Automatic tuning of respiratory model for patient-based simulation

Franck P. Vidal, Pierre-Frédéric Villard, Evelyne Lutton

Automatic tuning of respiratory model for patient-based simulation

Franck P. Vidal, Pierre-Frédéric Villard, and Évelyne Lutton

Abstract—This paper is an overview of a method recently published in a biomedical journal (IEEE Transactions on Biomedical Engineering[1]). The method is based on an optimisation technique called "evolutionary strategy" and it has been designed to estimate the parameters of a complex 15-D respiration model. This model is adaptable to account for patient's specificities. The aim of the optimisation algorithm is to finely tune the model so that it accurately fits real patient datasets. The final results can then be embedded, for example, in high fidelity simulations of the human physiology. Our algorithm is fully automatic and adaptive. A compound fitness function has been designed to take into account for various quantities that have to be minimised (here topological errors of the liver and the diaphragm geometries). The performance our implementation is compared with two traditional methods (downhill simplex and conjugate gradient descent), a random search and a basic real-valued genetic algorithm. It shows that our evolutionary scheme provides results that are significantly more stable and accurate than the other tested methods. The approach is relatively generic and can be easily adapted to other complex parametrisation problems when ground truth data is available.

Index Terms—Evolutionary computation, inverse problems, medical simulation, adaptive algorithm.

I. Introduction

The simulation of complex physiological phenomena often requires highly dimensional models. These models may be adaptable to account for patient's specificities. They can be used in various medical contexts, for example:

- Reducing motion artefacts in positron emission tomography (PET) and cone-beam computed tomography (CBCT) to improve the image quantification [1], [2]
- Accurate dose calculation in radiotherapy treatment planning [3]
- High fidelity computer-based training simulators [4]

The numerical accuracy of such simulations will depend on: i) the intrinsic limitations of the models and ii) on the parameters of the models. The calibration and parametrisation of the models are therefore critical to obtain the highest level of realism. For medical training simulators using virtual reality, the parameters are, however, often manually tuned using trial and error. This is very time consuming and it is not possible to make sure that the results are optimal.

F. Vidal is with the School of Computer Science, Bangor University, Dean Street, Bangor LL57 1UT, UK (e-mail: f.vidal@bangor.ac.uk).

P.-F. Villard is with LORIA, University of Lorraine, France (e-mail: pierre-frederic.villard@univ-lorraine.fr).

É. Lutton was with AVIZ, INRIA Saclay-Île-de-France, France. She is now with MALICES, INRA-AgroParisTech, Thiverval-Grignon, France (e-mail: evelyne.lutton@grignon.inra.fr).

[1]http://tbme.embs.org

In [5], [6] we presented methods to automatically tune such a model. The first method made use of a pure random search and the second made use of the evolutionary approach summarised here. The parametrisation corresponds to finely set 15 parameters of a patient specific respiration model. It will be used throughout the paper as the application example. The model takes into account the deformation of the diaphragm and liver [7]. We solve the parametrisation of this fifteen-dimensional (15-D) model as an inverse problem. The idea is to automatically fit the analytic model to experimental data using an optimisation technique. In our application example, two quantities are simultaneously minimised (topological errors of the diaphragm and liver). Generic approaches – such as downhill simplex, conjugate gradient descent, random search and basic real-valued genetic algorithm – were first deployed. These generic optimisation methods did not provide suitable solutions.

Our approach makes use of an adaptive evolutionary algorithm (EA) that is able to explore a search space with 15 dimensions. We choose an evolutionary framework because:

- EAs can be used when little is known about the function to optimise, e.g. when no derivative is known.
- This function does not need to be very smooth.
- EAs can work with any search space.
- EAs are less likely to stop at local optima than classical deterministic optimisation methods.

Our method is fully automatic and adaptive. It leads to significantly better tuning. The approach is generic and can be easily generalised to other models when ground truth is available and the differences between the model outputs and ground truth can be numerically measured.

