Image segmentation using Extended Topological Active Nets optimized by Scatter Search

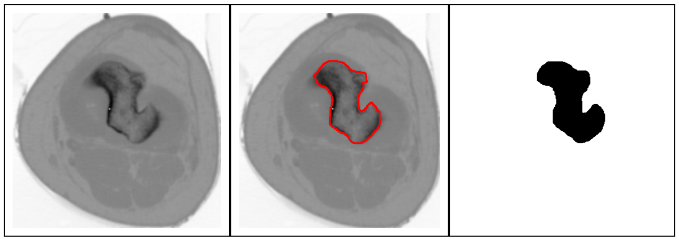
Nicola Bova, Óscar Ibáñez, Member, IEEE, and Óscar Cordón, Senior Member, IEEE

Abstract*—Image segmentation is the critical task of partitioning an image into multiple objects. Deformable Models are effective tools aimed at performing image segmentation. Among them, Topological Active Nets (TANs), and their extension, ETANs, are models integrating features of region-based and boundary-based segmentation techniques. Since the deformation of the meshes composing these models to fit the objects to be segmented is controlled by an energy functional, the segmentation task is tackled as a numerical optimization problem. Despite their good performance, the existing ETAN optimization method (based on a local search) can lead to result inaccuracies, that is, local optima in the sense of optimization. This paper introduces a novel optimization approach by embedding ETANs in a global search memetic framework, Scatter Search, thus considering multiple alternatives in the segmentation process using a very small solution population. With the aim of improving the accuracy of the segmentation results in a reasonable processing time, we introduce a global search-suitable internal energy term, a diversity function, a frequency memory population generator and two proper solution combination operators. In particular, these operators are effective in coalescing multiple meshes, a task previous global search methods for TAN optimization failed to accomplish. The proposal has been tested on a mix of 20 synthetic and real medical images with different segmentation difficulties. Its performance has been compared with two ETAN optimization approaches (the original local search and a new multi-start local search) as well as with the state-of-the-art memetic proposal for classical TAN optimization based on differential evolution. Our new method significantly outperformed the other three for the given set of images in terms of four standard segmentation metrics.

***Index Terms*—Deformable models, topological active nets, extended topological active nets, local search optimization, global search optimization, memetic algorithms, scatter search, differential evolution.**

I. Introduction

IMAGE SEGMENTATION is a key task in image processing aiming at partitioning a digital image into multiple objects which share some common properties. Image segmentation is a critical issue as the quality of its outcomes has a strong influence on the posterior image understanding task. Among its practical applications are medical imaging (where it is employed for tasks such as tumor location, computer-guided surgery, and diagnosis); traffic control systems; object location in satellite images (roads, forests, etc.); and machine vision.



(a) Input (b) DM segmentation (c) Output

Figure 1: Example of segmentation of a medical image by a DM. (a) is the image to be segmented, (b) is the final adjustment of the DM (in red), and (c) is the result image identifying two segments, the object (black) and the background (white).

Deformable Models (DMs) are a promising and vigorously researched computer-assisted image segmentation technique [1]. They have proven to be effective in segmenting digital images by exploiting features of the image data together with *a priori* knowledge about the structures of these images. Since the pioneering work of Kass et al. [2], a number of different kinds of Deformable Models have been proposed. Fig. 1 shows an example of a segmentation task by a DM.

The Topological Active Net (TAN) model is a geometric DM consisting of a discrete implementation of an elastic mesh with interrelated nodes [3]. It integrates features of *region based* and *boundary-based* segmentation techniques. Since the TAN deformation is controlled by an energy functional in such a way that the mesh reaches a minimum when the model is over the objects, the segmentation process is tackled as a numerical optimization (energy minimization) problem. The original optimization strategy is a Best Improvement Local Search (BILS) [4]. The advantage of this model is the capability of fitting the edges of the objects while at the same time detecting their inner topology. Conversely, the model is complex and has limitations regarding topological changes, local deformations, and the definition of the energy functional.

To overcome these limitations while keeping the promising features of TAN, Bova et al. [5] recently presented an Extended Topological Active Net (ETAN) model. The authors developed novel mechanisms tackling topological changes including external and internal link cuts, proposed a new external energy term to properly guide the model in case of complex concavities and highly non-convex shapes, and introduced node movement constraints to avoid crossing links. Besides, they designed a specific local search for ETAN optimization, the Extended BILS (EBILS) procedure, incorporating heuristics to correct the position of eventually misplaced nodes. Although the ETAN model is a significant improvement over

N. Bova, O. Ibáñez and O. Cordón are with the European Centre for Soft Computing, 33600-Mieres, Asturias, Spain.
E-mail: {nicola.bova, oscar.ibanez, oscar.cordon}@softcomputing.es.
O. Cordón is also with the Dept. of Computer Science and Artificial Intelligence (DECSAI) and the Research Center on Information and Communication Technologies (CITIC-UGR), University of Granada, 18071 Granada, Spain.
E-mail: ocordon@decsai.ugr.es.

the TAN, the local search nature of the EBILS optimization method can still make it get stuck in local optima, thus providing suboptimal segmentations.

The main approach developed in the TAN literature to overcome this limitation has been endowing TANs with a global search framework. Actually, two Memetic Algorithms (MAs) [6] were introduced in [7] and [8], respectively based on Genetic Algorithms (GAs) [9] and Differential Evolution (DE) [10]. Although the segmentation results obtained by the MAs for TAN optimization improve the BILS approach, their applicability is still limited in real world images and complex synthetic ones. In particular, those proposals failed to design proper evolutionary operators able to effectively combine nets and consequently required very large populations of solutions to operate, thus negating the main advantage of a global search approach. In addition, they lack a proper energy definition for a global optimization scenario.

The aim of this work is to provide an accurate, quick and robust segmentation technique by endowing ETANs with an effective global search method. To do so, we will embed ETANs in a flexible and powerful memetic framework, Scatter Search (SS) [11], carefully designing specific components to avoid getting stuck in local optima while considering small populations.

The proposed design technique will be tested over a dataset made up of a mix of 20 synthetic and real-world medical images. It will be compared with two other ETAN optimization methods, the original EBILS [5] and a new Multi-Start Local Search (MSLS) designed in this paper, as well as with the state-of-the-art DE-based MA for TAN optimization [8].

The structure of this paper is as follows. Section II introduces both the TAN and ETAN models and summarizes the main global optimization approaches dealing with TANs, focusing on the state-of-the-art evolutionary proposal based on DE. Section III describes the proposed global search framework while Section IV is devoted to the evaluation of the performance of our proposal and comparison with other methods. Finally, Section V summarizes some conclusions and future developments.

II. Background

A. Topological Active Nets

A TAN is a discrete implementation of an elastic two-dimensional mesh with interrelated nodes [3]. The structure of a TAN is depicted in Fig. 2. The model has two kinds of nodes: the external nodes fit the edges of the objects whereas the internal nodes model their internal topology. Hence, this model allows information based on both discontinuities and regions to be integrated in the segmentation process.

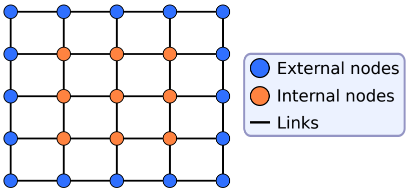


Figure 2: An example 5x5 mesh.

A TAN is defined parametrically as $v(r,s) = (x(r,s), y(r,s))$ where $(r,s) \in [0,1] \times [0,1]$. The mesh deformations are controlled by an energy function defined as follows:

$$E(v(r,s)) = \int_0^1 \int_0^1 \left[E_{\text{int}}(v(r,s)) + E_{\text{ext}}(v(r,s))\right] drds \quad (1)$$

where E_{int} and E_{ext} are the internal and the external energy of the TAN, respectively. The internal energy controls the shape and the structure of the mesh whereas the external energy represents the external forces governing the adjustment process.

The internal energy depends on first and second order derivatives which control contraction and bending, respectively. The internal energy term is defined as:

$$\begin{aligned} E_{\text{int}}(v(r,s)) = & \alpha(|v_r(r,s)|^2 + |v_s(r,s)|^2) + \\ & \beta(|v_{rr}(r,s)|^2) + |v_{rs}(r,s)|^2 + |v_{ss}(r,s)|^2) \end{aligned}$$

where subscripts represents partial derivatives, and α and β are coefficients that control the first and second order smoothness of the net. In order to calculate the energy, the parameter domain $[0,1] \times [0,1]$ is discretized as a regular grid defined by the internode spacing (k,l) and the first and second derivatives are estimated using the finite differences technique [3].

On the one hand, the first derivatives are computed using the following equations to avoid the central differences:

$$\begin{aligned} |v_r(r,s)|^2 &= \left[\|d_r^+(r,s)\|^2 + \|d_r^-(r,s)\|^2\right]/2 \\ |v_s(r,s)|^2 &= \left[\|d_s^+(r,s)\|^2 + \|d_s^-(r,s)\|^2\right]/2, \end{aligned}$$

where d^+ and d^- are respectively the forward and backward respectively, which are computed as follows:

$$\begin{aligned} d_r^+(r,s) &= \left[v(r+k,s) - v(r,s)\right]/k \\ d_r^-(r,s) &= \left[v(r,s) - v(r-k,s)\right]/k \\ d_s^+(r,s) &= \left[v(r,s+l) - v(r,s)\right]/l \\ d_s^-(r,s) &= \left[v(r,s) - v(r,s-l)\right]/l. \end{aligned}$$

On the other hand, the second derivatives are estimated by:

$$\begin{aligned} v_{rr}(r,s) &= \frac{v(r-k,s) - 2v(r,s) + v(r+k,s)}{k^2} \\ v_{ss}(r,s) &= \frac{v(r,s-l) - 2v(r,s) + v(r,s+l)}{l^2} \\ v_{rs}(r,s) &= \frac{v(r-k,s) - v(r-k,s+l) - v(r,s) + v(r,s+l)}{kl}. \end{aligned}$$

The external energy represents the features of the scene that guide the adjustment process. It is defined as:

$$\begin{aligned} E_{\text{ext}}(v(r,s)) = & \omega f[I(v(r,s))] + \\ & \frac{\rho}{|\aleph(r,s)|} \sum_{p \in \aleph(r,s)} \frac{1}{\|v(r,s) - v(p)\|} f[I(v(p))] \end{aligned}$$

where ω and ρ are weights, $I(v(r,s))$ is the intensity value of the original image in position $v(r,s)$, $\aleph(r,s)$ is the neighborhood of node (r,s), and f is a function, that is different for both types of nodes since the external nodes fit the edges whereas the internal nodes model the inner features of the objects. In this situation, function f is defined as:

$$f[I(v(r,s))] = \begin{cases} \gamma \overline{I(v(r,s))_n}, & \text{for internal nodes} \\ I_{\text{max}} - \overline{I(v(r,s))_n} + & \\ \xi(G_{\text{max}} - G(v(r,s))) + & \\ \delta GD(v(r,s)), & \text{for external nodes} \end{cases}$$

γ, ξ and δ are weighting terms, I_{max} and G_{max} are the maximum intensity values of the image I and the gradient image G, respectively, $I(v(r,s))$ and $G(v(r,s))$ are the intensity values of the original image and the gradient image in node position $v(r,s)$, and $\overline{I(v(r,s))_n}$ is the mean intensity in a $n \times n$ square mask[1]. If the objects to detect are dark and the background is bright, the energy of an internal node will be minimum when it is on a point with a low gray level. The energy of an external node will be minimum when it is on a discontinuity and on a light point outside the object. Notice that, since the work of Ibáñez et al. [7] the external energy also includes the gradient distance term, $GD(v(r,s))$, that is, the distance from position $v(r,s)$ to the nearest edge. This term introduces a continuous range in the external energy since its value diminishes as the node gets closer to an edge.

