Evolutionary Approach of Clustering to Optimize Hydrological Simulations

Elnaz Azmi[1(✉)][0000-0002-0073-8940], Marcus Strobl[1][0000-0001-8265-227X], Rik van Pruijssen[2][0000-0002-9337-2246], Uwe Ehret[2][0000-0003-3454-8755], Jörg Meyer[1][0000-0003-0861-8481], and Achim Streit[1][0000-0002-5065-469X]

[1] Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany
{elnaz.azmi,marcus.strobl,joerg.meyer2,achim.streit}@kit.edu
[2] Institute of Water and River Basin Management, Karlsruhe Institute of Technology, Karlsruhe, Germany
{rik.pruijssen,uwe.ehret}@kit.edu

Abstract. Modeling of hydrological systems and their dynamics in high spatio-temporal resolution leads to a better understanding of the hydrological cycle, thus it reduces the uncertainties in hydrologic forecasts. Simulation of such high-resolution, distributed and physically based models demands high performance computing resources. However, the availability of such computing resources is restricted in some domains. In this paper, we propose an approach to reduce computational costs by reducing hydrological model redundancies using similarities in functionality of hydrological model units. The approach applies K-Means clustering to detect similar model units and simulates only one representative unit of each cluster. The clustering is applied when rainfall is forced to the hydrological system and is based on the structure, current state and flux of the model units. Application of this evolutionary approach on a test case results in a 1.8x speedup over the original simulation run time and the RMSE of 0.0049 compared to the original simulation output.

Keywords: Clustering · K-Means · Time series analysis · Simulation.

1 Introduction

Physically based and highly detailed models of environmental processes are used to improve the understanding of the nature of hydrological systems [22]. Such models are spatially heterogeneous and consist of a hierarchy of units [7, 26]. Thus, the simulation of these models in high spatio-temporal resolution is compute-intensive. One of the popular methods to tackle this issue is to use high performance computing and parallel processing of hydrological model units [9, 12, 14]. However, the parallelization of these models is challenging because of their heterogeneous nature and a demand of partially sequential execution of the model units. Also the interconnection of model units can be very tight, and the necessary communication of the units per time step makes efficient parallelization challenging. Additionally, these methods require programming expertise of domain scientists and partially revision of the existing modeling software. In this work, we introduce an approach to make use of redundancies in the simulated hydrological systems in order to decrease the computational effort of such simulations. The redundancies are due to the natural hydrological behavior of the model units and the simplification caused by the model choice. The remainder of this paper is structured as follows: Sect. 2 provides further information about the study background, Sect. 3 is a survey of related work, the proposed approach is explained in Sect. 4. In Sect. 5, the processing results are presented, Sect. 6 is about the implementation environment and the conclusions are drawn in Sect. 7.

2 Background

In this paper we apply our method on the CAOS (Catchment as Organized Systems) model proposed in [26]. This model simulates water related dynamics in catchments up to hundreds of square kilometers. The CAOS model provides a high-resolution and distributed process based simulation of hydrological systems. In this model, functioning of catchments, defined as a closed area draining completely to a single point along a river (the catchment outlet), is controlled by a hierarchy of three major model units, namely, Elementary Functional Unit (EFU), Hillslope (HSL), and River element (RIV). EFUs are soil columns containing other sub-units to transfer water through soil layers and other EFUs. HSLs are subsets of hills, and independent from each other. They contain several EFUs and have a connection point to a RIV. Finally, RIVs are linear elements along the lower edge of a HSL. They are parts of a river, connected sequentially to each other and transport the water of a catchment to the lowermost point, the catchment outlet (Fig. 1).

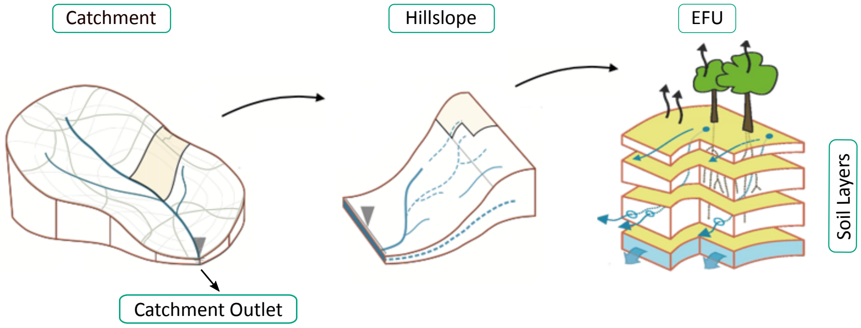

Fig. 1. Simplified hierarchy of the CAOS model units (modified after [26]).

The structure of the model is specified by domain knowledge. The resulting simulation dynamics depend on, 1. model units properties (static), 2. model units state (current discharge) and 3. forcing (rainfall or radiation). The underlying principle is that similar properties, states and forcings of the model units lead to similar simulation dynamics [26] which lead to redundancies that can be removed from the computation process. In order to examine this hypothesis, the Wollefsbach catchment [24] is used to develop and test of the evolutionary approach on the CAOS model. It is located in the Attert basin in Luxembourg with an area of 4.5 km^2 and 174 HSLs (Fig. 2). For this catchment we had access to the model and the required data set available in the CAOS project [26].

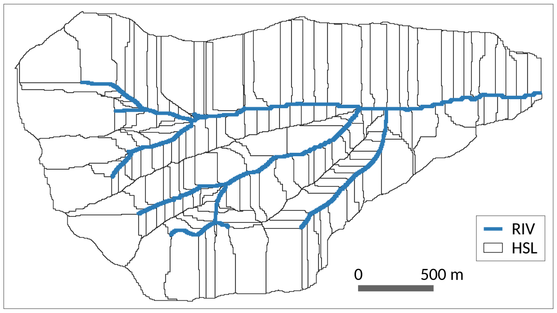

Fig. 2. Wollefsbach catchment divided into HSLs, delineated in black. Each HSL has an edge connected to a RIV in blue.

