2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

# Evolutionary Medical Image Registration using Automatic Parameter Tuning

Andrea Valsecchi*, Jérémie Dubois-Lacoste†, Thomas Stützle†, Sergio Damas*, José Santamaría ‡ and Linda Marrakchi-Kacem§
*European Centre for Soft Computing, Mieres, Spain {andrea.valsecchi, sergio.damas}@softcomputing.es
†IRIDIA, Université Libre de Bruxelles, Brussels, Belgium {jeremie.dubois-lacoste, stuetzle}@ulb.ac.be
‡University of Jaén, Jaén, Spain jslopez@ujaen.es
§Neurospin, CEA, Gif-Sur-Yvette. CRICM, UPMC Université Paris 6, France linda.marrakchi@gmail.com

***Abstract*—Image registration is a fundamental step in combining information from multiple images in medical imaging, computer vision and image processing. In this paper, we configure a recent evolutionary algorithm for medical image registration, r-GA, with an offline automatic parameter tuning technique. In addition, we demonstrate the use of automatic tuning to compare different registration algorithms, since it allows to consider results that are not affected by the ability and efforts invested by the designers in configuring the different algorithms, a crucial task that strongly impacts their performance. Our experimental study is carried out on a large dataset of brain MRI, on which we compare the performance of r-GA with four classic IR techniques. Our results show that all algorithms benefit from the automatic tuning process and indicate that r-GA performs significantly better than the competitors.**

## I. Introduction

Image registration (IR) refers to the process of geometrically aligning multiple images having a shared content [1]. The alignment is represented by a spatial transformation that overlaps the common part of the images. Most IR methods are based on an iterative optimization procedure, in which the quality of a solution is the degree of resemblance between the images after the transformation. Along with classic, gradient-based numerical optimization techniques, methods based on evolutionary computation and other metaheuristics have been successfully used to tackle image registration in different contexts [2]–[6], notably 3D modeling [7] and medical imaging [8].

In [9] an evolutionary IR method based on a genetic algorithm, called r-GA, has been introduced. The novelty of r-GA lies in its design, which combines a multi-resolution strategy with a restart and search space adaptation mechanism. In the original study, r-GA obtained the best performance in a comparison of different IR techniques, demonstrating its applicability in two medical studies. Often, however, the effectiveness of optimization algorithms is heavily dependent on the setting of their parameter values. This poses a series of challenges for both the design and the comparison of algorithms. Finding appropriate settings is a complex task. For many years, this was done manually, using a trial-and-error approach with preliminary experiments. As a consequence, algorithm development requires a lot of expertise and it is very time consuming for algorithm designers. Additionally, when comparing algorithms, the effort invested in finding appropriate parameter settings for each algorithm may be very different, and the potential uneven tuning may compromise the ability of experimental studies to assess the intrinsic quality of an algorithm.

In recent years, several methods have been proposed for automatically configuring algorithms parameters, a process also called parameter *tuning* [10]–[13]. These methods are increasingly acknowledged in the research community for being important to find effective parameters. However, these automatic algorithm configuration methods, or tuners, are still very rarely used for the comparison of algorithms. In this paper, we will use such methods to assess the performance of various algorithms for IR. The procedure we use for this task is the following. First, we automatically configure the different algorithms we compare, and from this process we obtain parameter configurations that have been found to be the best by the tuner. Then, in the actual comparison, we use these parameter settings instead of the default parameter settings that have been proposed in the literature. In this way, we expect to compare the different algorithms at stake without introduction of human bias.

This paper is structured as follows. Section II introduces image registration, the registration study that will be performed, r-GA and the other algorithms involved in the comparison. In Section III, we first review the automatic configuration of algorithms and the tool we use, and then we explain the experimental setup. In Section IV we report the improvement obtained from the tuning, and the experimental comparison of the tuned configurations along with the analysis of the results. Finally, we conclude and highlight some promising directions for future research in Section V.

## II. Image Registration

### A. Preliminaries

In a typical problem instance, we are given two images: a reference image, the *model*, and the image that will be transformed to reach the model geometry, called *scene* [1]. We will denote these two images by $I_M$ and $I_S$ respectively. The result of the registration process is a transformation $f$ such that the model $I_M$ and the transformed scene $f(I_S)$ are as similar as possible.

Three main components characterize an IR method: the *transformation model*, the *similarity metric* and the *optimization process*. The transformation model determines what

978-1-4799-0454-9/13/$31.00 ©2013 IEEE

kind of transformations can be used to align the images. Transformation models vary greatly in complexity, ranging from simple combinations of translation and rotation up to elastic transformations that can represent local deformation and warpings. The choice of the appropriate transformation model for a given application is often crucial.

The similarity metric is the component that measures the quality of an alignment. In medical applications, the most common approach, called *intensity-based*, is to compare the distribution of intensity values (e.g. the gray levels) between the scene and the model once a transformation has been applied. The degree of matching can be computed from the intensity distributions using measures such as the mean square error, the correlation coefficient or the mutual information [14]. In an alternative approach, called *feature-based*, the alignment is measured only on salient and distinctive features of the image, such as lines, corners and edges, ignoring the rest of the image contents. This can make the problem easier and speedup the registration provided these features can be reliably detected automatically. This is rarely the case in medical imaging because great precision and consistency is required; in the remainder of this article we focus on intensity-based methods.

The optimization procedure is the component responsible for finding an appropriate transformation to carry out the registration. Figure 1 shows the flow chart of the whole registration process.

