Stoch Environ Res Risk Assess (2010) 24:547–557
DOI 10.1007/s00477-009-0335-x

ORIGINAL PAPER

Modified dynamic programming approach for offline segmentation of long hydrometeorological time series

Abdullah Gedikli · Hafzullah Aksoy · N. Erdem Unal · Athanasios Kehagias

Published online: 20 August 2009

Abstract For the offline segmentation of long hydrometeological time series, a new algorithm which combines the *dynamic programming* with the recently introduced *remaining cost* concept of branch-and-bound approach is developed. The algorithm is called *modified dynamic programming* (mDP) and segments the time series based on the first-order statistical moment. Experiments are performed to test the algorithm on both real world and artificial time series comprising of hundreds or even thousands of terms. The experiments show that the mDP algorithm produces accurate segmentations in much shorter time than previously proposed segmentation algorithms.

Keywords Time series · Offline segmentation · Change point · Dynamic programming · Modified dynamic programming · Remaining cost concept

1 Introduction

Time series analysis has always been an important topic in hydrology and related sciences, such as climatology, meteorology, environmetrics, etc. For instance, analysis of streamflow, precipitation and temperature records has been conducted widely in the past and it is still attractive due to its importance in both research and practice. The past behavior of a time series is important toward understanding its future behavior: we forecast the future by looking at the past. In the same manner, the detection of abrupt or gradual changes in the time series has always been interesting. Therefore, variability in hydrometeorological time series, i.e. streamflow and precipitation, should be analyzed by proper means (Kundzewicz and Robson 2004; Radziejewski and Kundzewicz 2004; Xiong and Guo 2004; Koutsoyiannis 2006; Aksoy 2007; Dahamsheh and Aksoy 2007; Aksoy et al. 2008a). The recent emergence of interest in climate change scenaria makes the issue even more important. For example, global temperature records from the past centuries can be used to test the hypothesis of an ongoing "greenhouse effect" (which can possibly lead to an environmental disaster). The detection of gradual or abrupt changes (trend or jump, respectively) is of the highest importance in order to detect the natural or manmade causes changing the behavior of the hydrometeorological time series. The changing behavior must be taken into account when attempting to extrapolate the past into the future.

Structural characteristics of hydrometeorological variables (precipitation, streamflow, etc.) are important in modeling studies. For instance, if an autoregressive (AR) type model is to be applied to the observed time series of the variable of interest, it is required to remove any trend before the model is applied. Similarly, any time series discontinuities (abrupt shifts, jumps) should be removed prior to modeling. AR type models (and many others) assume that the data follow a prespecified probability distribution function, such as the normal distribution. In

A. Gedikli · H. Aksoy · N. Erdem Unal
Department of Civil Engineering, Istanbul Technical University, 34469 Maslak, Istanbul, Turkey

Present Address:
H. Aksoy (✉)
Fakultät III, Umwelt und Technik, Hydrologie und Wasserwirtschaft, Leuphana Universität Lüneburg, Herbert-Meyer-Str. 7, 29556 Suderburg, Germany
e-mail: haksoy@itu.edu.tr

A. Kehagias
School of Engineering, Aristotle University of Thessaloniki, 541 24 Thessaloniki, Greece

practice, information on the structural characteristics of precipitation data might signal for climate change or variability. Such information can be extracted from the trend analysis of the precipitation record (Dahamsheh and Aksoy 2007).

Analysis of streamflow characteristics can help in understanding possible effects of manmade or natural short- or long-term changes. For example; any change in the physical conditions of the measuring system causes shifts in the time series of the process analyzed. Another example is from water resources engineering; an abrupt change in the river cross-section due to a major flood is likely to cause a permanent shift in the discharge time series of this particular cross-section. In such cases, the stage-discharge relationship of the cross-section changes due to erosion and sedimentation (Tsakalias and Koutsoyiannis 1999). Such information can be extracted from the jump analysis of the streamflow record. Jump analysis of a time series is closely related to the segmentation of the time series.

In this paper, we study the offline time series segmentation problem. A given time series must be divided into several segments (i.e. blocks of contiguous data) so that each segment is homogeneous, while contiguous segments are heterogeneous; homogeneity and heterogeneity are defined in terms of some appropriate segment statistics. The problem has received considerable attention in hydrological literature. As a result, the development of fast and efficient segmentation algorithms emerges as a practically significant problem. An early landmark study on hydrological time series segmentation was made by Buishand (1982), and an extensive bibliography on segmentation methods was presented in Basseville and Nikiforov (1993). Hubert et al. (1989) worked on the segmentation of hydrometeorological time series with a continuous effort (Hubert 2000; Labbé et al. 2004; Hubert et al. 2007). Motivated by the pioneering work of Hubert (2000), Kehagias et al. (2006) used the dynamic programming (DP) optimization algorithm for segmentation of hydrological and environmental time series from the real world which then become motivation for this study. Some alternative approaches, for instance the Bayesian Markov Chain Monte Carlo (MCMC) approach of Fortin et al. (2004) and Kehagias and Fortin (2006) and the hidden Markov model (HMM) approach of Kehagias (2004) and Kehagias and Fortin (2006) should be noted.

The focus of the current paper is on *multiple segments*, *offline* segmentation. A key advance introduced in the current paper is the modification of the DP algorithm by the remaining cost concept of Gedikli et al. (2008). The unique offer of this study is the modification of the DP algorithm. The DP and its modified version yield optimal segmentations (in terms of a well defined segmentation criterion) and can segment long time series of over one thousand items in a few seconds. Both algorithms are evaluated on both real world (hydrometeorological) and artificial time series.