3 Related Work

In environmental science and especially in hydrology, the identification of similarities plays a key role to detect patterns in the environment, extract correlations and improve future events forecast [6, 26, 20, 1, 5]. In order to define the hydrologic similarity, [23] suggested a framework that is both descriptive and predictive. Their metrics to define hydrologic similarity or dissimilarity between catchments include static characteristics and dynamic response of a catchment to its forcing. They discussed the demand for a catchment classification system based on the structure and hydro-climatic conditions of catchments as well as their functional response to the precipitation input. Following this, [20] derived signatures from precipitation-temperature-streamflow data to apply a Bayesian clustering and to identify groups of similar catchments. In the evolutionary approach, we include such similarity metrics using drainage time series of the HSLs within a catchment to represent static properties like structure, size, slope, soil profile and drainage of the HSLs. Their detection of similarities has been done mostly at catchment level, while we apply this process at HSL level within a catchment in order to reduce the recurring simulation properties (statics and dynamics).

Classification and clustering are machine learning techniques that identify groups of similar objects using already labeled data or object neighborhood properties like distance and density [15, 11]. Such methods provide efficient pattern detection and help in a better understanding of hydrological systems [21, 13]. Due to the lack of labeled data we chose a clustering method to detect similar HSLs. Selection of an appropriate clustering method depends on several parameters like the type of input data and clustering output, scalability and robustness, and thus it is use case dependent. [10] compared the hierarchical clustering fuzzy C-mean (FCM) and K-Means to analyze regional flood frequency and its underlying distribution. The results of both clustering methods for their application were almost similar so they concluded the choice of the best clustering method depends on the individual use case. [16] presented a clustering-based classification of climate data that resulted in internally more homogeneous and externally more distinct climate types than the types in the rule-based Kppen-Geiger classification, which is the de facto standard in global climate classification. [27] successfully defined regions with clear boundaries of homogeneous precipitation regions with highly varied spatio-temporal patterns using K-Means on a gridded dataset for automatic delineation. [3] tested different clustering methods to identify similar hydrological model units on the CAOS model where K-Means performed best. Thus, for our use case, we use K-Means clustering as well.

4 Methodology

In this work, we introduce an evolutionary approach to speed up the hydrological simulation which consists of two steps, namely, initial clustering and evolutionary clustering. The idea behind this approach is to reduce the computational costs by reducing redundant computations and calculating a close approximation of the original model dynamics. Using hydrological similarity [6], we distinguish similar model units considering their structure, current state and flux to detect clusters in the whole system. In the hierarchy of the CAOS model, dynamics of each HSL are independent of the other HSLs. Thus, we use them as individual objects to apply the K-Means clustering. Afterwards, we select a representative HSL of each cluster and execute the simulation only on the representatives. The next step is to map the output of the representatives to the remaining cluster members. This way, we avoid running the simulation for all model units and consequently, reduce the computation time. The degree of fluctuation of the simulation output can be controlled by the number of clusters and the frequency of applying clustering. Finally, we compare our results from the evolutionary approach with the results from the original simulation respecting execution time and simulation output. In Fig. 3, the whole approach is delineated step by step in detail. The original simulation consists of the major steps shown in the white boxes (Fig. 3): 1. loading the catchment structure and creating a list of processes (dynamics) between the model units, 2. starting the simulation for a predefined number of time steps (n), 3. running the simulation according to the processes list, and 4. saving the output and finalizing the simulation. The evolutionary approach shown in the gray boxes (Fig. 3) is described further in detail.

4.1 Initial Clustering

In order to define a representation of the static properties of the HSLs, namely, structure, size, slope, soil profile and drainage, we run a drainage test [3]. The drainage test results in discharge time series for each HSL. These time series are

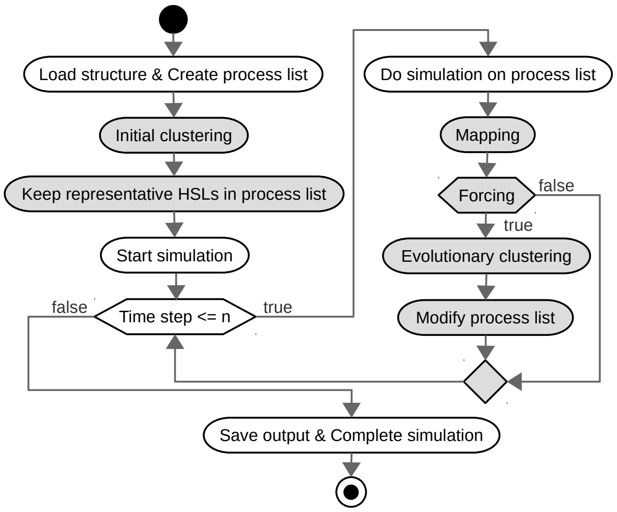
Load structure & Create process list
Initial clustering
Keep representative HSLs in process list
Start simulation
Time step <= n
false
true
Do simulation on process list
Mapping
Forcing
false
true
Evolutionary clustering
Modify process list
Save output & Complete simulation

Fig. 3. Simulation workflow consists of the original and evolutionary approach.

hydrologic characteristics that provide an insight into the functionality of the HSLs. We extract seven features from these time series which are the input data for the clustering method. The features are the four moments *Mean*, *Variance*, *Skewness*, *Kurtosis* and the three hydrologically significant features *1st Gradient*, *Active Storage* and *Time to Equilibrium* [3]. In this feature set we identified outliers in the *Mean*, *Kurtosis*, *1st Gradient* and *Active Storage* features (see arrows in Fig. 4). In the preprocessing step, we separate these outliers from the clustering and simulate them as single clusters. K-Means clustering requires the number of clusters (K) to be set. Further, we describe the parameter setting of the K-Means clustering used in the initial and evolutionary clustering.