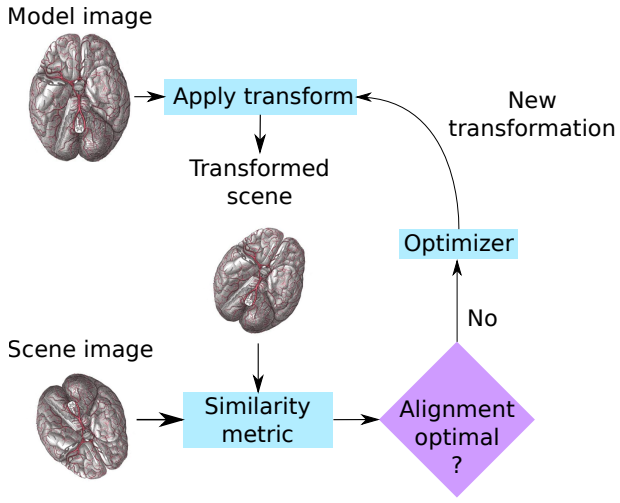


Figure 1. The interactions among the components of an image registration technique.

A transformation is specified by a series of parameters (e.g. as a translation vector and a rotation angle), which turns the registration into a continuous optimization problem. Classic numerical optimization algorithm such as gradient descent, Newton's method, Powell's method and discrete optimization [15], [16] are among the most common choices for the optimization component, together with approaches based on EC and other meta-heuristics [2]–[6], [17]–[19].

Another feature of IR methods is the use of multiple resolutions. Typically, a registration is initially performed using a simpler version of the input images, obtained through smoothing and downsampling. Once an initial solution has been found, the algorithm moves to a more detailed version of the images and continues the search for a suitable transformation. Each stage of this process is called *resolution*. The use of multiple resolutions aims both to reduce the computational cost of the registration and to facilitate the optimization by increasing the complexity of the problem gradually.

### *B. Registration study: atlas-based segmentation of deep brain structures*

In this study, image registration is part of a larger medical application called atlas-based segmentation. The aim is to segment a region of the human brain called the deep nuclei, which consists of caudate, putamen, globus pallidus and thalamus. We are provided an atlas (i.e. a typical or average image of the brain), in which the desired region has been already segmented. First, we register the atlas to the input image. The result of the segmentation process is indeed the region of the image that overlaps with the segmented region of the atlas after the registration. See Figure 2 for an example of this procedure.


Figure 2. An example of atlas-based segmentation. The figure also shows a slice of a 3D MRI brain image used in the study and the corresponding deep brain structure segmentation.

The quality of atlas-based segmentation depends closely on the accuracy of the registration step. By measuring the quality of the segmentation, therefore, we can evaluate the registration quantitatively, which otherwise can be a challenging and heavily application-dependent task. In this registration study, we perform atlas-based segmentation of deep brain structures [20]. Thirteen 3D $T_1$-weighted brain MRI were retrieved from the NMR database [21]. The deep nuclei structures in each image have been manually delineated by an expert in order to create the ground-truth data used to evaluate the registration.

Registration instances were created by selecting a pair of different images. No transformation was applied on the images; however, the location of the brain in each image is different due to the variability in the pose of the patient during the acquisition of the images. One image is used as an atlas, while the other is the image to be segmented. To evaluate the process, the segmented region obtained from the registration $V_R$ is compared with its ground-truth $V_{\text{GT}}$. The overlap of the two regions is measured using the Dice's coefficient [22], given by $\text{Dice}(V_R, V_{\text{GT}}) = 2|V_R \cap V_{\text{GT}}|/(|V_R| + |V_{\text{GT}}|)$ where $|\cdot|$ is the number of voxels. A value of 1 means perfect overlapping, while 0 means the two regions do not overlap at all.

For all algorithms, the transformation model is an affine transform, which involves rotation, translation, scaling and shearing. For 3D images, it can be represented using 12 real parameters. Affine transform is a popular choice in registration of medical images [23]. It is flexible enough to present a wide range of transformations and it does not produce anatomically unrealistic results, as it could happen with deformable models. As the images have the same modality, the similarity metric used is the normalized correlation coefficient.

### C. Image Registration Algorithms

This section describes the methodology that we aim to validate in this study, r-GA, and four gradient-based medical IR methods included in the comparison.

*1) r-GA:* r-GA is an evolutionary IR method for medical imaging [9]. The optimization component of r-GA is based on a Genetic Algorithm with a real-coded design. A solution is encoded as a real vector, storing the transformation parameters, and the variation operators are common choices for real-coded genetic algorithms: blend crossover (BLX-$\alpha$) [24] and random mutation [25]. The fitness value of a solution $t$ is simply the similarity metric between the two images when aligned according to $t$. No changes are required to handle different transformation models or similarity metrics.

A distinctive feature of r-GA is the use of multiple resolutions combined with a *restart* and a search space adaptation mechanism. The key idea is that if a low quality solution is carried over to the second resolution, the process is unlikely to recover and produce a good final solution. Therefore, restart is used at the end of the first resolution until a suitable solution if found. This process is computationally cheap, because in the first resolution the algorithm is using a small version of the imaging data, and most of the total effort is spent on the further resolutions. In addition, as the second resolution is meant to be a refinement phase, the search is focused around the best solution by restricting the range of the transformation parameters.

