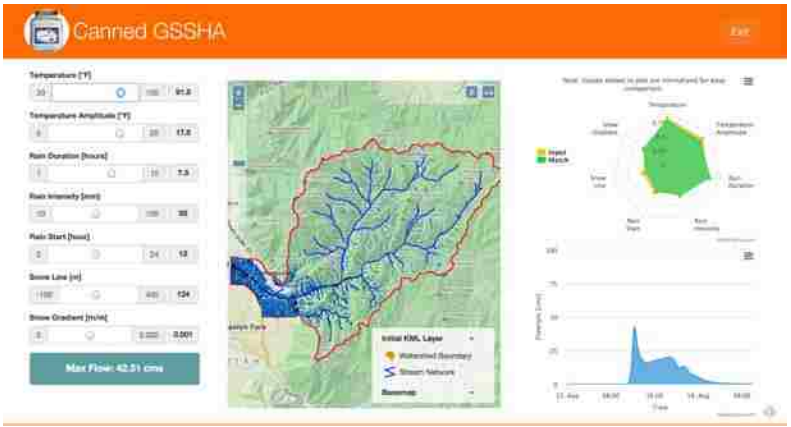
6. The method allows flood forecasting to be connected to different existing telemetry and forecast data, in various formats and resolutions.

7. When new telemetry and forecast data becomes available, the systems implemented with this method can add new scenarios in order to exploit it.

8. The alternative to the use of this method would be to develop or adjust a hydrologic model in real-time, which would require full-time availability of a skilled modeler, substantial computational resources, and sufficient time to wait for the simulation to run to completion. Furthermore, as a storm nears, forecasts may change hourly. This method results in instantaneous predictions that could be performed automatically.

9. Scenarios could be used to connect global and local models, leveraging the strengths of both.

10. Finally, this method allows extremely simple implementations, such as web services, which could grant many people access to flooding impact predictions of a current hydrologic forecast, in the form of hydrographs, flood maps, warning levels, etc. A first attempt in this direction was the development of a Tethys app (Jones et al.) named "Canned GSSHA" (Figure 67) which allows the user to interactively modify the parameters used in this model and immediately retrieves the corresponding hydrograph and flood map.



Temperature [°F] 30 100 91.8

Temperature Amplitude [°F] 6 20 17.8

Rain Duration [hours] 1 10 7.5

Rain Intensity [mm] 10 100 30

Rain Start [hour] 0 24 12

Snow Line [m] -1100 400 124

Snow Gradient [m/m] 0 0.003 0.001

Max Flow: 42.51 cms

Figure 67. Tethys "Canned GSSHA" App

7.2 By-products of the Research

This research also generated several by-products, and enabled new research lines for the future. One of the by-products was the development of "instantaneous models", which are at the core of the proposed flood forecasting systems. These "instantaneous models" can be in and of themselves good teaching tools for Hydrology education. Using them, the students could easily see the effects the change of certain parameters produce on the resulting hydrographs and flood areas. That can also help the student to understand the concepts of sensitivity and non-linearity.

Another by-product is the use of these systems in the training of emergency teams. The forecasts can be replaced by ad-hoc projections, or even “hindcasts”, that would help these teams to develop new strategies, or re-evaluate the ones used in the past.

In addition, I expect this research to open the doors to new research lines. For instance, given that this method relies on a set of discreet scenarios results, research on interpolating these results —hydrographs and flood maps— would be useful to evaluate the accuracy increase versus the number of scenarios required. Another line of research could be the use of the resulting hydrograph, interpolated or not, as an input for real-time hydraulic models, like SRH-2D, in areas of particular concern. A third line of research could consist in the incorporation of observed data of extreme events, following the adage “The best model is the watershed”, into the “database of scenarios” and its seamless integration (through interpolation) with the modeled results. Finally, the sequential approach would allow the research on “incremental calibration”, by which better model results could incrementally replace less accurate ones without having to halt the operation of the model.

7.3 Implementations

This research had produced one publication (Dolder, Jones, & Nelson, 2015) that describes the basic methodology presented in chapter 2. Several other manuscripts are in different stages of the publication process.

