Engineering Optimization (2012), 44 (3), 262-287.
DOI: http://dx.doi.org/10.1080/0305215X.2011.641718

A multi-objective approach for the segmentation issue

José Luis Guerrero*, Antonio Berlanga and José Manuel Molina

Universidad Carlos III de Madrid, Colmenarejo, Spain

Abstract: This work presents and formalizes an explicit multi-objective evolutionary approach for the segmentation issue according to Piecewise Linear Representation, which 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 required. Available techniques are focused on the minimization of the quality of the obtained approximation, being the cost of that approximation considered, in general, only for certain comparison purposes. The multi-objective nature of the problem is analysed and its treatment in available works reviewed, presenting an *a posteriori* approach based on an evolutionary algorithm. Three representative curves are included in the data set, comparing the proposed technique to nine different techniques. The performance of the presented approach is tested according to single and multiobjective perspectives. The statistical tests carried out show that the experimental results are, in general, significantly better than available approaches from both perspectives.

Keywords: polygonal approximation; time series; piecewise linear representation; multi-objective optimization; evolutionary algorithms

1. Introduction

The digital curves domain, lead by the importance of human processing and understanding visual information, established its roots with psychological studies performed in the mid-fifties (Attneave 1954). 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 (Mörchen 2003).

Segmentation processes may resort to different representations, the Piecewise Linear Representation (PLR, also named the Piecewise Linear Approximation, PLA, or polygonal approximation) being among the most extended options. This scope has been deeply analysed and used according to a data mining perspective (Keogh *et al.* 2004, Gionis and Mannila 2005, Liu *et al.* 2008) and also as a digitization method (Marji and Siy 2003, Sarfraz 2008). Several works have detailed

*Corresponding author. Email: jguerrer@inf.uc3m.es

the characteristics of PLR segmentation which have led to its extensive use: simplicity, locality, generality, compactness and ease of use (Keogh *et al.* 2004, Sarfraz 2008). 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 $K \ll n$), approximating each of these segments by a linear model. It can also be described as the process of searching the *dominant points* of a given curve, being those points at the edges of the segments.

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 sensitive to their segmentation criteria 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 control over the approximation error. Comparisons between different algorithms, especially in the data mining domain (Keogh *et al.* 2004) 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 Ray and Ray 1992, 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 (Coello *et al.* 2007).

The previous argument introduces segmentation as a multi-objective optimization problem (MOOP, Coello and Lamont 2004): 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 (Osyczka 1985). 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.

Consider the two different segmentations presented in Figure 1. Both 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 (Keogh *et al.* 2001) or data mining approaches (Keogh and Pazzani 1998) up to optical character recognition applications (Pavlidis and Ali 2007) or applications in the air traffic control domain (Guerrero *et al.* 2010a). 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 (Surry *et al.* 1995). This implies that the algorithm designer establishes the importance of the different

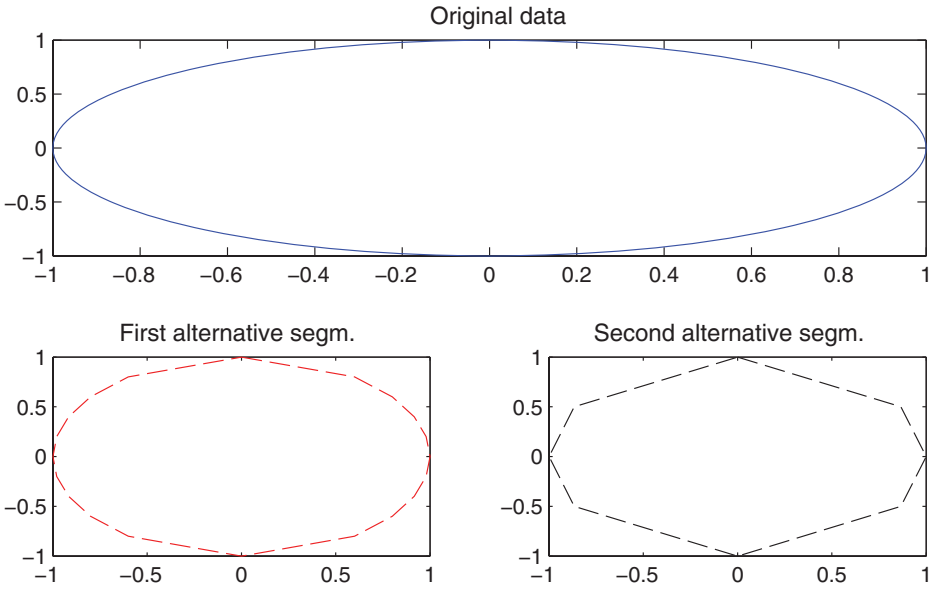

Figure 1. Alternative segmentations for a simple circle shape.

objectives, and afterwards codifies it into the algorithm running cycle. In other cases, 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 re-run 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 become very difficult to perform accurately.

Multi-objective evolutionary algorithms (MOEAs) are evolutionary algorithms (EAs) focused on a set of different objective functions which have to be optimized jointly. The objective of these algorithms is to find the optimal Pareto Front (the set of solutions where improving an objective function value cannot be performed without degrading the value of a different one, Ehrgott 2005). Evolutionary algorithms are a useful technique to deal with multi-objective problems, since they simultaneously deal with a set of possible solutions (the population) which allow them to find several members from the Pareto Front in a single run (Coello and Lamont 2004), instead of performing a series of separate runs, as had to be done with traditional mathematical techniques (Miettinen 1999). They also have the interesting property of being less susceptible to the shape or continuity of that Pareto Front (being able to deal with discontinuous and concave Pareto Fronts).

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).

The main contributions of this work are both theoretical and practical in their nature. First of all, the proposal and formalization of the segmentation issue as a multi-objective problem, along with the analysis of techniques available in the polygonal approximation literature regarding this multi-objective perspective and how it has been dealt with. This discussion includes the relevance of the decision-maker role and to whom this role has been assigned in available approaches. This analysis leads to the proposal of an *a posteriori* resolution method based on a standard MOEA, along with the required representation and operators (particularly focused on a specific initialization process regarding the nature of the objective functions). Finally, the proposed implementation is tested with its results comparison against a set of nine techniques from the polygonal approximation domain with a dataset of three standard curves, according to a single objective (quality of the individual elements of the obtained Pareto Fronts compared to other techniques results) and multi-objective (measured by quality indicators) perspectives, highlighting the statistical significance of the obtained results.

The following sections are organized as follows: Section 2 describes a number of segmentation techniques from the available literature on the topic, leading to Section 3 where, according to the presented characteristics of available approaches, the multi-objective nature of the problem is formalized and the proposed approach detailed. Section 4 deals with the choice and configuration of the evolutionary technique for the multi-objective formalization of the problem, presenting the obtained results in Section 5 and finally the conclusions which the work leads to in Section 6.