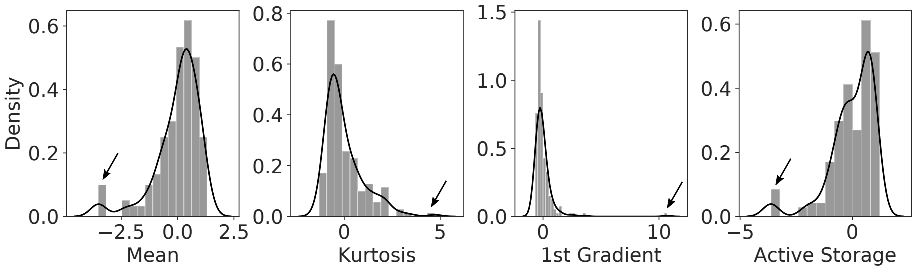

Fig. 4. Feature histograms and kernel density estimate fits.

4.2 Clustering Parameter Setting

The K-Means clustering package [18] that we used for our tests requires a set of parameters to be determined. The main parameter to be specified is the number of clusters (K). There are several methods like *elbow* [19], *silhouette* [25] and *RMSE-Computation-Time (rmse-ctime)* [3] methods to determine an appropriate value for K. The process of selecting an appropriate K is called here K-determiner. Another parameter to be set for the K-Means clustering is the *random seed* which determines the random number generation for centroid initialization. Thus, we use a fixed integer random seed (zero), to make K-Means deterministic, so that running it multiple times produces the same results. The following describes how the K-determiner works using the *elbow* and *rmse-ctime* methods.

elbow method The *elbow* method runs K-Means clustering on a given dataset for a range of K values, and for each value of K, calculates the average distance from data points to the centroid of each cluster. As K increases, the average distance to the centroid decreases rapidly until the elbow or maximum curvature of the calculated curve [19] which is the optimal K (Fig. 5, left). We determined the elbow by using the point with the maximum distance from the straight line connecting the end-points of the curve. In order to save time, K-determiner runs the clustering for a set of Ks in a defined interval, interpolates the results and calculates the elbow point (Fig. 5, left).

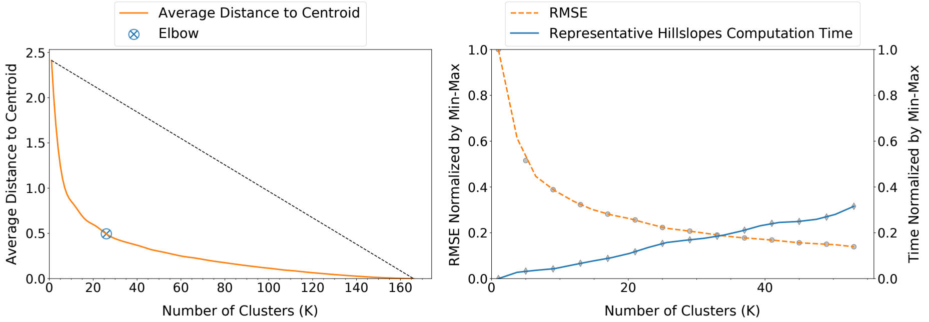

Fig. 5. Application of the *elbow* (left) and *rmse-ctime* (right) methods on the input feature set.

rmse-ctime method We have introduced another approach to determine the number of clusters at [3], that considers the balance between the number of clusters (K), Root Mean Square Error (RMSE) [4] of each cluster member HSL and the representative HSL of that cluster as well as the simulation computation time of HSLs. This technique is a customized *elbow* method, that allows scientists to decide which K to use whether they prefer a lower RMSE or a lower computation time. K-determiner applies this method for a set of Ks in an interval from one HSL to one third of all HSLs, and interpolates the points (Fig. 5, right). This reduces the run time of K-determiner through interpolation instead of clustering and calculating RMSE for all potential number of clusters. For our tests, K-determiner uses the intersection point of the curves, which is a balance of RMSE and computation time, as the appropriate K.

4.3 Representative Output Mapping

The initial clustering (Sect. 4.1) is followed by the selection of a representative HSL of each cluster. We define a cluster representative as the Medoid object whose average dissimilarity to other objects in the cluster is minimal. At each time step of the simulation, only the representative HSLs are simulated. The next step of the evolutionary approach is the mapping and scaling of the output of the representative HSL to the member HSLs of the same cluster. Therefore, the discharge (output) of the representative HSL, already computed in the simulation, is used to calculate the discharge of other HSLs using the following equation:

$$\mathrm{CMHD} = \mathrm{RHD} \times \frac{\mathrm{CMHA}}{\mathrm{RHA}} \tag{1}$$

where CMHD is the cluster member HSL discharge $[\frac{m^3}{s}]$, RHD the representative HSL discharge $[\frac{m^3}{s}]$, CMHA the cluster member HSL area $[m^2]$ and RHA the representative HSL area $[m^2]$.

