This document is published in:

“Andre Ponce de Leon F. de Carvalho. et al. (eds.) (2010) *Distributed Computing and Artificial Intelligence*: *7th International Conference*, (Advances in Intelligent and Soft Computing, 79) Springer, pp. 267- 274.
Doi: http://dx.doi.org/10.1007/978-3-642-14883-5_35**”**

Piecewise Linear Representation Segmentation as a Multiobjective Optimization Problem

José Luis Guerrero, Antonio Berlanga, Jesús García, and José Manuel Molina

Abstract Actual time series exhibit huge amounts of data which require an unaffordable computational load to be processed, leading to approximate representations to aid these processes. Segmentation processes deal with this issue dividing time series into a certain number of segments and approximating those segments with a basic function. Among the most extended segmentation approaches, piecewise linear representation is highlighted due to its simplicity. This work presents an approach based on the formalization of the segmentation process as a multiobjetive optimization problem and the resolution of that problem with an evolutionary algorithm

1 Introduction

Time series (sequences of data having, among other components, a timestamp for each of their points) are of great importance for a wide variety of domains, such as financial [1], medicine [13] of manufacturing applications [7]. In recent years, the fast development of storage and collection technologies has lead to an increasing role of time series in the industry. A clear example can be found in the tracking of stock prices [8], being constantly updated in the different markets all over the world.

The required amount processing for these huge volumes of data is unaffordable, and thus the need for an approximate representation emerges. Time series segmentation is a tool designed to deal with this issue, by means of dimensionality reduction. A segmentation technique basically divides a certain time series into a number of segments and approximates those segments with a basic function. According to the different choices for this function, several segmentation techniques can be defined:

José Luis Guerrero · Antonio Berlanga · Jesús García · José Manuel Molina
Group of Applied Artificial Intelligence (GIAA), Computer Science Department.
University Carlos III of Madrid. Colmenarejo, Spain.
e-mail: {joseluis.guerrero, antonio.berlanga, jesus.garcia, josemanuel.molina}@uc3m.es

Fourier Transforms [1], Wavelets [4], Symbolic Mappings [2], etc. Each of these approaches has its own advantages and handicaps, and also, according to them, has reached a determined level of use.

Among the different segmentation techniques, probably the most extended one is Piecewise Linear Representation (PLR, also named Piecewise Linear Approximation, PLA), [11, 15]. This segmentation technique is highlighted by its ease of use, since the basic function used to approximate the different segments is a linear function. Due to its wide usage, several processes have been designed regarding the result of this segmentation technique, such as fast similarity search [10] or the definition of data mining approaches [12].

Traditionally, the results of the different proposed segmentation techniques were compared according to the final error obtained, regardless of the number of segments required to obtain that error. Recently, this fact has been pointed out [15] and new approaches at least consider this number of segments as a quality metric over the final results. Considering this from an optimization point of view, this means that originally only one objective function was considered (the measured error), but the introduction of the number of segments as an additional objective function has turned this problem into a multi-objective optimization problem (MOOP) [6].

MOOPs are complex problems which require that a set of objective functions (usually in conflict) are optimized (maximized or minimized) jointly. In the PLR segmentation problem defined, the two objective functions are approximation error and number of segments. These objectives are in conflict, since a greater number of segments implies a finer approximation, and thus a lower error value. Both objective functions have to be minimized. Evolutionary algorithms (EAs) have obtained remarkable results applied to MOOPs, being classified as Multi-objective Evolutionary Algorithms (MOEAs) [5].

The objective of this work is to define the proper problem dependent items for the application of a MOEA to the PLR segmentation issue, define the required configuration for the algorithms and compare the obtained results with one of the techniques specifically designed for this purpose: bottom up segmentation. The test set used will be a set of trajectories coming from the Air Traffic Control domain.

The organization of the paper will introduce the segmentation techniques in general and the bottom up technique used in the second section, followed by the required MOEA definition and configuration, performed in the third section. After the definition of the two techniques, their results and comparison are presented, along with the conclusions which these results point to.

2 Piecewise Linear Representation segmentation techniques

Segmentation techniques can be defined as the process which divides a given trajectory into a series of segments and afterwards approximates each of these segments with a given basic function. PLR segmentation techniques in particular, use a piecewise linear model as its basic function.

Several classifications can be made over PLR segmentation techniques, being probably the two most important ones their online or offline nature and the use of linear interpolation or regression functions. Online segmentation techniques perform their time series division processing the data as it is received, being the sliding window one of its most known examples [15]. Offline segmentation techniques, on the other hand, require the trajectory to be complete prior to the application of the technique, allowing them to use global information about its behavior. This characteristic allows them, in general, to obtain better segmentation results, but it also makes their complexity order higher, especially according to the trajectory size. The most extended approaches are top down and bottom up techniques [11].

