Multiobjective Evolutionary Polygonal Approximation for Identifying Crude Oil Reservoirs

José Luis Guerrero
Universidad Carlos III de Madrid
Madrid, Spain
jguerrer@inf.uc3m.es

Luis Martí
Universidade Federal Fluminense
Niterói, Brazil
lmarti@ic.uff.br

Nayat Sanchez-Pi
Universidade do Estado do Rio de Janeiro
Rio de Janeiro, Brazil
nayat@ime.uerj.br

Antonio Berlanga
Universidad Carlos III de Madrid
Madrid, Spain
aberlan@ia.uc3m.es

Jose Manuel Molina
Universidad Carlos III de Madrid
Madrid, Spain
molina@ia.uc3m.es

ABSTRACT

This work formalizes a multi-objective evolutionary approach for the segmentation issue according to Piecewise Linear Representation. It consists in the approximation of a given digital curve by a set of linear models minimizing the representation error and the number of such models. This solution allows the final user to decide from the best array of best found solutions considering the different objectives jointly. The proposed approach eliminates the difficult *a-priori* parameter choices in order to satisfy the user restrictions (the solution choice is performed *a-posteriori*, from the obtained array of solutions) and allows the algorithm to be run a single time (since the whole Pareto front is obtained with a single run and different solutions may be chosen at different times from that Pareto front in order to satisfy different requirements). This solution will be applied to Petroleum Industry in particular to the problem of identifying resources from extraction areas in order to optimize their operational costs and production capacity.

CCS CONCEPTS

• **Information Systems**; • **Applied computing-Operations research**; • **Computing methodologies-Machine learning-Machine learning approaches-Bio-inspired approaches**;

KEYWORDS

polygonal approximation; evolutionary multi-objective optimization; visual information

ACM Reference Format:
José Luis Guerrero, Luis Martí, Nayat Sanchez-Pi, Antonio Berlanga, and Jose Manuel Molina. 2018. Multiobjective Evolutionary Polygonal Approximation for Identifying Crude Oil Reservoirs. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205651.3208290

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208290

1 INTRODUCTION

Digital curves domain, leaded by the importance of human processing and understanding of visual information, established its roots with the psychological studies performed in the middle fifties [2]. One of the main keys to the study of this domain is the representation performed over the original data. The goal of this representation is to cover the main characteristics of a given shape with the least amount of data. This dimensionality reduction performs several objectives. On the one hand, it reduces the storage capacity required for the obtained time series, and, on the other hand, it has an immense impact on the efficiency of the subsequently applied methods, such as feature extraction [24].

Segmentation processes may resort to different representations, being Piecewise Linear Representation (PLR, also named Piecewise Linear Approximation, PLA, or polygonal approximation) among the most extended options. This scope has been deeply analyzed and used according to a data mining perspective [12, 18, 21] and also as a digitization method [22, 32]. Several works have detailed the characteristics of PLR segmentation which have led to its extensive use: simplicity, locality, generality, compactness and ease of use [18, 32]. PLR segmentation is based on the approximation of a curve (or, more generally, a certain time series) T with length n by means of a set of K segments (where $K << n$), approximating each of these segments by a linear model. It can be also described as the process of searching the *dominant points* of a given curve, being these points the edges of the segments in the previous definition.

Polygonal approximation techniques are offline segmentation processes (since they require the whole curve they will be applied to) which can be divided into three different categories: sequential approaches, split and merge approaches and heuristic search approaches. Sequential and split and merge approaches have a strong dependency on the initial steps of their algorithms (either in the form of the starting point for the scanning or the initial segmentation performed). The outcome of these methods is extremely sensible to their segmentation criterion parameters (such as error tolerance), values which may not be easy to determine. On the other hand, heuristic based approaches are computationally expensive, being not guaranteed to be optimal.

Most of the different presented techniques share the lack of a direct mechanism to control the number of segments obtained (and through it, the compression performed over the original data), even though indirect mechanisms may exist (e.g., error tolerance indirectly controls segment length, which along with the number of elements in the original data determines the number of segments in the final representation). Other alternatives, such as evolutionary approaches, allow the choice of the number of segments but lose the control over the approximation error. Comparisons between different algorithms, especially in the data mining domain [18] are usually performed according to the error value obtained by the representation, not considering the cost of that error. Some techniques do take into account the number of segments of the obtained representation (such as in [29], where each cycle tries to obtain the longest possible segments with the lowest possible error value) but, since those objectives are in conflict, it is performed by what, in the multi-objective community, is usually referred to as *a priori* techniques: in order to deal with different objectives in conflict jointly, a decision maker (DM) determines the importance of each of the objectives and, according to that importance, their joint value is calculated and used by underlying algorithms[4].