The paper is organized as follows. In the following section, definitions about and general formulation of the segmentation problem are presented, which will be used in subsequent sections. The DP segmentation algorithm and its modified version (mDP) are then presented. After the issue of optimal segmentation order is discussed, the algorithms are evaluated (on both artificial and real world hydrometeorological time series) and compared to each other. Finally, conclusions are drawn and future studies foreseen are presented.

2 Definitions and formulation of the problem

The aim in the segmentation process is to determine time points where changes are observed in the time series characteristics. These time points are called *change points*; the interval included between two change points is defined as a *segment* (of the time series); and the procedure by which the segments of a time series are determined is named "*time series segmentation*". In this study, an *offline* segmentation (see definition below) algorithm is presented, which is based on the DP optimization technique (Kehagias et al. 2006); also a modified version (mDP) is developed, which is based on the remaining cost concept of Gedikli et al. (2008). In the offline segmentation an entire time series $(x_1,x_2,\ldots,x_T)$ is given and must be divided into segments. In the online segmentation, on the other hand, the data points $(x_1,x_2,\ldots,x_t,\ldots)$ arrive one at a time and, at every time step t, it is required to decide whether x_t belongs to the previous segment or assigned to a new segment which starts at t (Dobigeon and Tourneret 2007).

Assume that the time series $\mathbf{x} = (x_1,x_2,\ldots,x_T)$ is given. The segmentation can be described by a sequence $\mathbf{t} = (t_0,t_1,\ldots,t_K)$ satisfying $0 = t_0 < t_1 < \cdots < t_{K-1} < t_K = T$. The intervals of integers $[t_0 + 1,t_1]$, $[t_1 + 1,\ldots,t_2],\ldots,$ $[t_{K-1} + 1,t_K]$ are called *segments*, the times $t_0,t_1,\ldots,t_K$ are called *segment boundaries* and K, the number of segments, is called the *order of the segmentation*. The set of all segmentations of $\{1, 2,\ldots,T\}$ is denoted by $\mathbf{T}$ and the set of all segmentations of order K by $\mathbf{T}_K$. Clearly, $\mathbf{T} = \cup_{K=1}^{T}\mathbf{T}_K$. The number of all possible segmentations of $\{1,2,\ldots,T\}$ is 2^{T-1}.

Offline segmentation can be formulated as an optimization problem. The segmentation cost $J(\mathbf{t})$ is defined by

$$J(\mathbf{t}) = \sum_{k=1}^{K} d_{t_{k-1}+1,t_k} \tag{1}$$

where $d_{s,t}$ (for $0 \le s < t \le T$) is the segment error corresponding to segment $[s,t]$. The segment error depends on the data vector $\mathbf{x} = (x_s,x_{s+1},\ldots,x_t)$. A variety of $d_{s,t}$ functions can be used. In this study,

$$d_{s,t} = \sum_{\tau=s}^{t} \left(x_\tau - \mu_{s,t}\right)^2 \tag{2}$$

is used in which the segment-mean is given by

$$\mu_{s,t} = \frac{\sum_{\tau=s}^{t} x_\tau}{t-s+1} \tag{3}$$

The optimal segmentation, denoted by $\widehat{\mathbf{t}} = (\widehat{t}_0, \widehat{t}_1, \ldots, \widehat{t}_K)$ is defined as $\widehat{\mathbf{t}} = \arg\min_{\mathbf{t}\in\mathbf{T}} J(\mathbf{t})$ and the optimal segmentation of order K, denoted by $\widehat{\mathbf{t}}^{(K)} = \left(\widehat{t}_0^{(K)}, \widehat{t}_1^{(K)}, \ldots, \widehat{t}_K^{(K)}\right)$, is defined as $\widehat{\mathbf{t}}^{(K)} = \arg\min_{\mathbf{t}\in\mathbf{T}_K} J(\mathbf{t})$. The optimal segmentation can be found by exhaustive enumeration of all possible segmentations (and computation of the corresponding $d_{s,t}$). However, this is computationally infeasible, because the total number of segmentations increases exponentially in T. Hubert (2000) uses a branch-and-bound approach to search efficiently the set of all possible segmentations and states that this approach "currently" (in 2000) can segment time series with several tens of terms but is not able "… to tackle series of much more than a hundred terms…" because of the combinatorial increase of computational burden. In Sects. 3 and 4 below, algorithms which can segment time series with hundreds of terms in a few seconds are presented.

In order to obtain these *fast* algorithms, it will be useful to develop a fast method for computing the costs $d_{s,t}$. The recursive formulation of

$$d_{s,t+1} = d_{s,t} + (t-s+1)\left(\mu_{s,t} - \mu_{s,t+1}\right)^2 + \left(x_{t+1} - \mu_{s,t+1}\right)^2 \tag{4}$$

where

$$\mu_{s,t+1} = \frac{(t-s+1)\mu_{s,t} + x_{t+1}}{t-s+2}. \tag{5}$$

is easily proved as a special case of the results in (Kehagias et al. 2006).

3 The DP algorithm

The DP segmentation algorithm of Kehagias et al. (2006) computes efficiently the optimal segmentation of order k for $k = 1,2,\ldots K$ in the following manner.

Consider the optimal segmentation $(t_1,t_2,\ldots,t_k)$ of $(x_1,x_2,\ldots,x_t)$ which contains k segments and suppose its last segment is $[s+1,t]$. Then the first $k-1$ segments form an optimal segmentation $(t_1,t_2,\ldots,t_{k-1})$ of $(x_1,x_2,\ldots,x_s)$. More specifically, if c_t^k is the minimum segmentation cost of $(x_1,x_2,\ldots,x_t)$ into k segments then

$$c_t^k = c_s^{k-1} + d_{s+1,t} \tag{6}$$

is satisfied. Equation 6 allows the use of a typical DP approach to compute the optimal costs and the corresponding optimal segmentations, as illustrated in the Dynamic Programming pseudocode in Appendix A. The algorithm is based on standard dynamic programming arguments and should be clear to the reader; let us only note that the variable z_t^K denotes the optimal Kth segment break of the sub-time series $(x_1,x_2,\ldots,x_t)$ (and $z_0^1 = z_0^2 = \cdots = z_0^K = 0$, corresponding to a fictitious segment preceding the first actual segment $[1, t_1]$ in segmentation of every order).