2. Overview of segmentation techniques

This section provides 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 time, provide a considerable understanding of the approaches which have been taken to deal with the segmentation issue. For formalization purposes, Equation (1) defines the components of a given curve, where x_i and y_i are the plane coordinates of the point and i is the point's number.

$$t = \{\vec{p}_i\}, \quad \vec{p}_i = (x_i, y_i, i), \quad i = 1, \ldots, n. \tag{1}$$

From the definition of the input data included in Equation (1), the definition of a segmentation process may be formalized with Equation (2), where each B_m is one of the resultant segments, composed of a set of $\vec{p}_i$ points. Dominant points are those at the extremes of each of these segments, $k_{\min}$ and $k_{\max}$, which delimit them.

$$S(t) = \{B_m\}, \quad B_m = \{\vec{p}_i\}, \quad i = k_{\min}, \ldots, k_{\max}, \quad m \in [1, \ldots, n-1]. \tag{2}$$

Polygonal approximation algorithms can be divided into sequential, split-and-merge and heuristic search approaches. Sequential approaches are constructive methods based on a given local search over the current data, trying to obtain, at each step, a new segment division (where the length of these segments is sequentially increased) which satisfies a certain criterion. Examples of the used criteria may be finding the longest possible segments (Sklansky and Gonzalez 1980) or a combination of finding the longest possible segments with the minimum possible error (Ray and Ray 1992). Split-and-merge approaches perform an initial segmentation over the given time series and afterwards start an iterative process to merge the initial segments until a certain criterion is met. According to their definition, these approaches have to deal with two different issues: the initial segmentation procedure and the merging criterion. An example of

these techniques is Ramer (1972), which performs an initial boundary segmentation, followed by a sequence of steps where the segment is split at the point with the furthest distance from the corresponding segment unless the approximation error is lower than, or equal to, a specified error tolerance.

Heuristic search approaches are based on the development of heuristic methods in order to avoid the exhaustive search of the optimal *dominant points* for the given curve (which is a process with an exponential complexity). Different techniques may be used for this purpose, such as dynamic programming (Dunham 1986, Sato 1992) or several metaheuristics, among them solutions based on genetic algorithms (Goldberg 1989, Yin 1998, 1999, Pal *et al.* 2002, Tsai 2006), which will be highlighted in this work due to their relationship with the proposed solution. The idea proposed by these works is to codify the time series as a chromosome with n genes, associating each of these genes with one of the points in the original data. If the gene value is a '1', it is considered a *dominant point*, and the algorithm tries to find the ideal codification of the chromosome according to a fitness function which evaluates the quality of the given codified segmentation in the chromosome.

2.1. *Teh and Chin algorithm*

Teh and Chin algorithm (Teh and Chin 2002) is based on the concept of the *region of support* (Langridge 1972): this concept states that each boundary point of a closed curve must have its own view of the curve, dominant points being those that have a meaningful view of the curve and that block the view of other non-dominant points.

In Teh and Chin (2002) the proposal is based on the difficulty of determining the curvature of a digital curve. The functions to determine discrete curvature are named *measures of significance* (Rosenberg 1972). Three different measures of significance are used: the k cosine measure, the k curvature measure and the *1* curvature measure. The algorithm starts with the calculation of the region of support for a given point p_i. This calculation is performed determining the length of the chord joining the points p_{i-k} and p_{i+k} (l_{ik}, shown in Equation 3) and the perpendicular distance of the points contained in the chord to their respective ones in the original data, d_{ik}. This process is continued until the value of the length of the chord stops growing or until the mean distance starts growing (represented in Equation 4).

$$l_{ik} = \overline{|p_{i-k}p_{i+k}|} \tag{3}$$

$$\begin{cases} \dfrac{d_{ik}}{l_{ik}} \geq \dfrac{d_{i,k+1}}{l_{i,k+1}}, & \text{if } d_{ik} > 0 \\ \dfrac{d_{ik}}{l_{ik}} \leq \dfrac{d_{i,k+1}}{l_{i,k+1}}, & \text{if } d_{ik} < 0. \end{cases} \tag{4}$$

The second step of the algorithm calculates the three measures of significance. Finally, according to the previous data, dominant points are calculated suppressing non-maximal points from the previous sets. This is performed following an iterative process which sequentially filters the points according to their measure of significance value. This process changes depending on the concrete measure used.

2.2. *Marji and Siy algorithm*

The Marji and Siy algorithm (Marji and Siy 2003) relies on the concept of *support arms*. This implies that the region of support is not used to calculate a significance measure of the boundary points, but instead it is used to 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.

To determine both support arms, the function shown in Equation (5) 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 iteratively 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. Variable k has an initial value of $j + 2$ or $j - 2$, depending on which support arm is being calculated.

$$F = L_{jk} - E_{jk}. \tag{5}$$

The algorithm proceeds by calculating the support arms of all points, and determining their classification as dominant depending on the classification of the points surrounding them, their distance to the segment delimited by the dominant points immediate to their left and right and the possible overlapping of regions of support.

2.3. *Genetic approach based algorithms*

Genetic algorithms (Goldberg 1989) have been used to deal with the polygonal approximation issue in a variety of ways (Yin 1998, 1999, Pal *et al.* 2002, Tsai 2006). These different approaches share many characteristics, such as the codification used, while differing in specific choices, such as the crossover or mutation operators used. This section will be focused in the Yin algorithm (Yin 1998, 1999) and the speed-up modification introduced by Tsai (2006) to provide the required overview of the topic.

In the Yin algorithm, from the formulation of the problem presented in Equation (1), the codification proposed is a string of 1's and 0's as presented in Equation (6), where $a_i = 1$ implies that a_i is a dominant point. The required fitness function of the genetic algorithm is expressed in Equation (7), 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 article, the maximum error (E_∞, Equation 8) and the integral square error ($E_2(\alpha)$, Equation 9). In both cases, $e_i(\alpha)$ is the distance between p_i and the nearest line segment.

$$\alpha = a_1, \quad a_2, \ldots, a_n \tag{6}$$

$$f(\alpha) = R - E(\alpha) \tag{7}$$

$$E_\infty(\alpha) = \max_{1 \le i \le n} e_i(\alpha) \tag{8}$$

$$E_2(\alpha) = \sum_{i=1}^{n} [e_i(\alpha)]^2. \tag{9}$$

In the algorithm, the R value is adapted in order to prevent an outstanding individual taking a significant proportion of the following generation by adopting the selectivity concept (Singh *et al.* 1997): selectivity is the ratio of the maximally and minimally fitted solutions in the population. The crossover operator forces the offspring to have the same number of dominant points as their parents (Gen and Cheng 1997, Pal *et al.* 2002), with a probability adapted according to the generation value (Loncaric and Dhawan 1995). The mutation operator performs a cyclic shift to preserve the number of dominant points, being also adapted according to the generation value. Finally, the algorithm uses an elitist strategy (Goldberg 1989), with a population size and number of generations of 100.