Section II briefly presents related work. The analytic model of respiration is described in Section III. Details about our evolutionary algorithm are provided in Section IV. In Section V, the performance of the algorithm is analysed and compared with the performance of the other methods. Some conclusions are finally presented in Section VI.

II. Related Work

A. Optimization Based on Artificial Evolution

Artificial Evolution is the generic name of a large set of techniques that rely on the computer simulation of natural evolution mechanisms. Since the pioneering works of A. Fraser, H.-J. Bremermann, and after them, J. Holland and I. Rechenberg, Artificial Darwinism has progressively gained a major importance in the domain of stochastic optimisation and artificial intelligence [8].

The basic idea in Artificial Evolution is to copy, in a very rough manner, the principles of natural evolution, which let a population be adapted to its environment. According to Darwin's theory, adaptation is based on a small set of very simple mechanisms:

- Random variations
- Survival/Reproduction of the fittest individuals

Computer scientists have transposed this scheme into optimisation algorithms.

Considering evolutionary optimisation as a "black box" is not a good strategy in general because one may lose an opportunity to adapt to the problem. Adapting the evolutionary mechanisms to the specificities of the problem usually improves the efficiency of the algorithms and reduces its computation time. In particularly difficult problems it is always helpful to compare evolutionary approaches to a pure random optimisation to evaluate the improvement due to the "intelligence" set in the genetic operators. This is what we do in Section V. In our context, the random search algorithm generates a given number of sets of 15 random parameters [5]. The set of parameters that provides the lowest fitness is extracted and corresponds to the solution of the optimisation problem.

B. Breathing Simulation

Computer-based simulations of respiration can be found in different areas. The level of fidelity that needs to be reached depends on the application domain. For example, in radiotherapy the model accuracy is the first consideration. To precisely represent the real organ behaviour, complex mechanical laws are used and their equations are solved using the finite-element method [9]. In computer animation for movies a high degree of fidelity is not essential. For example, in [10] a mass-spring system is used in realtime to simulate the thoracic muscles.

For medical training simulators there is a tradeoff between accuracy and speed: i) the perception should be realistic enough to train medical students on cases that are close to the reality and ii) the computation has to be performed in realtime to integrate haptics and graphics. For example, Hostettler *et al.* use a method based on geometrical constraints and the viscera are only modelled using a single envelope that wraps all the organs [11]. The method we propose here takes into account the motion of each organ due to the respiration. It relies on a geometrical method that takes into account mechanical-based parameters.

III. RESPIRATION MODEL

A. Real behavior Analysis

The diaphragm and the intercostal muscles are the main muscles that are involved in the breathing process [12]. In the case of abdominal-surgery simulators, a key challenge is to precisely take them into account as they follow the respiration movement. The liver has an up-and-down displacement while being compressed by the diaphragm deformations.

The diaphragm is composed of two parts: i) a very rigid tendon and ii) muscle parts that contracts and relaxes. The

diaphragm is also attached to the floating ribs and to the spine. In our study we assume that the influence of the intercostal muscle is negligible because the patient is lying on his/her back.

B. Organ behavior modeling

We choose to focus on the liver because it is the targeted organ in [7] and on the diaphragm because it is the active muscle (see above). We use the generalised extension [13] of the 3D ChainMail [14] to simulate the soft-tissue deformation. The main reason for this choice is the relatively small computing time. Instead of computing the deformation field based on time integration of the forces, as other physically-based methods [15], the 3D ChainMail only uses geometrical equations that are quickly computed. It also preserves the use of parameters that have links with bio-mechanical approaches: i) the compression (S_i), ii) the stretching (Str_i) and iii) the shearing (Sh_i) (where i designates the organ of interest).

In Section III-A, we saw the the diaphragm is decomposed into two parts. The diaphragm is split using the Cartesian equation of a plane ($a.x + b.y + c.z + d = 0$). As boundary conditions, we chose to impose a null displacement when the diaphragm is close to the ribs (defined by a distance D_{ribs}) and to impose a uniform 3D force to the whole tendon part (defined by (F_x, F_y, F_z)). The muscle part can deform. The deformation is governed by the muscle elasticity. The muscle stretches from the point attached to the ribs to the moving tendon.