The previous argumentation introduces segmentation as a multi-objective optimization problem (MOOP, [3]): segmenting a digital curve implies optimizing a set of objective functions in conflict (the considered error of the segmentation and the compression required in order to obtain that error) obtaining values for them which are acceptable to the decision maker [25]. This definition leads to the question of who should play the decision maker role in a segmentation algorithm. Most presented approaches assign this role to the algorithm designer.

Two different segmentations show different values for their objective functions, namely the error function and the number of segments. The suitability of the representation depends on its particular application. Some may require a certain maximum error value, while others, due to their costly processing, may require a number of segments as low as possible. The range of possible processes is huge, from fast similarity search [17] or data mining approaches [19] up to optical character recognition applications [27] or applications in the air traffic control domain [15]. Also, each of these processes may require different priorities for the different objective functions, and these requirements may change over time (e.g, different classifications may be preferred according to different available computer resources). This argumentation leads to the assignment of the decision maker role to the final user of the algorithm, considering as well that this DM may have changing preferences at different instants of time.

Available algorithms generally assume the algorithm designer to be also the DM, performing an *a priori* dealing of the objectives in conflict, usually by means of an aggregating function [35]. This implies that the algorithm designer establishes the importance of the different objectives, and afterwards codifies it into the algorithm running cycle. In other cases, the control over the secondary objective function may be implicit: as explained before, algorithms with a certain error tolerance as one of their input parameters may vary the compression value accordingly to that parameter value. This would imply that, for a scenario where the requirements of the decision maker (the final user) may change over time, the original data would have to be stored and the algorithm rerun with different parameters in order to deal with those different requirements. It is important to highlight that the choice of those parameters in order to meet certain requirements (especially regarding the implicit objective function values) can get to be very difficult to be performed accurately.

The objective of this work is to propose a multi-objective solution based on genetic algorithms for the PLR segmentation problem to cope with the previous requirements: allowing the final user to decide from the best array of best found solutions considering the different objectives jointly (which will constitute the Pareto Front of the problem). The proposed approach eliminates the difficult *a priori* parameter choices in order to satisfy the user restrictions (the solution choice is performed *a posteriori*, from the obtained array of solutions) and allows the algorithm to be run a single time (since the whole Pareto front is obtained with a single run and different solutions may be chosen at different times from that Pareto front in order to satisfy different requirements).

2 OVERVIEW OF SEGMENTATION TECHNIQUES

One of the difficulties of detailing with the state of the art for the segmentation domain are the different naming conventions which similar algorithms receive in the different domains where they are applied [18]. A clear example of these different naming conventions may the Ramer algorithm, [28]. That name is used in the image processing field, while in cartography is known as the Douglas Peucker algorithm [8], or the Iterative End-Point Fits algorithm, usually referred to in the machine learning community [9]. Another commonly used name for this approach is the Top-Down algorithm [18].

The objective of this section is to provide an insight into some different alternatives available in the segmentation domain following the classification provided in the previous introduction section. This description of different algorithms will be used as the basis for the proposal of the multi-objective technique presented in this work, and at the same there provide a considerable understanding of the approaches which have been taken to deal with the segmentation issue. For formalization purposes, we will start defining the components of the given time series with equation 1, where x_i and y_i are the plane coordinates of the point and t_i is the timestamp of the point's reception. If we are dealing with a closed curve without an explicit timestamp, that equation can be adapted following equation 2.

$$t = \{\vec{p_i}\}, \vec{p_i} = (x_i, y_i, t_i), i = 1, \tag{1}$$

$$t = \{\vec{p_i}\}, \vec{p_i} = (x_i, y_i, i), i = 1, \ldots, n \tag{2}$$

Teh and Chin algorithm [36] is based on the concept of the *region of support* [20]: this concept states that each boundary point of a closed curve must have its own view of the curve, being relevant points those which have a meaningful view of the curve which blocks the view of other non-relevant points.

In [36] the proposal is based on the difficulty of determining the curvature for a digital curve, which, in the real Euclidean plane, can be easily defined with equation 3. The functions to determine

discrete curvature are named measures of significance [30]. Three different measures of significance are used: the k cosine measure, the k curvature measure and the *1* curvature measure. The k cosine measure was introduced in [31] and is shown in equation 4. The k curvature measure was introduced in [14] and is shown in equation 5. Finally, the *1* curvature measure is derived from the previous measure (where $k = 1$), and is shown in equation 6.

$$\frac{\frac{d^2y}{dx^2}}{[1 + (\frac{dy}{dx})^2]^{3/2}} \tag{3}$$

$$cos_{ik} = \frac{\vec{a_{ik}} \cdot \vec{b_{ik}}}{|\vec{a_{ik}}||\vec{b_{ik}}|} \tag{4}$$

$$CURik = \frac{1}{k} \sum_{j=-k}^{-1} f_{i-j} - \frac{1}{k} \sum_{j=0}^{k-1} f_{i-j} \tag{5}$$

$$CURi1 = f_{i+1} - f_i \tag{6}$$

Marji and Siy algorithm [22] relies on the concept of *support arms*. This means that they do not use the region of support to calculate a significance measure of the boundary points, but instead compute the strength of the end points of their calculated regions of support, both in clockwise and counterclockwise directions. This strength is determined by the frequency of their choice. The idea is supported on an ideal corner shape, where the corner point would be chosen as an endpoint for all the different points in the shape, and thus, chosen as the relevant point.