The adjustment process consists of minimizing these energy functions. In the original TAN proposal [4], the mesh is placed over the whole image and, then, the energy of each node is minimized using a BILS algorithm (called greedy search in previous papers). In each step of the algorithm, the energy of each node is computed in its current position and in its nearest neighborhood. The position with the lowest energy value is selected as the new position of the node. The algorithm stops when there is no node in the mesh that can move to a position with lower energy.

B. Related works: TAN global optimization

Evolutionary Computation has been broadly applied in medical image segmentation during the last thirty years [12]. In particular, evolutionary-based approaches for DMs optimization recently gained much attention [13], [14]. However, after more than twenty years of TAN development there are just a few proposals for its optimization due to the inherent complexity of the model. In [7] the authors defined and adapted the classical genetic operators to deal with this problem, with emphasis on a mutation operator that produces models with no crossings in their definition nodes. The GA showed superiority in the minimization of the probability of getting stuck in local minima, especially with noisy images. Later, they proposed a MA that hybridizes the previous GA and the BILS [4]. That method clearly outperformed the accuracy of the segmentation of the previous proposals. However, it required a high computation time as a consequence of the huge population sizes required to operate properly. Additionally, in [15] the authors used a multi objective evolutionary approach to avoid the complex energy parameter tuning.

The proposal by Novo et. al in [8] is probably the state of the art in TAN optimization. It consists of a MA that hybridizes a DE [10] and the BILS in [4]. The classical DE operator, based on discrete recombination and differential mutation [10], is considered to create new candidate solutions. The mutation scale factor F is experimentally established as a random value between 0.2 and 0.6, specifically computed for each node. The crossover rate CR is fixed to 1. The base vector x_1 is selected using tournament selection of size 3% of the population size, which was established to 1000 individuals, thus keeping the requirement of handling large size populations.

The BILS algorithm is applied to every mesh in the population only in specific moments of the evolutionary process, typically every 10 generations, with a random number of steps in $[0,4]$. A more frequent and/or deeper (in the sense of number of steps) application would require a non affordable computation time without a significant improvement of the results. The BILS not only achieves the best local adjustment but also allows topological changes in the TANs. However, the topological changes are only enabled for the best individual of the population. Then, the resulting topology is extrapolated to the entire population. The presence of just one topology in such a large population is due to the incapacity of the crossover operator to produce feasible offspring nets (e.g. without crossing links) by combining parents with different topologies.

C. Extended Topological Active Nets

This new Deformable Model extends TANs in several ways [5]. The following sections give a summary of their main features, as well as of the EBILS proposed to optimize them.

1) External energy: ETANs employ a new external energy term to guide the mesh in case of complex concavities and highly convex shapes. It is calculated for every pixel p from the Extended Vector Field Convolution (EVFC) [16] by equalizing a distance to gradient image DG_{evfc} such that:

$$DG_{\text{evfc}}(p) = \frac{1}{\sum\limits_{q \in \aleph_w(p)} |q|}, \tag{2}$$

where q is a vector of the EVFC field belonging to the squared neighborhood of size w of the pixel p. Fig. 3 shows an example of the construction of this image. Therefore, the final energy function becomes the one in Section II-A changing the term $GD(v(r,s))$ by DG_{evfc} in Eq. (2).

2) Topological changes: If the shape of the object(s) calls for the need of cutting links and, eventually, changing the topology of the ETAN, it is necessary to properly adapt the structure of the net. The previously existing solution [7] does not take into account the underlying image and it cannot open holes into the mesh.

To solve these issues, the underlying idea in ETANs is to cut the links which bear an energy higher than a threshold. The energy of a link is calculated as

$$E_{link} = \left(\sum_{p \in A} DG_{\text{evfc}}(p) \cdot \frac{I(p)}{I_{\text{max}}} \right) \Big/ |A|, \tag{3}$$

where p is a pixel belonging to the area A over which the energy is computed, I is the original image, I_{max} is the

[1] Here and in the rest of the paper, the part of the mask falling out of image boundaries is ignored when approaching those limits.

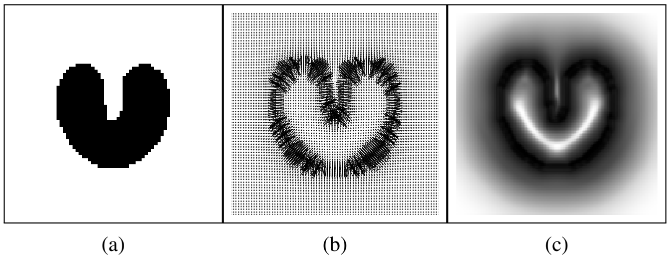


Figure 3: (a) original image; (b) the EVFC of (a); (c) equalized energy term derived from (b).

maximum intensity value in the original image, and $|A|$ is the size of the area A.

The cutting threshold is not fixed, in fact it depends on the mean energy the links of a mesh are experiencing, E_{mean}. However, not all links in a mesh can be cut, even if they bear a high energy, as the topology of the net can be damaged [7]. Therefore the only links whose energy is measured and contributes to E_{mean} are those which passed some *cuttability tests* and can actually be cut [5].

The holes in a mesh are recognized starting from misplaced *internal* nodes. For every internal node n, $r(n)$ is derived as

$$r(n) = \frac{E_{\text{ext}}(n)}{E_{\text{ext}}(n) + E_{\text{int}}(n)}, \qquad (4)$$

where $E_{\text{ext}}(n)$ is the external energy and $E_{\text{int}}(n)$ is the internal energy of node n. The node n_h with the highest ratio is selected and if $r(n_h) > th_{\text{holes}}$, a hole is opened in the net starting from this node. If this is the case, the values of the energy of the links connecting n_h and its neighbors are calculated and the highest one is chosen. This internal link is removed and the mesh can now cut more links and adjust to the internal edge of the object.

3) ETAN optimization process: The adjustment of the mesh to the object is a procedure that comprises several steps. After the mesh initialization, the first step is an EBILS algorithm that optimizes the position of every node in the mesh. To do so, a new location (i.e. a pixel on the image) with a lower energy is searched for each node in a square window centered on it, according to Eq. (1). If a better position is found, the node is immediately moved to this location. The search is performed sequentially from the first to the last node of the net and the whole process is repeated until no node can be further moved. To avoid crossings of the links connecting nodes, EBILS test if the node is located outside the safe area, that is the polygonal area delimited by the node neighbors, repositioning it if that is the case. When it is not possible to stretch or compress the mesh anymore, the link cutting procedure is activated. A specific heuristic procedure is called after this phase to correct the position of *gradient-misplaced* nodes. At this point the mesh should be adjusted to the contour of the object and it is now possible to segment the holes which eventually exist inside the object. The last step of the process is the activation of a less constrained version of the cutting procedure to finalize the segmentation. Fig. 4 depicts the overall scheme, that will be used as the local optimization component in the SS-based method proposed in this contribution.

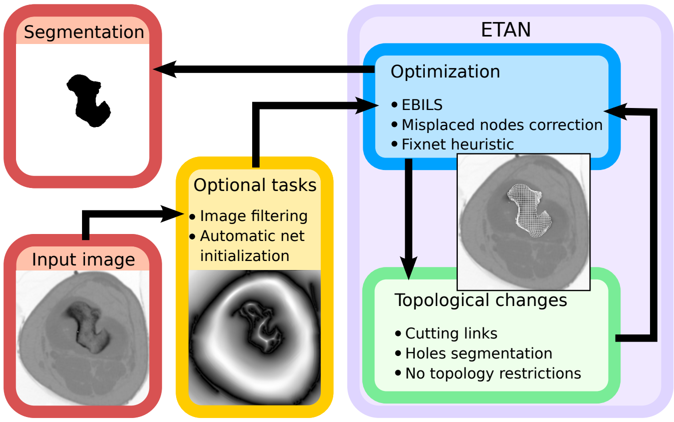


Figure 4: The segmentation process using ETAN and EBILS.

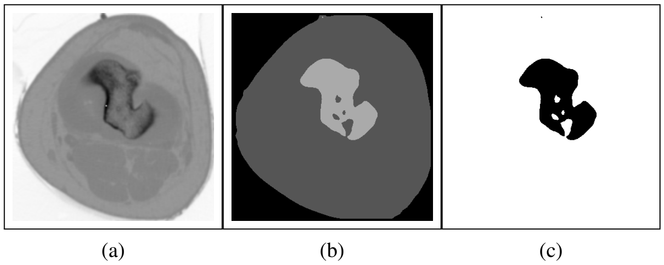


Figure 5: (a) original image; (b) resulting image after clustering image (a); (c) resulting segmentation from (b).

4) Image filtering: The gradient and distance to the gradient images are usually constructed from the application of an edge detector over the the original image. As an alternative, ETANs can use a K-means clustering generated pre-segmentation [5] (see Fig. 5). Moreover, the bounding box of this pre-segmentation can be used to automatically initialize the mesh.

III. A Scatter Search Framework for Extended Topological Active Nets optimization

The results obtained by ETANs were encouraging [5]. They outperformed TANs and state-of-the-art snake models [16], while needing lower computational resources. Moreover, the robustness achieved was significantly better than the previous TAN method, and the ETAN together with the EBILS optimization procedure less sensitive to parameter values changes.