The attachments of the liver to the diaphragm are modelled by a distance (D_{diaph}). It is used to define the points that directly follow the diaphragm. It mimics a rigid link that simulates the compression of the liver by the diaphragm during the respiration process. The other parts of the liver deforms following the ChainMail algorithm.

C. Parameter Analysis

Various parameters have been extracted from the model previously described (see Fig. 1). These parameters are unique to each patient and need to be individually customised. There are bio-mechanical parameters, anatomy-based geometrical constraints and respiration pattern information. More details on these parameters can be found in [5].
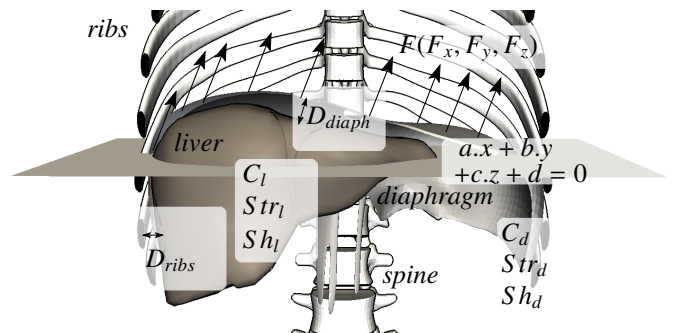

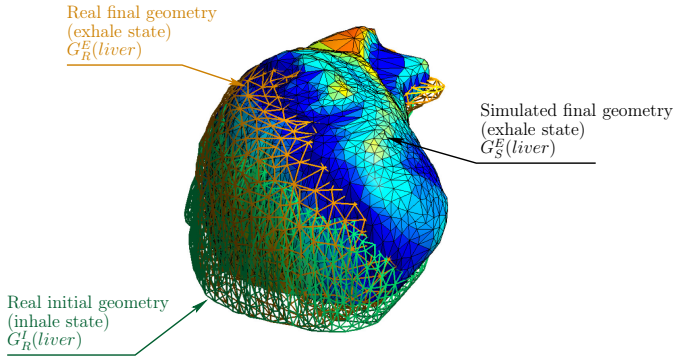Fig. 1. List of the respiration model parameters.

Fig. 2.   Illustration of the vocabulary used for geometries.

Fig. 3.   Evolutionary loop.

D. Model Evaluation

To optimise the model parameters we need datasets of different patients but also metrics to evaluate the accuracy of the respiration model. Three geometrical models are in used at any one time (see Fig. 2). The simulation always starts at the real inspiration state (Geometry G_R^I) extracted by segmenting the real data. The aim is to reach the real expiration state (Geometry G_R^E) with as little numerical errors as possible. G_R^E is also extracted by segmentation of the real data. In the 4D CT case (Geometry G_R^t), going through intermediate states t is also achievable. The metrics assesses the accuracy of a simulated geometry G_S^t compared to the ground truth geometry extracted at the same time step t.

To compute the difference between two geometries, we choose to analyse for each vertex of each mesh the point-to-surface distance [16]. It is based on a distance measure $d(p, G')$ between a point p belonging to a surface G and a surface G' as follows:

$$d(p, G') = \min_{p' \in G'} \|p - p'\| \tag{1}$$

In this application, the root mean square error ($\mathcal{E}_{RMS}\left(G_S^t, G_R^t\right)$) defined by Eq. (2) is used.

$$\mathcal{E}_{RMS}\left(G_S^t, G_R^t\right) = \sqrt{\frac{1}{G_S^t} \iint_{p \in G_S^t} d\left(p, G_R^t\right)^2 \, \mathrm{d}S} \tag{2}$$