But even more rewarding to me is that these ideas are already being implemented. For instance, the Red Cross is implementing a decision system based on the results presented in the Chapter 6. Several hundreds of scenarios —potential outflows from the dam— have been pre-computed to predict floods downstream to the Nangbeto Dam on the Mono River (Figure 68), and the depth of water was determined for different locations.

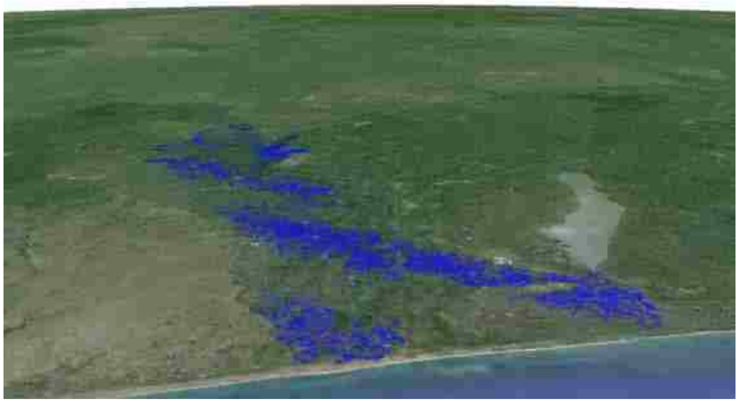
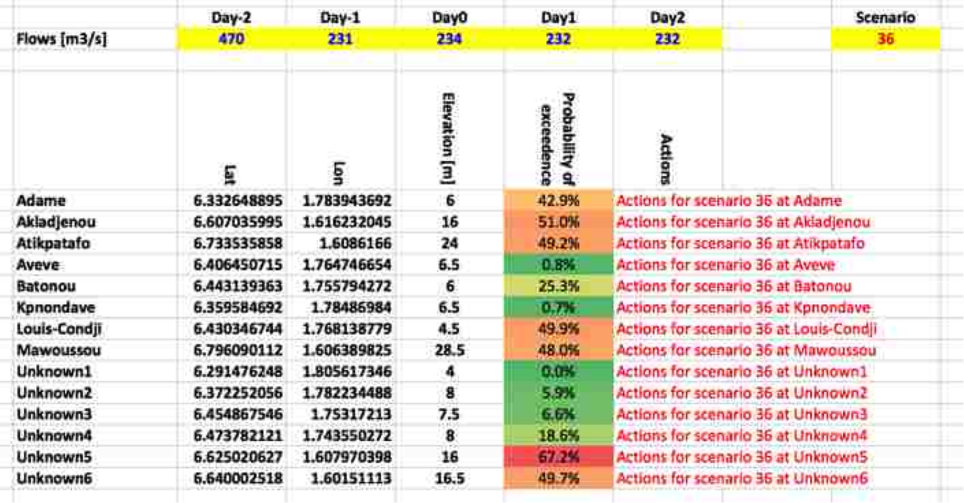


Figure 68. Frame of the Animation of the Flood Corresponding to One of the Scenarios

Each of those scenarios would be connected to a set of actions resulting from the analysis of the flood they would produce. All the data has been synthesized in a simple spreadsheet shown on Figure 69. In this spreadsheet the operator would input the observed outflow of the last two days and the predicted flow for the following three days, as provided from the dam operators. The spreadsheet would then retrieve the closest pre-computed scenario to that time-series, the probability of exceedence (flood) at different locations, and the set of actions predefined for that location and scenario.