However, since the ETANs were optimized using a local search procedure, the model can reach wrong segmentations, local minima from the optimization viewpoint, due to the presence of noise and/or artifacts or simply to the complexity found in the images. Fig. 6 shows two of these cases. A feasible solution is to complement the EBILS optimizer with a

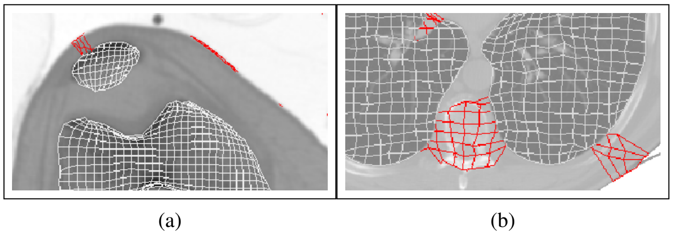


Figure 6: Two cases of ETANs inaccuracies (in red).

global search. Indeed, such a global optimizer could consider multiple meshes at the same time, combining them to generate more accurate ones until approaching the global minimum of the energy function.

In this section we describe our proposal of such an ETAN global search method. First of all, we introduce the basis of SS. Then we provide a motivation for the use of this specific MA. Finally, we deal with the customization of the SS general framework to fit our ETAN optimization problem, describing every designed component in detail.

A. The Scatter Search template

SS fundamentals were originally proposed by Fred Glover [17] and later developed in [11]. The main idea of this technique is based on a systematic combination between solutions (instead of a randomized one like that usually done in GAs) *taken from a considerably reduced evolved pool of solutions named Reference set* (between five and ten times lower than usual GA population sizes) as well as on the typical use of a local optimizer. This way, an efficient and accurate search process is encouraged thanks to the latter and to other innovative components we will describe below.

The fact that the mechanisms within SS are not restricted to a single uniform design allows the exploration of strategic possibilities that may prove effective in a particular implementation. These observations and principles led the authors in [11] to propose the following "five methods" template for implementing SS:

1) A *diversification generation method* to generate a collection of diverse trial solutions.
2) An *improvement method* to transform a trial solution into one or more enhanced trial solutions.
3) A *Reference set update method* to build and maintain a Reference set consisting of the b "best" solutions found in terms of quality and diversity.
4) A *subset generation method* to operate on the Reference set, to produce a subset of its solutions as a basis for creating combined solutions.
5) A *solution combination method* to transform a given subset of solutions into one or more combined solution vectors.

The SS design starts with the creation of a population P, of size P_{size}, by means of the diversification generation method. The solutions, generated by a diversity criterion, are enhanced by the improvement method. Then, a total of b solutions are selected to form the *RefSet*, that is partitioned into two subsets of high quality and diverse solutions. This strategy is called a two-tier design [11] and is used to proactively inject diversification into the search. The *quality subset* of the RefSet, *RefSet1*, is created by selecting the b_1 ($b_1 \leq b$) fittest solutions from P, from which they are removed. For every solution remaining in P, the minimum of the distances from the elements in RefSet is calculated, according to a proper *diversity function*. Then, the solution with the maximum of these minimum distances is removed from P and inserted in *RefSet2*, the *diversity subset* of the RefSet. Finally, all the distances are updated. This procedure is repeated $b_2 = b - b_1$ times, until completing RefSet2.

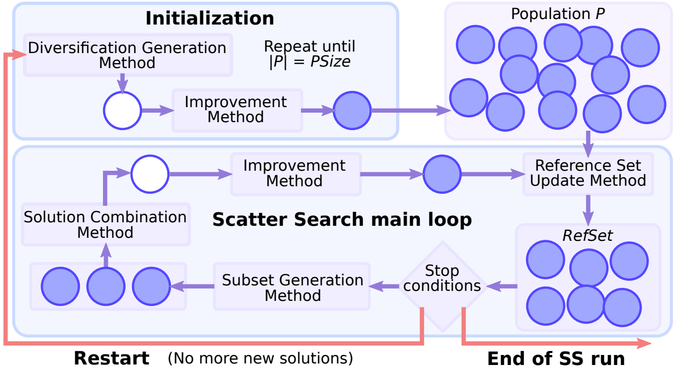


Figure 7: The control diagram of SS.

The subset generation method builds different subsets of RefSet solutions to be later combined to produce new ones. This method is not limited to combining just pairs of solutions. However, it has been shown in [18] that the simple design considering all the subsets of two solutions usually provides a very high performance. Therefore, there is no need to consider subsets of larger cardinality as recommended in early SS designs.

The next step is combining the solutions of each subset by means of the solution combination method. Since the SS framework does not rely on mutations, like other evolutionary approaches do, this is a crucial component of the process. In fact, an effective combination method is often heavily problem-dependent.

Finally, the combined solutions are enhanced by the improvement method and processed by the Reference set update method which will refresh the RefSet according to quality and diversity metrics. If none of the combined solutions is worth inserting in the RefSet, SS performs a restart: only the best solution is kept and P is created again. The whole process ends considering problem-specific stop conditions. Fig. 7 shows the interaction among the five methods and puts in evidence the central role of the RefSet.

B. Motivation for the use of Scatter Search

Despite the huge dimensionality of the search space in our problem, we can rely on an effective local search, EBILS (see Sec. II-C3). In fact, ETANs outperformed TANs in every experiment we did in [5]. The counterpart is that ETAN EBILS-based optimization process is two orders of magnitude slower than the TAN-BILS one. These time constraints have to be taken into account when designing *how* our global search will consider multiple alternatives in the segmentation process.

The simplest option we implemented was a MSLS [19]. In this case, we initialized a large number of meshes, different in size and location in the image plane, run the EBILS on them and chose, as final result, the one with the lowest energy. This method is simple and fast, but it lacks the capability of mixing solutions. In our case, this problem is particularly evident in the case of images with multiple, distant objects. In fact, it is unlikely that a single mesh is able to divide and move toward distant objects, without getting stuck in local minima, e.g. noise, along the way. Other more advanced approaches as *iterated local search* and *variable neighborhood search* would show the same problem [19].

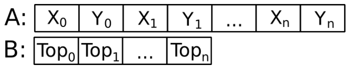
Another possibility could be to use an evolutionary algorithm, or even better, its hybridization with a local search, a MA [6]. Contrary to the MSLS, they have the ability of combining candidate solutions and of improving them to provide high quality candidates, thus providing a better intensification-diversification tradeoff. According to [8], the segmentation results obtained by the DE-based memetic approach (using TANs) improve previous global and local proposals. However, it implies a high computational cost. Indeed, it is a huge effort since the employed evolutionary algorithm used a population of 1000 individuals, each one storing a complete mesh and relative topology. Actually, the proposed DE applies the BILS just every 10 generations, and for only a few steps, because the authors needed to reduce the heavy computational burden implied by those huge populations.

There are two main reasons for the need of such large populations: firstly, the absence of an adequate combination method. The existing ones employ simple flavors of arithmetical combination operators which prove to be inefficient in combining meshes with moderate differences in shape, or even small differences in topology. In fact, the decision to force a single topology in [8] or to employ niches of topologies in [7] arise from the inability to deal with meshes with different topologies. This implies that those proposals fail in segmenting objects with complex shapes or, even worse, multiple, distant objects, above all when separated by heavy noise. Ad-hoc combination procedures to generate feasible and usable offsprings are then strongly required.

Secondly, large populations are required due to the evolutionary framework employed, where randomized combinations of individuals are considered. Within the large umbrella of MAs [6], SS is endowed with some specific and very attractive capabilities [11], as introduced in the previous section. In fact, SS has been successfully applied to other computer vision tasks, such as image registration [20]. The SS methodology is very flexible since each of its elements can be implemented in a variety of ways and degrees of sophistication. Besides, the SS approach relies on systematically injecting diversity in the RefSet to achieve better exploration and, therefore, avoiding the need for a large population. Considering the time constraints imposed by ETANs (which are effective but slower than TANs), employing a reduced population of high-quality solutions is quite an advantage in order to deal with a global search for our problem. Thanks to this fact, the EBILS can be deeply applied at every generation, thus getting a better intensification-diversification tradeoff. The aggregation of this intensive EBILS application and a problem-specific solution combination method allowing us to properly mix nets with different topologies would become a very convenient way to deal with ETAN optimization. These are the reasons why we considered SS the metaheuristic which fits our problem the most.

C. Scatter Search-based ETAN optimization implementation

Each individual in the SS population encodes a different ETAN definition, using a double encoding, A and B, as shown in Fig. 8. The A part stores two real values, the x and y coordinates in the image plane for every node in the mesh. The B part stores the topology of the net for every node, that is, the presence or absence of links to the four possible neighbors and the type of node (external or internal).



A:	X_0	Y_0	X_1	Y_1	...	X_n	Y_n
B:	Top_0	Top_1	...	Top_n			

Figure 8: The coding scheme of our SS proposal.

The overall scheme of the designed algorithm is shown in Fig. 9. The first step is performing some *preliminary operations* to incorporate image-specific information into the SS process, with two objectives: i) generating proper *gradient* and *distance gradient images*, which are employed in the external energy term; and ii) achieving a rough *pre-segmentation*, by the K-means clustering described in Sec. II-C4, whose result is employed to define the reference net size and to bias the population initialization.

The *diversification generation method* is implemented using a *frequency memory*, with the purpose of creating an initial population of diverse meshes, P. The generated solutions, coherently with the SS framework, are enhanced through the use of the *improvement method*, in our case the whole EBILS procedure described in Sec. II-C3. Then, the *Reference set update method* selects the best meshes in terms of quality and diversity and insert them in the RefSet. We thus consider a two-tier RefSet approach (see Sec. III-A).

In the next step, the *subset generation method* generates all possible solution pairs to perform *structured combinations* of them by means of the *solution combination method*. The obtained results are also enhanced applying the improvement method, i.e. the EBILS. The best solutions obtained are selected to replace the worst ones in the Reference set.

The main SS loop is repeated until one of the following events happens:

- the RefSet did not change in the last iteration;
- the diversity among *RefSet1* solutions is below a threshold;
- a new population has not been generated in the last th_{pop} iterations.

Then, a restart is performed. All but the best solution are removed from the RefSet and a new base population is reinitialized in order to inject diversity. The algorithm stops if the fitness of the best individual did not improve after a given number of restarts (N_R) or the maximum number of SS iterations (N_{SS}) has been reached.

The remaining specific SS components are described in the next subsections.