IV. Optimization Algorithm

The inverse problem considered here corresponds to an error minimisation problem. The optimisation algorithm will therefore aim at reducing the fitness function (see Section IV-H). A Basic Real-Valued Genetic Algorithm has been first implemented using an elitist generational approach (see Fig. 3). Based on natural selection, different individuals will be competing to produce offsprings. During an optimisation loop, a new population is created from the previous generation of individuals. To create a new individual a genetic operator is randomly chosen. Such operators are: i) *elitism*, ii) *mutation*, iii) *crossover*, and iv) *new blood*. A probability of occurrence is assigned to each operator (W, X, Y, and Z respectively). When the algorithm stops improving the results, there is stagnation and the individual that is the most adapted to the environment is extracted. This is the individual with the smaller fitness function.
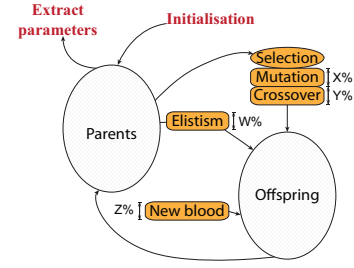
A. Genotype

Each individual embeds 15 parameters (coded as floating point numbers). They correspond to the model unknown values described in Sections III-B and III-C and illustrated in Fig. 1.

B. Tournament Selection

To create a new individual using mutation or crossover, one or several individuals of the previous generation need to be selected. This is performed using a "tournament". A given number of random (x) individuals from the whole population are assessed. The best individual amongst x is selected. It is the one with the lowest fitness.

C. Mutation

$X\%$ of the new individuals are created using the mutation operator. An individual is selected. The new individual is a slightly modified version of the selected individual to introduce spontaneous and random changes:

$$C_i' = C_i + \frac{Range_i}{2} \times k(\sigma) \tag{3}$$

with C_i the i-th parameter of the selected individual, C_i' the new parameter, $Range_i$ the range of possible values for a given parameter (i), k a random number in the interval $[-1, 1]$, and σ a parameter of the evolutionary algorithm that control the range of possible random changes.

D. Crossover

$Y\%$ of the new individuals are created using BLX (for blend crossover) [17]. Two individuals (1 father and 1 mother) are obtained via tournaments. The crossover corresponds to:

$$C_i' = R \times C1_i + (1 - R) \times C2_i \tag{4}$$

with $C1_i$ is the i-th parameter of the father and $C2_i$ is the i-th parameter of the mother, C_i is the i-th parameter of the new individual, and R a random number between 0 and 1. This operation is performed on every gene.

E. Elitism

All the individuals of the population are ranked. $W\%$ of the individuals of the new population correspond to the best individuals of the previous population.
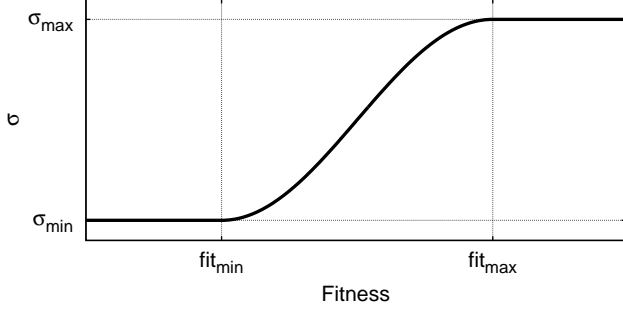
Fig. 4. Adaptive range of mutation controlled by the fitness value.

F. New Blood (or Immigration)

Z% of the individuals of the new population are completely new. Their genes are purely randomly generated.

G. Adaptive Mutation Variance

In classical implementations, the σ value is fixed. A variable, adaptive or self-adaptive mutation is beneficial in optimisation [18]. Adaptive mutation is actually a scheme that has been experimentally observed in natural population, for example stress mutation in bacteria populations [19].