4.4 Evolutionary Clustering

According to the domain knowledge, the dynamics of the simulation depends on the static properties, current discharge of HSLs and the amount of rainfall enforced to the HSLs. In the evolutionary approach, first we include the static features of the HSLs by running the initial clustering. In the next steps of the approach, while the simulation is running, we add two features, namely, current discharge and flux to our feature set. Flux is defined as the volume of rainfall enforced to the area of HSLs in a given time. Hence, we define our evolutionary clustering as a clustering method that uses a new feature set dynamically during the simulation. The detailed steps of the evolutionary approach are:

a) Determine the initial K using the K-determiner and do the initial clustering.
b) Select representative HSL of each cluster and simulate the first time step.
c) Do the output mapping and update the status of all HSLs.
d) If there is no forcing at the next time steps, continue running the simulation with the already defined representatives.
e) When forcing starts, use the K-determiner, run the evolutionary clustering and continue the simulation with a new set of representatives.
f) Do step e) without the K-determiner for time steps in the time frame with active forcing (forcing time block) because the values of the feature set change strongly with variable forcing over time.
g) When forcing stops, use K-determiner, run the evolutionary clustering and continue the simulation with a new set of representatives until the next forcing time block is reached.

5 Processing Results

In this section, we show the results of the proposed approach applied on the study case (Wollefsbach). There are metrics available to evaluate the quality of the evolutionary approach, i.e. how close our results are to the original simulation. We use three metrics, alone or in combination, to show the quality of the evolutionary approach, namely, RMSE, Pearson Correlation Coefficient (PCC) [17] and the Kling-Gupta Efficiency (KGE) [8]. KGE is a measure of the goodness-of-fit, commonly used in hydrological modeling. In addition to the quality metrics, the computational efficiency of the evolutionary approach is presented as the simulation run time speedup in comparison to the run time of the original simulation. Values of the RMSE closer to zero shows a better estimation of the model results. PCC and KGE values closer to one indicate higher efficiency. All evaluation results are shown in tables 1-6 which are sorted based on RMSE in ascending order. In addition, as simulation of time blocks with forcing is more compute intensive, we also show the RMSE for time blocks with forcing (RMSE-WF) and without forcing (RMSE-WOF) separately. The best values of the metrics are shown in bold type.

5.1 Influence of Random Seed

The original simulation executes processes of the model units in random order to keep the model close to the natural behavior of a hydrological system. Change of the random seed in the original simulation results in slightly different curves. We run the original simulation four times with different random seeds to test the randomness in the nature of the model. In Fig. 6, the horizontal axis shows the simulation time of one week (1st to 7th of January). The left vertical axis shows the discharge at the catchment outlet after the simulation and the right vertical axis from top to bottom the amount of rainfall during the simulation. The gray band shows the minimum and maximum of all tests. The curve labeled “Original” is used as ground truth for all following tests, hence all evaluation results (Tbl. 1-6) are relative to this curve. Tbl. 1 shows the evaluation of the remaining three tests relative to the ground truth.

Table 1. Evaluation results of the original simulation with different random seeds.

Tests	RMSE	RMSE-WOF	RMSE-WF	PCC	KGE
Test-1	**0.00155**	**0.00136**	**0.00214**	**0.990**	0.960
Test-3	0.00163	0.00143	0.00221	0.988	**0.983**
Test-2	0.00213	0.00191	0.00281	0.984	0.954

In addition to the randomness of the original simulation, the evolutionary approach uses the K-Means clustering that generates its initial centroids randomly. In order to retain reproducible test results, we set the *random seed* parameter from K-Means clustering to zero as well. However, to evaluate the influence of a variable random seed on the simulation result, we run six simulations (Test-1

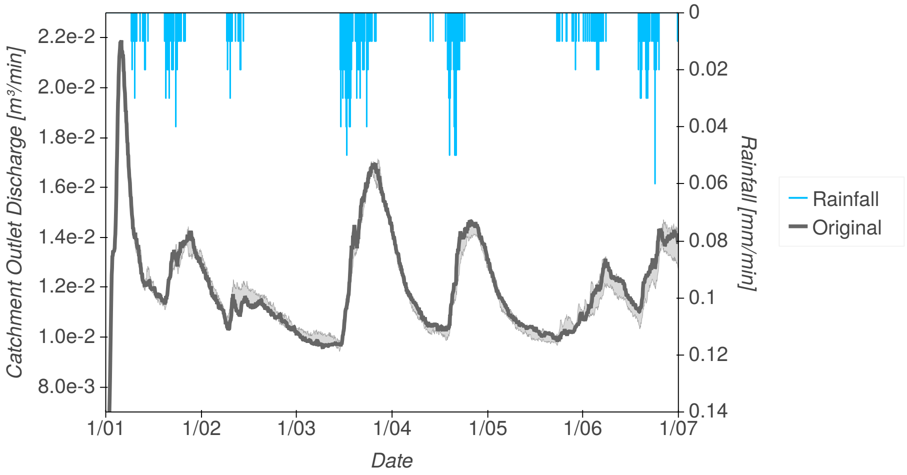

Fig. 6. Output of the original simulation with different random seeds.

to Test-6, Tbl. 2) with different random seeds set for the clustering (Fig. 7). Only the initial clustering with $K = 9$ without the evolutionary clustering was applied to these simulations. All tests, represented by the gray band, show a similar trend like the original simulation, and an acceptable KGE from the hydrological perspective (Tbl. 2). The following sections describe detailed tests to evaluate our evolutionary approach with automatically set parameters.

Table 2. Evaluation results of the tests with different random seeds.