D. Objective function definition: internal energy terms

When tackling image segmentation as an energy minimization task, the correlation between fitness function and segmentation quality plays a critical role. Indeed, if this correlation is

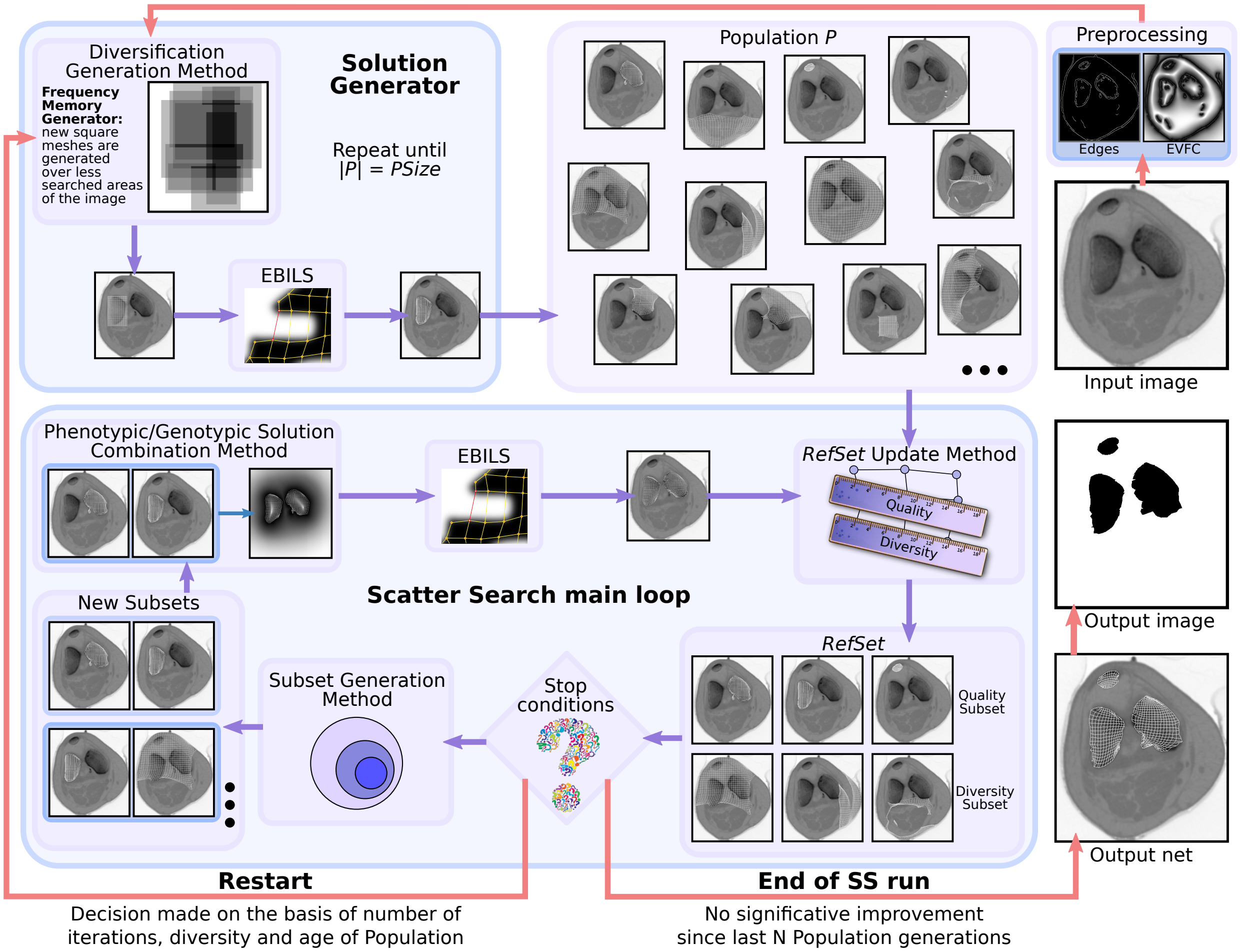

Diversification Generation Method
Frequency Memory Generator: new square meshes are generated over less searched areas of the image
Solution Generator
Repeat until |P| = PSize
EBILS
Population P
Preprocessing
Edges
EVFC
Input image
Phenotypic/Genotypic Solution Combination Method
EBILS
RefSet Update Method
Quality
Diversity
Scatter Search main loop
New Subsets
Subset Generation Method
Stop conditions
RefSet
Quality Subset
Diversity Subset
Output image
Output net
Restart
Decision made on the basis of number of iterations, diversity and age of Population
End of SS run
No significative improvement since last N Population generations

Figure 9: The complete segmentation process using ETANs and SS.

loose, even the perfect optimizer would lead to suboptimal segmentation results. Although the mesh adjustment to the object contour (local search) and the evaluation of a segmentation through its mesh position (global search) are both tackled as minimization problems, they actually are quite different tasks. Hence, it is not surprising that the most suitable fitness functions for the two tasks do not match. As a consequence, we employ two different energy function definitions.

The fitness function is the sum of the internal and external energy of every node (see Sec. II-A and Eq. (1)). While we opted for the same external energy formulation for both local and global searches, the internal energy function has been redesigned for the global search in order to solve some specific problems which are described below.

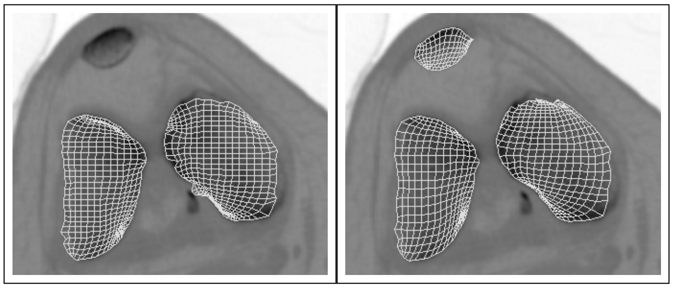
In fact, in (E)TANs (including previous evolutionary TAN optimization proposals) the energy function is derived from the original formulation, designed with the consideration of a local minimization approach. In that formulation, the contraction term gives energy values directly proportional to the distance among the nodes, thus forcing the mesh to contract. The contraction is stopped by the external nodes in presence of edges. However, the contraction term is not suitable in a global search framework because it penalizes big nets regardless of the size of the target segmentation object. To deal with this issue we propose to substitute the contraction term of the internal energy with an area-related one. Different from the other terms which are calculated on a per-node basis, this term only takes into account the total area of the meshes. Its magnitude is proportional to the ratio of the area of the candidate net $A(n_c)$ (the net whose fitness is being calculated) and the reference area A_r, taken from the K-means pre-segmentation (see Sec. II-C4). The obtained value of the ratio is the input of a proper function f_α, computed as follows:

$$E_{int_\alpha, n_t} = f_\alpha\left(\frac{A(n_c)}{A_r}\right); \; f_\alpha(x) = \begin{cases} \psi\frac{1}{x} - \psi & \text{if } x < 1 \\ \psi x - \psi & \text{if } x \geq 1, \end{cases}$$

where ψ is a constant (with a typical value of 5).

Another important change we performed is the removal of the bending term in the global search fitness function. The aim of this term is ensuring a smooth mesh shape and this plays a central role in the local adjustment to the object, helping to keep the net together. However, this term is not suitable for the evaluation of the meshes in a global search framework. It strongly penalizes meshes which divide into parts to adapt their topology to the objects, in particular when the segmentation target is composed of many objects.

In some cases, the desired segmentation is made up of several objects of different sizes. If the size of the smaller objects is negligible compared to the bigger ones, the contribution provided by the global area energy term is not enough to distinguish among nets segmenting only big objects and nets segmenting all the objects. This implies that, in these cases,



(a) A mesh with 2 submeshes (b) A mesh with 3 submeshes

Figure 10: The submesh reward procedure. For the image in (a): α-term = 20, E_{ext} = 310, reward = -66, total = 264. For the image in (b): α-term = 10, E_{ext} = 350, reward = -108, total = 252. With the reward, (b) has a lower energy than (a).

the fitness values are only dependent on the external energy value. Moreover, if the smaller objects are brighter[2] than the bigger ones, the fitness values of the meshes segmenting them will be worse than the fitness values of meshes segmenting only the bigger, darker objects.

The solution we propose involves the use of *submeshes*, i.e. isolated parts of a mesh resulting from topological changes to adapt to more than one object. Every node in a mesh is part of only one submesh. When evaluating the fitness of a mesh, we calculate its *submesh relevance*, that is $g = \sum_i g_i$, where g_i refers to every submesh the mesh is divided into. For every submesh i, $g_i = 1$ if the following three conditions hold:

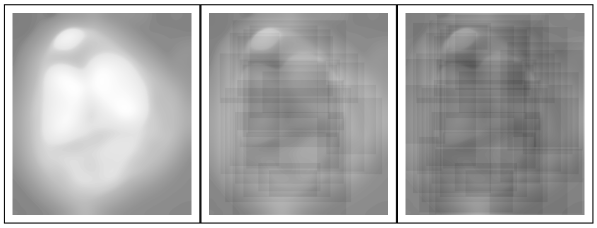
1) the ratio between the covered area s_i and the total area of the image is above a threshold, $G_{\text{min}} \in (0, 1)$;
2) the ratio between the area of the *bounding box* containing the submesh and its area is lower than a threshold, G_a;
3) the form factor of the bounding box, defined as the ratio between the longest and shortest edges of the rectangle, is below a threshold, G_{ff}.

The G_{min} threshold defines the minimum area of a structure in the image to be considered an object. Together with the net reference size, they are the only kind of *prior knowledge* we insert into the process, thanks to the SS flexibility to do so. If one of the three previous conditions does not hold, then $g_i = -s_i/G_{\text{min}}$. Experimentally found proper values for the three thresholds are 0.005, 8 and 5, respectively.

Once the submesh relevance g has been determined, we reward the net on the basis of g, calculating $f_r = f \cdot (1 - r_g \cdot g)$, where f is the fitness value calculated so far, f_r is the fitness value considering the submesh reward, and r_g is a weighting coefficient in $(0, 1)$. The rationale is rewarding (that is, lowering the fitness value of) the meshes with many relevant submeshes and penalizing the meshes with many non-relevant ones. In particular, the non-relevant submeshes penalize the mesh depending on their size: the larger, the worse. An experimentally found proper value for r_g is 0.1. Fig. 10 shows an example of the submesh reward term.