To determine both support arms, the function shown in equation 7 is maximized, where L_{jk} is the length of the segment joining points p_j and p_k and E_{jk} is the sum of the squared perpendicular distances of the points contained between p_j and p_k to that segment. This is performed increasing the length of the region until that increase makes the function obtain a lower value. When that happens, the previous end point is considered the support point. k variable has an initial value of $j + 2$ or $j - 2$, depending on which support arm is being calculated.

$$F = L_{jk} - E_{jk} \tag{7}$$

Genetic algorithms have been used to deal with the polygonal approximation issue in a variety of ways [13, 26, 37, 39, 40]. These different approaches share many characteristics, such as the codification used, while they differ in specific choices, such as the crossover or mutation operators used.

In Yin algorithm, from the formulation of the problem presented in equations 1 and 2, the codification proposed is a string of 1's and 0's as presented in equation 8, where $a_i = 1$ implies that a_i is a dominant point. The required fitness function of the genetic algorithm is expressed in equation 9, where R is a constant and $E(\alpha)$ is the approximation error between the segmentation result and the original data. Two different approximation error functions are proposed in the paper, the maximum error (E_∞, equation 10) and the integral square error ($E_2(\alpha)$, equation 11). In both cases, $e_i(\alpha)$ is the distance between p_i and the nearest line segment.

$$\alpha = a_1, a_2, \ldots, a_n \tag{8}$$

$$f(\alpha) = R - E(\alpha) \tag{9}$$

$$E_\infty(\alpha) = \max_{1 \le i \le n} e_i(\alpha) \tag{10}$$

$$E_2(\alpha) = \sum_{i=1}^{n} [e_i(\alpha)]^2 \tag{11}$$

The algorithm uses an elitist strategy [13], where the fittest string in each generation is always taken to the following one. The rest of the genetic algorithm parameters are a population size of 100 and a number of generations of 100.

[37] proposes several modifications over Yin algorithm, mainly to increase the speed required to obtain the solution. An additional table is added to the genetic algorithm, determining the probability of point p_i to be a *break point* regarding the current population. This probability is based on the k-cosine measure of significance (equation 4). The proposed probability function is shown in equation 12, where Z is the population size.

$$P_B(i) = \frac{\sum_{j=1}^{Z} cos_{ikj} + 1}{2Z} \tag{12}$$

The algorithm uses the same operators presented in Yin algorithm, but adds a divide-and-conquer technique based on the break point detection. Once a point has been determined to be a break point, the GA divides the chromosome in two parts according to the break point position and continues to be executed over both parts separately. The final solution is built upon the partial solutions of the different GAs built in this manner. The configuration parameters used are also different (a fact which does affect the number of generations required, even though no discussion was included in the work), setting the initial values of the population size to 60, the crossover probability to 0.6 and the mutation probability to 0.3.

It is remarkable that both algorithms require an input parameter: the number of segments in the solution. This fixed number of segments is the factor which creates the need for operators which do not alter the number of dominant points in the parents (if we are dealing with the crossover operator) or in the original individual (in the case of the mutation operator).

3 MULTI-OBJECTIVE APPROACH TO SEGMENTATION PROCESSES

The traditional criteria used in the data mining community to determine the quality of a segmentation process [18, 21], 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*, and
(3) minimizing the number of segments so that the total representation error does not exceed, *total_error*.

These criteria highlight the importance of the number of segments, but the comparisons performed, for instance, in the one of the source works for those criteria, [18], are based only on the quality of the segmentation obtained, neglecting the cost of that quality. From the definition of the input data included in equations 1 and 2,

we may formalize the definition of a segmentation process of (13), where each B_m would be the set of resultant segment, delimited by the dominant points at their extremes, k_{min} and k_{max}, and the number of those segments must be lower than n, the number of points in the original data.

$$\begin{aligned} S(t) = \{B_m\}, & B_m = \{\vec{p_i}\}, \text{ with} \\ & i = k_{min}, \ldots, k_{max}, \\ & m \in [1, \ldots, n-1]. \end{aligned} \tag{13}$$

Considering the previously stated criteria, we need to perform that segmentation according to a set of different objective functions which have to be minimized jointly, and which are in conflict. That problem matches perfectly the definition for a multi-objective optimization problem. The textual definition for these problems by [25] states that a multi-objective optimization problem can be defined as the problem of finding a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term optimize means finding such a solution which would give the values of all the objective functions acceptable to the decision maker. As seen in section **??**, it may be formalized following equation 14.