In practice, stress mutation is used in our implementation to use an adaptive strategy for σ. It allows the algorithm to control search space. σ is bigger when fitness (f) is high and smaller when fitness is low. The idea is to favour large exploration around bad individuals, whilst performing fine tuning in the vicinity of good individuals. It makes use of a scaled and stretched cosine function (see Fig. 4). σ in Eq. 3 is replaced by:

$$\sigma(f) = \begin{cases} \sigma_{min}, & f < fit_{min} \\ \sigma_{max}, & f > fit_{max} \\ \sigma_{min} + (\sigma_{max} - \sigma_{min}) \times \\ \dfrac{\cos\left(\pi \times \frac{f - fit_{min}}{fit_{max} - fit_{min}}\right) + 1.0}{2.0}, & \text{otherwise} \end{cases} \quad (5)$$

f corresponds to the fitness of the individual who will undergo a mutation. $\sigma(f)$ smoothly varies between the smaller (fit_{min}) and the larger (fit_{max}) fitness thresholds respectively. If f is smaller than fit_{min}, σ is then σ_{min}; if the individual's fitness is greater than fit_{max}, σ is then σ_{max} (with σ_{min} and σ_{max} two constant values set by the user).

H. Adaptive Fitness Function

A metric ($\mathcal{E}_{RMS}(M_0, M_1)$) is presented in Section III-D to evaluate the discrepancies between two polygon meshes M_0 and M_1. It is therefore possible to quantify the difference between the mesh $\left(G_S^t(i)\right)$ simulated using the deformable model with the parameters corresponding to Individual i, and the real mesh $\left(G_R^t\right)$ extracted from the patient's dataset at State t. For each individual, two metrics are computed (one for the diaphragm, and one for the liver). These values can be used to define the fitness function ($fitness$) corresponding to i.

The optimisation consists in minimising $fitness$. The simplest function is:

$$fitness(i) = \alpha \mathcal{E}_{RMS}\left(G_S^t(i, diaph), G_R^t(diaph)\right) +$$
$$(1 - \alpha)\mathcal{E}_{RMS}\left(G_S^t(i, liver), G_R^t(liver)\right) \quad (6)$$

with $0 \le \alpha \le 1$ to give more or less weight to the diaphragm or the liver. Selecting the value of α is not trivial because the numerical quantity of the error for the diaphragm and the liver can be significantly different. If the same weight is applied to both tissue error measurement (i.e. α is equal to 0.5), the predominant quantity will then have more influence on the optimisation process. We would actually expect instead the importance of both tissues to be the same during the minimisation. In an application such as the real-time simulation of liver puncture, a higher level of fidelity is required for the liver than the diaphragm. Scaling factors on errors are introduced to give the same relative weight to the diaphragm and liver. Eq. 6 becomes:

$$fitness(i) = \frac{\alpha}{\mathcal{E}_{RMS}^{diaph}}\mathcal{E}_{RMS}\left(G_S^t(i, diaph), G_R^t(diaph)\right) +$$
$$\frac{(1 - \alpha)}{\mathcal{E}_{RMS}^{liver}}\mathcal{E}_{RMS}\left(G_S^t(i, liver), G_R^t(liver)\right) \quad (7)$$

with $\mathcal{E}_{RMS}^{liver}$ and $\mathcal{E}_{RMS}^{diaph}$ the error metrics of the best individual provided by the previous generation for the liver and diaphragm respectively. For each iteration of the evolution loop, these metrics are updated. For practical reasons, α is rescaled so that the sum $\frac{\alpha}{\mathcal{E}_{RMS}^{diaph}} + \frac{(1-\alpha)}{\mathcal{E}_{RMS}^{liver}}$ is equal to one.

V. RESULTS AND VALIDATION

The results of the evolutionary algorithm are compared with the outputs of:

- Pure random search: to evaluate the improvement provided by the genetic operators (presented in [5])
- Basic real-valued genetic algorithm (GA) used as a black box evolutionary optimisation (i.e. without adaptive mutation variance and without adaptive fitness function): to assess the efficiency of our new genetic operators
- Two more traditional methods for further comparisons:
  - Downhill simplex method [20]
  - Powell's conjugate gradient descent method [21]

For each optimisation process, the errors were recorded. It allows us to ascertain the effectiveness and usefulness of our adaptive evolutionary algorithm, i.e. to demonstrate that it outperforms the brute force algorithm, the black box EA, and the classic methods.