E. Diversity function

Our diversity function is meant to measure how much a net solution is different from another one, with the aim of selecting candidate nets for the diversity subset of the RefSet. We intend this subset as a *reservoir* of meshes which are not good enough to be considered as a possible outcome of the segmentation process but that, nevertheless, segment some objects (or parts of them) in a proper way. Moreover, these objects should be located far away, on the image plane, from the ones segmented by the nets in the quality subset of the RefSet. With this in mind, we designed the following diversity function:



(a) k-means bias (b) After 20 meshes (c) After 40 meshes

Figure 11: The frequency-memory population initialization.

$$d(m_1, m_2) = \frac{\sum_{i=1}^{n} \sqrt{(m_{1,i,x} - m_{2,i,x})^2 + (m_{1,i,y} - m_{2,i,y})^2}}{E_{\text{ext}}(m_1) + E_{\text{ext}}(m_2)},$$

where m_1 and m_2 are two meshes, n is the number of nodes in a mesh, and $m_{a,i,k}$ is the component $k = \{x, y\}$ of the ith node of mesh a. In this way, the numerator of $d(\cdot)$ implies that the farther the meshes are located on the image plane, the higher the $d(\cdot)$ measure will be. Besides, the denominator implies that the poorer the meshes adjustments, the lower the $d(\cdot)$ measure. We only consider the external energy because we are especially interested in small, distant objects. In fact, the internal energy of nets only segmenting small objects is usually high because their areas are probably different from the reference area.

F. Diversification generation method

Our diversification generation method employs controlled randomization and frequency-based memory, typically used in SS, to generate an initial set of diverse and good quality solutions. When a population is initialized, we generate P_{size} candidate rectangular nets with uniformly spaced nodes. The EBILS is applied to each of them and we select the b_1 fittest nets and the b_2 most diverse nets to form the RefSet.

Each new mesh M_n is generated by randomly selecting the size and the ratio between the two sides of the rectangle. Then, we place M_n in the image plane on the basis of the frequency-memory image, I_{fm}, that has the same size of the image to be segmented. To keep record of the already searched areas, we lower the intensity values of I_{fm} pixels covered by every mesh we generated. Therefore, we always place M_n over the brightest area of I_{fm} in which the new mesh could fit.

To improve the convergence time of the segmentation process, we bias the search toward the most promising areas by initializing I_{fm} with a blurred version of the K-means presegmentation (see Fig. 11(a)).

As the brightest area of I_{fm} is always chosen and its intensities lowered, after the generation of some meshes, I_{fm} becomes flatter. Therefore, new meshes are placed over the image more uniformly (see Fig. 11(b,c)), permitting the exploration of the whole search space.

[2] In this article the target objects are dark and the background is bright.

G. Solution combination method

In previous works as [7], [8], either the DE operator [10] or the arithmetic crossover [21] were employed for TAN combination. The formulation of the latter is as follows:

$$m_{o,i} = \theta m_{1,i} + (1-\theta) m_{2,i},$$

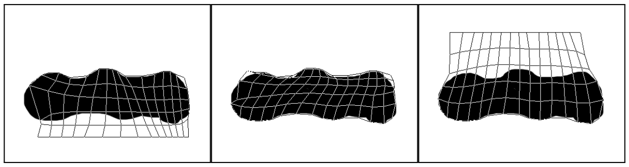
where $m_{a,i}$ is the ith node of the mesh a and θ is a real number randomly generated in $[0,1]$, the same for all the nodes of the two combined meshes.

Unfortunately, this operator is only useful at the very beginning of the search process, producing nets worsening their parents' fitnesses whenever the search process starts to converge. In addition, it does not incorporate the same information the parents hold and infeasible offsprings are obtained when combining nets with different topologies. To overcome this problem, we propose an advanced solution combination method based on two different solution combination operators (SCOs):

1) The genotypic SCO: The rationale of this operator is trying to combine two nets which perform a good segmentation of different parts of the object(s). Regardless of the topology of the parents, the combination will have a basic topology, with every link in place. The EBILS, always called after the combination, will eventually take the function of cutting links and/or open holes in the mesh.

The genotypic SCO calculates a different θ (a combination weight) for every pair of homologous nodes of the parent nets. This value is inversely proportional to the local energy of the nodes, in such a way that the location of the corresponding offspring node will be more similar to the parent node with a lower local energy (hence, it works as a heuristic real-coded crossover [21]). The θ weights are only derived for the external nodes located in the four edges of the net. The genotypic SCO performs a "boost" of the combination weights by means of the f_{cw} function (Eq. (5)) to further increase high θ values and further decrease lower θ values. The idea is to keep the position of a parent node placed over an edge in the offspring net, since a final node position depends on the θ value of both parents. Moreover, the weights are smoothed substituting them with the mean of their external neighbors (including the node itself) to prevent link crossings. The relations are:

$$\begin{aligned}
\theta_i &= \frac{e_{2,i}}{e_{1,i}+e_{2,i}} \\
\theta_{b,i} &= f_{cw}(\theta_i) = \frac{1}{2}\sin\left(\pi\theta_i - \frac{\pi}{2}\right) + \frac{1}{2} \qquad (5) \\
\theta_{\mu,i} &= \overline{\theta_{b,\aleph_E(i)}}; \quad m_{o,i} = \theta_{\mu,i} m_{1,i} + (1-\theta_{\mu,i}) m_{2,i}.
\end{aligned}$$



(a) P1 (42474) (b) Offs. (31781) (c) P2 (69772)

Figure 12: The genotypic SCO. (a) and (c) show the parents, while (b) depicts the offspring. Net energies in parenthesis.

The θ weights obtained with this procedure for the parent nets shown in Fig. 12(a,c) are shown in Fig. 13(a). These θ_i weights multiply the parent 1 net, while those multiplying the parent 2 are computed as $1-\theta_i$. The weights for the x coordinates are obtained interpolating the extremes of the row while the weights for the y coordinates are calculated interpolating the extremes of the column. The results of the interpolations for the weights of Fig. 13(a) are shown in Fig. 13(b,c).

Finally, Fig. 12(b) shows the result of the genotypic SCO applied to the nets shown in Fig. 12(a,c). Note how the energy of the offspring net (shown in the caption) and the segmentation obtained are better.

	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15
r1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.95	0.95
r2	0.82	-	-	-	-	-	-	-	-	-	-	-	-	-	0.95
r3	0.49	-	-	-	-	-	-	-	-	-	-	-	-	-	0.89
r4	0.16	-	-	-	-	-	-	-	-	-	-	-	-	-	0.58
r5	0.00	-	-	-	-	-	-	-	-	-	-	-	-	-	0.25
r6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.02

(a) Weights of the external nodes for the net in Fig. 12 (6 × 15 nodes).

	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15
r1	1.00	0.99	0.99	0.99	0.98	0.98	0.98	0.97	0.97	0.97	0.97	0.96	0.96	0.96	0.95
r2	0.82	0.83	0.84	0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.92	0.93	0.93	0.94	0.95
r3	0.49	0.52	0.55	0.58	0.61	0.63	0.66	0.69	0.72	0.75	0.78	0.81	0.84	0.87	0.89
r4	0.16	0.19	0.22	0.25	0.28	0.31	0.34	0.37	0.40	0.43	0.46	0.49	0.52	0.55	0.58
r5	0.00	0.02	0.04	0.06	0.07	0.09	0.11	0.12	0.14	0.16	0.18	0.19	0.21	0.23	0.25
r6	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02

(b) Interpolation of the external weights along the columns (x coordinate).

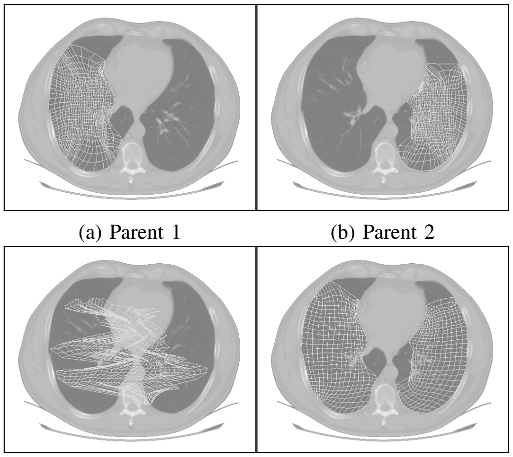
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15
r1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.95	0.95
r2	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.76	0.77
r3	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.57	0.58
r4	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.38	0.40
r5	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.21
r6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.02

(c) Interpolation of the external weights along the rows (y coordinate).

Figure 13: The weights of the genotypic SCO for Fig. 12(a,c)

2) The phenotypic SCO: Despite its good performance in combining good segmentations of different parts of the object(s), the genotypic SCO does not perform properly when trying to combine nets which perform good segmentations of different objects. Fig. 14(c) shows an example.

As a solution to this problem, we propose a *phenotypic* SCO. This SCO tackles the problem of combining two meshes with a top-down approach, the opposite of the bottom-up



(a) Parent 1 (b) Parent 2

(c) Genotypic offspring (d) Phenotypic offspring

Figure 14: The results of the two SCOs on nets which perform good segmentations of different objects.

approach of the genotypic SCO. The first step is to derive the segmentation images of the two parents, as if they were the final results of the process. A two-step filtering is applied to the binary images in order to remove the “segmentation noise”. First, a submesh filtering is applied to remove any submesh which is not considered *relevant*. Second, a morphological closing followed by an opening are applied to further smooth the resulting shape.

The union of the resulting two binary images is calculated, merging them through a logic OR. The following step is adjusting a mesh to the shape of the object(s) in the union image, a task the EBILS has been demonstrated capable of. With this in mind, we initialize the offspring net using the bounding box of the synthetic object of the union image. Then, we run EBILS to fit the mesh to the synthetic object(s). The resulting net will have the same shape of the union of the two parent nets, including a new proper topology, calculated by EBILS. While this SCO has only been applied to subsets of size two, it can be extended easily to combinations of more than two solutions. Fig. 15 depicts the phenotypic SCO process while Fig. 14(d) shows the result of the combination of the nets in Figs. 14(a,b).

3) Application of the SCOs: The phenotypic and genotypic SCOs are fully complementary. The former is appropriate in combining high quality meshes segmenting *different* objects while the latter is useful to derive better nets while combining solutions segmenting the *same* object(s). Therefore, we use both of them in the solution combination method. We propose to alternate them on the basis of the RefSet1. In order to do so, at every iteration of the algorithm, we test if

$$f_{\text{margin}} = \frac{f_{\text{mean}} - f_{\text{best}}}{f_{\text{best}}} < \text{th}_{\text{margin}},$$

where f_{best} is the fitness value of the best mesh in the RefSet, f_{mean} is the mean fitness value of the meshes in the quality subset of the RefSet, and $\text{th}_{\text{margin}}$ is a proper threshold experimentally set to 0.1. If the inequality is true, we apply the genotypic SCO, otherwise we apply the phenotypic version. We also apply the phenotypic SCO after every population generation, to exploit its superior capabilities in merging different nets. Moreover, if the best solution improved in the last SS iteration, we apply again the same SCO to the whole population in the current iteration. Fig. 16 depicts an extract of a convergence graph that shows how both SCOs synergically contribute to the fitness improvement.