Tests	RMSE	RMSE-WOF	RMSE-WF	PCC	KGE
Test-6	**0.0045**	**0.0044**	**0.0050**	**0.920**	0.765
Test-4	0.0056	0.0050	0.0073	0.891	0.776
Test-3	0.0056	0.0054	0.0064	0.868	0.729
Test-1	0.0078	0.0075	0.0087	0.907	**0.803**
Test-5	0.0096	0.0091	0.0110	0.858	0.748
Test-2	0.0097	0.0091	0.0117	0.814	0.713

5.2 Constant-K

In order to reveal the similarities in the static model unit properties of the hydrological model, we have designed a test that applied only initial clustering at the first time step of the simulation without running the evolutionary clustering. The simulation continues using the representative HSLs of the initial clustering. The catchment outlet discharge is calculated as output for each time step (Fig. 8) and the evaluation results are shown in Tbl. 3. We have repeated the approach for a range of Ks defined as a percentage of total HSLs (5 - 30% of HSLs corresponds to $K = 9$ - 50) to test the effect of parameter K on the simulation output. The curves follow the trend of the original simulation (Fig. 8). Evaluation of the

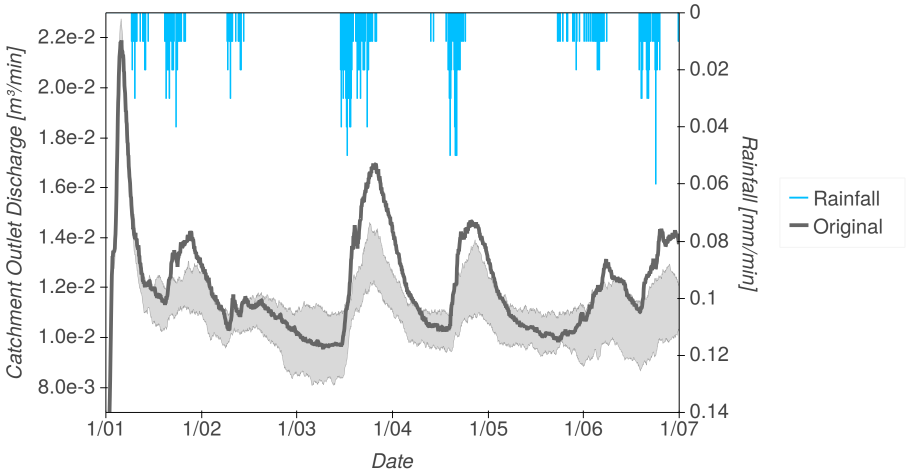

Fig. 7. Simulation output of the tests with different random seeds for K-Means.

results reveals that our quality measures have no strong correlation with K, and the RMSE order does not fit one by one to the KGE metric (Tbl. 3). This means that the test with a lower RMSE is not always the more efficient one. However, the smaller the K is, the higher the speedup is. To further increase the speedup by keeping the RMSE value low and the PCC as well as KGE values high, we designed the following Variable-K tests.

Table 3. Evaluation results of the Constant-K tests.

Tests	RMSE	RMSE-WOF	RMSE-WF	PCC	KGE	Speedup
K-42	**0.0037**	**0.0035**	0.0042	**0.954**	**0.818**	2
K-50	0.0039	0.0039	**0.0040**	0.943	0.794	1.9
K-9	0.0045	0.0044	0.0050	0.920	0.765	**3.2**
K-34	0.0066	0.0061	0.0081	0.909	0.775	2.3
K-17	0.0070	0.0070	0.0068	0.927	0.806	2.8
K-25	0.0092	0.0088	0.0103	0.923	0.792	2.4

5.3 Variable-K

In Sect. 5.2, we have shown tests running the initial clustering with the static properties of the model units. In the following, we include the effect of the current state and flux of the model units into our approach with the evolutionary clustering. In order to define the frequency of running the evolutionary clustering during the simulation, we considered the intensity of the dynamics occurred. Since more dynamic processes run when there is forcing due to rainfall in the simulation, we split the simulation time into "with forcing (WF)" and "without forcing (WOF)" time blocks and run the evolutionary clustering only by the

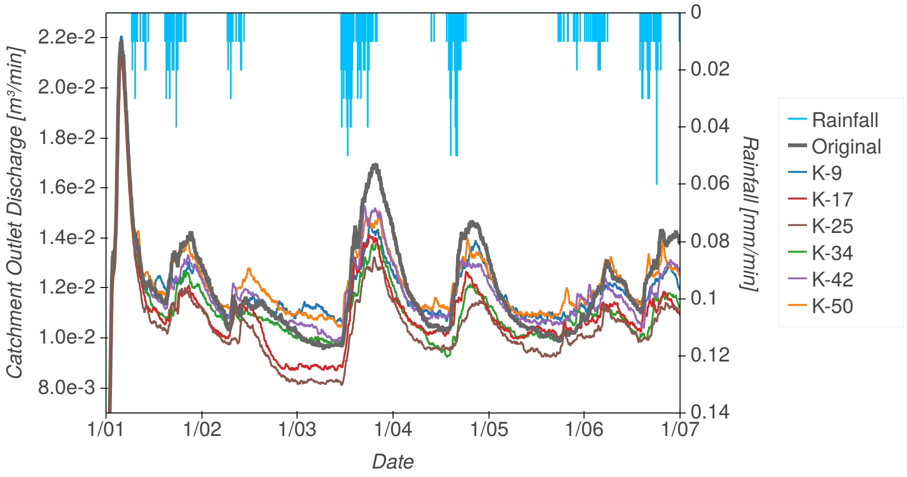

Fig. 8. Simulation output of the Constant-K tests.

forcing time blocks. Tbl. 3 shows a tendency to better results for higher K, which are associated with a higher run time. Thus, using the results of Constant-K, we designed the Variable-K tests that execute the simulation using different representatives from the evolutionary clustering. In order to obtain high quality results during high dynamics combined with a short run time, we changed K during the simulation, according to the on- and offset of forcing (rainfall) (Fig. 9). This means we split the simulation run into using a high K at WF time blocks and a low K at WOF time blocks. We paired higher Ks and lower RMSE values for WF time blocks and lower Ks and lower RMSE values for WOF time blocks according to the results of Sect. 5.2. The simulation is started with the initial clustering using the best K of WOF time block. Then according to the forcing, the simulation continues with a low K at WOF time blocks and it uses evolutionary clustering with a high K at WF time blocks (Fig. 9). Because of the clustering overhead, we use the evolutionary clustering only when switching between WF and WOF time blocks and back. The trend of the curves for Variable-K is difficult to interpret, so the tests show a tendency to a higher RMSE and lower PCC as well as KGE than Constant-K (Tbl. 4).