Tsai (2006) proposes several modifications to the 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 being a *break point* regarding the current population, based on the k cosine measure of significance. 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 proposed configuration values set the population size to 60, crossover probability to 0.6 and the mutation probability to 0.3.

3. Multi-objective approach to segmentation processes

The traditional criteria used to determine the quality of a segmentation process (Keogh *et al.* 2004, Liu *et al.* 2008) 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 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 one of the sources works for these criteria (Keogh *et al.* 2004), are based only on the quality of the segmentation obtained, neglecting the cost of that quality. Considering these criteria, segmentation has to be performed 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 (Osyczka 1985), which may be formalized following Equation (10):

$$f_p : \chi \to \Re, \quad F(x) = (f_1(x), \ldots, f_k(x)), \quad \min_{x \in \Re} F(x)$$
$$\text{such that } \begin{cases} g_i(x) \leq 0 & i = [1, \ldots, n] \\ h_j(x) = 0 & j = [1, \ldots, m]. \end{cases} \tag{10}$$

Combining the segmentation problem formulation with the general multi-objective problem formulation according to the previous criteria, one obtains Equation (11), which is the general formulation for the problem. In Equation (11), $E(S(t), t)$ 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 and the original points contained in B_m:

$$B_m = \vec{x}_j, \quad j \in [k_{\min}, \ldots, k_{\max}], \quad m \in [1, \ldots, p], \quad p < n \to \min\{E(S(t), t), p\}$$
$$\text{such that } \begin{cases} E(S(t), t) \leq total\_error \\ \forall m, E(S(B_m), B_m) \leq max\_segment\_error. \end{cases} \tag{11}$$

The basis for this formalization can be found in Guerrero *et al.* (2011), where it is applied for comparison issues over single-objective algorithms, with specific quality measures for the air traffic control domain. Once the problem has been formalized, it is interesting to analyse the way in which this multi-objective formulation has been tackled in the available algorithms. There are, basically, three different paths to deal with a multi-objective problem (Coello *et al.* 2007).

- ***A priori* techniques** These techniques require the DM, in general, to define the importance of the different objective functions in the MOOP. The MOOP 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} (Horn 1997), 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 one objective function cannot be improved without degrading the value of another objective function. The Pareto Optimal Front is the set of objective function values associated to the Pareto Optimal Set. Their formal definition may be looked up in Coello *et al.* (2007). Applied to the segmentation issue, the Pareto Optimal Set would be the set of best possible segmentation solutions with different compression levels (a compression level being 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 convert the multi-objective problem into a single-objective problem, and optimize that single-objective problem with their specific techniques. Different a priori techniques include lexicographic ordering (Fourman 1985), aggregative functions (Surry *et al.* 1995) 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. Aggregative functions build a single fitness value combining the different objective function values. Finally, converting an objective 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.

The Teh and Chin algorithm (Section 2.1) uses both aggregative functions and lexicographic ordering techniques. Aggregative functions are used at different steps. Their first use is the computation of the region of support, which continues to grow while the mean distance value does not increase. That mean distance value (Equation 4) is also an aggregative function, using the length of the segments and the approximation error. The non-maximal suppression processes use a combination of different objective functions (the measure of significance and the length of the region of support) for their decision, along with a final lexicographic ordering if the *1 curvature* measure was used (where the measure of significance is the main objective function and the size of the region of support is the secondary objective function).

The Marji and Siy algorithm (Section 2.2) uses aggregative functions both explicitly and implicitly. The function to determine the length of a supporting arm (Equation 5) is an aggregative function using again the length of the support arm and the approximation error as the combined objective functions. Also, the process for determining whether a candidate point must be considered a dominant point or not chooses a non-explicit aggregative 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 presented evolutionary techniques (Section 2.3) deal with the multi-objective nature of the problem of converting the *number_of_segments* objective function into an input parameter determined by the user. This allows the algorithm to focus on a concrete compression value, but establishing this value may be difficult and lead to unfeasible results in different objective values. In Yin (1998, 1999) these difficulties are met by providing different solutions for different

number_of_segments parameter values. Each of these solutions runs the evolutionary algorithm from an initial random population.

4. Multi-objective evolutionary algorithm for segmentation processes

A priori techniques have a series of difficulties in their treatment of MOOPs: difficulties regarding continuity and shape, the need to be run several times to obtain several individuals from the Pareto Front, etc. In the segmentation domain, difficulties regarding the configuration of the techniques for obtaining the different elements of that Pareto Front must also be taken into account (the choice of input parameter values in order to obtain certain values of the objective functions is clearly not trivial, such as determining a certain maximum error in the approximation segments in order to obtain a segmentation solution with a certain number of dominant points).

The purpose of this work is to deal with the segmentation issue by using an *a posteriori* technique, according to the formulation presented in Equation (11). The first ideas of this approach can be found in Guerrero *et al.* (2010b), where a different representation is presented (based on integer values), performing a simple comparison with a time series segmentation technique on a restricted dataset for the particular air traffic control domain. The presented formulation introduces some restrictions which may help to reduce the search space, but the choice of those boundaries may be problem dependent and also (according to previous argumentations) not trivial to establish. Thus, the segmentation issue will be faced as a multi-objective problem without restrictions, obtaining as many elements as possible from the *Pareto Set* and *Pareto Front* and letting the DM choose from those final solutions. An important consideration is that this choice is made from final solutions with all their characteristics, rather than *a priori* configuration values which may lead to unfeasible results in other components of the solution vector, allowing the DM to make simple choices (and also to vary them according to different needs for different processes, as was discussed in the introduction section).

The focus of this work is not to prove 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, the present authors will choose a very extended MOEA: Strength Pareto Evolutionary Algorithm 2 (SPEA-2) (Zitzler *et al.* 2001), according to its implementation in the JMetal integrated development environment (IDE) (Durillo and Nebro 2011). The choice of this algorithm has been made because of its extended implementations in different languages and IDEs, which can ease the comparison with the results presented by 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, Deb *et al.* 2002) owing to its use of an *archive* to preserve the best solutions among different generations, which suits the requirements of segmentation algorithms.

The main characteristic of Strength Pareto Evolutionary Algorithms (SPEAs) is the concept of *strength*, which defines their name. These algorithms define an external archive, where non-dominated solutions are saved (called the external non-dominated set). This archive is updated with non-dominated individuals after each generation is processed, and the *strength* value of each of these individuals is computed. The computation of this value was originally proportional to the number of solutions which a certain individual dominates (in the original SPEA algorithm) but in SPEA2 it was changed to a value which depends on both the number of individuals which a certain individual dominates and the number of individuals which dominate it.