The use of linear interpolation or regression functions usually depends on the need to obtain continuous piecewise lines. Linear interpolation uses a model defined by the first and last point of the segment, so that contiguous segments will always have one point in common. This characteristic may be a requirement in some domains. Linear regression, on the other hand, obtains the regression line considering all the different points belonging to the segment, obtaining, thus, discontinuous piecewise lines. The overall error of the regression functions is always less than or equal to the one obtained with linear interpolation [11], leading to its usual choice as the approximation function (it must be mentioned, though, that the required complexity for its calculation is also considerably higher). Both techniques can be applied along with any of the previously mentioned segmentation algorithms.

The traditional criteria to determine the quality of a segmentation process [11, 15] are the following:

1. Minimizing the overall representation error (*total_error*)
2. Minimizing the number of segments such that the representation error is less than a certain value (*max_segment_error*)
3. Minimizing the number of segments such that the representation error is less than a certain value (*max_segment_error*)

where *total_error* and *max_segment_error* are user defined parameters for the algorithm. The segmentation problem, seen as a multiobjective optimization problem, can be defined with (1)

$$T = \{\mathbf{x}_k\} \rightarrow S(T) = \{B_m\} \rightarrow B_m = \{\mathbf{x}_k\} j \in [k_{min}...k_{max}] \rightarrow \underset{max}{min} f_{quality}(\{B_m\}) \quad (1)$$

where T is the original trajectory, $\mathbf{x}_k$ are the points belonging to it, *S(T)* is the segmentation process, B_m is a given resultant segment from that process and $f_{quality}(\{B_m\})$ are the quality metrics used. Particularizing this general formulation to the criteria presented, the segmentation problem is defined in (2)

$$S(T) = \{B_m\} \rightarrow B_m = \{\mathbf{x}_k\} j \in [k_{min}...k_{max}] m \in [1...seg_{num}]$$
$$\begin{cases} d(S(T),T) \leq total_{error} \\ \forall m, d(f_{ap}(B_m),B_m) \leq max\_segment\_error \end{cases} \quad (2)$$

where *d(x, y)* is a distance error function between segments x and y, $f_{ap}(x)$ is the approximation function result over segment x (in PLR the resulting line which approximates the data in segment x), and seg_{num} is the number of segments obtained by the applied segmentation algorithm.

Among offline algorithms, the bottom up technique is usually reported to produce the best results [11], so this will be the chosen technique to compare its results with the multi-objective approach presented. The heuristic applied by this technique consists in an initial division of the trajectory into its finest possible set of segments, followed by an iterative merge of these segments until no pair of segments can be merged without obtaining a segment with an error above the *max_segment_error boundary*. An overview of this process is shown in figure 1.

3 Multi Objective Evolutionary Algorithms configuration for the PLR segmentation problem

Multi-objective evolutionary algorithms have reached an enormous expansion in their use in the recent years, helped by their implementation in tools such as PISA [3] or the general metaheuristics environment PARADISEO [14], which allow the user to focus on his particular problem. This section will cover the problem dependent issues which must be implemented under these tools in order to resolve the PLR problem.

The PLR segmentation problem requires a codification which allows expressing a variable number of segments (with values ranging from one to the series length) represented by the position of their boundaries in the time series. According to these boundaries the chosen codification was a vector of integer values (which represent the number of a measurement in the time series) with a length of the number of measurements in the time series, n, minus two. These values represent the intermediate

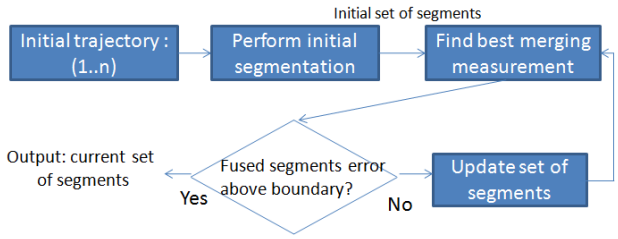

Fig. 1 Overview of the different operations in the bottom up segmentation algorithm

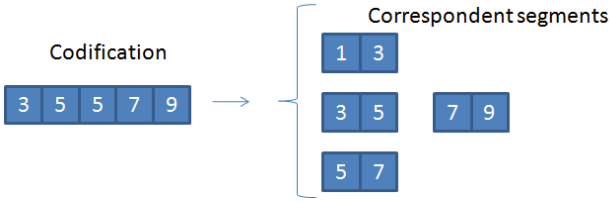

Fig. 2 Overview of the different operations in the bottom up segmentation algorithm

segmentation points in the trajectory. This representation must be sorted, ignoring repeated gene values.