On termination, the algorithm has computed the optimal segmentation cost $c_T^K = J(K) = \min_{\mathbf{t}\in T_K} J(\mathbf{t})$ and, by backtracking, the optimal segmentation $\widehat{\mathbf{t}}^{(K)} = \left(\widehat{t}_0^{(K)}, \widehat{t}_1^{(K)}, \ldots, \widehat{t}_K^{(K)}\right)$; these quantities have been computed for $K = 1,2,\ldots,K_{\max}$, in other words a *sequence* of minimization problems has been solved recursively.[1]

4 The modified DP algorithm

As suggested by Hubert (2000), the upper bound, u, of the kth segment in the Kth order segmentation can trivially be given as

$$t_k \le u = T - K + k \tag{7}$$

The easiest but the most time-consuming way to determine optimal segmentations of any order from $K = 2$ to $T-1$ requires 2^{T-1} computational loops and it is therefore not effective in obtaining all optimal segmentations. In this way, the loops are always completed from $K = 2$ to $T-1$ and then a comparison and update is made to minimize the cost which initially is taken equal to $d_{1,T}$. This also means that the cost $c_{t_k}^k$ (where $t < T$) of any kth-order segmentation of the first t_k elements, is not considered. By taking this cost into account and also reducing the upper bound of segments as defined in Eq. 7, a more efficient way is obtained to further eliminate segmentations. For this purpose, it is easy to check that

$$c_{t+1}^k \ge c_t^k \ge (c_t^{k+1} \text{ and } c_{t+1}^{k+1}) \tag{8}$$

[1] The recursive solution of an entire sequence of minimization problems makes the DP algorithm very attractive for online operation; the same feature, however, is the main difficulty in converting the algorithm to online operation. Indeed, the two inner loops in the Minimization section of the algorithm show that, if a new datum x_{T+1} is added to the time series, the costs $e_{s,t}$ must be recomputed for every pair (s,t) with $s < t \le T+1$. Hence, for every new datum $T+1$ additional computation must be performed; as T (i.e., the length of the time series) increases the amount of computation required becomes prohibitive, especially for online operation.

is valid for $t = 2,\ldots,T-1$ and $k = 1,\ldots,t$. A detailed derivation of Eq. 8 can be found in Gedikli et al. (2008), where four lemmas—one with proof—are given. In addition to Eq. 8, it is also known that any k sequential segments extracted from the optimal segmentation are also optimal; i.e., if the cost of the optimal segmentation is $J(\hat{\mathbf{t}})$, then the cost $J(\hat{\mathbf{t}}_k)$ with change points $\mathbf{t}_k = (t_0,t_1,\ldots,t_k)$ also satisfies the optimality condition. It then becomes clear that a kth-order segmentation of $(x_1,x_2,\ldots,x_t)$ with cost $c_t^{k-1} > c_T^K$ cannot be optimal (Gedikli et al. 2008).

In order to reduce the upper bound u in this way, the *remaining cost concept* is defined by Gedikli et al. (2008) as

$$R_{T,t}^{K,k} = c_T^K - c_t^k \tag{9}$$

where $k \le K$ and $t \le T$. Considering Eq. 8, the reduced upper bound of the kth segment, e, can be obtained as the largest integer satisfying

$$s \le e \le T - K + k \tag{10}$$

and

$$d_{s,e} \le R_{T,s-1}^{K,k-1} \tag{11}$$

where s is the starting point of the kth segment.[2] Based upon Eq. 11, it is seen that the cost of the kth segment must be less than or equal to the remaining cost. When Eqs. 9 and 11 are combined, it is noted, for $k = 1$, that Eq. 11 takes the form of

$$d_{1,e} \le c_T^K \tag{12}$$

since $R_{T,0}^{K,0} = c_T^K - c_0^0$ and $c_0^0 = 0$, therefore

$$R_{T,0}^{K,0} = c_T^K \tag{13}$$

is obtained. Considering the kth-order segmentation of the subseries made of the first r items, and using Eq. 11

$$d_{s,r} \le R_{e,s-1}^{k,k-1} \tag{14}$$

can be written and hence a new upper bound, r, satisfying

$$s \le r \le e \tag{15}$$

can be obtained.

The above detailed remaining cost concept of the AUG segmentation algorithm (Gedikli et al. 2008) was coupled to the DP algorithm developed by Kehagias et al. (2006) as in the pseudocode in Appendix B to obtain the modified DP segmentation algorithm which is the unique original method developed in this study.

[2] Note that here too the assumption of offline segmentation is crucial. Namely, to implement the reduction of upper bound u is possible only if the costs $d_{s,t}$ are known for every value of s,t and, in particular, for $t = T$, which implies that the final time T is known; this would not be the case for online segmentation.