The environmental selection determines how this archive is updated. Originally, a clustering technique was used, but this process tended to lose boundary solutions when the size of the archive was too small for the required number of non-dominated solutions. An enhanced truncation technique is present in SPEA2, which invokes an iterative process that eliminates at each

stage the individual with the minimum distance to another individual (in the case of ties, the distance to the second closest individual is considered, and so on) until the number of non-dominated individuals in the archive fits its maximum size. This process allows the algorithm to retain the boundary solutions through its different generations. The size of the archive in the SPEA2 algorithm remains constant, implying that if the number of non-dominated solutions at a certain generation is less than the archive size, this archive is filled up to its size with dominated solutions.

The computational complexity of the algorithm is dominated by the environmental selection procedure, with a worst case complexity for the truncation operator of $O(M^3)$, where M is the population size plus the archive size. On average, that complexity is reduced to $O(M^2 \log M)$, which is also the complexity presented by the fitness assignment process. In the application case presented in this work, the established configuration parameters, which will be detailed later, guarantee that the truncation operator will not be required, setting the overall algorithm complexity to this reduced form. The pseudo-algorithm of SPEA2 can be defined as follows.

(1) **Initialization** Generate initial population P_0 and create the initial empty archive $\overline{P}_0$.
(2) **Fitness assignment** Calculate fitness values for individuals in P_t and $\overline{P}_t$.
(3) **Environmental selection** Update $\overline{P}_{t+1}$ according to the explained procedure.
(4) **Termination** Check stopping criteria. If it is met, output non-dominated individuals in $\overline{P}_{t+1}$, otherwise continue.
(5) **Mating selection** Binary tournament selection with replacement on $\overline{P}_{t+1}$ to fill the mating pool.
(6) **Variation** Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Go to step 2.

The use of the multi-objective approach has been presented from the perspective of the problem formulation and also as a mechanism to prevent some of the difficulties found in available methods in the literature. Once an evolutionary algorithm has been chosen for this task, one may also determine whether the MOEA approach is suitable for handling this issue. The key to this approach is the fact that the different solutions of the Pareto Front can share valuable information among them. This can be seen in Figures 2–4, where different approximations with a different number of dominant points share similar positions of their key dominant points, and increasing their number leads to a more refined approximation of the zones with more abrupt changes. This implies that the use of the MOEA for this purpose also balances the high computational cost of using an evolutionary algorithm with the advantage of obtaining the whole set of solutions of the Pareto Front at a speed much faster than the time required to obtain them individually.

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 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, the present authors used the Wilcoxon test (Corder and Foreman 2009) over the hypervolume result (Zitzler *et al.* 2003) of the obtained Pareto Fronts, with 30 runs for each configuration over the three curves in the used dataset. In Table 1, 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

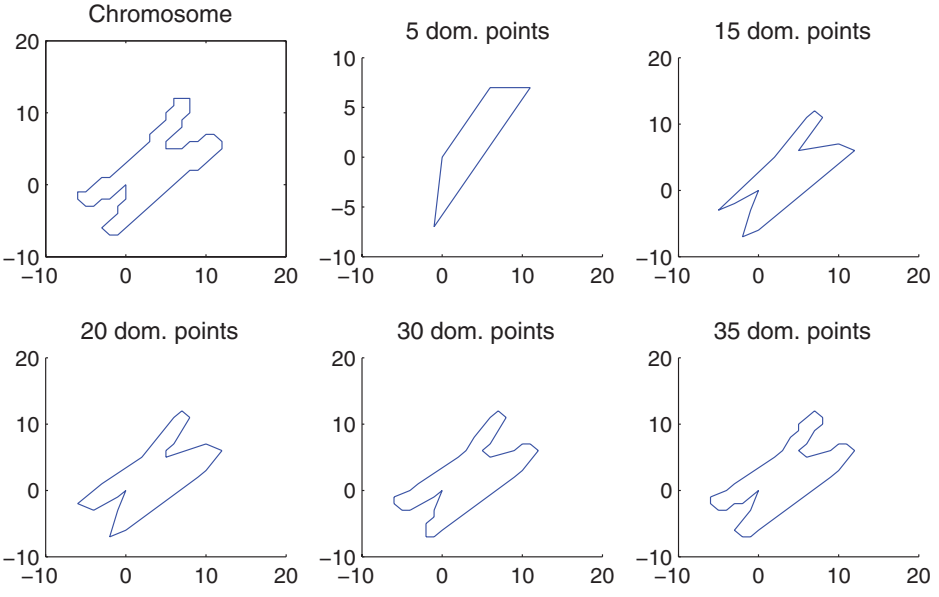

Figure 2. Chromosome curve.

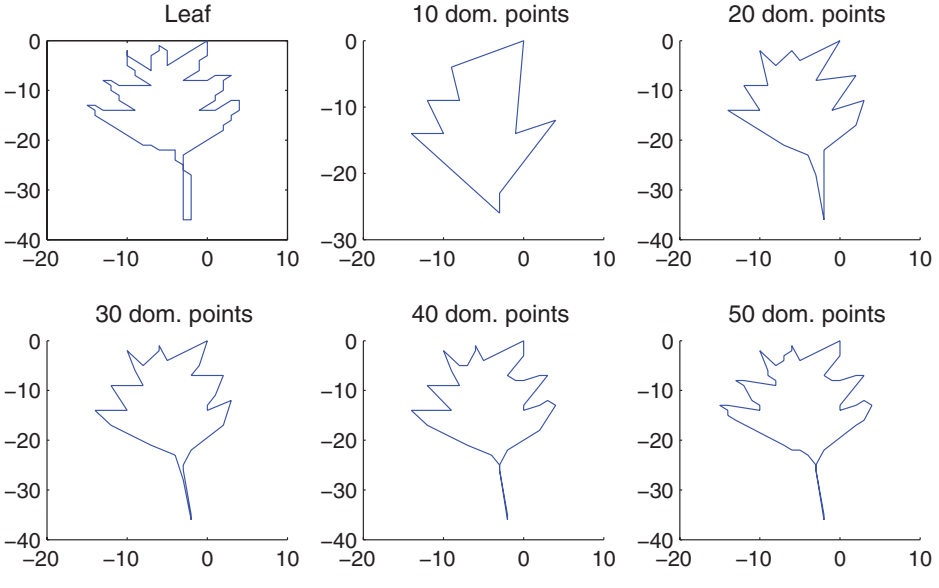

Figure 3. Leaf curve.

values for each configuration number with a population size of 100 are shown in Table 2. Configuration numbers 7–12 share the same growing generation values with population size 200, and configuration numbers 13–18 with population size 500.