The evaluation function is defined according to the codification presented and the two objective functions required for the problem. The distance function used to measure the error is the Euclidean Distance calculating the sum of squares over the least squares regression line. To evaluate a given solution, the algorithm analyzes the codification components sequentially, adding one to the number of segments and the calculated Euclidean distance value to the error whenever it finds a different value, until the maximum possible value is found (n) or the vector ends. This sequential evaluation of the chromosome requires it to be sorted (in order to obtain the output segments of the solution and be able to calculate the objective functions values). This leads to the application of a sorting procedure after chromosome modifications.

The initialization function seeks to introduce the highest possible diversity into the random initial population of the MOEA. According to the genotype, that was approached by means of a random choice of an integer value for every gene. However, as shown in figure 3, this lead to a very poor diversity in the phenotype values (especially regarding the number of segments) which lead to poor final results. To resolve this issue, the following alternative initialization was designed: a certain number of segments are randomly chosen, followed by the random choice of the extreme values for those segments, duplicating the values where necessary to fill the codification vector completely. The results obtained were better, but were highly dependant on the initial boundaries values, leading to a final initialization function which uses one or the other alternative randomly.

The crossover transformation function is a standard crossover with two split points. The mutation function however, presented similar difficulties to the ones introduced in the initialization. The initial choice was to mutate a certain number of genes according to a *gene_mutation_probability* to a random integer value defined by an epsilon percentage (referred to the trajectory length value). This mutation biased the evolution towards those solutions with the highest number of segments. A complementary method was introduced: whenever a gene was chosen to mutate its value, it could either change according to the random mutation exposed or change its value to one of its surrounding genes. This mechanism was used to allow the mutation operator to increase or decrease the number of segments in the mutated chromosome. In practice, this approach obtained more disperse final Pareto fronts than the random mutation, but the evolution was not satisfactory (the results ob-

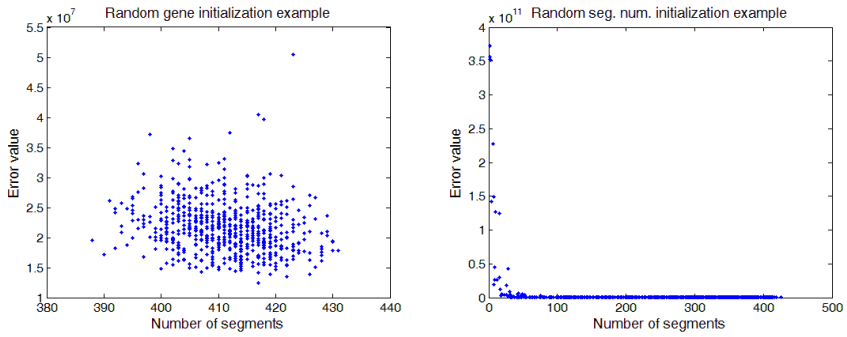

Fig. 3 Comparison of the objective function values obtained for an initial population of size 650 with the proposed initialization methods individually applied

tained were worse than those of the bottom up technique). The final implemented mutation operation applies one or the other mechanism randomly to the whole chromosome (instead of a random application to every individual gene).

Several MOEAs from the PARADISEO framework have been used to test the proposed operations, (their individual results cannot be presented due to space restrictions), obtaining SPEA2 [16] the best results (probably due to its archive use). This will be the chosen algorithm for the final results, with an archive size equal to the time series length minus one (to be able to store, ideally, one solution for every possible number of segments).

4 Experimental validation

The proposed MOEA configuration has been tested on an Air Traffic Control test set similar to the one used in [9]. This test set includes the measurements recorded by sensor devices of different trajectories performed by aircrafts (with an added measuring error). Due to space restrictions, the results for only two of these trajectories can be shown, being the chosen ones a racetrack (the trajectory performed by aircrafts during landing procedures) and a turn trajectory, shown in figure 4

Along with the introduced codification vector, a different one with size *n/2* was also tested, in order to focus in solutions with a smaller number of segments. Table 1 shows the configuration parameters and figure 5 the obtained results. For the bottom up algorithm, the presented front was obtained by a trial process with different *max_segment_error* values, focusing in the search space zone corresponding to a number of segments around 50% of the number of measurements in the time series. The MOEA solution is composed of the non-dominated solutions obtained both in the whole codification and the reduced one. This approach exhibits better results than the bottom up alternative in the whole Pareto front.