Table 4. Evaluation results of the Variable-K tests.

Tests	RMSE	RMSE-WOF	RMSE-WF	PCC	KGE	Speedup
K-17-50	**0.0061**	**0.0058**	**0.0071**	**0.823**	**0.719**	1.9
K-17-42	0.0079	0.0076	0.0090	0.720	0.714	2
K-9-42	0.0101	0.0095	0.0122	0.693	0.591	**2.1**
K-9-50	0.0121	0.0123	0.0113	0.509	0.462	**2.1**

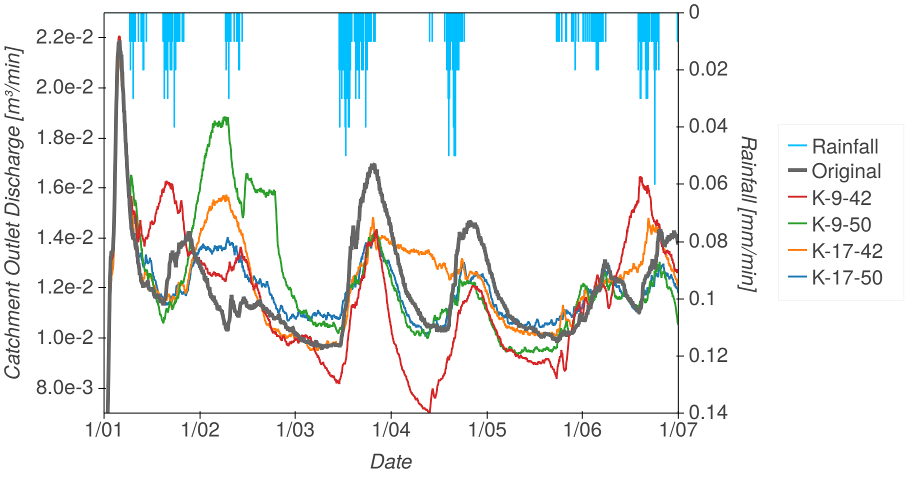

Fig. 9. Simulation output of the Variable-K tests.

5.4 Auto-K

The Variable-K test showed a high variation by changing K during the simulation. Thus, in Auto-K, we apply the K-determiner during the simulation in order to select an appropriate K automatically and reduce the high fluctuations. The Auto-K tests have the same settings as Variable-K tests with the difference that for Auto-K, we do not set K manually and use the K-determiner once at the beginning of each forcing time block. The K-determiner selects an appropriate K automatically using the evolving feature set generated based on the dynamics occurring during the simulation. In the Auto-K tests, K-determiner applies the *elbow* (K-AEL) and *rmse-ctime* (K-ARC) methods respectively. The trend of their curves fits well to the original. The gray band, representing six tests for K-ARC with different random seeds, becomes thinner in the forcing time blocks, thus shows a reproducible peak discharge (Fig. 10). The RMSE for both tests, K-ARC and K-AEL, is acceptably low. Although K-ARC results in a lower RMSE and higher PCC as well as KGE than that of K-AEL, its speedup is lower than K-AEL since K-ARC uses higher Ks (Tbl. 5).

Table 5. Evaluation results of the Auto-K tests.

Tests	RMSE	RMSE-WOF	RMSE-WF	PCC	KGE	Speedup
K-ARC	**0.0049**	**0.0046**	0.0059	**0.894**	**0.80**	1.8
K-AEL	0.0061	0.0061	**0.0058**	0.828	0.72	**2**

Additionally, to test the variability of the best K, we run the Auto-K with K-determiner at all time steps of the simulation and got a narrow frequency distribution of the selected best Ks. K ranges between 17 and 30 with the most frequent $K = 25$ for the K-determiner with the *elbow* method (Fig. 11, left) and the

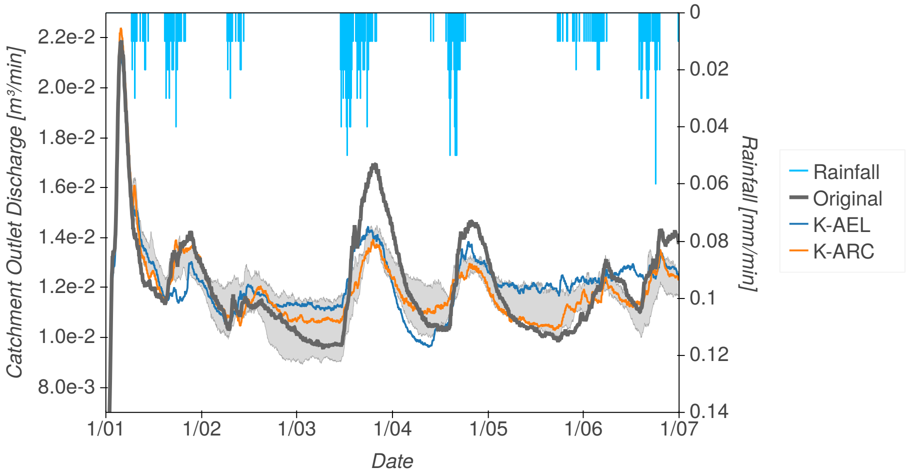

Fig. 10. Simulation output of the Auto-K tests.