$$\begin{aligned} & f_p : \chi \rightarrow \Re, \; F(x) = (f_1(x), \ldots, f_k(x)) \; min_{x \in \Re} F(x) \\ & \text{such that } \begin{cases} g_i(x) \leq 0 & i = [1 \ldots n] \\ h_j(x) = 0 & j = [1 \ldots m] \end{cases} \end{aligned} \tag{14}$$

Combining the segmentation problem formulation with the general multi-objective problem formulation according to the previous criteria, we obtain equation 15, which is the general formulation for the problem. $E(S(t), t)$, is the approximation error between the output segments of the process and the original data and $E(S(B_m), B_m)$ is the approximation error between the segment created by the dominant points of segment B_m (the edges of the segment) and the original points contained in B_m.

$$\begin{aligned} B_m = & \vec{x}_j, \text{ such that} \\ & j \in [k_{min}, \ldots, k_{max}], m \in [1, \ldots, p], \\ & p < n \rightarrow \min\{E(S(t), t), p\} \\ & E(S(t), t) \leq total\_error \\ & \forall m, E(S(B_m), B_m) \leq max\_segment\_error \end{aligned} \tag{15}$$

Once the problem has been formalized (this formalization had been initially required in chapter **??**, to define the proper quality metrics), it is interesting to analyze the ways in which this multi-objective formulation has been tackled in the available algorithms. There are, basically, three different ways to deal with a multi-objective problem, [4]. The definitions in the reference are restricted to multi-objective problems solved by means of evolutionary algorithms, but most of the definitions can be generalized to different approaches:

- *A priori* techniques: These techniques require the DM, in general, to define the importance of the different objective functions in the MOP. The MOP is, with the use of these importance factors, reduced to a single objective optimization problem.
- Progressive techniques: These techniques require the direct interaction of the DM during the search process, combining cycles of search and decision making.
- *A posteriori* techniques: *A posteriori* techniques seek for P_{true} and PF_{true} [16], trying to perform a search as widespread as possible to generate as many elements as possible from the Pareto Set.

P_{true} is the *Pareto Optimal Set* and PF_{true} is the *Pareto Optimal Front*. The Pareto Optimal Set is the set of solutions where, changing their values, cannot improve one of the objective functions without degrading the value of another objective function. The Pareto Optimal Front is the set of objective function values associated to the Optimal Pareto Set. Their formal definition may be looked up in [4]. Applied to the segmentation issue, the Pareto Optimal Set would be the set of different segmentation solutions (each of them with a different number of dominant points) where changing the number of dominant points in any of those solutions would result in a solution with a worse approximation error than one of the solutions already included in the Pareto Set. This means that the output for a segmentation process seeking that Pareto Optimal Set would be the best possible segmentation solutions with different compression levels (being a compression level the rate between the original points in the curve and the dominant points in that particular element of the Pareto Set).

The different techniques presented in section 2 deal with the problem according to *a priori* techniques. This means that they turn, with different mechanisms, the multi-objective problem into a single objective problem, and optimize that single objective problem with their particular techniques. Different *a priori* techniques include lexicographic ordering [11], aggregation functions [35] or converting objective functions to input parameters. Lexicographic ordering imposes an order among the different objective functions, and the best fitted individual is obtained according to the most important objective function, using the others as secondary fitness values to solve tie situations. Aggregation functions build a single fitness value combining the different objective function values. Finally, converting an objecting function into an input parameter focuses the search of the algorithm into a single element of the Pareto Set, leaving the DM with the responsibility of determining the rest of the characteristics of that element.

Teh and Chin algorithm [36] uses both aggregation functions and lexicographic ordering techniques. Aggregation functions are used at different steps: computing the region of support, it continues to grow while the mean distance value does not increase. That mean distance value (equation **??**) is an aggregation function, using the length of the segments and the approximation error. Also, the suppression condition in equation **??**, uses a combination of different objective functions (the measure of significance and the length of the region of support) for its decision. Finally, the suppression process performed as a final step when the *1 curvature* measure of significance was chosen, uses lexicographic ordering to determine which is the dominant point in surviving groups with only two points, using the measure of significance as the priority objective

function and the size of the region of support as the secondary objective function.

Marji and Siy algorithm [22] uses aggregation functions both explicitly and implicitly. Function (7) to determine the length of a supporting arm is an aggregation function using again the length of the support arm and the approximation error as the combined objective functions. Also, the process to determine whether a candidate point must be considered a dominant point or not, chooses a non-explicit aggregation function, since choosing it as a dominant point would reduce the length of the segments on the output, and that choice is taken according to a threshold over the approximation error.

The proposed evolutionary techniques deal with the multi-objective nature of the problem converting the *number_of_segments* objective function into an input parameter determined by the user. This choice can be analyzed from two opposite points of view: if the user knows which is the compression level he requires for his application, this allows the calculation of the best solution focused only on that compression ratio. This idea can be implemented to perform automated batch processing of data sets according to the multiplication of the compression ratio by the number of measures in the time series. However, the results obtained for the error may not be feasible for the application of the results, leading to the need of individual choices for the number of segments in each input time series, and requiring the constant feedback from the DM during the whole process.