5 The optimal segmentation

Both the DP algorithm and its modified version (mDP) compute a *sequence* of optimal segmentations; $\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2, \ldots, \hat{\mathbf{t}}_K$ where $\hat{\mathbf{t}}_k$ is the kth-order optimal segmentation. Determining the *optimal order* of segmentation; i.e. selecting the number of segments, is however a subsequent step in the segmentation procedure to be performed for which the *Scheffe's hypothesis test* (Scheffe 1959), which, in short, is a very general multiple means comparison test, was employed in this study. For a given segmentation ($\hat{\mathbf{t}}_k$ for instance), the hypothesis that the means of consecutive segments are significantly different is tested. The test was applied on the optimal segmentations $\hat{\mathbf{t}}^{(1)}, \hat{\mathbf{t}}^{(2)}, \ldots, \hat{\mathbf{t}}^{(K)}$. If $\widehat{\mathbf{t}}^{(k+1)}$ is the *first lowest order* segmentation which is rejected by the Scheffe test (i.e. the first segmentation for which at least two consecutive segments do not show a statistically significant difference in their means), then $\widehat{\mathbf{t}}^{(k)}$ is accepted as the optimal segmentation in Hubert (2000). In this study, the optimal segmentation order is selected differently. The segmentation process does not stop as soon as a rejection was decided by the Scheffe test but continues to search for higher order segmentations; because it is possible to have a higher order segmentation that can pass the Scheffe test and hence be accepted as the optimal segmentation. In other words, not the first lowest but the highest order segmentation which is accepted by the Scheffe test is considered instead.

6 Experiments

In this section, using several real-world and artificial data sets, the performance of the DP and mDP algorithms is studied. Previously used by Aksoy et al. (2008b) the real data sets (annual total precipitation data at Fortaleza, Brazil and the minimum water level data of the River Nile) have been again used in this study, since series of length in the order of hundred or even thousand years are very rare and expensive to construct. A longer time series (precipitation data of Nevada) was added in this study; because validation on real-world data is crucial for the applicability of the proposed algorithm. Also three artificial data sets were used. The lengths of the artificial data has been selected as 100, 1,000 and 8,000. Details of the generation of the artificial data are provided in Sect. 6.2.

In all experiments presented here the DP and mDP algorithms have obtained identical segmentations of all orders. Therefore, experimental results are presented in a single table which contains the segmentations of all orders up to the highest order accepted by the Scheffe test. In the artificial data case, also the change points of the true segmentation are provided. Results of the experiments

Table 1 Change points in the optimal segmentations of the Fortaleza annual total precipitation data (1849–1979), for orders $k = 2$–4

k	Change points				
2	1848	1962	1979		
3	1848	1949	1960	1979	
4	1848	1893	1897	1962	1979

performed using hydrometeorological time series of precipitation, temperature, streamflow and river water level are given below together with a short analysis that does not concentrate on hydrological, meteorological or climatological process itself but only on how the DP and mDP algorithms perform time series segmentation as computational tools.

6.1 Real data experiments

Annual total precipitation data (in mm) at Fortaleza, Brazil, has a length of 131 years for the period 1849–1979. The time series has been previously presented in Morettin et al. (1987). The DP and mDP algorithms were applied on the time series. The segmentations obtained are listed in Table 1 up to the fourth-order, which is the highest order segmentation accepted by the Scheffe test. In Fig. 1, both the original time series and segment means corresponding to the fourth-order segmentation are plotted together with long-term average. When Fig. 1 is analyzed, it is seen that the annual precipitation ranges approximately from 500 to 2,500 mm. However, for the last segment, the minimum annual precipitation remains higher than 1,000 mm; which is the reason for having this upward shift in the time series mean.

The DP and mDP algorithms were also applied on the time series of minimum water level data of the River Nile for the years 622–1918. These data can also be found in Hipel and McLeod (1994). It has been previously used, among others, in Kehagias (2004), Kehagias et al. (2006, 2007), Aksoy et al. (2007, 2008b), and Gedikli et al. (2008). The segmentations obtained are listed in Table 2 up to order 16, which is again the highest order segmentation accepted by the Scheffe test.[3] In Fig. 2, the 16th-order segmentation of the time series is plotted together with the original time series and its long-term average. As previously observed by Gedikli et al. (2008), a very long stable period was located for a period of 294 years starting very early in the ninth century. A segment of constant values was discovered by both algorithms starting with 1528, which can be considered a kind of verification that the algorithms work properly. This data set can be considered a "hockey-stick" graph as an increasing trend is observed in last decades of the time series.

Next, the DP and mDP algorithms were applied on the time series of raw precipitation data of Nevada, a long time series, consisting of 7,996 annual data points extending back to 6000 BC. These data were obtained using the multi-millennial-length tree ring chronology network available in the World Data Center (WDC) for Paleoclimatology in the US (Hughes and Graumlich 1996). The segmentations obtained by the DP and mDP algorithms are listed in Table 3 up to order 6, the highest order segmentation accepted by the Scheffe test. In Fig. 3, the 6th-order segmentation of the time series is plotted together with the original time series and its long-term average. It can be seen that precipitation in Nevada looked stable around its mean value when a few short fluctuations were ignored.

6.2 Artificial data experiments

Next the mDP and DP algorithms were applied on three *artificial* time series. The advantage of using an artificial time series is that its true segmentation is known and hence an evaluation can be made on how efficient the segmentation algorithms are in approaching the true change points.

The artificial time series were generated by the following procedure.

1. The length T of the time series is selected (In this study, $T = 100$, 1,000 and 8,000 were selected).
2. Then K segment lengths l_k (for $k = 1,2,\ldots,K$) are generated, following a normal distribution with mean μ_L and standard deviation σ_L, and rounding off fractional lengths to the closest integer. For each time series K is chosen large enough that $l_1 + l_2 + \cdots + l_K \geq T$. The relationship of the lengths l_k to the change points t_k is that $t_1 = l_1$, $t_2 = l_1 + l_2, \ldots, t_K = l_1 + \ldots + l_K$.
3. For $k = 1,2,\ldots,K$ we choose a mean value μ_k with uniform probability from the set $\{1,2,\ldots,6\}$.
4. Next, the values x_t are generated for $t = 1,2,\ldots,l_1 + l_2 + \cdots + l_K$. For the k-th segment (i.e., for $t \in [l_{k-1} + 1, l_k]$) x_t is chosen from a normal distribution with mean μ_k and standard deviation σ (note that each segment has a different mean, but all have the same standard deviation). This results in a time series which has length *at least* T.
5. Finally, the time series is truncated by dropping all x_t values with $t > T$.