K range between 30 and 38 with the most frequent $K = 34$ for the K-determiner with the *rmse-ctime* method (Fig. 11, right). Using the K-determiner at each time step instead of only once at the beginning of each forcing time block showed slightly better results in the quality evaluation than the Auto-K tests, though it is such inefficient, that the whole simulation run time will be longer than the original simulation (Tbl. 6).

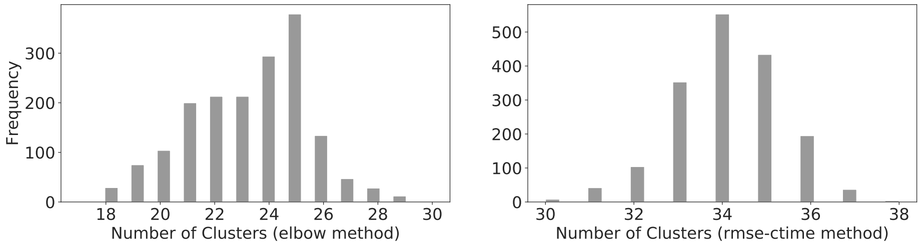

Fig. 11. Histogram of the selected Ks by K-determiner when it is applied at all time steps of the simulation.

5.5 Summary of Results

We applied three different test settings (Constant-K, Variable-K and Auto-K) to speedup a hydrological simulation with K-Means clustering. Our metrics show minor differences for the Auto-K and Constant-K tests, considering several test runs with different random seeds set for the K-Means clustering (Tbl. 2, 3 and 5). The results of Auto-K, that clusters at every time step of each forcing time block, show lower fluctuations, hence a more reliable output at these time blocks

Table 6. Evaluation results of the Auto-K tests when the K-determiner is applied at all time steps of the simulation.

Tests	RMSE	RMSE-WOF	RMSE-WF	PCC	KGE	Speedup
K-AEL	**0.0051**	0.005	**0.0055**	0.950	0.825	0.6
K-ARC	0.0053	0.005	0.0060	**0.954**	**0.843**	0.6

compared to the Constant-K tests (Fig. 7 and 10). The Auto-K tests use a combination of the initial and evolutionary clustering with the K-determiner to determine the appropriate K dynamically based on the given feature set and configure the clustering parameters automatically during the simulation. The additional clustering steps of the Auto-K tests result in a better RMSE, PCC and KGE, and a lower speedup. Since K-ARC takes $K = 34$ frequently as the appropriate K, its comparison with K-34 test from Constant-K tests confirms this statement (Tbl. 3 and 5). Constant-K tests showed the potential for a higher speedup (Fig. 12), although the ideal choice for a high speedup together with a low RMSE appears to be unpredictable. This means, although K-42 from Constant-K tests has the lowest RMSE value of all tests and a speedup of 2, the prediction of such a favorable K without running tests for a particular use case is not possible (Fig. 12). As a solid solution, we recommend K-ARC from Auto-K tests with a speedup of 1.8, and acceptable RMSE, PCC and KGE values.

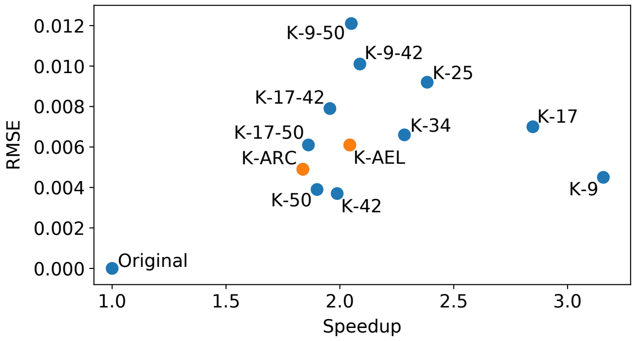

Fig. 12. Simulation run time speedup and RMSE of all tests. The orange markers highlight the Auto-K tests.

6 Implementation Environment

The simulation scripts are written and executed in Matlab R2019a. The analysis methods are implemented in Python. All tests are executed on a Red Hat Enterprise Linux Server release 7.4 on a 16-core Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00 GHz processor. All scripts, data files and requirements of the analyses are available as a GitLab repository named “hyda” [2].

7 Conclusions

In this work we introduced an approach to make use of landscape properties and dynamics of hydrological models to reduce computational redundancies in hydrological simulations. The approach consists of several steps, mainly, initial and evolutionary clustering and scaling of the simulation output of the cluster representatives to the remaining cluster members. We have used the K-Means clustering together with the K-determiner that automatically defines a suitable number of clusters using the *elbow* and *rmse-ctime* methods. The results of our tests demonstrated that the K-ARC approach has a promising RMSE of 0.0049, PCC of 0.89 and KGE efficiency of 0.8 which is a close approximation of the original simulation output. Additionally, K-ARC has a simulation run time speedup of 1.8 that is close to half of the original simulation run time.