The last column of Table 1 shows that there is no statistical significance in the difference of the presented results between the runs with population size 500 and 1000/2000 generations (doubling the computational effort does not provide additional improvement in the quality obtained). If similar tests are run over the other two figures in the dataset, that increase in the run generations value

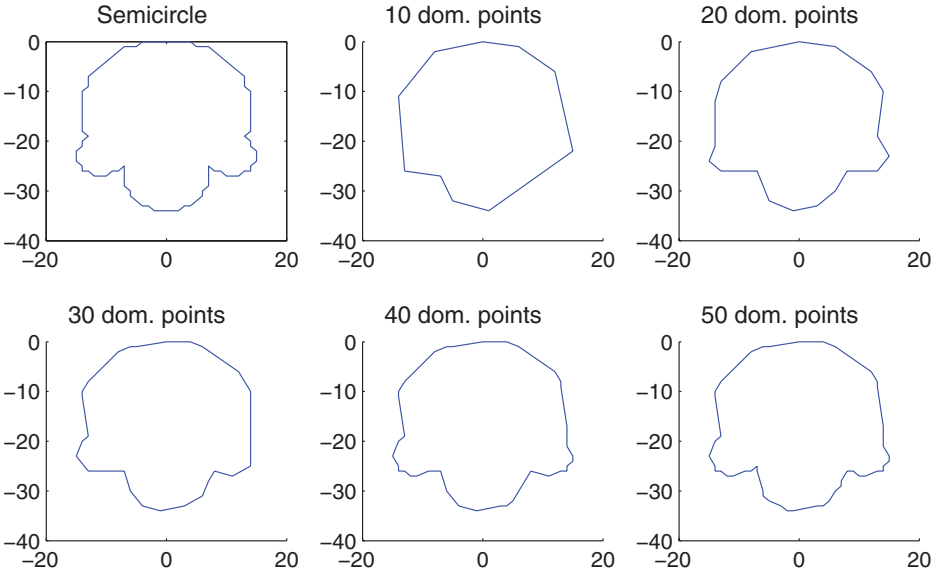

Figure 4. Semicircle curve.

Table 1. Wilcoxon test results for different MOEA configurations applied to the Chromosome curve.

Config. no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	–	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
2	–	–	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1
3	–	–	–	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
4	–	–	–	–	0	1	1	0	0	1	1	1	0	1	1	1	1	1
5	–	–	–	–	–	0	1	1	0	0	0	1	0	0	1	1	1	1
6	–	–	–	–	–	–	1	1	1	0	0	1	1	0	0	0	1	1
7	–	–	–	–	–	–	–	1	1	1	1	1	1	1	1	1	1	1
8	–	–	–	–	–	–	–	–	0	1	1	1	0	1	1	1	1	1
9	–	–	–	–	–	–	–	–	–	0	1	1	0	0	1	1	1	1
10	–	–	–	–	–	–	–	–	–	–	0	1	0	0	1	1	1	1
11	–	–	–	–	–	–	–	–	–	–	–	1	1	0	0	1	1	1
12	–	–	–	–	–	–	–	–	–	–	–	–	1	1	0	0	0	1
13	–	–	–	–	–	–	–	–	–	–	–	–	–	0	1	1	1	1
14	–	–	–	–	–	–	–	–	–	–	–	–	–	–	1	1	1	1
15	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0	1	1
16	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0	1
17	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0
18	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

does provide improvements in the results; so, in order to set the same configuration parameters for the three curves in the dataset, the authors will use a population size of 500 and a generation number of 2000.

There is another important configuration feature not addressed yet: the initialization process. The use of the default initialization process (choosing randomly the values of each of the genes individually) leads to an initial population which covers only a concrete region of the search space. The ideal initialization process would provide an initial population as spread as possible along the two objective functions (namely the dominant points and the integral squared error). This can be obtained using a two-phase initialization: the first process chooses the number of dominant points randomly (according to the number of points in the curve used) and the second

Table 2. MOEA configurations detail for population size 100.

Config. no.	1	2	3	4	5	6
Population size	100	100	100	100	100	100
Generations	100	300	500	700	1000	2000

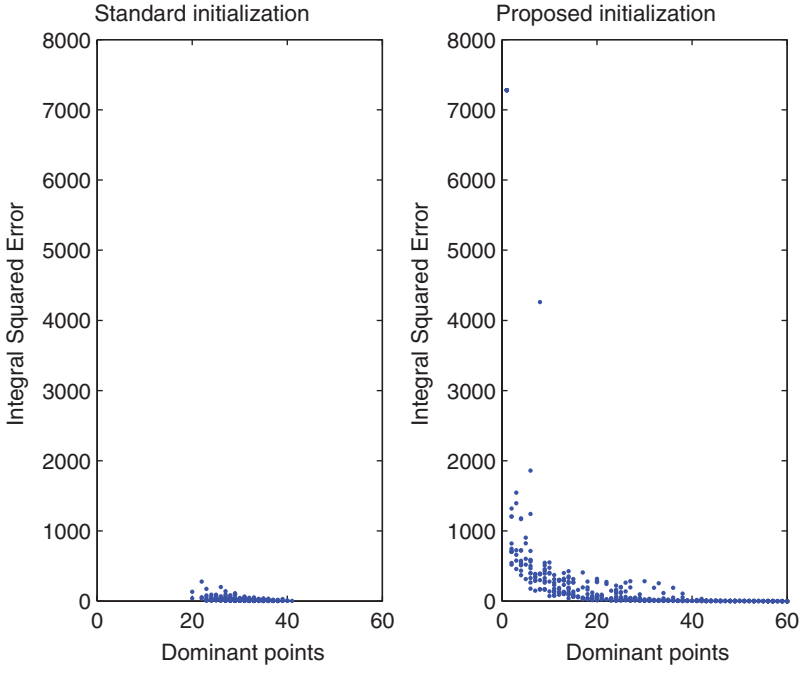

Figure 5. Initialization processes comparison.

one chooses the value of the genes randomly, but according to the previously chosen number of dominant points. This initialization leads to a faster evolution and better final results. The comparison between initial populations obtained with these two different initialization processes is shown in Figure 5. Using the same axis ranges in both initializations, it can be appreciated that the results with traditional initialization processes provide a much more compacted initial population regarding both objective functions, which, as has been pointed out, leads to a worse performance of the optimization algorithm.

The summary of the proposal is presented in Table 3.

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: Marji and Siy (2003), Teh and Chin (2002), Sarfraz *et al.* (2004), Cronin (1999), Ansari and Huang (1991), Ray and Ray (1992), Sarkar (1993), Wu (2003) and finally a special comparison with the evolutionary technique by Yin (1999).

Table 3. Multi-objective segmentation algorithm summary.