Note that this procedure generates a time series of length *exactly* T but with a random number of segments. By using this procedure three time series, characterized by the parameters listed in Table 4 were generated. The time

[3] In previous applications of this data set (Aksoy et al. 2007; Gedikli et al. 2008), the optimal segmentation was mistakenly printed as 14 instead of 16.

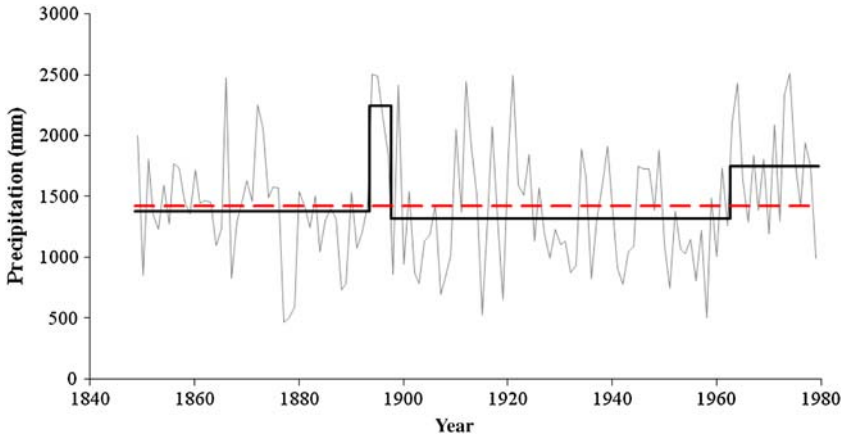

Fig. 1 Segmentation of the Fortaleza annual total precipitation data (1849–1979) for $k = 4$

Table 2 Change points in the optimal segmentations of the minimum water level data of the River Nile (622–1918), for orders $k = 2$–16

k	Change points																
2	621	1857	1918														
3	621	1527	1583	1918													
4	621	1527	1583	1857	1918												
5	621	1426	1527	1583	1857	1918											
6	621	1017	1428	1527	1583	1857	1918										
7	621	1081	1196	1426	1527	1583	1857	1918									
8	621	1081	1196	1426	1527	1583	1836	1887	1918								
9	621	731	804	1081	1196	1426	1527	1583	1857	1918							
10	621	731	804	1081	1196	1426	1527	1583	1836	1887	1918						
11	621	731	804	1098	1131	1196	1426	1527	1583	1836	1887	1918					
12	621	731	804	1098	1131	1196	1426	1527	1583	1619	1836	1887	1918				
13	621	731	804	1098	1131	1196	1353	1396	1426	1527	1583	1836	1887	1918			
14	621	731	804	1098	1131	1196	1353	1396	1426	1527	1583	1619	1836	1887	1918		
15[a]	621	731	804	1098	1131	1196	1356	1357	1396	1426	1527	1583	1619	1836	1887	1918	
16	621	731	804	1098	1131	1196	1353	1396	1426	1527	1583	1619	1798	1822	1857	1889	1918

[a] Rejected by the Scheffe test

series (along with their segmentations) are plotted in Figs. 4, 5, 6.

Both DP and mDP algorithms are applied to these time series. The segmentations obtained are listed in Tables 5, 6, 7 (for time series no. 1–3, respectively) and the highest order segmentation passing the Scheffe test is also plotted in Figs. 4, 5, 6 (for time series no. 1–3, respectively). It can be seen that in all cases, both the DP and mDP algorithm (in combination with the Scheffe test) have determined the correct number of segments. In addition the overwhelming proportion of time series data has been placed in the true segment. This can be seen in Figs. 4, 5, 6 and also be stated more precisely as follows. For every time step ($t = 1,2,\ldots,T$) define z_t to be the number of the true segment and $\widehat{z}_t$ to be the number of the *estimated* segment to which x_t belongs; then define u_t to be the *error indicator*, i.e.,

$$u_t = \begin{cases} 1 & \text{if } z_t = \widehat{z}_t \\ 0 & \text{else} \end{cases} \tag{16}$$

Finally, define *segmentation accuracy* c as

$$c = 1 - \frac{\sum_{t=1}^{T} u_t}{T}. \tag{17}$$

In other words, segmentation accuracy is 1 minus the fraction of data points which have been misclassified. The c values for the three artificial time series are listed in Table 8 and are always very close to 1, indicating the high accuracy of the segmentations achieved.

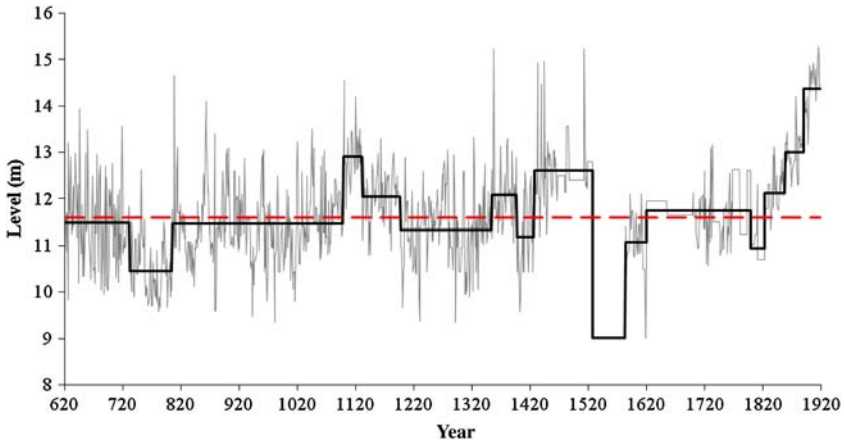

Fig. 2 Segmentation of the minimum water level data of the River Nile (622–1918) for $k = 16$