References

1. Ali, G., Tetzlaff, D., Soulsby, C., McDonnell, J.J., Capell, R.: A comparison of similarity indices for catchment classification using a cross-regional dataset. Adv. Water Resour. **40**, 11–22 (2012). https://doi.org/10.1016/j.advwatres.2012.01.008
2. Azmi, E.: Hydrological data analysis (March 2020), `https://gitlab.com/elnazazmi/hyda`
3. Azmi, E., Ehret, U., Meyer, J., van Pruijssen, R., Streit, A., Strobl, M.: Clustering as approximation method to optimize hydrological simulations. In: European Conference on Parallel Processing. pp. 256–269. Springer (2019). https://doi.org/10.1007/978-3-030-29400-7_19
4. Barnston, A.G.: Correspondence among the correlation, rmse, and heidke forecast verification measures; refinement of the heidke score. Weather Forecast. pp. 699–709 (1992)
5. Corzo, G., Solomatine, D.: Baseflow separation techniques for modular artificial neural network modelling in flow forecasting. Hydrolog. Sci. J. **52**(3), 491–507 (2007). https://doi.org/10.1623/hysj.52.3.491
6. Ehret, U., Zehe, E., Scherer, U., Westhoff, M.: Dynamical grouping and representative computation: a new approach to reduce computational efforts in distributed, physically based modeling on the lower mesoscale. In: presented at the AGU Chapman conference, 23–26 September, 2014 (2014)
7. Grayson, R., Blöschl, G.: Spatial patterns in catchment hydrology: observations and modelling. CUP Archive (2001)
8. Gupta, H.V., Kling, H., Yilmaz, K.K., Martinez, G.F.: Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. J. Hydrol. **377**(1-2), 80–91 (2009)
9. Jones, J.E., Woodward, C.S.: Newton-krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. Adv. Water Resour. pp. 763–774 (2001). https://doi.org/10.1016/S0309-1708(00)00075-0
10. Kar, A.K., Goel, N., Lohani, A., Roy, G.: Application of clustering techniques using prioritized variables in regional flood frequency analysis-case study of mahanadi basin. J. Hydrol. Eng. **17**(1), 213–223 (2012)
11. Kassambara, A.: Practical guide to cluster analysis in R: Unsupervised machine learning, vol. 1. STHDA (2017)

12. Kollet, S.J., Maxwell, R.M., Woodward, C.S., Smith, S., Vanderborght, J., Vereecken, H., Simmer, C.: Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources. Water Resour. Res. **46**(4) (2010). https://doi.org/10.1029/2009WR008730
13. Ley, R., Casper, M., Hellebrand, H., Merz, R.: Catchment classification by runoff behaviour with self-organizing maps (som). Hydrol. Earth Syst. Sci. **15**(9), 2947–2962 (2011). https://doi.org/10.5194/hess-15-2947-2011
14. Maxwell, R., Condon, L., Kollet, S.: A high-resolution simulation of groundwater and surface water over most of the continental us with the integrated hydrologic model parflow v3. Geosci. Model Dev. p. 923 (2015)
15. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
16. Netzel, P., Stepinski, T.: On using a clustering approach for global climate classification. J. Clim. pp. 3387–3401 (2016). https://doi.org/10.1175/JCLI-D-15-0640.1
17. Pearson, K.: Vii. mathematical contributions to the theory of evolution.iii. regression, heredity, and panmixia. Philos. Trans. R. Soc. A pp. 253–318 (1896). https://doi.org/10.1098/rsta.1896.0007
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
19. Satopaa, V., Albrecht, J., Irwin, D., Raghavan, B.: Finding a ”kneedle” in a haystack: Detecting knee points in system behavior. In: 2011 31st international conference on distributed computing systems workshops. pp. 166–171. IEEE (2011)
20. Sawicz, K., Wagener, T., Sivapalan, M., Troch, P.A., Carrillo, G.: Catchment classification: empirical analysis of hydrologic similarity based on catchment function in the eastern usa. Hydrol. Earth Syst. Sci. **15**(9), 2895–2911 (2011). https://doi.org/10.5194/hess-15-2895-2011
21. Sawicz, K., Kelleher, C., Wagener, T., Troch, P., Sivapalan, M., Carrillo, G.: Characterizing hydrologic change through catchment classification. Hydrol. Earth Syst. Sci. **18**(1), 273–285 (2014). https://doi.org/10.5194/hess-18-273-2014
22. Schulz, K., Seppelt, R., Zehe, E., Vogel, H.J., Attinger, S.: Importance of spatial structures in advancing hydrological sciences. Water Resour. Res. **42**(3) (2006). https://doi.org/10.1029/2005WR004301
23. Wagener, T., Sivapalan, M., Troch, P., Woods, R.: Catchment classification and hydrologic similarity. Geography compass **1**(4), 901–931 (2007). https://doi.org/10.1111/j.1749-8198.2007.00039.x
24. Wrede, S., Fenicia, F., Martínez-Carreras, N., Juilleret, J., Hissler, C., Krein, A., Savenije, H.H., Uhlenbrook, S., Kavetski, D., Pfister, L.: Towards more systematic perceptual model development: a case study using 3 luxembourgish catchments. Hydrol. process. **29**(12), 2731–2750 (2015). https://doi.org/10.1002/hyp.10393
25. Zaki, M.J., Meira Jr, W., Meira, W.: Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press (2014). https://doi.org/10.1017/CBO9780511810114
26. Zehe, E., Ehret, U., Pfister, L., Blume, T., Schroeder, B., Westhoff, M., Jackisch, C., Schymanski, S.J., Weiler, M., Schulz, K., et al.: Hess opinions: From response units to functional units: a thermodynamic reinterpretation of the hru concept to link spatial organization and functioning of intermediate scale catchments. Hydrol. Earth Syst. Sci. pp. 4635–4655 (2014). https://doi.org/10.5194/hess-18-4635-2014
27. Zhang, Y., Moges, S., Block, P.: Optimal cluster analysis for objective regionalization of seasonal precipitation in regions of high spatial-temporal variability: application to western ethiopia. J. Clim. **29**(10), 3697–3717 (2016)