*2) Comparison methods:* To validate the results of r-GA, we considered four well-established medical IR algorithms: gradient descent (GD), quasi-Newton (QN), nonlinear conjugate gradient (NCG) and adaptive stochastic gradient descent (ASGD). In [16], the authors review and compare IR methods for the registration of follow-up chest CT scans and found these methods to yield the best results. All four algorithms are gradient-based; they consist of an iterative optimization process

$$\mu_{k+1} = \mu_k + a_k d_k$$

where $d_k$ is the search direction at iteration $k$, and $a_k$ is a gain factor that controls the step size along the search direction. The search directions and gain factors are chosen such that the sequence $\mu_k$ converges to a local minimum of the similarity metric. The difference between the four optimizers lies in the way the search direction and the gain factor are computed.

The gradient descent method takes steps in the direction of the negative gradient of the cost function

$$\mu_{k+1} = \mu_k - a_k g(\mu_k)$$

where $g$ is the derivative of the cost function and the gain factor is the decaying function $a_k = a/(k+A)^\alpha$ with $a > 0$, $A \geq 1$ and $0 \leq \alpha \leq 1$.

The quasi-Newton method also moves along the negative gradient direction. The gain factor is an approximation of the inverse Hessian matrix $[H(\mu_k)]^{-1}$, computed using the Broyden–Fletcher–Goldfarb–Shanno method.

In the nonlinear conjugate gradient method, the search direction is a linear combination of the gradient and the previous search direction, i.e. $d_k = -g(\mu_k) + \beta_k d_{k-1}$. Several expressions for gradient type $\beta_k$ have been proposed in the literature. The gain factor is determined by an inexact line search routing, the Moré–Thuente algorithm.

Table I. DEFAULT PARAMETER SETTINGS OF THE ALGORITHMS.

| Parameter | Value |
|---|---|
| **Genetic Algorithm** | |
| PopulationSize | 50 |
| CrossoverProbability | 0.7 |
| MutationProbability | 0.1 |
| TournamentSize | 3 |
| MaximumNumberOfIterations | 50 |
| Restarts | 5 |
| NumberOfResolutions | 2 |
| **Gradient Descent** | |
| NumberOfResolutions | 3 |
| MaximumNumberOfIterations | 1000 |
| a | 400 |
| A | 50 |
| $\alpha$ | 0.602 |
| **Quasi-Newton** | |
| NumberOfResolutions | 2 |
| MaximumNumberOfIterations | 1000 |
| LBFGSUpdateAccuracy | 5 |
| **Nonlinear Conjugate Gradient** | |
| NumberOfResolutions | 3 |
| MaximumNumberOfIterations | 1000 |
| ConjugateGradientType | DaiYuanHestenesStiefel |
| **Adaptive Stochastic Gradient Descent** | |
| NumberOfResolutions | 3 |
| MaximumNumberOfIterations | 1000 |

The adaptive stochastic gradient descent [26] follows the same scheme as the regular gradient descent, but implements an adaptive step size mechanism and an automatic estimation procedure for the parameters $a$ and $A$. ASGD considers the solutions $\mu_{k+1} = \mu_k - \gamma_k(t_k) g_k$ where $t_{k+1} = \max(0,\ t_k + \text{sigm}(-g_k \cdot g_{k-1}))$ and $\gamma_k = \frac{a}{t_k + A}$. The "time" $t_k$ is adapted depending on the inner product between the current and the previous gradients. If the gradients have the same direction, the time is reduced, leading to a larger step size.

*3) Implementation details:* All comparison algorithms are written in C++ and integrated in Elastix [27], a toolbox for intensity-based medical image registration. Elastix is free, open-source and it has been used in over one hundred publications in medical imaging [28].

*4) Default configurations of the Algorithms:* The default settings that we also include in our final comparison are presented in Table I. These settings are those recommended from the literature.

## III. AUTOMATIC CONFIGURATION OF THE ALGORITHMS

The automatic configuration of algorithms, also called automatic *tuning*, has received a strong attention in recent research. In particular, in recent years a number of new algorithmic tools for the automatic configuration of algorithms have been developed. These include methods such as ParamILS [10], SMAC [11], SPO [12], SPO$^+$ [29], or iterated race [13], [30], which is available as an R [31] package and which we used in this paper. In this section, we explain the goal of the automatic configuration process, and we briefly overview the tool that we used to perform this process.

### A. Offline Automatic Configuration

Automatic parameter configuration can be done *online* to set the values of the parameters while the algorithm is running. This online adaptation of parameters is sometimes referred to as parameter adaptation. It is typically applied only to a small subset of key parameters, since it implies a significant overhead for the algorithm to "learn" good values for the parameters, in addition to the exploration of the search space of the instance to be tackled.

In this paper, we use *offline* automatic configuration. In this case, the purpose is to automatically configure optimization algorithms before they are deployed, that is, before they are applied to instances that are not yet known. Two clearly delimited phases are involved in this process. In a primary tuning phase, an algorithm configuration is selected, given a set of training instances. In a secondary production (or testing) phase, the selected algorithm configuration is used to solve unknown instances of the same problem. The goal is to find, during the tuning phase, an algorithm configuration that optimizes some cost measure over the set of instances that will be seen during the production phase. In other words, the ultimate purpose is that the high-quality configuration of the algorithm found during the tuning phase generalizes to similar but unknown instances.