Table 3 Change points in the optimal segmentations of Nevada precipitation data for orders $k = 2$–6

k	Change points						
2	−6001	−5956	1995				
3	−6001	−5997	−5956	1995			
4	−6001	−5956	−852	−821	1995		
5	−6001	−852	−821	−267	359	1995	
6	−6001	−5956	−852	−821	−267	359	1995

Table 4 The characteristics of the three artificial time series: T (Length), μ_L (average segment length), σ_L (standard deviation of segment length), K (number of segments), μ_k (average value in k-th segment), σ (standard deviation of noise—same for all segments)

Time series	T	μ_L	σ_L	K	μ_k	σ
1	100	25	4	4	(1, 4, 1, 6)	3
2	1000	200	40	5	(3, 5, 2, 6, 4)	4
3	8000	800	40	10	(2, 5, 3, 6, 4, 6, 5, 3, 5, 3)	4

6.3 Comparison

Based on the experimental results detailed above, both the DP and mDP algorithms are seen to minimize, in an exact manner (without approximation), the segmentation cost defined by Eqs. 1–3. Since the minimization is exact, both algorithms naturally give the same results. Hence the only comparison possible between the two algorithms is in terms of the execution time. This is listed in Table 8, where it can be seen that both algorithms (mDP in particular) are very fast and segment time series with even thousands of terms (on which the algorithm of Hubert (2000) does not terminate). It can also be seen that the mDP is always faster than the DP algorithm and this becomes especially obvious in the long time series. For example, for the Nevada time series, execution time is 42 min 41 s for the DP and 21 min 38 s for mDP; for artificial time series no. 3 the respective