The use of constrains in the evolutionary approaches might be a solution to deal with this issue, but the choice of those constrains would be individual for each input. In [39, 40] these difficulties are met providing different solutions for different *number_of_segments* parameter values. Each of these solutions runs the evolutionary algorithm from an initial random population.

This requirement for different possible solutions is not only met in evolutionary techniques. Traditionally, Pareto fronts were built by mathematical techniques for multi-objective optimization artificially by performing several runs with different parameters [23]. In non-evolutionary techniques for segmentation purposes, input parameters are commonly based on the approximation error rather than the number of segments, being also a representative amount of non-parametric techniques (which obviously can never produce a Pareto front, since they can only provide a single solution for each problem instance).

In parametric techniques, in order to build a complete Pareto front, the user must determine the approximation errors to obtain the required number of segments in the approximations. The choice of these values may be an optimization issue itself, and clearly problem dependent. This implies that, in the cases where such a solution is possible (parametric techniques) it is difficult and computationally costly to obtain a Pareto front for a segmentation problem with the available approaches.

4 MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM FOR SEGMENTATION PROCESSES

The first issue regarding representation of the problem is the choice of the related structure. In traditional approaches, the representation was based on the detection of relevant points, codifying each problem instance as a string of 0's and 1's, representing each gene a point in the problem instance and whether this was a dominant point or not. Figure 1 shows the relationship between the genotype and its represented phenotype.

An alternative possible representation can be based on integer values, representing each of these integer values the number of the point in the input problem instance. This representation could be based on a fixed or variable size chromosome. The chosen alternative could be a fixed size chromosome where this size was equal to the input problem instance size (such as in the previous approach), such that dominant point might be repeated in that structure. Figure 2 shows an example of this approach.

This representation attempts to provide a representation anchor to the importance of certain key dominant points, which are present in almost all the different possible segmentations, regardless of the number of dominant points used (this can be seen in figures ??-??). By storing several copies of those important dominant points in a chromosome, they would become more resilient to the changes introduced by transformation operators. Also, it introduces a series of handicaps: first of all, the chromosome has to be ordered in order to provide efficient transformation operators, and reordering has to be applied after the application of any transformation operator, affecting the performance of the algorithm. An even more important handicap is related to the fact that the search space is much more extensive than the one obtained using a binary representation. Also, there is no direct genotype to phenotype relationship, since now several different genotypes can represent the same phenotype. This fact may make the search slower and affect the efficiency of transformation operators.

The focus of this work is not to proof the benefits of a particular technique (even though one has been chosen for the results presentation and comparison), but rather of the whole approach itself. To do so, we will choose a very extended MOEA: Strength Pareto Evolutionary Algorithm 2 (SPEA-2) [41], according to its implementation in the JMetal integrated development environment (IDE) [10]. The choice of this algorithm has been made according to its extended implementations in different languages and IDE's which can ease the comparison with the results presented for different authors, along with its wide usage in research works. Also, it was chosen over alternative algorithms which share similar wide usage characteristics (such as NSGA-II, [7]) due to its use of an *archive* to preserve the best solutions among different generations, which suits the requirements of segmentation algorithms.

The configuration required for the chosen technique implies the mutation and crossover probabilities, population size and number of generations (the rest of the parameters are chosen according to their standard values: 1-point crossover, bit-flip mutation and binary tournament selection). The first two probabilities have been chosen according to standard values (0.9 for the crossover probability and $1/chromosome\_length$ for the mutation one). Population

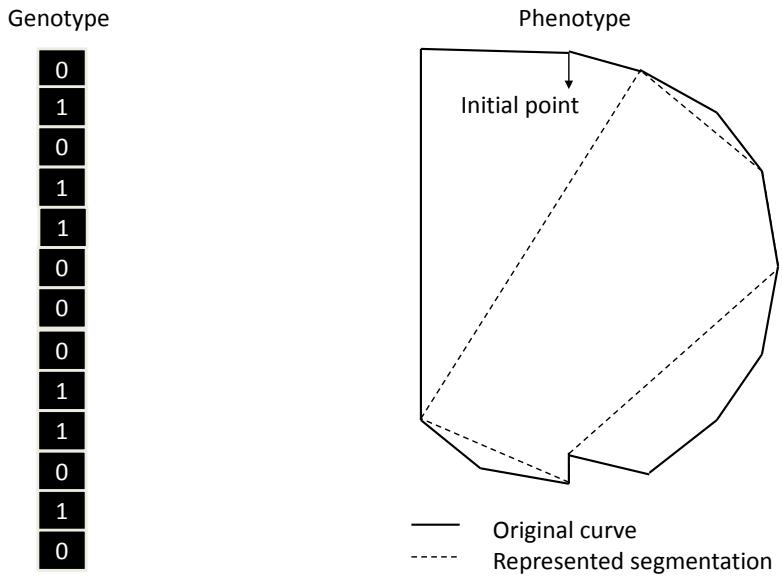

Figure 1: Genotype to phenotype mapping.