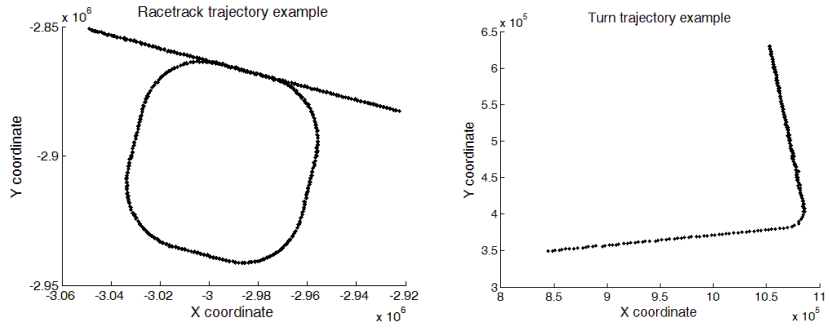

Fig. 4 Trajectories chosen for the application of the techniques exposed

Table 1 Parameter configuration for the MOEA algorithm

Parameter	Value	Parameter	Value
initial population	3000	mutation epsilon	0.2
mut. probability: chrom. / gene	0.3 / 0.01	crossover probability	0.5
reduced codif. iterations	500	complete codif. iterations	1000

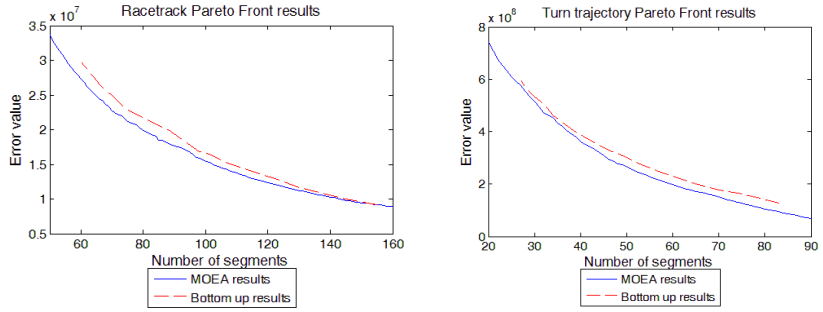

Fig. 5 Trajectories chosen for the application of the techniques exposed

5 Conclusions

Segmentation is a requirement to process the huge amount of data in actual time series. Among the variety of techniques which can be applied for this process, Piecewise Linear Representation is the most extended approach, probably due to its easy implementation. The results presented in this work show that this process can be faced with a Multi objective evolutionary algorithm obtaining better results than a classical offline technique reported to be very accurate: bottom up segmentation. Obviously, due to its computational complexity, these approaches cannot be used as a general segmentation technique, but their results can be useful for the develop-

ment of new heuristics or as a tool for quality assessment. Future lines include the analysis of the results over a wider set of test problems, the comparison with curve approximation algorithms and the optimization of the configuration (for example with the inclusion of a global stopping criteria).

Acknowledgements This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

References

1. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: Proceeding of the 4th Conference of Data Organization and Algorithms, pp. 69–84 (1993)
2. Agrawal, R., Lin, K., Sawhey, H., Shim, K.: Fast similarity search in the presence of noise, scaling and translation in time-series databases. In: Proceedings of 21st International Conference on Very Large Databases, pp. 490–501 (1995)
3. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: Pisa - a platform and programminglanguage independent interface for search algorithms. In: Evolutionary Multi-Criterion Optimization (EMO 2003), pp. 494–508 (2003)
4. Chan, K., Fu, W.: Efficient time series matching by wavelets. In: Proceedings of the 15th International Conference on Data Engineering, pp. 126–133 (1999)
5. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer (2007)
6. Ehrgott, M.: Multicriteria optimizaction. Lecture notes in Economics and Mathematical Systems **491** (1999)
7. Ge, X., Smyth, P.: Segmental semi-markov models for endpoint detection in plasma etching. IEEE Trans. Semiconductor Engineering (2001)
8. Gionis, A., Mannila, H.: Segmentation algorithms for time series and sequence data. Tutorial in SIAM International conference in Data Mining (2005)
9. Guerrero, J., Garcia, J.: Domain transformation for uniform motion identification in air traffic trajectories. In: Advances in Soft Computing, vol. 50, pp. 403–409 (2008)
10. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Journal of Knowledge and Information Systems pp. 263–286 (2001)
11. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Mining in Time Series Databases, chap. Segmenting Time Series: A Survey and Novel Approach, pp. 1–21. World Scientific (2003)
12. Keogh, E., Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining (1998)
13. Koski, A., Juhola, M., Meriste, M.: Syntactic recognition of ecg signals by attributed finite automata. Pattern Recognition pp. 1927–1940 (1995)
14. Liefooghe, A., Jourdan, L., Talbi, E.G.: A unified model for evolutionary multiobjec-tive optimization and its implementation in a general purpose software framework: Paradiseo-moeo. Research report rr-6906, INRIA (2009)
15. Liu, X., Z., L., Wang, H.: Novel online methods for time series segmentation. IEEE Trans. On Knowledge and Data Engineering **20**(12) (2008)
16. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutio-nary approach. In: EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, pp. 95–100 (2002)