### B. The IRACE software package

Birattari et al. [32]–[34] proposed an automatic configuration approach, F-Race, based on *racing* [35], with the use of Friedman's non-parametric two-way analysis of variance by ranks, to test for significantly inferior candidate configurations. This proposal was later improved by repeating the race process, refining iteratively the sampling distribution. The resulting automatic configuration approach was called Iterated F-race, and formally described in (I/F-Race) [30], [36]. However, no implementation of it has been made publicly available at that time. Later, the `irace` package has been proposed, that implements a general *iterated racing* procedure, which includes I/F-Race as a special case. It also implements several extensions, some described in [33], [34], such as the use of the paired $t$ test instead of Friedman's test. Several original contributions were also implemented to improve further the effectiveness of the tuning procedure. For more details, the reader can refer to [13].

The `irace` package has already been extensively tested in several successful research projects. For instance, Dubois-Lacoste et al. [37]–[39] used `irace` for tuning new state-of-the-art algorithms for the permutation flow-shop scheduling problem. However, to the best of our knowledge, this paper is the first to apply an automatic configuration tool to image registration, and to the field of medical images, in general.

The advantage of the `irace` tool is that it handles several parameter types: continuous, integer, categorical, and ordered. Continuous and integer parameters take values within a range specified by the user. Categorical parameters can take any value among a set of possible ones explicitly given by the user. An ordered parameter is a categorical parameter with a pre-defined strict order of its possible values. We also relied on `irace`'s capability to parallelize the configuration process in order to reduce considerably the amount of time required for it.

Table II. LIST OF THE PARAMETERS THAT ARE AUTOMATICALLY CONFIGURED. GIVEN IS THE NAME OF THE PARAMETER, THE TYPE AND THE DOMAIN OF THE PARAMETERS.

| Parameter | Type | Domain |
|---|---|---|
| **Genetic Algorithm** | | |
| `CrossoverProbability` | real | $[0.5, 0.9]$ |
| `MutationProbability` | real | $[0.05, 0.2]$ |
| `TournamentSize` | integer | $[2, 6]$ |
| `MaximumNumberOfIterations` | integer | $[25, 100]$ |
| `Restarts` | integer | $[2, 8]$ |
| `NumberOfResolutions` | integer | $[2, 4]$ |
| **Gradient Descent** | | |
| `NumberOfResolutions` | integer | $[2, 5]$ |
| `MaximumNumberOfIterations` | integer | $[500, 2000]$ |
| `a` | integer | $[400, 1600]$ |
| `A` | integer | $[50, 200]$ |
| $\alpha$ | real | $[0.5, 0.7]$ |
| **Quasi-Newton** | | |
| `NumberOfResolutions` | integer | $[2, 5]$ |
| `MaximumNumberOfIterations` | integer | $[500, 2000]$ |
| `LBFGSUpdateAccuracy` | integer | $[20, 50]$ |
| **Nonlinear Conjugate Gradient** | | |
| `NumberOfResolutions` | integer | $[2, 5]$ |
| `MaximumNumberOfIterations` | integer | $[500, 2000]$ |
| `ConjugateGradientType` | categorical | {SteepestDescent, FletcherReeves PolakRibiere, DaiYuan, HestenesStiefel, DaiYuanHestenesStiefel} |
| **Adaptive Stochastic Gradient Descent** | | |
| `NumberOfResolutions` | integer | $[2, 5]$ |
| `MaximumNumberOfIterations` | integer | $[500, 2000]$ |

### C. Tuning Setup

In the context of our study, the instances are defined by a pair of different images (see section II-B). The study is performed using 10-fold cross validation. We used 70 instances, that were randomly partitioned in 10 subsets. We performed 10 independent comparisons; for each comparison, all algorithms were tuned using 9 subsets of instances (that is, 63 instances) for the tuning phase, and the remaining subset (7 instances) was used for the comparison of the configurations obtained from the tuning. That is, each instance is used for testing in exactly one independent comparison, and used for training in all the others. In this way, we perform 10 repetitions of the whole tuning process for each algorithm, always keeping a clear separation between the training and the testing instances, so the algorithm automatically tuned are compared on instances that were never seen during the tuning.

We allowed a tuning budget of 1000 evaluations, i.e. `irace` can call the algorithm to be tuned a maximum of 1000 times to find the best possible configuration.

In our comparison, we compare both deterministic and stochastic algorithms. Unlike deterministic algorithms, which are evaluated during the tuning process through a single run, stochastic ones (r-GA and ASGD) are run 3 times independently and the average cost is returned as the result of the evaluation to the tuner. Table II presents the list of parameters that were automatically configured, with their types. In case of real or integer parameters, the table shows the ranges considered as candidate values, and for categorical variable all possible values are given explicitly.

## IV. Experimental Comparison

### A. Setup for the Comparison

Following the 10-fold cross validation setup, the algorithms are tested on the testing data using the configurations obtained from the tuning. Non-deterministic algorithms are run 30 times on each instance, and their average overlap value is used in the comparison with deterministic algorithms. Our analysis examines several aspects of the results. The per-instance performance is evaluated by ranking the algorithms according to their overlap value. The overall performance is assessed by computing the average ranking over all the instances. In addition, we count the number of instances in which one algorithm performs better than another, called *wins*, to allow a pairwise comparison.

In the final part of the analysis, statistical tests are used to determine which results are significantly different. We used the tests and the procedures recommended in [40] for comparing algorithms over multiple problems. Non-parametric tests are used to avoid relying on any assumption about the distribution of the results.