	Day-2	Day-1	Day0	Day1	Day2		Scenario
Flows [m3/s]	470	231	234	232	232		36
	Lat	Lon	Elevation [m]	Probability of exceedence	Actions		
Adame	6.332648895	1.783943692	6	42.9%	Actions for scenario 36 at Adame		
Akladjenou	6.607035995	1.616232045	16	51.0%	Actions for scenario 36 at Akladjenou		
Atikpatafo	6.733535858	1.6086166	24	49.2%	Actions for scenario 36 at Atikpatafo		
Aveve	6.406450715	1.764746654	6.5	0.8%	Actions for scenario 36 at Aveve		
Batonou	6.443139363	1.755794272	6	25.3%	Actions for scenario 36 at Batonou		
Kpnondave	6.359584692	1.78486984	6.5	0.7%	Actions for scenario 36 at Kpnondave		
Louis-Condji	6.430346744	1.768138779	4.5	49.9%	Actions for scenario 36 at Louis-Condji		
Mawoussou	6.796090112	1.606389825	28.5	48.0%	Actions for scenario 36 at Mawoussou		
Unknown1	6.291476248	1.805617346	4	0.0%	Actions for scenario 36 at Unknown1		
Unknown2	6.372252056	1.782234488	8	5.9%	Actions for scenario 36 at Unknown2		
Unknown3	6.454867546	1.75317213	7.5	6.6%	Actions for scenario 36 at Unknown3		
Unknown4	6.473782121	1.743550272	8	18.6%	Actions for scenario 36 at Unknown4		
Unknown5	6.625020627	1.607970398	16	67.2%	Actions for scenario 36 at Unknown5		
Unknown6	6.640002518	1.60151113	16.5	49.7%	Actions for scenario 36 at Unknown6		

Figure 69. Prototype of the Lower Mono Warning System

Figure 70. Pablo Suarez (Red Cross Red Crescent Climate Centre Associate Director for Research and Innovation), Janot Mendler de Suarez (Red Cross Red Crescent Climate Centre Consultant) and Joachim Schroeder (German Red Cross, Donor) Presenting to Mr. Ferdinand Sessou (Director of the Nangbeto Dam) a Quantitative Analysis for the Red Cross Standard Operating Procedures to be Based on Thresholds Defined by the Scenarios-based Model.

A major advantage of this implementation is that the system can become the "memory" of the watershed. If the predefined actions resulted ineffective, the operator can update the set of actions in the spreadsheet so that if that scenario is retrieved again in the future, a more effective set of actions will be proposed. In the long run, this system can become a "model-less system" in which the parameters (time-series, in this case) are directly connected to the actions to perform, independent of the results of the model.

7.4 The Future

The field of flood forecast is in many regards still in its infancy. Several efforts are being exerted around the world to address the unresolved problem of floods. One of those was the National Flood Interoperability Experiment (NFIE). During the NFIE Summer Camp, I had the opportunity to talk with some early responders and to present to them some of the concepts contained in this dissertation, and I was impressed by the enthusiasm they showed. There is a real need of tools to help making decision as early as possible, and there is where the research I performed can really make a difference.

REFERENCES

Aggarwal, C., Hinneburg, A., & Keim, D. (2001). On the Surprising Behavior of Distance Metrics in High Dimensional Space. In J. Van den Bussche & V. Vianu (Eds.), *Database Theory — ICDT 2001* (Vol. 1973, pp. 420-434): Springer Berlin Heidelberg.

Ago, E. E., Petit, F., & Ozer, P. (2005). Analysis of flood downstream from the Nangbeto dam on the Mono River (Togo and Benin).

Amoussou, E. (2010). *Variabilité pluviométrique et dynamique hydro-sédimentaire du bassin versant du complexe fluvio-lagunaire Mono-Ahémé-Couffo (Afrique de l'ouest).* Université de Bourgogne.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., . . . Stoica, I. (2010). A view of cloud computing. *Communications of the ACM, 53*(4), 50-58.

Bar-Joseph, Z., Gerber, G., Gifford, D. K., Jaakkola, T. S., & Simon, I. (2002). *A new approach to analyzing gene expression time series data.* Paper presented at the Proceedings of the sixth annual international conference on Computational biology.

Bellman, R. (1956). Dynamic programming and Lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America, 42*(10), 767.

Bentley, J. L., Weide, B. W., & Yao, A. C. (1980). Optimal expected-time algorithms for closest point problems. *ACM Transactions on Mathematical Software (TOMS), 6*(4), 563-580.

Beven, K. J. (2011). *Rainfall-runoff modelling: the primer*: John Wiley & Sons.

Chen, F., Crow, W. T., Starks, P. J., & Moriasi, D. N. (2011). Improving hydrologic predictions of a catchment model via assimilation of surface soil moisture. *Advances in Water Resources, 34*(4), 526-536.