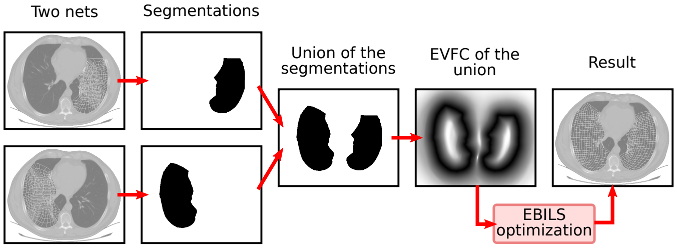


Figure 15: The phenotypic SCO process.

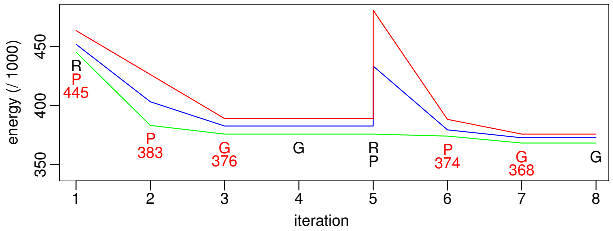


Figure 16: A convergence graph of RefSet1. The green, blue and red lines show the fitness of the best, mean and worst solutions, respectively. “P” stands for phenotypic SCO, “G” for genotypic SCO, and “R” for restart.

IV. Results and analysis

A. Experimental design

We tested four different segmentation algorithms: the TAN-DE [8], the ETAN-EBILS [5], initialized by the bounding box of the K-means pre-segmentation, an ETAN-MSLS considering EBILS as local search method, and our ETAN-SS proposal. In this section we will refer to the four algorithms as DE, LS, MS and SS, respectively. The three ETAN-based algorithms employ the same fitness function and set of parameters. DE, MS and SS are run ten times each while LS, being deterministic, is executed just one time.

MS considers multiple alternative solutions applying the EBILS starting from different positions. The meshes are initialized with the same frequency-memory procedure described in Section III-F. Since the time needed to calculate the value of the objective function is much lower than the duration of the EBILS itself, the stopping condition of the MS is the mean run time of the ten executions of the SS.

The considered image dataset is made up of a mix of 20 synthetic and real-world medical images. The images show various difficulties and have a *ground truth* allowing us to properly evaluate the segmentation performance.

In order to quantitatively assess our results on the dataset, we compute the spatial accuracy index *S*, which is a similarity index based on the overlapping rate between the segmentation result and the ground truth [22]:

$$S = 2 \cdot \frac{\text{Card}(R \cap T)}{\text{Card}(R) + \text{Card}(T)}, \tag{6}$$

where R is the segmentation result, T is the ground truth, and $\text{Card}(X)$ is the cardinality of the X set, that is the number of pixels it contains. Therefore, this index is dimensionless and varies in the range $[0, 1]$. The higher its value, the better.

We also compute the mean distance $M_{d_{\text{RT}}}$ between the contours of the segmentation results and the ground truth as well as the mean distance $M_{d_{\text{TR}}}$ between the ground truth and the segmentation results:

$$M_{d_{\text{RT}}} = \frac{\sum\limits_{r \in R} \min\limits_{t \in T} d(r,t)}{\text{Card}(R)}, \qquad M_{d_{\text{TR}}} = \frac{\sum\limits_{t \in T} \min\limits_{r \in R} d(r,t)}{\text{Card}(T)},$$

where $d(x, y)$ is the Euclidean distance.

Finally, we compute the Hausdorff distance [23], that is the maximum distance between the two contours:

$$d_H(R,T) = max\left\{ \sup_{r \in R} \inf_{t \in T} d(r,t), \;\; \sup_{t \in T} \inf_{r \in R} d(r,t) \right\}. \tag{7}$$

The latter three distances are measured in pixels. While zero is the lower bound, the upper bound depends on the size of the image. In any case, the lower the value, the better the obtained segmentation. Among the three distance metrics, $d_H(R,T)$ is the most relevant one since it measures the worst case of the segmentation. A low value in this metric implies *per se* an effective segmentation.

The parameters used to run the algorithms over the 20 images in the dataset are shown in Table I. Some, related to ETANs, are covered in [5]. As shown, the ETAN-based algorithms have all the parameters of the TAN and some new ones. Although this could suggest that tuning an ETAN is harder than a TAN, we experimentally arrived at an opposite conclusion. Indeed, in order to achieve an appropriate performance for the TAN, there is a need to develop a specific experimental design for each individual image, testing different values for each parameter in a wide range. For instance, for the TAN case, using a net size of 30×30 nodes provided poor results and it was necessary to manually set this parameter to lower vales, different for each image. Moreover, often small adjustments of the parameter values lead to significant changes in the behavior of the nets. Conversely, ETANs are far less sensitive to changes in those parameters [5].

The three ETAN-based algorithms have been implemented in C++, while the DE in C. The tests were run on an Intel® Core$^{\text{TM}}$2 Quad CPU Q8400 at 2.66 GHz with 4 GB RAM.

B. Image dataset

The 20 images selected try to cover the most typical difficulties in images segmentation: concavities, complex shapes, fuzzy borders, holes, noise, and multiple objects. They are shown in Fig. 17 with the SS results superposed to save space. We divided them into two categories:

1) Real-world medical images: Two groups of images belong to this category. Images from *k1* to *k4* (Fig. 17(a-d)) are real-world CT images of a human knee. The gray value of all pixels have been inverted so the bone becomes the darker object in the image. Images from *l1* to *l6* (Fig. 17(e-j)) are CT images of human lungs. The target objects are, respectively, the bones and the lungs. The ground truth has been derived manually. The images have sizes up to 432 × 470 pixels.

2) Synthetic images: We drew these images trying to cover the mentioned segmentation difficulties. In addition, these images have been artificially perturbed with three different kinds of noise: Gaussian (with $\sigma = 20$), Lorentz (salt-and-pepper, with $\gamma = 7$), and *tiny-objects*, that is, small dots or lines which are not part of the target objects. The only exception is image s1, which is only affected by the latter kind on noise. Since they are synthetic images, they have been generated starting from the ground truth. All the images have size 375 × 375 pixels.

Parameter	**DE**	**ETAN-based**
α	$\in [0.01, 1.0]$	$\in [5.0, 15.0]$
β	$\in [0.1, 0.5]$	3.0
ω	$\in [1.0, 20.0]$	3.0
ρ	1.0	1.0
ξ	$\in [1.0, 2.0]$	3.0
δ	$\in [2.0, 9.0]$	15.0
γ	1.0	$\in [1.0, 7.0]$
Net size	$\in [12 \times 12, 20 \times 20]$	$\in [20 \times 20, 30 \times 30]$
th_{cut}	-	3.4
th_{holes}	-	$\in [0.6, 1.0]$
S_{Canny}	-	$\in [15, 150]$
P_{size}	1000	20
$b_1,\ b_2$	-	3, 3

Table I: The parameters used in the experimentation.

C. Analysis of the results

Table II shows the numeric results obtained by the four methods in the four considered metrics for every image. Since DE, MS and SS are run ten times for each image, the table shows the corresponding *mean* (μ) and *standard deviation* (σ). It also shows their overall values on the whole dataset. Since it is not possible to normalize the values provided by the three distance metrics, the corresponding overall figures are only an indication. For every image, and for the global μ and σ, the best result for every metric is highlighted in boldface.

It can be clearly seen that our proposal achieves the best mean results in the four quality metrics in almost every case. It also showed the best behavior regarding robustness. Notice that our SS method achieved the lowest standard deviation values in the four segmentation metrics.

With regards to the execution time, while LS is quite fast, with a mean time of 1.18 s, SS and MS are almost 300 times slower, on average. DE is slightly faster, at about 150 times slower. This is an expected result, given their global search nature and the use of a simpler LS with a very limited number of applications (see Sec. II-B) in the case of DE. It is worth noting that any other MA employing the introduced operators and the EBILS would be slower than our SS proposal which uses a very reduced population. In addition, our approach is a fully automatic segmentation method while LS needs a good initialization, with the consequent time consumption.

In all the images, the target objects are structures, generally darker than the background, surrounded by a wide range of other structures. Segmentations including both the target objects and part of the background correspond to local optima of the energy function. They often have high values of the d_{RT} metric. Incomplete segmentations, lacking part of the target objects, are another kind of local optima. Typically, these segmentations showed high values of the d_{TR} metric. Conversely, segmentations including all and only the target objects are close to the global optimum, with low values of the d_{RT}, d_{TR}, and, consequently, d_H metrics.

For LS, it got trapped in local optima on the knee images, triggered by the tissue part around the bone. On the lung CTs, the results were similar. In these cases, the local optima are caused by the presence of the ribs, the vertebral column, and the interface between the external air and the tissue. Figs. 18(a)-(c) show some examples of the problems described. On the synthetic images, LS obtained slightly better results. It was able to segment the holes in the images and to successfully filter the two kinds of punctual noise. However, the algorithm tends to segment the small dots and other structural noise, even getting stuck into them, like in the case of image s8.