As we aim to validate the performance of r-GA, its results are compared with that of the remaining algorithms (that is, we perform a multiple comparison against a control method), a procedure that has more power than a pairwise comparison of all algorithms. The test we used is Nemenyi's test [41], and the sign-test. The first is a post hoc procedure of Friedman's rank sum test [42] and it is based on the ranks of the algorithms. The sign-test, instead, compares the algorithms using only the number of wins and losses. As multiple comparison are performed, the p-values of the tests have been adjusted using Holm's method [43] in order to control the family-wise error rate.

### B. Results

The detailed results of the experiments for 8 of the 70 instances are reported in Table III. The mean overlap and standard deviation values for each instance are reported. The average ranks (Table IV) and the count of wins (Table V) provides an overall view of the results of the comparison.

*1) Overall effect of the tuning:* First, we discuss the impact of the tuning on the performance of the algorithms, with respect to their default settings (see section II-C4). As shown in Table IV, the tuning leads to a clear improvement in terms of average ranking for all algorithms but NCG. In fact, when each tuned configuration is compared to the default one in terms of number of instances (see Table V) it is also true for NCG. Thus, all tuned algorithms performed better than their default version in more than half of the cases. r-GA benefited the most of the tuning, improving its performance in 57 out of 70 cases. ASGD improved in 52 cases, which is remarkable given that the algorithm uses an online adaptation mechanism for its parameters and, thus, was not a priori supposed to be subject to improvements from the tuning. GD, QN and NCG improved in 49, 38 and 37 cases, respectively. Overall, the use of automatic tuning lead to improvement in $57 + 52 + 49 + 38 + 37 = 233$ cases over 350. We tested this result using a binomial test to assess its significance, which confirms the benefits of the tuning, with a $p$-value of $5.537 \times 10^{-10}$.

Table III. Detailed results of the experiment. For each instance, the table reports the average overlap, the standard deviation and the ranking of the algorithms in the comparison.

| Instance | Algorithm | $\bar{x}$ | $s$ | Rank |
|---|---|---|---|---|
| 1 | ASGD | 0.762 | 0.001 | 2 |
| | ASGD-def | 0.755 | 0.002 | 7 |
| | NCG | 0.756 | | 6 |
| | NCG-def | 0.757 | | 5 |
| | r-GA | 0.771 | 0.008 | 1 |
| | r-GA-def | 0.759 | 0.015 | 3 |
| | GD | 0.758 | | 4 |
| | GD-def | 0.740 | | 9 |
| | QN | 0.742 | | 8 |
| | QN-def | 0.708 | | 10 |
| 2 | ASGD | 0.301 | 0.005 | 10 |
| | ASGD-def | 0.680 | 0.003 | 5 |
| | NCG | 0.702 | | 1 |
| | NCG-def | 0.598 | | 9 |
| | r-GA | 0.686 | 0.016 | 4 |
| | r-GA-def | 0.676 | 0.014 | 6 |
| | GD | 0.701 | | 2 |
| | GD-def | 0.669 | | 7 |
| | QN | 0.686 | | 3 |
| | QN-def | 0.628 | | 8 |
| 3 | ASGD | 0.272 | 0.043 | 10 |
| | ASGD-def | 0.595 | 0.005 | 5 |
| | NCG | 0.616 | | 2 |
| | NCG-def | 0.503 | | 8 |
| | r-GA | 0.629 | 0.026 | 1 |
| | r-GA-def | 0.611 | 0.023 | 3 |
| | GD | 0.589 | | 6 |
| | GD-def | 0.575 | | 7 |
| | QN | 0.488 | | 9 |
| | QN-def | 0.614 | | 4 |
| 4 | ASGD | 0.734 | 0.001 | 3 |
| | ASGD-def | 0.723 | 0.002 | 6 |
| | NCG | 0.707 | | 7 |
| | NCG-def | 0.635 | | 10 |
| | r-GA | 0.744 | 0.012 | 1 |
| | r-GA-def | 0.731 | 0.018 | 2 |
| | GD | 0.727 | | 5 |
| | GD-def | 0.729 | | 4 |
| | QN | 0.689 | | 9 |
| | QN-def | 0.702 | | 8 |
| 5 | ASGD | 0.705 | 0.002 | 4 |
| | ASGD-def | 0.694 | 0.002 | 8 |
| | NCG | 0.711 | | 3 |
| | NCG-def | 0.676 | | 9 |
| | r-GA | 0.713 | 0.012 | 2 |
| | r-GA-def | 0.697 | 0.011 | 6 |
| | GD | 0.724 | | 1 |
| | GD-def | 0.699 | | 7 |
| | QN | 0.702 | | 5 |
| | QN-def | 0.637 | | 10 |
| 6 | ASGD | 0.676 | 0.005 | 4 |
| | ASGD-def | 0.658 | 0.006 | 8 |
| | NCG | 0.382 | | 10 |
| | NCG-def | 0.672 | | 5 |
| | r-GA | 0.693 | 0.016 | 2 |
| | r-GA-def | 0.696 | 0.024 | 1 |
| | GD | 0.665 | | 7 |
| | GD-def | 0.678 | | 3 |
| | QN | 0.670 | | 6 |
| | QN-def | 0.637 | | 9 |
| 7 | ASGD | 0.721 | 0.001 | 6 |
| | ASGD-def | 0.716 | 0.001 | 7 |
| | NCG | 0.708 | | 8 |
| | NCG-def | 0.727 | | 4 |
| | r-GA | 0.738 | 0.022 | 2 |
| | r-GA-def | 0.705 | 0.011 | 9 |
| | GD | 0.706 | | 10 |
| | GD-def | 0.729 | | 3 |
| | QN | 0.722 | | 5 |
| | QN-def | 0.743 | | 1 |
| 8 | ASGD | 0.676 | 0.001 | 4 |
| | ASGD-def | 0.667 | 0.001 | 6 |
| | NCG | 0.663 | | 7 |
| | NCG-def | 0.691 | | 1 |
| | r-GA | 0.652 | 0.017 | 9 |
| | r-GA-def | 0.630 | 0.029 | 10 |
| | GD | 0.684 | | 2 |
| | GD-def | 0.669 | | 5 |
| | QN | 0.681 | | 3 |
| | QN-def | 0.657 | | 8 |