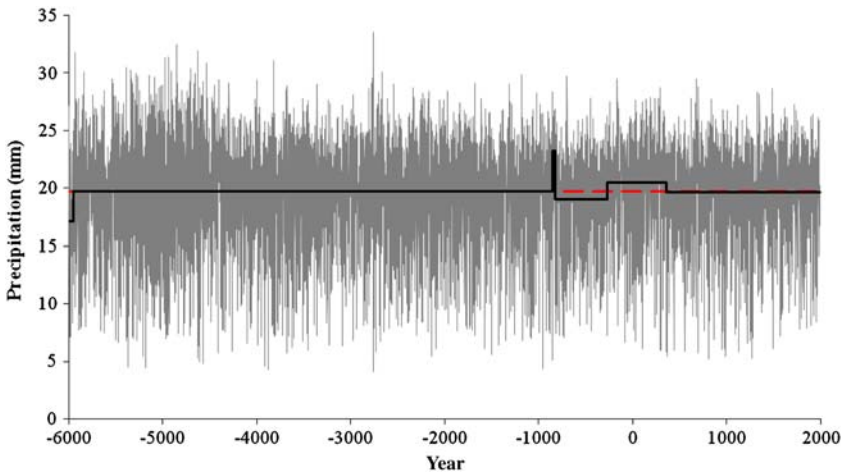

Fig. 3 Segmentation of the Nevada precipitation data for $k = 6$

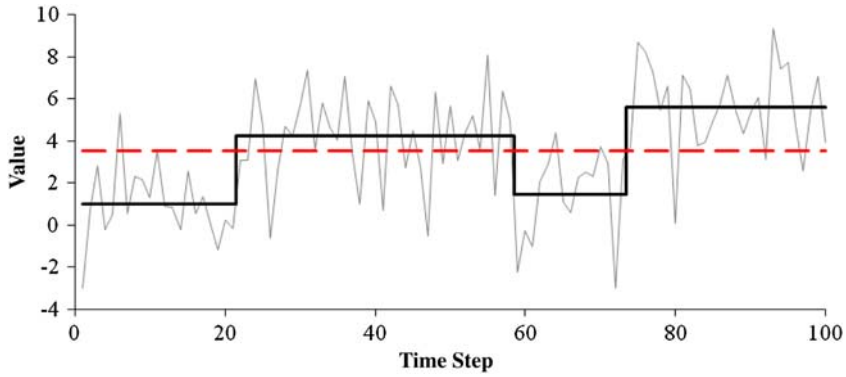

Fig. 4 Segmentation of the artificial time series No. 1 for $k = 4$

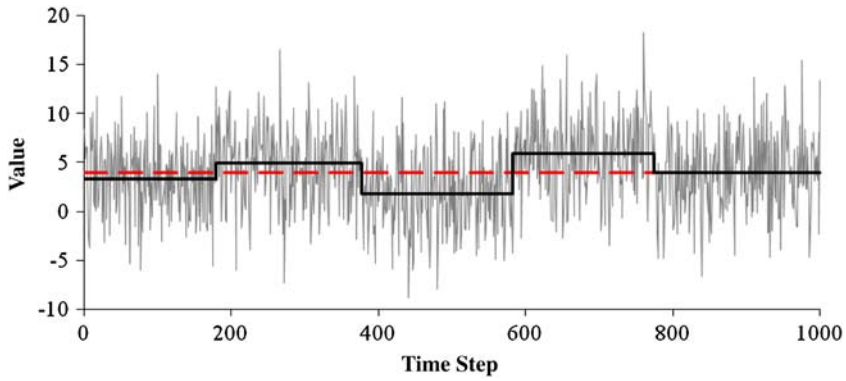

Fig. 5 Segmentation of the artificial time series No. 2 for $k = 5$

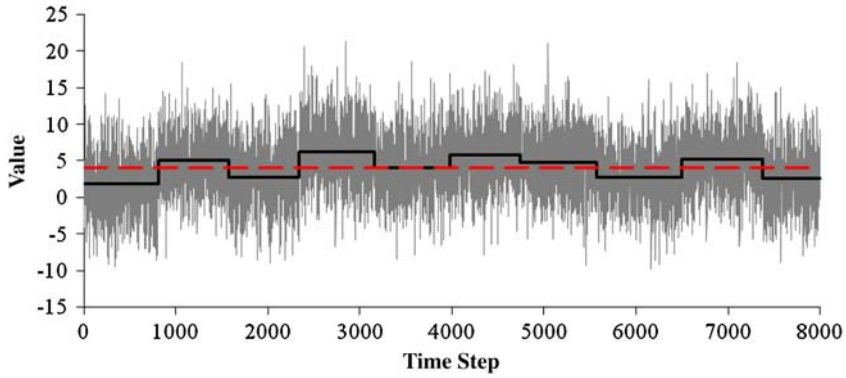

Fig. 6 Segmentation of the artificial time series No. 3 for $k = 10$

times are 50 min 01 s and 25 min 04 s, respectively.[4] These results show the superiority of the mDP algorithm, namely that it can segment long time series significantly faster than the DP algorithm.

Looking at the artificial time series (Table 8) for which the true segmentations are known it can also be seen that both DP and mDP achieve excellent (over 0.95) segmentation accuracy. Finally, it is worth mentioning that the algorithms defined in this study have been presented in user friendly software and applied on long time series by Gedikli et al. (2009).

Table 5 Change points in the optimal segmentations of the artificial time series No. 1 for orders $k = 2$–4 and change points in the true segmentation

k	Change points				
2	0	74	100		
3	0	21	74	100	
4	0	21	58	73	100
True segmentation	0	22	55	74	100

[4] All experiments were performed by running a Microsoft Visual Studio 2005 C# implementation of DP and mDP. The executable was run on a Windows PC with HT processor running at 3.00 GHz (CPU) and 2 GB memory (RAM).

Table 6 Change points in the optimal segmentations of the artificial time series No. 2 for orders $k = 2$–5 and change points in the true segmentation

k	Change points					
2	0	584	1000			
3	0	378	583	1000		
4	0	378	583	775	1000	
5	0	179	378	583	775	1000
True segmentation	0	178	379	584	777	1000

7 Conclusion

Using the remaining cost concept (or the upper bound reduction), the DP segmentation approach is changed to obtain the modified DP segmentation algorithm. The remaining cost concept is used to determine and eliminate (with minimal computation) a number of segmentations which do not satisfy the minimum cost condition. The DP algorithm is made faster after the upper bound reduction is used, i.e. the remaining cost concept of the AUG segmentation algorithm is incorporated into the DP algorithm. Efforts to link the AUG segmentation algorithm with the DP and mDP algorithms as a set of programs in a user-friendly interface are in progress.

Acknowledgments The authors thank the IAHS Secretary General Dr. Pierre Hubert of Université P. & M. Curie, Paris, France, for sharing his software of automatic segmentation algorithm online. The user-friendly version of the algorithms can be supplied to those who show interest and make a request to the authors. This manuscript has been submitted when the second author (H. Aksoy) was working at Leuphana Universität Lüneburg, Campus Suderburg in Germany as an experienced researcher invited by the Alexander von Humboldt Foundation of Germany.

Table 7 Change points in the optimal segmentations of the artificial time series No. 3 for orders $k = 2$–10 and change points in the true segmentation

k	Change points										
2	0	806	8000								
3	0	806	5569	8000							
4	0	796	2336	5569	8000						
5	0	806	1573	2336	5569	8000					
6	0	796	2336	5569	6499	7374	8000				
7	0	806	1573	2336	5569	6499	7374	8000			
8	0	806	1573	2336	3081	5569	6499	7374	8000		
9	0	806	1573	2336	3161	3978	5569	6499	7374	8000	
10	0	806	1573	2336	3161	3978	4742	5569	6499	7374	8000
True segmentation	0	808	1579	2339	3157	3978	4744	5563	6494	7377	8000

Table 8 Segmentation accuracy for artificial TS (time series) and execution time (h:min:s) required for the DP and mDP algorithms

Experiment	Length (years)	Segmentation accuracy c	Computer run time (h:m:s)	
			DP	mDP
Fortaleza	131	Not applicable	00:00:00.094	00:00:00.094
Nile	1297	Not applicable	00:00:10.531	00:00:02.594
Nevada	7996	Not applicable	00:42:41.328	00:21:38.547
Artificial TS No. 1	100	0.950	00:00:00.063	00:00:00.063
Artificial TS No. 2	1000	0.995	00:00:04.980	00:00:02.938
Artificial TS No. 3	8000	0.996	00:50:01.000	00:25:04.063

All times refer to running a Microsoft Visual Studio 2005 C# implementation of DP and mDP on a Windows PC with HT processor running at 3.00 GHz (CPU) and 2 GB memory (RAM)

Appendix A (Dynamic Programming)

Dynamic Programming
Input
The time series $\mathbf{x}=(x_1,x_2,...,x_T)$
The errors $d_{s,t}$ $(0\le s<t\le T)$
The max number of segments K_{max}

Initialization
$c_0^1=0$
For t = 1 to T
$c_t^1=d_{1,t}$
$z_t^1=0$
Next t

Minimization
For K = 2 to K_{max}
$c_0^K=0$
For t = 1 to T
For s = 1 to $t-1$
$e_{s,t}=c_s^{K-1}+d_{s+1,t}$
Next s
$c_t^K=\min_{1\le s\le t-1}(e_{s,t})$
$z_t^K=\arg\min_{1\le s\le t-1}(e_{s,t})$
Next t
Next K