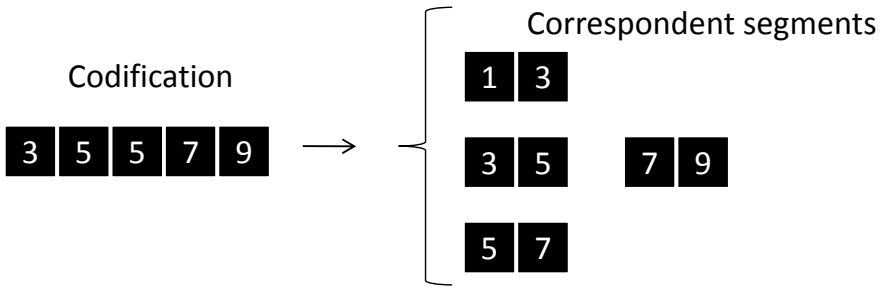

Figure 2: Integer representation genotype.

size and number of generations did not have a clear choice, a set of experiments was run with population sizes ranging from 100 to 500 and generation values ranging from 100 to 2000. In order to determine whether there were significant improvements between the different configurations, we used the Wilcoxon test [5] over the hypervolume result [42] of the obtained Pareto Fronts, with 30 runs for each configuration over the three curves in the used dataset. In table **??** the results for this comparison over the chromosome curve are shown, where 0 means that there is no statistical significance at 1% level, 1 means that there is statistical significance and "-" that the comparison is not applicable or already covered. The configuration values for each configuration number with a population size of 100 are shown in table 1. Configuration numbers 7-12 share the same growing generation values with population size 200, and configuration numbers 13-18 with population size 500.

Table 1: MOEA configurations detail for population size 100

Config. number	1	2	3	4	5	6
Population Size	100	100	100	100	100	100
Generations	100	300	500	700	1000	2000

5 EXPERIMENTAL RESULTS

The dataset used will be based on the three most extended curves for polygonal approximation testing, usually named chromosome, leaf and semicircle. We will compare the results obtained with a set of nine representative techniques, some of which have been detailed in previous sections: [22], [36], [33], [6], [1], [29], [34], [38] and finally a special comparison with the evolutionary technique by [39].

The dataset, along with some segmentation results from the obtained Pareto Fronts, is presented where introduces the chromosome curve, which has 60 boundary points, along with five results from the Pareto Set obtained by the technique. Also the same results for the leaf curve (with 120 boundary points), and for the semicircle one (with 102 boundary points).

Table 2 presents the results of the first eight techniques to be compared. These technique results are either non-parametric or the included results are those presented in their reference works according to their default configuration. This means that each of these techniques provides only a single solution for each problem in the dataset. Table 3 presents the statistical comparison of these techniques with the MOEA technique used. To perform this comparison, the solution with the appropriate number of dominant points (the same as the single solution provided by the compared technique) is extracted from the resultant Pareto front in the 30 independent executions performed, and a Student's t-test with 5% confidence level is performed over the difference of those values, determining whether the difference is statistically significant or not. If the difference is statistically significant, the best technique is indicated, including the '-' symbol in any other case.

The statistical comparison shown in table 3 determines that the MOEA technique is significantly better than the other alternatives in 21 out of 24 test cases, being significantly worse only in one case (Cronin's result for the semicircle curve). Also, the differences between its results and the alternatives are very significant, which can be observed in the different graphical comparisons presented in the figures and the low p-values contained in the tables. The dataset is rather scarce, but without standard implementations of the techniques or a framework to properly test them with novel data, the comparison has resorted to the results in their reference papers, which only included these figures. The good performance results of the evolutionary technique against a set of techniques specialized for this particular domain are, in any case, remarkable

6 CONCLUSIONS

This work has been focused on the segmentation issue by means of Piecewise Linear Representation, which is present in the polygonal approximation domain, highlighting its unresolved issues. One of those issues is the multi-objective nature of segmentation processes, where several objective functions have to be optimized jointly. This fact has not received the proper attention in terms of algorithm development (only for certain comparison purposes). Even so, any technique available has to deal with this multi-objective nature of the problem, even if this nature is not explicitly declared. Four representative algorithms have been detailed, covering their implicit treatment of that multi-objective nature, based on a-priori approaches. This discussion has lead to the explicit formulation of segmentation as a proper multi-objective problem and its resolution by means of an a-posteo approach using a multi-objective evolutionary algorithm. For the results presentation, the chosen algorithm is SPEA2, along with default variation operator values.