Table V. THE NUMBER OF INSTANCES IN WHICH THE ALGORITHM ON THE ROW HAS A BETTER MEAN OVERLAP VALUE THAN THAT ON THE COLUMN.

| | ASGD | ASGD-def | NCG | NCG-def | r-GA | r-GA-def | GD | GD-def | QN | QN-def |
|---|---|---|---|---|---|---|---|---|---|---|
| ASGD | - | 53 | 46 | 42 | 15 | 32 | 26 | 41 | 41 | 41 |
| ASGD-def | 17 | - | 44 | 40 | 13 | 29 | 14 | 27 | 31 | 37 |
| NCG | 24 | 26 | - | 37 | 14 | 28 | 16 | 22 | 27 | 32 |
| NCG-def | 28 | 30 | 33 | - | 11 | 29 | 21 | 19 | 33 | 29 |
| r-GA | 55 | 57 | 56 | 59 | - | 57 | 42 | 51 | 45 | 50 |
| r-GA-def | 38 | 41 | 42 | 41 | 13 | - | 26 | 31 | 36 | 41 |
| GD | 44 | 56 | 54 | 49 | 28 | 44 | - | 49 | 43 | 52 |
| GD-def | 29 | 43 | 48 | 51 | 19 | 39 | 21 | - | 38 | 44 |
| QN | 29 | 39 | 43 | 37 | 25 | 34 | 27 | 32 | - | 38 |
| QN-def | 29 | 33 | 38 | 41 | 20 | 29 | 18 | 26 | 32 | - |

Table IV. RESULT OF NEMENYI'S POST-HOC PROCEDURE WHEN COMPARING R-GA WITH THE OTHER ALGORITHMS. THE TABLE REPORTS THE AVERAGE RANKINGS OF THE ALGORITHMS AND THE ADJUSTED P-VALUE FOR EACH COMPARISON.

| Algorithm | Mean Rank | p-value |
|---|---|---|
| r-GA | 3.24 | |
| GD | 4.06 | 0.0257 |
| ASGD | 5.16 | 0.0000 |
| r-GA-def | 5.23 | 0.0000 |
| GD-def | 5.37 | 0.0000 |
| QN | 5.67 | 0.0000 |
| QN-def | 6.26 | 0.0000 |
| ASGD-def | 6.43 | |
| NCG-def | 6.77 | |
| NCG | 6.81 | |

Table VI. RESULT OF SIGN TEST. THE TABLE LISTS THE ALGORITHMS ALONG WITH THEIR NUMBER OF INSTANCES IN WHICH THEY HAVE BEEN OUTPERFORMED BY R-GA (TABLE V) AND THE ASSOCIATED ADJUSTED P-VALUE.

| Algorithm | Losses | p-value |
|---|---|---|
| ASGD | 54 | 0.0000 |
| ASGD-def | 56 | 0.0000 |
| NCG | 57 | 0.0000 |
| NCG-def | 59 | 0.0000 |
| r-GA-def | 57 | 0.0000 |
| GD | 44 | 0.0828 |
| GD-def | 52 | 0.0002 |
| QN | 44 | 0.0828 |
| QN-def | 50 | 0.0013 |

*2) Comparison of the tuned algorithms:* The overlap values can differ considerably across the instances, reflecting the fact that the effectiveness of this kind of segmentation can vary depending on the concrete anatomy of the patients. From the highest to the lowest average rank we have the order NCG, QN, ASGD, GD and r-GA. NCG delivered the worst performance of the group: it was outperformed by all other tuned algorithms in 43 over 70 scenarios and scored the worst average ranking (6.81).

QN and ASGD have similar average ranking (5.67 and 5.16, respectively), but rather different behaviors. While the results of QN are quite consistent throughout the instances, ASGD occasionally delivered a mean overlap value below half of those of the other algorithms, as can be seen in the case of instances 2 and 3 in Table III. It is interesting to point out that this inconsistent behavior of ASGD was not observed at all in previous comparisons without tuning [9], [44].

GD scored consistently close to the best results, scoring the second best mean ranking after r-GA (4.06) and beating the remaining algorithms in 43 instances. We remark that under tuning, this "simplest" algorithm among those compared is able to deliver a rather good performance.

Finally, r-GA performed better than all other algorithms in 44 of 70 instances and it reached the lowest average rank (3.24), delivering, thus, the best overall performance. The significance of the results is confirmed by two statistical tests. Table IV reports the p-value of Nemenyi's test comparing r-GA against the best ranking algorithms. In all three cases, the test confirms that the performance of r-GA is significantly better than those of the competitors, with the highest p-value being that of GD, 0.0257. The sign-test with respect to the number of wins (Table VI) confirms that the difference between r-GA and the others algorithms is significant, although with less confidence.