Parameter	Description
MOEA algorithm	SPEA2
Representation	Binary vector (0 = non-dominant, 1 = dominant)
Objective functions	2, dominant points and Integral Squared Error
Initialization process	Problem specific
Crossover operator	1-point crossover
Mutation operator	bit-flip mutation
Crossover probability	0.9
Mutation probability	$1/n$
Population size	500
Generation number	2000
Archive size	$n - 1$

Table 4. Freeman chain code representation of the figures in the dataset.

Chromosome
00665 65560 01010 76555 45555 55555 43112 12233 45432 01101 11112 11212
Leaf
66656 65500 01005 66565 50011 06656 56555 55666 76666 66666 42222 22222 22322 44343 33333 23070 00033 23230 70000 33232 22677 72221 27666 11111
Semicircle
00007 00777 77766 76666 66665 76766 56454 43436 66656 55454 44434 33232 22254 54434 23221 21322 22222 21221 11111 0010000

The dataset, along with some segmentation results from the obtained Pareto Fronts, is presented in Figures 2–4. Figure 2 introduces the chromosome curve, which has 60 boundary points, along with five results from the Pareto Set obtained by the technique. Figure 3 shows the same results for the leaf curve (with 120 boundary points), and Figure 4 for the semicircle one (with 102 boundary points). The numerical description of these figures, according to their Freeman chain code (Freeman 1961), is presented in Table 4.

Table 5 presents the concrete results for the Pareto Fronts obtained with the presented configuration of the technique for the three curves of the dataset, showing the number of dominant points in the element and the integral squared error of that element. There are several interesting facts in those results: first of all, the technique is able to find the number of segments that produces a lossless approximation over the different curves. Secondly, the leaf curve Pareto Front approximation results have no value for 55 dominant points, while it reaches its lossless approximation with 56 dominant points. This does not mean that the algorithm was not able to find a solution with 55 dominant points, but rather that the solution found did not improve those found with a lower number of them, and according to that (it is a dominated solution according to Pareto dominance) it was removed from the Pareto Front. Mean and standard values corresponding to 30 different executions are provided for all the different elements in the dataset.

Table 6 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 7 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,

Table 5. Pareto Front dominant points/integral squared error results for the dataset.

	Chromosome		Semicircle		Leaf	
Dom. points	Mean	Std	Mean	Std	Mean	Std
1	7037.33	1329.14	88,648.00	0.00	56,705.63	10,709.98
2	502.67	94.94	11,200.00	0.00	3682.00	695.42
3	343.57	64.89	2436.10	38.57	437.78	82.68
4	133.13	25.14	1093.39	19.67	342.40	64.73
5	96.75	18.56	559.70	30.86	288.78	55.04
6	25.18	4.76	142.49	1.43	217.89	41.57
7	18.84	3.56	116.43	1.31	183.14	34.71
8	12.99	2.45	91.05	1.34	154.78	29.69
9	11.68	2.21	74.07	2.29	134.69	25.54
10	7.80	1.47	60.51	1.64	107.29	20.32
11	6.83	1.29	39.76	3.90	89.89	17.11
12	5.62	1.06	30.71	2.33	63.60	12.79
13	4.65	0.88	23.58	1.48	47.76	9.33
14	4.03	0.76	17.90	0.79	42.45	8.06
15	3.67	0.69	14.60	0.57	31.26	6.14
16	3.34	0.63	13.54	0.11	26.10	4.99
17	3.03	0.57	12.30	0.22	21.17	4.05
18	2.74	0.52	11.23	0.06	16.14	3.07
19	2.47	0.47	10.10	0.10	14.64	2.77
20	2.28	0.43	9.06	0.07	13.05	2.47
21	2.08	0.40	8.10	0.07	11.59	2.20
22	1.89	0.36	7.19	0.09	10.35	1.96
23	1.70	0.32	6.34	0.15	9.19	1.74
24	1.50	0.29	5.57	0.22	8.39	1.59
25	1.31	0.25	4.97	0.26	7.75	1.47
26	1.13	0.22	4.51	0.15	7.13	1.35
27	1.00	0.19	4.15	0.11	6.51	1.23
28	0.88	0.17	3.80	0.11	6.00	1.13
29	0.77	0.15	3.45	0.11	5.54	1.05
30	0.65	0.12	3.10	0.11	5.08	0.96
31	0.54	0.10	2.75	0.11	4.69	0.89
32	0.46	0.09	2.51	0.09	4.33	0.82
33	0.34	0.06	2.32	0.08	3.98	0.75
34	0.30	0.06	2.15	0.08	3.63	0.69
35	0.15	0.03	1.98	0.07	3.37	0.64
36	0.15	0.03	1.79	0.08	3.13	0.59
37	0.00	0.00	1.63	0.07	2.87	0.54
38			1.47	0.06	2.63	0.50
39			1.28	0.08	2.37	0.45
40			1.17	0.03	2.14	0.41
41			1.08	0.00	1.95	0.37
42			0.93	0.02	1.75	0.33
43			0.86	0.00	1.56	0.30
44			0.77	0.00	1.38	0.26
45			0.62	0.00	1.21	0.23
46			0.62	0.00	1.05	0.20
47			0.46	0.00	0.90	0.17
48			0.46	0.00	0.77	0.15
49			0.31	0.00	0.67	0.13
50			0.31	0.00	0.53	0.10
51			0.15	0.00	0.44	0.08
52			0.15	0.00	0.30	0.06
53			0.00	0.00	0.28	0.05
54					0.15	0.03
55					–	–
56					0.00	0.00

Table 6. Results of comparable techniques for the dataset.

	Chromosome		Semicircle		Leaf	
Technique	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 7. Statistical result comparison.

	Chromosome		Semicircle		Leaf	
L Technique	*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

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 7 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 sparse, but without standard implementations of the techniques or a framework to test them properly with novel data, the comparison has resorted to the results in their reference articles, 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.

The evolutionary technique presented in Yin (1999) did not provide results for the leaf curve, so the comparisons will be focused on the remaining two figures of the dataset. Also, as explained in Section 2.3, several solutions with different numbers of dominant points (building artificially a set of solutions similar to a Pareto Front) are presented for each of the curves in its dataset. Tables 8 and 9 present these results for the chromosome and semicircle curves, respectively. In those tables, a statistical significance test is also presented, treating each of the solutions provided by Yin's algorithm individually and with the same parameters used for the comparison with the previous techniques.

Additionally, a statistical comparison from a multi-objective perspective has been carried out. This has been performed with the extraction of the Pareto Fronts contained in Yin's solutions (removing one dominated solution) and the computation of the hypervolume values from those fronts. These hypervolume computations required the choice of the corresponding *nadir* points (the worst possible solution points). These points have been adapted to the portion of the Pareto Front covered by the solutions. For the chromosome curve, their values are 19 dominant points

Table 8. Yin's results and statistical comparison for the Chromosome curve.