The segmentations obtained by MS are expressions of different local minima (see Figs. 18(d)-(f) for illustrative examples). In these cases, the algorithm found meshes with a lower energy than LS. However, in doing so, it lost some

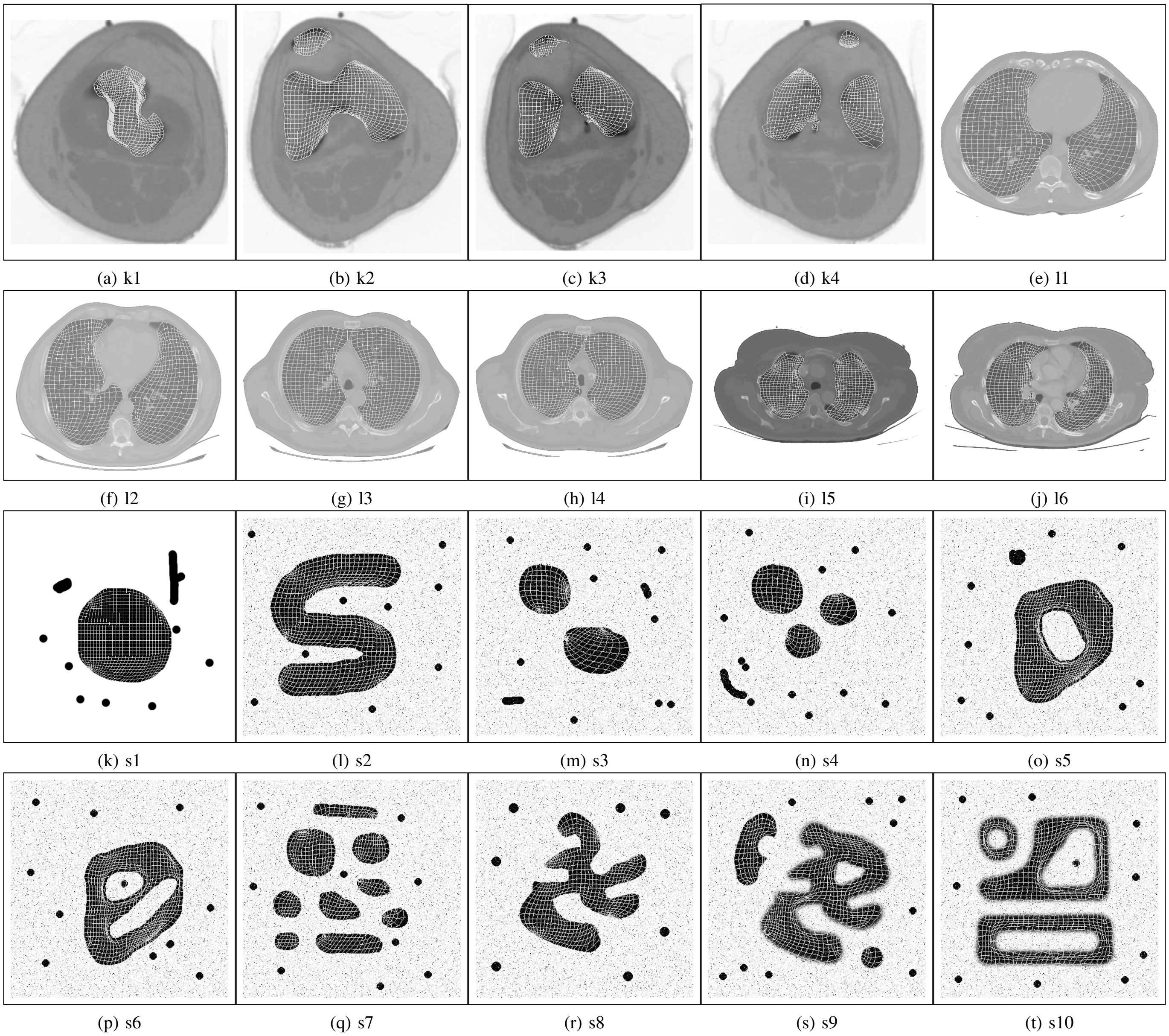


Figure 17: SS results on: (a)-(j) real-world CT images; (k)-(t) synthetic images. For every image, the mesh shown is the most similar to the mean one, resulting from the ten runs, according to the S index and Hausdorff distance statistics.

high energy objects, not being able to cut the connections between these objects and the background. Indeed, it often was unable to segment the smallest object in images *k2*, *k3* and *k4*. Moreover, the segmentation of the lungs are often incomplete, lacking some important parts. Connections with external borders are also present, as in the case of LS, but to a lesser extent. The values of the d_{RT} and the d_{TR} distances for the LS and MS algorithms confirm the analysis of the segmentation defects. Unsurprisingly, on the synthetic images, MS was able to filter the structural noise better than LS, but it showed the same tendency to undersegment the objects. It successfully filtered the two kinds of punctual noise.

The results obtained by DE are poor. The meshes got stuck in both kinds of local optima exposed so far. In addition, they failed in locating the objects, eventually taking degenerated shapes. The only successful result was s1, which is only

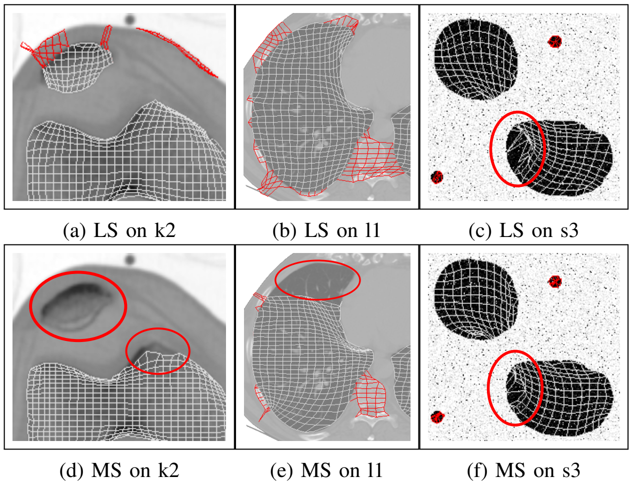


Figure 18: Examples of LS and MS inaccuracies (in red).

			Real-world										Synthetic										Overall	
Metric			k1	k2	k3	k4	l1	l2	l3	l4	l5	l6	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	μ	σ
S		LS	0.92	0.96	0.95	0.92	0.89	0.97	0.96	0.98	0.92	0.87	0.97	0.97	0.94	0.93	0.97	0.97	0.96	0.47	0.94	0.93	0.92	0.10
	μ	MS	0.90	0.92	0.90	0.93	0.86	0.97	0.95	0.97	0.94	0.83	0.98	0.98	0.97	0.96	0.98	0.96	0.95	0.95	0.89	0.90	0.93	0.04
		DE	0.44	0.40	0.57	0.28	0.63	0.56	0.80	0.82	0.54	0.64	0.98	0.65	0.64	0.57	0.77	0.72	0.47	0.61	0.56	0.50	0.61	0.16
		SS	**0.93**	**0.97**	**0.96**	**0.94**	**0.98**	**0.99**	**0.99**	**0.99**	**0.97**	**0.94**	**1.00**	**0.99**	**0.99**	**0.98**	**0.99**	**0.98**	**0.97**	**0.98**	**0.96**	**0.93**	**0.97**	**0.02**
	σ	MS	0.05	0.03	0.01	0.01	0.10	0.01	0.03	0.02	0.02	0.06	0.01	0.00	0.02	0.01	0.00	0.02	0.02	0.02	0.05	0.04		
		DE	0.20	0.18	0.07	0.07	0.03	0.11	0.04	0.01	0.10	0.05	0.01	0.01	0.04	0.12	0.02	0.02	0.03	0.02	0.03	0.04		
		SS	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.02		
$M_{d_{\mathrm{RT}}}$		LS	**2.84**	10.17	10.17	17.14	10.86	4.13	5.39	3.80	6.01	7.41	18.68	2.76	20.64	21.52	7.33	2.52	2.19	51.54	5.91	2.59	10.68	10.27
	μ	MS	12.93	5.14	4.39	7.84	6.62	4.89	5.14	3.22	5.05	5.64	11.17	2.37	10.15	4.91	5.80	3.04	2.17	5.29	5.47	3.06	5.71	2.79
		DE	14.10	8.83	21.86	23.80	17.90	25.03	17.73	18.27	39.70	21.80	1.89	12.75	3.77	3.90	14.24	12.13	9.55	11.27	7.28	11.38	14.86	8.65
		SS	3.03	**1.45**	**2.13**	**2.90**	**1.57**	**0.94**	**0.89**	**0.64**	**1.34**	**2.17**	**0.57**	**0.59**	**0.84**	**0.74**	**0.99**	**1.61**	**0.72**	**0.83**	**1.62**	**2.14**	**1.39**	**0.73**
	σ	MS	8.12	5.69	1.05	4.05	1.58	1.18	1.39	0.83	0.81	0.82	7.44	0.61	3.37	1.33	3.97	0.65	0.84	3.54	1.09	0.67		
		DE	10.68	2.64	0.98	3.71	4.23	4.83	1.60	1.41	12.39	4.77	1.10	1.23	1.61	1.31	2.16	2.98	1.86	1.44	0.64	3.15		
		SS	0.43	0.27	0.26	0.35	0.38	0.11	0.09	0.08	0.27	0.89	0.03	0.02	0.49	0.23	0.38	0.42	0.10	0.10	0.14	0.25		
$M_{d_{\mathrm{TR}}}$		LS	4.46	1.63	**1.77**	**2.00**	6.27	1.49	2.25	**1.29**	2.81	7.96	0.91	1.00	1.09	1.34	0.87	0.86	0.84	59.33	**1.84**	1.82	5.09	11.54
	μ	MS	3.39	11.30	13.73	9.30	31.01	2.51	3.92	5.37	**2.09**	14.64	0.94	0.77	1.08	1.03	0.84	1.12	1.06	2.48	6.25	3.01	5.79	7.19
		DE	28.99	53.59	24.97	32.84	80.76	66.43	34.86	31.31	34.23	35.22	1.56	36.76	41.61	37.58	22.56	24.68	35.19	29.53	42.42	40.03	36.76	15.91
		SS	**2.73**	**1.51**	2.04	3.50	**1.60**	**1.42**	**1.84**	1.30	2.72	**3.87**	**0.56**	**0.60**	**0.67**	**0.63**	**0.80**	**0.83**	**0.66**	**0.88**	1.89	**1.71**	**1.59**	**0.96**
	σ	MS	1.25	4.90	2.45	5.28	31.44	0.68	2.68	4.92	0.69	16.25	0.41	0.10	0.78	0.22	0.05	0.51	0.55	2.15	5.23	2.59		
		DE	11.29	13.31	1.75	3.25	4.37	14.29	2.45	2.50	7.70	9.50	0.83	2.08	3.10	4.46	2.12	2.50	9.49	1.95	3.22	4.26		
		SS	0.40	0.31	0.35	3.46	0.24	0.16	0.20	0.07	0.19	0.74	0.02	0.02	0.24	0.15	0.05	0.04	0.10	0.15	0.79	0.32		
d_{H}		LS	28.60	62.10	72.44	83.01	54.45	44.94	37.44	45.28	**31.83**	48.80	98.99	34.89	135.91	126.57	76.69	30.41	35.90	121.81	71.06	36.00	63.86	30.35
	μ	MS	74.43	86.58	92.29	91.58	121.94	49.79	42.05	53.10	38.40	59.88	70.03	32.73	96.88	72.92	78.33	34.43	38.25	74.86	82.13	38.10	66.43	24.42
		DE	74.60	127.19	88.09	86.67	234.70	183.04	118.06	104.81	94.33	104.61	17.23	98.95	126.73	100.05	70.44	81.21	95.99	83.25	123.24	100.63	105.69	42.50
		SS	**17.17**	**8.48**	**17.49**	**30.12**	**24.19**	**22.64**	**25.72**	**14.81**	38.15	**27.08**	**2.38**	**4.08**	**8.92**	**7.19**	**9.52**	**24.01**	**10.79**	**8.65**	**21.23**	**31.41**	**17.70**	**9.82**
	σ	MS	16.98	10.66	16.28	12.15	77.92	9.45	11.43	15.02	14.90	43.55	28.34	4.60	24.63	13.40	21.73	14.01	19.41	12.19	18.12	8.87		
		DE	18.58	24.48	6.15	4.55	7.76	24.90	6.25	5.96	10.01	24.57	12.34	4.48	4.96	6.99	2.72	7.47	24.87	11.97	8.35	13.67		
		SS	1.35	2.28	10.24	23.81	2.82	2.75	3.09	0.40	15.34	3.86	1.88	1.28	12.52	3.67	9.12	2.07	3.72	3.40	13.01	7.76		

Table II: Numeric results of the four segmentation metrics for all the algorithms on all the images.

affected by *tiny-objects* noise, showing how the global search is able to filter these kinds of structures. Conversely, the algorithm proved to be heavily affected by the punctual noise on the synthetic images. Although DE achieved good performance in [8], the images considered there were significantly simpler than the ones in our dataset. The lack of an energy term rewarding the segmentation of multiple objects, the loss of the topology information of every net but the best individual at every new generation, the inability of the BILS to adjust the mesh to objects with complex shapes, and the absence of crossover operators considering the characteristics of the problem strongly limit the performance of that proposal.