## V. CONCLUSION

In this paper, we studied algorithms for the registration of medical images. In particular, we compared a recently proposed evolutionary algorithm [9], against four other well-established algorithms. Our study has been carried out on a benchmark set of brain MRI data, retrieved from the NMR database [21].

Unlike many comparisons of algorithms proposed in the literature, we relied on automatic configuration techniques to set appropriately the parameter values of all the algorithms at hand, a crucial aspect of their effectiveness. Our experimental setup consisted of two phases. In a first training phase, we applied an automatic configuration tool [13] to all algorithms involved in the comparison. Our experimental results showed a consistent improvement through the automatic configuration procedure with respect to the default parameter settings. In the second phase, the testing phase, we compared all algorithms with the settings that were obtained in the first phase. This comparison is made on unseen instances, that is, instances that are different from the training instances used in the first phase. Our experimental comparison, based on statistical tests to assess the significance of the observed differences, shows that r-GA performs significantly better than the other algorithms in terms of the quality of the results.

An interesting direction for future research would be to extend the comparison to a larger set of candidate algorithms and also to examine the performance of the tested algorithms on other registration tasks. In addition, it may be interesting trying to improve the best performing algorithms further. This could include considering restarts of the tested local search methods from appropriately chosen initial solutions using, for example, ideas from iterated local search [45]. Another possibility is to further improve the evolutionary algorithm, which was found to be the currently best performing algorithm from a solution quality perspective.

## ACKNOWLEDGMENTS

Andrea Valsecchi and Jérémie Dubois-Lacoste acknowledge support from the MIBISOC network, an Initial Training Network funded by the European Commission, grant PITN–GA–2009–238819. Thomas Stützle acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Associate. The NMR database is the property of CEA/I2BM/NeuroSpin and can be provided on demand to cyril.poupon@cea.fr. Data were acquired with PTK pulse sequences, reconstructed with PTK reconstructor package and post-processed with Brainvisa/Connectomist software, freely available at http://brainvisa.info.

## REFERENCES

[1] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image Vision Comput.*, vol. 21, pp. 977–1000, 2003.

[2] C. K. Chow, H. T. Tsui, and T. Lee, "Surface registration using a dynamic genetic algorithm," *Pattern Recogn.*, vol. 37, pp. 105–117, 2004.

[3] O. Cordón, S. Damas, and J. Santamaría, "A Fast and Accurate Approach for 3D Image Registration using the Scatter Search Evolutionary Algorithm," *Pattern Recogn. Lett.*, vol. 27, no. 11, pp. 1191–1200, 2006.

[4] O. Cordón, S. Damas, and J. Santamaría, "Feature-based image registration by means of the CHC evolutionary algorithm," *Image Vision Comput.*, vol. 22, pp. 525–533, 2006.

[5] E. Lomonosov, D. Chetverikov, and A. Ekart, "Pre-registration of arbitrarily oriented 3D surfaces using a genetic algorithm," *Pattern Recogn. Lett.*, vol. 27, no. 11, pp. 1201–1208, 2006.

[6] L. Silva, O. R. P. Bellon, and K. L. Boyer, *Robust range image registration using genetic algorithms and the surface interpetenetration measure*. World Scientific, 2005.

[7] J. Santamaría, O. Cordón, and S. Damas, "A comparative study of state-of-the-art evolutionary image registration methods for 3D modeling," *Comput. Vis. Image Underst.*, vol. 115, no. 9, pp. 1340–1354, 2011.

[8] 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, nov. 2011.

[9] A. Valsecchi, S. Damas, J. Santamaría, and L. Marrakchi-Kacem, "Genetic algorithms for voxel-based medical image registration," in *IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2013)*, 2013.

[10] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle, "ParamILS: an automatic algorithm configuration framework," *Journal of Artificial Intelligence Research*, vol. 36, pp. 267–306, Oct. 2009.

[11] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization, 5th International Conference, LION 5*, ser. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, 2011.

[12] T. Bartz-Beielstein, *Experimental Research in Evolutionary Computation: The New Experimentalism*. Berlin, Germany: Springer, 2006.

[13] M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, and M. Birattari, "The irace package, iterated race for automatic algorithm configuration," IRIDIA, Université Libre de Bruxelles, Belgium, Tech. Rep. TR/IRIDIA/2011-004, 2011. [Online]. Available: http://iridia.ulb.ac.be/IridiaTrSeries/IridiaTr2011-004.pdf

[14] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *IEEE T. Med. Imaging*, vol. 22, no. 8, pp. 986–1004, 2003.

[15] F. Maes, D. Vandermeulen, and P. Suetens, "Comparative evaluation of multiresolution optimization strategies for image registration by maximization of mutual information," *Med. Image Anal.*, vol. 3, no. 4, pp. 373–386, 1999.

[16] S. Klein, M. Staring, and J. P. W. Pluim, "Evaluation of optimization methods for nonrigid medical image registration using mutual information and b-splines," *Image Processing, IEEE Transactions on*, vol. 16, no. 12, pp. 2879–2890, 2007.

[17] J. M. Rouet, J. J. Jacq, and C. Roux, "Genetic algorithms for a robust 3-D MR-CT registration," *IEEE T. Inf. Technol. B.*, vol. 4, no. 2, pp. 126–136, 2000.