Dom. points	ISE	*p*-value	Stat. best
8	17.41	8.67E−11	MOEA
9	13.82	1.02E−05	MOEA
12	7.99	6.06E−13	MOEA
14	5.47	2.82E−11	MOEA
15	5.22	6.00E−13	MOEA
17	4.58	4.17E−15	MOEA
18	4.17	2.93E−15	MOEA

Table 9. Yin's results and statistical comparison for the Semicircle curve.

Dom. points	ISE	*p*-value	Stat. best
10	52.95	2.73E−21	Yin
12	42.85	8.77E−23	MOEA
14	29.93	4.42E−36	MOEA
17	17.41	3.80E−41	MOEA
18	14.80	6.21E−54	MOEA
19	14.94	2.19E−50	MOEA
22	12.91	1.99E−53	MOEA
27	7.04	5.70E−43	MOEA
30	6.61	1.17E−45	MOEA

with 19.15 ISE and for the semicircle curve 31 dominant points with 66 ISE. To obtain these values, 1 was added to the maximum value of dominant points in the fronts, and an additional 10% to the maximum ISE value. To obtain the hypervolume values from the MOEA solutions, one individual is extracted for each one provided by the artificial Yin Pareto Front it is being compared to, building a distinct Pareto Front for each execution. Finally, a *t*-test is run over the hypervolume results to test their statistical significance. It must be noted that the different MOEA runs provided solutions for dominant point values not contained in Yin's solutions, which were not included to make the comparison fairer.

The comparison to Yin's results is shown in Figures 6 and 7. These figures present the whole section of the Pareto Front of the present article according to the highest and lowest number of dominant points presented in Yin's results. This section of the Pareto Front contains more points than the ones included for the statistical multi-objective comparison, according to the procedure previously explained, where only those points with a number of dominant points contained in non-dominated Yin's solutions were included. To facilitate that comparison, the points included in the quality indicators for those Pareto Fronts have been highlighted with a circle marker. The individual comparison to Yin's results shows significantly better results for the MOEA technique in 15 out of 16 cases. This statistical difference is corroborated with the hypervolume results shown in Table 10, where the MOEA technique is significantly better in both curves.

Table 10. Multi-objective hypervolume comparison from the reduced Pareto Front and Yin's algorithm.

Curve	MOEA mean	MOEA std	Yin	*p*-value	Stat. best
Chromosome	0.3690	0.0397	0.3055	1.19E−09	MOEA
Semicircle	0.5089	0.0036	0.4573	3.23E−35	MOEA

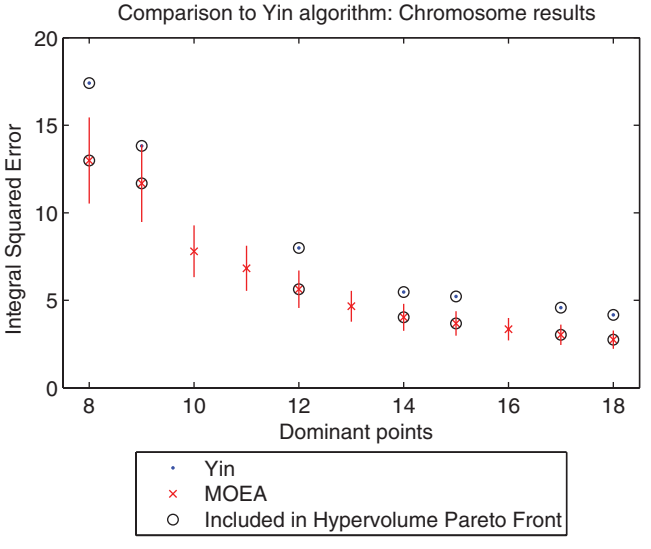

Figure 6. Yin's chromosome results comparison.

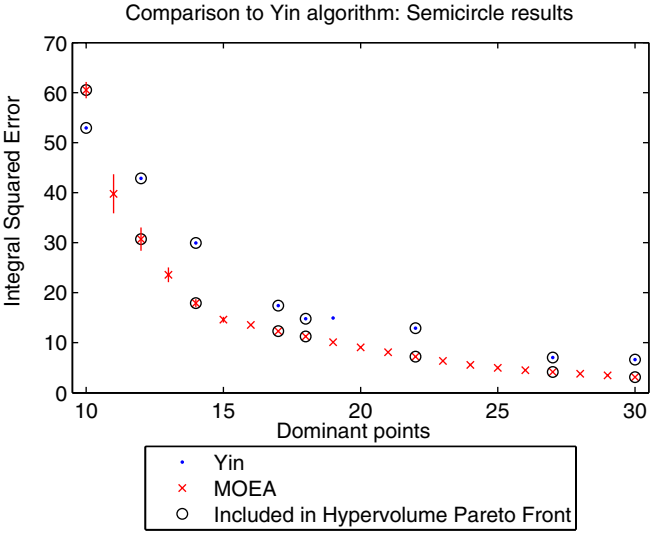

Figure 7. Yin's semicircle results comparison.

The overall results show that the MOEA solution to the segmentation issue is extremely competitive with the available works in the literature in terms of the quality of the solutions obtained on the Pareto Front.

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 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 posteriori* approach using a multi-objective evolutionary algorithm. For the results presentation, the chosen algorithm is SPEA2, along with default variation operator values. The segmentation domain characteristics, along with the representation used, allow the introduction of a specialized initialization operator which, in order to improve the algorithm performance, obtains an initial population with a better coverage of the search space. The population size and number of generation values are chosen according to Wilcoxon test results over a set of possible configurations with increasing 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, each of these solutions being 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 on 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 on 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.

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

Ansari, N. and Huang, K., 1991. Non-parametric dominant point detection. *Pattern Recognition*, 24 (9), 849–862.

Attneave, F., 1954. Some informational aspects of visual perception. *Psychological review*, 61 (3), 183–193.

Coello, C. and Lamont, G., 2004. *Applications of multi-objective evolutionary algorithms*. Singapore: World Scientific.

Coello, C., Lamont, G., and Van Veldhuizen, D., 2007. *Evolutionary algorithms for solving multi-objective problems*. New York: Springer-Verlag.

Corder, G. and Foreman, D., 2009. *Nonparametric statistics for non-statisticians: a step-by-step approach*. Chichester, UK: Wiley.

Cronin, T., 1999. A boundary concavity code to support dominant point detection. *Pattern Recognition Letters*, 20 (6), 617–634.

Deb, K., *et al.*, 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2), 182–197.