Finally, the results obtained by SS are clearly the best ones. For all the images, it performed better than the other three methods in almost every statistic. Indeed, it respectively ranked first 20 and 19 out of 20 times, according to the mean of the S and d_H index values, as shown in Table II. The low values of the d_{RT} and the d_{TR} distances, smaller than the three competitors, are in line with the quality of the segmentations. SS gets the best result in 19 cases for d_{RT} and in 15 cases for d_{TR}. Focusing on the medical images, the resulting nets on images *k2*, *k3* and *k4* properly segment the small objects and there are few connections to the background on all the images of this category. The segmentations are almost complete but, in some cases, SS was not able to segment some small structures, as in Figs. 17(g, h, i). As for the synthetic images, the segmentations are complete while there is almost no presence of structural noise.

D. Statistical analysis of the results

In the previous section, we provided a detailed analysis of the numerical results obtained, giving an insight of the performance of the four compared methods. To prove the significant superiority of the segmentation capabilities of our proposal, in this section we provide a statistical analysis of the obtained results. With this aim, we performed a two-tailed Wilcoxon signed-rank test [24] for each of the four segmentation metrics as follows.

Let $N = 20$ be the sample size, that is, the number of images in the dataset and hence the number of pairs in the test. We compare the mean performance over the ten runs for each image. Each pair is composed of the result of our SS proposal and the best result among those achieved by the other three algorithms. This is the hardest case since, for every image, we always compare SS against an aggregate algorithm whose performance is the best one obtained by the set $\mathrm{C} = \{\mathrm{LS}, \mathrm{MS}, \mathrm{DE}\}$. Since we are now comparing only two algorithms (C and SS), we are allowed to use the Wilcoxon test, as outlined in [25]. The null and alternative hypothesis are defined, respectively, as:

- H_0: median difference between the pairs is zero,
- H_1: median difference is not zero.

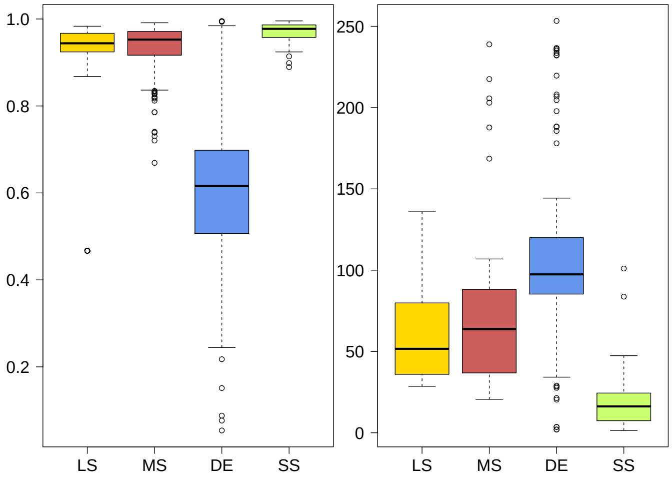
The p-values obtained for the four metrics are shown in Table III. For SS, the values of the medians are higher for the S index and lower for the three distances. Given the obtained p-values, we found enough evidence to reject the null hypothesis with a confidence level of 0.05 for every metric. It is worth noting that the confidence level could have been much lower for the S, $M_{d_{RT}}$ and d_H metrics, for example 0.01.

Although the Wilcoxon test is significant enough to prove the superiority of the performance of our proposal with respect to the other three methods, for the sake of clarity we also show a boxplot of the two most relevant metrics, S and d_H, in Fig. 19. These boxplots are a quick way to graphically examine graphically the distributions of the 200 results obtained by the

Metric	p-value	median(C)	median(SS)
S	$1.907 \cdot 10^{-6}$	0.958	0.977
$M_{d_{RT}}$	$3.815 \cdot 10^{-6}$	4.266	1.164
$M_{d_{TR}}$	$2.958 \cdot 10^{-2}$	1.700	1.465
d_H	$5.722 \cdot 10^{-6}$	47.036	17.328

Table III: The results obtained by the Wilcoxon test.

three stochastic algorithms over the 20 images, considering the 10 runs (being deterministic, LS has been run just once per image).



(a) Index S (b) $d_{\text{Hausdorff}}$

Figure 19: The distributions of the results (1/10 runs for 20 images), for S and d_H metrics obtained by the four methods.

V. Conclusions and future work

In this work we have proposed an accurate, robust and automatic segmentation method that is able to perform in a reasonable time. It embeds the ETAN model into a customized SS global search framework. We designed several new specific components for the method. They become a crucial outcome allowing us to really take advantage of the population-based optimization framework as none of the previous approaches were able to do. The obtained results were encouraging. Our SS proposal significantly improved the accuracy of the segmentation on real-world medical images, as well as on synthetic ones, in comparison with the original ETAN-EBILS, an ETAN-MSLS and the state-of-the-art DE-based MA for TANs. Moreover, the robustness achieved was significantly better than the previous methods.

Nevertheless, regardless of its good performance, our method still has room for improvement. In a few cases the obtained segmentations were not fully correct. To better take advantage of the region-based segmentation capabilities, we could endow the ETAN with a texture analysis system. Indeed, texture is one of the important characteristics used in identifying objects or regions of interest in an image.

Acknowledgment

This work is supported by the European Commission with the contract No. 238819 (MIBISOC Marie Curie ITN) and by the Spanish Ministerio de Educación y Ciencia (ref. TIN2009-07727), both including EDRF funds.

Part of the code related to DE was provided by J. Novo and the VARPA group, University of A Coruña, Spain.

We would like to thank A. Valsecchi for his support in performing the statistical analysis.

References

[1] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: a survey." *Medical image analysis*, vol. 1, no. 2, pp. 91–108, 1996.

[2] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.

[3] M. Bro-Nielsen, "Active nets and cubes," *IMM Technical Report*, 1994.

[4] F. M. Ansia, M. Penedo, C. Mariño, and A. Mosquera, "A new approach to active nets," *Pattern Recognition Image Analysis 2*, pp. 76–77, 1999.

[5] N. Bova, O. Cordón, and O. Ibáñez, "Extended topological active nets," European Centre for Soft Computing, Mieres, Spain, Tech. Rep. AFE 2012-01, 2012, submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, available at http://docs.softcomputing.es/public/afe/Tech_rep_2012-06.pdf.

[6] Y.-S. Ong, M. H. Lim, and X. Chen, "Research frontier: memetic computation-past, present & future," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–31, May 2010.

[7] O. Ibáñez, N. Barreira, J. Santos, and M. G. Penedo, "Genetic approaches for topological active nets optimization," *Pattern Recognition*, vol. 42, no. 5, pp. 907–917, 2009.

[8] J. Novo, J. Santos, and M. Penedo, "Topological active models optimization with differential evolution," *Expert Systems with Applications*, vol. 39, no. 15, pp. 12 165 – 12 176, 2012.

[9] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag, 2003.

[10] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution A Practical Approach to Global Optimization*, ser. Natural Computing Series, G. Rozenberg, T. Bäck, A. E. Eiben, J. N. Kok, and H. P. Spaink, Eds. Berlin, Germany: Springer-Verlag, 2005.

[11] M. Laguna and R. Martí, *Scatter Search: Methodology and Implementations in C*. Norwell, MA, USA: Kluwer Academic Publishers, 2003.

[12] U. Maulik, "Medical image segmentation using genetic algorithms," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 2, pp. 166–173, 2009.

[13] C. McIntosh and G. Hamarneh, "Medial-based deformable models in nonconvex shape-spaces for medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 31, no. 1, pp. 33–50, 2012.

[14] Y. Fan, T. Jiang, and D. J. Evans, "Volumetric segmentation of brain images using parallel genetic algorithms," *IEEE Transactions on Medical Imaging*, vol. 21, no. 8, pp. 904–909, 2002.

[15] J. Novo, M. Penedo, and J. Santos, "Evolutionary multiobjective optimization of topological active nets," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1781–1794, 2010.

[16] T. Radulescu and V. Buzuloiu, "Extended vector field convolution snake for highly non-convex shapes segmentation," *2009 9th International Conference on Information Technology and Applications in Biomedicine*, pp. 1–4, 2009.

[17] F. Glover, "Heuristics for integer programming using surrogate constraints," *Decision Sciences*, vol. 8, no. 1, pp. 156–166, 1977.

[18] R. Martí, A. Duarte, and M. Laguna, "Advanced scatter search for the max-cut problem," *INFORMS Journal on Computing*, vol. 21, no. 1, pp. 26–38, 2009.

[19] F. Glover and G. A. Kochenberger, Eds., *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.

[20] S. Damas, O. Cordón, and J. Santamaría, "Medical image registration using evolutionary computation: An experimental survey," *IEEE Computational Intelligence Magazine*, vol. 6, no. 4, pp. 26–42, 2011.

[21] F. Herrera, M. Lozano, and A. M. Sánchez, "A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study," *International Journal of Intelligent Systems*, vol. 18, pp. 309–338, 2003.

[22] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric analysis of white matter lesions in MR images: method and validation," *IEEE Transactions on Medical Imaging*, vol. 13, no. 4, pp. 716–724, 1994.

[23] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[24] D. F. Bauer, "Constructing confidence sets using rank statistics," *Journal of the American Statistical Association*, vol. 67, no. 339, pp. 687–690, 1972.

[25] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.