Deb, K., & Saxena, D. (2006). *Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems.* Paper presented at the Proceedings of the World Congress on Computational Intelligence (WCCI-2006).

Dolder, H. G., Jones, N. L., & Nelson, E. J. (2015). Simple Method for Using Precomputed Hydrologic Models in Flood Forecasting with Uniform Rainfall and Soil Moisture Pattern. *Journal of Hydrologic Engineering*, 04015039.

Dorigo, W., Xaver, A., Vreugdenhil, M., Gruber, A., Hegyiová, A., Sanchis-Dufau, A., . . . Drusch, M. (2013). Global automated quality control of in situ soil moisture data from the International Soil Moisture Network. *Vadose Zone Journal, 12*(3).

Downer, C. W., & Ogden, F. L. (2004). GSSHA: Model to simulate diverse stream flow producing processes. *Journal of Hydrologic Engineering, 9*(3), 161-174.

Falcone, J. A., Carlisle, D. M., Wolock, D. M., & Meador, M. R. (2010). GAGES: a stream gage database for evaluating natural and altered flow conditions in the conterminous United States: Ecological Archives E091-045. *Ecology, 91*(2), 621-621.

Fu, T.-c., Chung, F.-l., Ng, V., & Luk, R. (2001). *Pattern discovery from stock time series using self-organizing maps.* Paper presented at the Workshop Notes of KDD2001 Workshop on Temporal Data Mining.

Green, I. R. A., & Stephenson, D. (1986). Criteria for comparison of single event models. *Hydrological Sciences Journal, 31*(3), 395-411. doi: 10.1080/02626668609491056

Harms, S. K., Goddard, S., Reichenbach, S. E., Waltman, W. J., & Tadesse, T. (2002). *Data mining in a geospatial decision support system for drought risk management.* Paper presented at the Proceedings of the 1st National Conference on Digital Government.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Applied statistics*, 100-108.

Henonin, J., Russo, B., Mark, O., & Gourbesville, P. (2013). Real-time urban flood forecasting and modelling--a state of the art. *Journal of Hydroinformatics, 15*(3).

Homer, C., Dewitz, J., Fry, J., Coan, M., Hossain, N., Larson, C., . . . Wickham, J. (2007). Completion of the 2001 national land cover database for the counterminous United States. *Photogrammetric Engineering and Remote Sensing, 73*(4), 337.

Honda, R., Wang, S., Kikuchi, T., & Konishi, O. (2002). Mining of moving objects from time-series images and its application to satellite weather imagery. *Journal of Intelligent Information Systems, 19*(1), 79-93.

Hounam, D., & Werner, M. (1999). *The Shuttle Radar Topography Mission (SRTM).* Paper presented at the Proceedings of ISPRS workshop-Sensors and Mapping from Space.

Jones, N., Nelson, J., Swain, N., Christensen, S., Tarboton, D., & Dash, P. Tethys: A Software Framework for Web-Based Modeling and Decision Support Applications.

Kennedy, A. B., Westerink, J. J., Smith, J. M., Hope, M. E., Hartman, M., Taflanidis, A. A., . . . Smith, T. (2012). Tropical cyclone inundation potential on the Hawaiian Islands of Oahu and Kauai. *Ocean Modelling, 52*, 54-68.

Keogh, E., & Lin, J. (2005). Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems, 8*(2), 154-177.

Köppen, M. (2000). *The curse of dimensionality.* Paper presented at the 5th Online World Conference on Soft Computing in Industrial Applications (WSC5).

Mahfouf, J.-F. (1991). Analysis of soil moisture from near-surface parameters: A feasibility study. *Journal of applied meteorology, 30*(11), 1534-1547.

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics, 21*(2), 239-245.

Metropolis, N., & Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association, 44*(247), 335-341.

Nash, J., & Sutcliffe, J. (1970). River flow forecasting through conceptual models part I—A discussion of principles. *Journal of Hydrology, 10*(3), 282-290.