Backtracking
For K = 1 to K_{max}
$\hat{t}_K^K=T$
For k = K to 1
$\hat{t}_{k-1}^K=z_{\hat{t}_k^{(K)}}^k$
Next k
Next K

Appendix B (Modified Dynamic Programming)

Modified Dynamic Programming
mDP()
For K = 2 to T
$\hat{t}_K^K=T$
update(K, T, c)
For s = K to T
R = $c_K^T-c_{K-1}^{s-1}$
e = reduce(s, T, R, d)
For t = s to e
$C=c_{s-1}^{K-1}+d_{s,t}$
if $C\le c_t^K$ then
$\hat{t}_t^K=s-1$
$c_t^K=C$
endif
Next t
Next s
Next K
Bactracking
For K = 2 to T
q = T
For t = K - 1 to 0 by step -1
$q=\hat{t}_q^{t+1}$
$\hat{t}_t^K=q$
Next t
Next K
return

Upper bound reduction
reduce(s, T, R, d)
P = s
e = T
While (e > p + 1)
q = (e + p) / 2; *Round down to the nearest integer*
If $d_{s,j}>R$ then e = q else p = q
Endwhile
return e

Global update of the cost matrix (by lemmas)
update(K, T, c)
For k = K + 1 to T
if $c_k^{K-1}<c_{k-1}^{K-1}$ then $c_{k-1}^{K-1}=c_k^{K-1}$
if $c_{k-1}^{K-1}<c_k^K$ then $c_k^K=c_{k-1}^{K-1}$
Next k
return

References

Aksoy H (2007) Hydrological variability of the European part of Turkey. Iran J Sci Technol Trans B 31(B2):225–236

Aksoy H, Unal NE, Gedikli A (2007) Letter to the editor. Stoch Environ Res Risk Assess 21(4):447–449

Aksoy H, Unal NE, Alexandrov V, Dakova S, Yoon J (2008a) Hydrometeorological analysis of northwestern Turkey with links to climate change. Int J Climatol 28(8):1047–1060

Aksoy H, Gedikli A, Unal NE, Kehagias A (2008b) Fast segmentation algorithms for long hydrometeorological time series. Hydrol Process 22:4600–4608

Basseville M, Nikiforov IV (1993) Detection of abrupt changes: theory and application. PRT Prentice Hall, Englewood Cliffs, NJ

Buishand TA (1982) Some methods for testing the homogeneity of rainfall records. J Hydrol 58:11–27

Dahamsheh A, Aksoy H (2007) Structural characteristics of annual precipitation data in Jordan. Theor Appl Climatol 88(3–4): 201–212

Dobigeon N, Tourneret JY (2007) Joint segmentation of wind speed and direction using a hierarchical model. Comput Stat Data Anal 51:5603–5621

Fortin V, Perreault L, Salas JD (2004) Restropective analysis and forecasting of streamflows using a shifting level models. J Hydrol 296:135–163

Gedikli A, Aksoy H, Unal NE (2008) Segmentation algorithm for long time series analysis. Stoch Environ Res Risk Assess 22(3):291–302

Gedikli A, Aksoy H, Unal NE (2009) AUG-Segmenter: a user-friendly tool for segmentation of long time series. J Hydroinformatics (in press)

Hipel KW, McLeod AI (1994) Time series modelling of water resources and environmental systems. Elsevier, Amsterdam

Hubert P (2000) The segmentation procedure as a tool for discrete modeling of hydrometeorological regimes. Stoch Environ Res Risk Assess 14:297–304

Hubert P, Carbonnel JP, Chaouche A (1989) Segmentation des series hydrométéorologiques—application à des séries de précipitations et de débits de l'afrique de l'ouest. J Hydrol 110(3–4):349–367

Hubert P, Bader J-C, Bendjoudi H (2007) Un siècle de débits annuels du fleuve Sénégal. J Hydrol Sci 52(1):68–73

Hughes MK, Graumlich LJ (1996) Climatic variations and forcing mechanisms of the last 2000 years, vol 141. Multi-millenial dendroclimatic studies from the western United States. NATO ASI Series, pp 109–124

Kehagias A (2004) A hidden Markov model segmentation procedure for hydrological and environmental time series. Stoch Environ Res Risk Assess 18:117–130

Kehagias A, Fortin V (2006) Time series segmentation with shifting means hidden Markov models, *Nonlin*. Process Geophys 13: 339–352

Kehagias A, Nidelkou E, Petridis V (2006) A dynamic programming segmentation procedure for hydrological and environmental time series. Stoch Environ Res Risk Assess 20:77–94

Kehagias A, Petridis V, Nidelkou E (2007) Reply by the authors to the letter by Aksoy et al. Stoch Environ Res Ris Assess 21: 451–455

Koutsoyiannis D (2006) Nonstationarity versus scaling in hydrology. J Hydrol 324:239–254

Kundzewicz ZW, Robson AJ (2004) Change detection in hydrological records—a review of the methodology. J Hydrol Sci 49(1):7–19

Labbé C, Labbé D, Hubert P (2004) Automatic segmentation of texts and corpora. J Quant Linguist 11(3):193–213

Morettin PA, Mesquita AR, Rocha JGC (1987) Rainfall at Fortaleza in Brazil revisited. Time Ser Anal Theory Pract 6:67–85 Editor: O.D. Anderson

Radziejewski M, Kundzewicz ZW (2004) Detectability of changes in hydrological records. J Hydrol Sci 49(1):39–51

Scheffe M (1959) The analysis of variance. Wiley, New York

Tsakalias G, Koutsoyiannis D (1999) A comprehensive system for the exploration and analysis of hydrological data. Water Resour Manag 13:269–302

Xiong L, Guo S (2004) Trend test and change-point detection for the annual discharge series of the Yangtze River at the Yichang hydrological station. J Hydrol Sci 49(1):99–112