The final objective of the multi-objective evolutionary approach is obtaining the whole Pareto front of possible segmentation results for a given problem. Parametric techniques can obtain artificial Pareto fronts with several different runs configured with different input parameters, being each of these solutions independent. This is computationally inefficient and can lead to additional optimization problems (such as the determination of the proper error approximation value in order to obtain a certain number of segments in the solution). These problems are inherently solved with the use of the MOEA approach presented in this work. Also, the different solutions in the Pareto front of a segmentation problem share valuable information in the form of dominant point position, leading to faster and better solutions when compared to obtaining individual elements from that Pareto front.

The results obtained in the Pareto front with the chosen technique in the polygonal approximation dataset used are extremely competitive with the available works in the literature, having obtained statistically significant improvements in 36 out of the 40 individual results, and also in the two curves compared under a multi-objective perspective by means of the hypervolume quality indicator, showing that treating the multi-objective nature of the problem explicitly allows the algorithm to obtain better solutions. It is important to highlight that this technique is able to cope with the requirements presented in the introduction, allowing the final user to regain its role as the decision maker of the problem and to change which solutions fit its requirements at different moments (provided by obtaining the whole Pareto Front in a single execution). Future lines include the application and comparison of the presented technique with time series datasets.

ACKNOWLEDGMENTS

This work was supported in part by Project MINECO TEC2017-88048-C2-2-R, FAPERJ APQ1 Project 211.500/2015, FAPERJ APQ1 Project 211.451/2015, CNPq Universal 430082/2016-9, FAPERJ JCNE E-26/203.287/2017, Project Prociência 2017-038625-0, CNPq PQ 312792/2017-4.

REFERENCES

[1] N. Ansari and K. Huang. 1991. Non-parametric dominant point detection. *Pattern Recognition* 24, 9 (1991), 849–862.
[2] F. Attneave. 1954. Some informational aspects of visual perception. *Psychological review* 61, 3 (1954), 183–193.
[3] C.A.C. Coello and G.B. Lamont. 2004. *Applications of multi-objective evolutionary algorithms*. World Scientific Pub Co Inc.
[4] C.A.C. Coello, G.B. Lamont, and D.A. Van Veldhuizen. 2007. *Evolutionary algorithms for solving multi-objective problems*. Springer-Verlag New York Inc.
[5] G.W. Corder and D.I. Foreman. 2009. *Nonparametric statistics for non-statisticians: a step-by-step approach*. John Wiley & Sons Inc.
[6] T.M. Cronin. 1999. A boundary concavity code to support dominant point detection. *Pattern Recognition Letters* 20, 6 (1999), 617–634.
[7] K. Deb, A. Anand, and D. Joshi. 2002. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary computation* 10, 4 (2002), 371–395.
[8] David H. Douglas and Thomas K. Peucker. 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. *The Canadian Cartographer* 10, 2 (1973), 112–122.
[9] R. O. Duda and P. E. Hart. 1973. *Pattern Classification and Scene Analysis*. Wiley.
[10] Juan J. Durillo and Antonio J. Nebro. 2011. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software* 42, 11 (2011), 760–771. https://doi.org/DOI:10.1016/j.advengsoft.2011.05.014
[11] M.P. Fourman. 1985. Compaction of symbolic layout using genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*. L. Erlbaum Associates Inc., 141–153.
[12] A. Gionis and H. Mannila. 2005. Segmentation algorithms for time series and sequence data. Tutorial on SIAM International Conference in Data Mining. (2005).
[13] D.E. Goldberg et al. 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley Reading Menlo Park.
[14] FCA Groen and PW Verbeek. 1978. Freeman-code probabilities of object boundary quantized contours. *Computer Graphics and Image Processing* 7, 3 (1978), 391–402.
[15] Josï£¡ Luis Guerrero, J. Garcia, and Josï£¡ Manuel Molina. 2010. Air Traffic Control: A Local Approach to the Trajectory Segmentation Issue. In *Proceedings for the 23rd International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, part III. Lecture Notes in Artificial Intelligence*, Vol. 6098. Springer, 498–507.
[16] J. Horn. 1997. *Handbook of evolutionary computation*. Oxford University Press, Chapter Multicriterion decision making.

Table 2: Comparable techniques results for the dataset

Technique	Chromosome		Semicircle		Leaf	
	Dom. Points	ISE	Dom. Points	ISE	Dom. Points	ISE
SAMAPA	12	5.82	19	12.90	21	13.60
Ansari and Huang	16	20.30	28	17.80	30	25.60
Teh and Chin	15	7.20	22	20.60	29	14.96
Cronin	17	3.18	30	2.91	28	7.30
Marji and Siy	11	9.96	18	24.20	21	14.10
Ray and Ray	18	5.57	29	11.80	32	14.70
Sarkar	19	3.86	19	17.40	23	13.10
Wu	17	5.01	27	9.01	23	20.34