Dunham, J., 1986. Optimum uniform piecewise linear approximation of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8 (1), 67–75.
Durillo, J.J. and Nebro, A.J., 2011. jMetal: a Java framework for multi-objective optimization. *Advances in Engineering Software*, 42 (11), 760–771.
Ehrgott, M., 2005. *Multicriteria optimization*. Berlin: Springer-Verlag.
Fourman, M., 1985. Compaction of symbolic layout using genetic algorithms. *In*: *Proceedings of the 1st international conference on genetic algorithms*, Hillsdale, NJ: Lawrence Erlbaum Associates, 141–153.
Freeman, H., 1961. On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers*, 10 (2), 260–268.
Gen, M. and Cheng, R., 1997. *Genetic algorithms and engineering design*. New York: Wiley-Interscience.
Gionis, A. and Mannila, H., 2005. Segmentation algorithms for time series and sequence data. *In*: *Tutorial in the SIAM international conference on data mining*. Newport Beach, CA: SIAM.
Goldberg, D., 1989. *Genetic algorithms in search, optimization, and machine learning*. Boston, MA: Addison-Wesley.
Guerrero, J., *et al.*, 2010. Piecewise linear representation segmentation as a multiobjective optimization problem. *In*: A.P. de Leon, *et al.* eds. *Distributed computing and artificial intelligence: 7th international symposium*, 7–10 September 2010, Valencia, Spain. Advances in intelligent and soft computing Vol. 79. Berlin: Springer-Verlag, 267–274.
Guerrero, J., García, J., and Molina, J., 2010. Air traffic control: a local approach to the trajectory segmentation issue. *In*: *Proceedings of the twenty-third international conference on industrial, engineering & other applications of applied intelligent systems – Part III*, ,. Lecture notes in artificial intelligence Vol. 6098. Berlin: Springer-Verlag, 498–507.
Guerrero, J., García, J., and Molina, J., 2011. Piecewise linear representation segmentation in noisy domains with a large number of measurements: the air traffic control domain. *International Journal on Artificial Intelligence Tools*, 20 (2), 367–399.
Horn, J., 1997. Multicriterion decision making. *In*: *Handbook of evolutionary computation*. Oxford: Oxford University Press.
Keogh, E., *et al.*, 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3 (3), 263–286.
Keogh, E., *et al.*, 2004. Segmenting time series: a survey and novel approach. *In*: M. Last, *et al.*, eds. *Data mining in time series databases*, Series in machine perception and artificial intelligence Vol. 57. Singapore: World Scientific, 1–21.
Keogh, E. and Pazzani, M., 1998. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. *In*: *Proceedings of the 4th international conference on knowledge discovery and data mining*, 27–31 August, New York. Palo Alto, CA: AAAI Press, 239–241.
Langridge, D., 1972. On the computation of shape. *In*: S. Watanabe (ed.), *Frontiers of Pattern Recognition*, 347–366.
Liu, X., Lin, Z., and Wang, H., 2008. Novel online methods for time series segmentation. *IEEE Transactions on Knowledge and Data Engineering*, 20 (12), 1616–1626.
Loncaric, S. and Dhawan, A., 1995. Near-optimal MST-based shape description using genetic algorithm. *Pattern Recognition*, 28 (4), 571–579.
Marji, M. and Siy, P., 2003. A new algorithm for dominant points detection and polygonization of digital curves. *Pattern Recognition*, 36 (10), 2239–2251.
Miettinen, K., 1999. *Nonlinear multiobjective optimization*. Boston, MA: Kluwer Academic.
Mörchen, F., 2003. Time series feature extraction for data mining using DWT and DFT. Technical report 33. Department of Mathematics and Computer Science, Philipps-University Marburg.
Osyczka, A., 1985. Multicriteria optimization for engineering design. *Design Optimization*, (), 193–227.
Pal, N., Kundu, M., and Nandi, S., 1998. Application of a new genetic operator in feature selection problems. *In*: *1998 IEEE Region 10 international conference on global connectivity in energy, computer, communication and control (TENCON'98)*, 17–19 December, New Delhi, India, Vol. 1. Piscataway, NJ: IEEE, 37–40.
Pavlidis, T. and Ali, F., 2007. Computer recognition of handwritten numerals by polygonal approximations. *IEEE Transactions on Systems, Man and Cybernetics*, 5 (6), 610–614.
Ramer, U., 1972. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1 (3), 244–256.
Ray, B. and Ray, K., 1992. Detection of significant points and polygonal approximation of digitized curves. *Pattern Recognition Letters*, 13 (6), 443–452.
Rosenberg, B., 1972. The analysis of convex blobs. *Computer Graphics and Image Processing*, 1 (2), 183–192.
Sarfraz, M., 2008. Linear capture of digital curves. *In*: *Interactive curve modeling*. London: Springer-Verlag, 241–265.
Sarfraz, M., Asim, M., and Masood, A., 2004. Piecewise polygonal approximation of digital curves. *In*: *Proceedings of the eighth international conference on information visualisation (IV 2004)*, 14–16 July, Washington, D.C. New York: IEEE, 991–996.
Sarkar, D., 1993. A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves. *Pattern Recognition Letters*, 14 (12), 959–964.
Sato, Y., 1992. Piecewise linear approximation of plane curves by perimeter optimization. *Pattern Recognition*, 25 (12), 1535–1543.
Singh, M., Chatterjee, A., and Chaudhury, S., 1997. Matching structural shape descriptions using genetic algorithms. *Pattern Recognition*, 30 (9), 1451–1462.
Sklansky, J. and Gonzalez, V., 1980. Fast polygonal approximation of digitized curves. *Pattern Recognition*, 12 (5), 327–331.

Surry, P., Radcliffe, N., and Boyd, I., 1995. A multi-objective approach to constrained optimisation of gas supply networks: the COMOGA method. *In: Proceedings of the AISB-95 Workshop on Evolutionary Computing, 3–7 April*, Sheffield, UK. London: Springer, 166–180.

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

Tsai, Y., 2006. Fast polygonal approximation based on genetic algorithms. *In*: *Proceedings of the 5th IEEE/ACIS international conference on computer and information science and the 1st IEEE/ACIS international workshop on component-based software engineering, software architecture and reuse*. Washington, D.C.: IEEE Computer Society Press, 322–326.

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

Yin, P.Y., 1998. A new method for polygonal approximation using genetic algorithms. *Pattern Recognition Letters*, 19 (11), 1017–1026.

Yin, P., 1999. Genetic algorithms for polygonal approximation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, 13 (), 1061–1082.

Zitzler, E., *et al.*, 2003. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7 (2), 117–132.

Zitzler, E., Laumanns, M., and Thiele, L., 2001. SPEA2: improving the Strength Pareto Evolutionary Algorithm. *In*: K. Giannakoglou, *et al.*, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty, eds. *Evolutionary methods for design, optimization and control with applications to industrial problems (EUROGEN 2001)*, 19–21 September, Athens. Barcelona: International Center for Numerical Methods in Engineering (CIMNE), 95–100.