Njoku, E. G., Jackson, T. J., Lakshmi, V., Chan, T. K., & Nghiem, S. V. (2003). Soil moisture retrieval from AMSR-E. *Geoscience and Remote Sensing, IEEE Transactions on, 41*(2), 215-229.

Oates, T. (1999). *Identifying distinctive subsequences in multivariate time series by clustering.* Paper presented at the Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining.

Ochoa-Rodríguez, S., Thraves, L., & Johnston, A. Surface water flood warnings in England: An overview, assessment and recommendations based on survey responses and workshops.

Park, J.-S. (1994). Optimal Latin-hypercube designs for computer experiments. *Journal of statistical planning and inference, 39*(1), 95-111.

Peleg, D. (2000). Distributed computing. *SIAM Monographs on discrete mathematics and applications, 5*.

Raymond, M., Peyron, N., & Martin, A. (2006). *ESPADA, A unique flood management tool: first feedback from the September 2005 flood in Nîmes.* Paper presented at the 7th International Conference on Hydroinformatics, HIC.

Refsgaard, J. C. (1997). Parameterisation, calibration and validation of distributed hydrological models. *Journal of Hydrology, 198*(1–4), 69-97. doi: http://dx.doi.org/10.1016/S0022-1694(96)03329-X

Schaefer, G. L., & Paetzold, R. F. (2001). *SNOTEL (SNOwpack TELemetry) and SCAN (soil climate analysis network).* Paper presented at the Proc. Intl. Workshop on Automated Wea. Stations for Appl. in Agr. and Water Resour. Mgmt.

Schwarz, G. E., & Alexander, R. (1995). State Soil Geographic (STATSGO) data base for the conterminous United States.

SCS. (1975). Urban hydrology for small watersheds. Technical Release No. 55: Engineering Division, Soil Conservation Service, U.S. Department of Agriculture, Washington, DC.

Serreze, M. C., Clark, M. P., Armstrong, R. L., McGinnis, D. A., & Pulwarty, R. S. (1999). Characteristics of the western United States snowpack from snowpack telemetry (SNOTEL) data. *Water Resources Research, 35*(7), 2145-2160.

Taflanidis, A. A., Kennedy, A. B., Westerink, J. J., Smith, J., Cheung, K. F., Hope, M., & Tanaka, S. (2012). Rapid Assessment of Wave and Surge Risk during Landfalling Hurricanes: Probabilistic Approach. *Journal of Waterway, Port, Coastal, and Ocean Engineering, 139*(3), 171-182.

Thain, D., Tannenbaum, T., & Livny, M. (2005). Distributed computing in practice: The Condor experience. *Concurrency-Practice and Experience, 17*(2-4), 323-356.

Titov, V., González, F., Bernard, E. N., Eble, M., Mofjeld, H., Newman, J., & Venturato, A. (2005). Real-Time Tsunami Forecasting: Challenges and Solutions. In E. N. Bernard (Ed.), *Developing Tsunami-Resilient Communities* (pp. 41-58): Springer Netherlands.

Uehara, K., & Shimada, M. (2002). Extraction of primitive motion and discovery of association rules from human motion data *Progress in Discovery Science* (pp. 338-348): Springer.

Verleysen, M., & François, D. (2005). The Curse of Dimensionality in Data Mining and Time Series Prediction. In J. Cabestany, A. Prieto & F. Sandoval (Eds.), *Computational Intelligence and Bioinspired Systems* (Vol. 3512, pp. 758-770): Springer Berlin Heidelberg.

Wächter, J., Babeyko, A., Fleischer, J., Häner, R., Hammitzsch, M., Kloth, A., & Lendholt, M. (2012). Development of tsunami early warning systems and future challenges. *Natural Hazards and Earth System Science, 12*(6), 1923-1935.

Xian, G., Homer, C., & Fry, J. (2009). Updating the 2001 National Land Cover Database land cover classification to 2006 by using Landsat imagery change detection methods. *Remote Sensing of Environment, 113*(6), 1133-1147.

Ye, Q.-Z. (2011). Forecasting cloud cover and atmospheric seeing for astronomical observing: application and evaluation of the global forecast system. *Publications of the Astronomical Society of the Pacific, 123*(899), 113-124.