Table 3: Statistical result comparison

Technique	Chromosome		Semicircle		Leaf	
	p-value	stat. best	p-value	stat. best	p-value	stat. best
SAMAPA	3.21E-01	-	1.67E-43	MOEA	2.52E-05	MOEA
Ansari and Huang	3.15E-43	MOEA	4.55E-63	MOEA	2.27E-40	MOEA
Teh and Chin	1.80E-22	MOEA	3.68E-64	MOEA	1.57E-29	MOEA
Cronin	1.54E-01	-	2.09E-10	Cronin	7.23E-07	MOEA
Marji and Siy	6.93E-14	MOEA	3.43E-70	MOEA	8.06E-07	MOEA
Ray and Ray	2.47E-23	MOEA	2.20E-56	MOEA	8.18E-34	MOEA
Sarkar	4.36E-16	MOEA	1.47E-55	MOEA	4.75E-13	MOEA
Wu	6.62E-18	MOEA	1.67E-49	MOEA	2.54E-25	MOEA

[17] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems* 3, 3 (2001), 263–286.

[18] E. Keogh, S. Chu, D. Hart, and M. Pazzani. 2003. Segmenting time series: A survey and novel approach. *Data mining in time series databases* (2003), 1–21.

[19] E. Keogh and M. Pazzani. 1998. 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*. AAAI Press, 239–241.

[20] DJ Langridge. 1972. On the computation of shape. *Frontiers of Pattern Recognition* (1972), 347–366.

[21] X. Liu, Z. Lin, and H. Wang. 2008. Novel online methods for time series segmentation. *IEEE Transactions on Knowledge and Data Engineering* 20, 12 (2008), 1616–1626.

[22] M. Marji and P. Siy. 2003. A new algorithm for dominant points detection and polygonization of digital curves. *Pattern recognition* 36, 10 (2003), 2239–2251.

[23] K. Miettinen. 1999. *Nonlinear multiobjective optimization*. Kluwer Academic Publisher, Boston.

[24] F Mörchen. 2003. *Time series feature extraction for data mining using DWT and DFT*. Technical Report 33. Departement of Mathematics and Computer Science Philipps-University Marburg.

[25] A. Osyczka. 1985. Multicriteria optimization for engineering design. *Design Optimization* (1985), 193–227.

[26] N.R. Pal, M.K. Kundu, and S. Nandi. 1998. Application of a new genetic operator in feature selection problems. In *IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control*, Vol. 1. IEEE, 37–40.

[27] T. Pavlidis and F. Ali. 2007. Computer recognition of handwritten numerals by polygonal approximations. *Systems, Man and Cybernetics, IEEE Transactions on* 5, 6 (2007), 610–614.

[28] U. Ramer. 1972. An Iterative Procedure for the Polygonal Approximation of Plane Curves. *Computer Graphics and Image Processing* 1 (1972), 244–256.

[29] B.K. Ray and K.S. Ray. 1992. Detection of significant points and polygonal approximation of digitized curves. *Pattern Recognition Letters* 13, 6 (1992), 443–452.

[30] B. Rosenberg. 1972. The analysis of convex blobs. *Computer Graphics and Image Processing* 1, 2 (1972), 183–192.

[31] A. Rosenfeld and E. Johnston. 1973. Angle Detection on Digital Curves. *IEEE Trans. Comput.* 22, 9 (1973), 875–878.

[32] M. Sarfraz. 2008. Linear Capture of Digital Curves. In *Interactive Curve Modeling*. Springer London, 241–265.

[33] M. Sarfraz, MR Asim, and A. Masood. 2004. Piecewise polygonal approximation of digital curves. In *Proceedings of the 8th International Conference on Information Visualisation*. IEEE, 991–996.

[34] D. Sarkar. 1993. A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves. *Pattern Recognition Letters* 14, 12 (1993), 959–964.

[35] P. Surry, N. Radcliffe, and I. Boyd. 1995. A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA method. *Evolutionary Computing* (1995), 166–180.

[36] C.H. Teh and RT Chin. 2002. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 8 (2002), 859–872.

[37] Y.H. Tsai. 2006. Fast Polygonal Approximation Based on Genetic Algorithms. In *Proceedings of the 5th International Conference on Computer and Information Science and 1st International Workshop on Component-Based Software Engineering, Software Architecture and Reuse*. IEEE, 322–326.

[38] W.Y. Wu. 2003. An adaptive method for detecting dominant points. *Pattern Recognition* 36, 10 (2003), 2231–2237.

[39] P.Y. YIN. 1999. Genetic algorithms for polygonal approximation of digital curves. *International journal of pattern recognition and artificial intelligence* 13, 7 (1999), 1061–1082.

[40] P. Y. Yin. 1998. A New Method for Polygonal Approximation Using Genetic Algorithms. *Pattern Recognition Letters* 19, 11 (1998), 1017–1026.

[41] E. Zitzler, M. Laumanns, and L. Thiele. 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Proceedings of the Conference on Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems*.

[42] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* 7, 2 (2003), 117–132.