[18] R. He and P. A. Narayana, "Global optimization of mutual information: application to three-dimensional retrospective registration of magnetic resonance images," *Comput. Med. Imag. Grap.*, vol. 26, pp. 277–292, 2002.

[19] P. Chalermwat, T. El-Ghazawi, and J. LeMoigne, "2-phase GA-based image registration on parallel clusters," *Future Gener. Comp. Sy.*, vol. 17, pp. 467–476, 2001.

[20] B. C. Vemuri, J. Ye, Y. Chen, and C. M. Leonard, "Image registration via level-set motion: Applications to atlas-based segmentation," *Medical Image Analysis*, vol. 7, no. 1, pp. 1–20, 2003.

[21] C. Poupon, F. Poupon, L. Allirol, and J.-F. Mangin, "A database dedicated to anatomo-functional study of human brain connectivity," in *Proceedings of the 12th Annual Meeting of the Organization for Human Brain Mapping*, no. 646, Florence, Italy, 2006.

[22] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[23] D. Rueckert and J. A. Schnabel, "Medical image registration," in *Biomedical Image Processing*, ser. Biological and Medical Physics, Biomedical Engineering, T. M. Deserno, Ed. Springer Berlin Heidelberg, 2011, pp. 131–154.

[24] L. J. Eshelman, "Real-coded genetic algorithms and interval schemata," in *Foundations of Genetic Algorithms 2*, L. D. Whitley, Ed. San Mateo, USA: Morgan Kaufmann, 1993, pp. 187–202.

[25] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. IOP Publishing Ltd and Oxford University Press, 1997.

[26] S. Klein, J. Pluim, M. Staring, and M. Viergever, "Adaptive stochastic gradient descent optimisation for image registration," *International Journal of Computer Vision*, vol. 81, pp. 227–239, 2009.

[27] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. W. Pluim, "elastix: A toolbox for intensity-based medical image registration," *IEEE Trans. Med. Imaging*, vol. 29, no. 1, pp. 196–205, 2010.

[28] "Elastix webpage," http://elastix.bigr.nl, 2012.

[29] F. Hutter, H. H. Hoos, K. Leyton-Brown, and K. P. Murphy, "An experimental investigation of model-based parameter optimisation: SPO and beyond," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2009*, F. Rothlauf, Ed. New York, NY: ACM Press, 2009, pp. 271–278.

[30] P. Balaprakash, M. Birattari, and T. Stützle, "Improvement strategies for the F-race algorithm: Sampling design and iterative refinement," in *Hybrid Metaheuristics*, ser. Lecture Notes in Computer Science, T. Bartz-Beielstein, M. J. Blesa, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, Eds. Springer, Heidelberg, Germany, 2007, vol. 4771, pp. 108–122.

[31] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[32] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp, "A racing algorithm for configuring metaheuristics," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*, W. B. Langdon *et al.*, Eds. Morgan Kaufmann Publishers, San Francisco, CA, 2002, pp. 11–18.

[33] M. Birattari, "The problem of tuning metaheuristics as seen from a machine learning perspective," Ph.D. dissertation, Université Libre de Bruxelles, Brussels, Belgium, 2004.

[34] Birattari, *Tuning Metaheuristics: A Machine Learning Perspective*, ser. Studies in Computational Intelligence. Berlin/Heidelberg, Germany: Springer, 2009, vol. 197.

[35] O. Maron and A. W. Moore, "The racing algorithm: Model selection for lazy learners," *Artificial Intelligence Research*, vol. 11, no. 1–5, pp. 193–225, 1997.

[36] M. Birattari, Z. Yuan, P. Balaprakash, and T. Stützle, "F-race and iterated F-race: An overview," in *Experimental Methods for the Analysis of Optimization Algorithms*, T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, Eds. Berlin, Germany: Springer, 2010, pp. 311–336.

[37] J. Dubois-Lacoste, M. López-Ibáñez, and T. Stützle, "Effective hybrid stochastic local search algorithms for biobjective permutation flowshop scheduling," in *Hybrid Metaheuristics*, ser. Lecture Notes in Computer Science, M. J. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, and A. Schaerf, Eds. Springer, Heidelberg, Germany, 2009, vol. 5818, pp. 100–114.

[38] J. Dubois-Lacoste, M. López-Ibáñez, and T. Stüetzle, "A hybrid TP+PLS algorithm for bi-objective flow-shop scheduling problems," *Computers & Operations Research*, vol. 38, no. 8, pp. 1219–1236, 2011.

[39] J. Dubois-Lacoste, M. López-Ibáñez, and T. Stützle, "Automatic configuration of state-of-the-art multi-objective optimizers using the TP+PLS framework," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2011*, N. Krasnogor and P. L. Lanzi, Eds. New York, NY: ACM Press, 2011, pp. 2019–2026.

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

[41] P. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Princeton University, 1963.

[42] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940. [Online]. Available: http://dx.doi.org/10.2307/2235971

[43] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979. [Online]. Available: http://dx.doi.org/10.2307/4615733

[44] A. Valsecchi, S. Damas, and J. Santamaría, "An image registration approach using genetic algorithms," in *Proceedings of the IEEE World Congress On Computational Intelligence - Congress on Evolutionary Computation 2012*, 2012, pp. 1–8.

[45] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search," in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds. Kluwer Academic Publishers, 2003, pp. 321–353.