To allow fair comparisons, the same "evolved fitness" is used for:

- Downhill simplex
- Powell's conjugate gradient descent
- Random search

and the same computing time is used with:

- Random search

Name	Image size	Spacing [mm^3]	Protocol
Patient A	512×512×136	1.08×1.08×2.5	Breath hold
Patient B	512×512×75	0.98×0.98×5	Breath hold
Patient C	512×512×139	1.17×1.17×2	Breath hold
Patient D	512×512×141	0.98×0.98×2	4D CT scan
Patient E	512×512×287	0.71×0.71×1	4D CT scan

TABLE I

PATIENT DATASET PROPERTIES.

population size (n)	200
α	33%
tournament size	5% of the population size
elitism (W)	9%
mutation probability (X)	55%
crossover probability (Y)	35%
new blood probability (Z)	1%
σ_{min}	0.001
σ_{max}	0.2

TABLE II

EVOLUTIONARY ALGORITHM PARAMETERS.

A. Input Data

Five patient specific datasets have been selected (see Table I). Three datasets (Patients A, B and C) have been acquired with the "breath hold" protocol, i.e. with only two time steps corresponding to the inhale and exhale states. The patients are asked to hold their breath following the "ABC" protocol [22]. Two datasets (Patients D and E) correspond to 4D CT scans with ten time steps each. The data was acquired over the respiratory cycle while the patient breathes normally. For Patients A, B and C, a single optimisation problem each needs to be solved. For patients with 4D CT scans, one optimisation problem per time step needs to be solved. Every CT scan has been segmented to extract the organs that are required by the simulation model. Polygon meshes were then exported using the Marching Cube algorithm [23]. Meshes were decimated and smoothed to have about 2,000 vertices per organ.

B. Parameters of the Evolutionary Algorithm

Table II provides a summary of the algorithm's parameters. The size of population is 200 individuals. α is set to 33% to give more weight to the liver than the diaphragm, without neglecting the diaphragm.

## C. Performance Comparison of the Different Methods

Fig. 5 and Fig. 6 show on the y-left axis the average root mean square error ($\mathcal{E}_{RMS}$) of the liver and diaphragm for each patient and for each optimisation technique. They show on the y-right axis the average number of fitness evaluations that was needed to minimise errors. Every stochastic optimisation process (adaptive EA, pure random search, and basic real-valued GA) has been repeated 15 times. The results in Fig. 5 and Fig. 6 show that only our evolutionary algorithm can minimise successfully both the error of the liver and diaphragm. Classic optimisation methods fail to explore the 15-D search space to minimise the two error values. This is particularly true for the downhill simplex method. The basic real-valued genetic algorithm is very slow and fails to converge in some cases (see
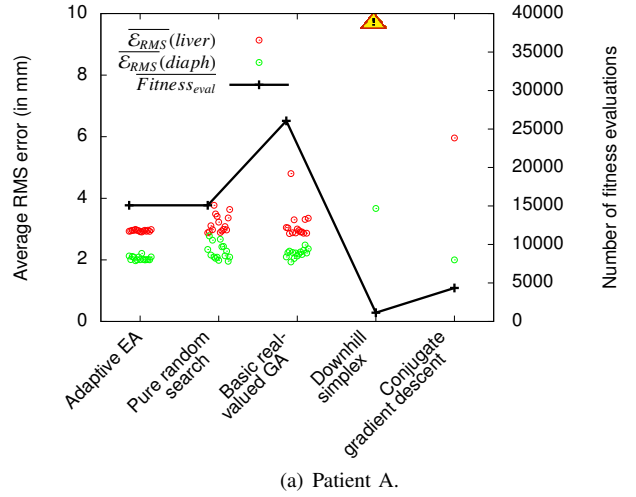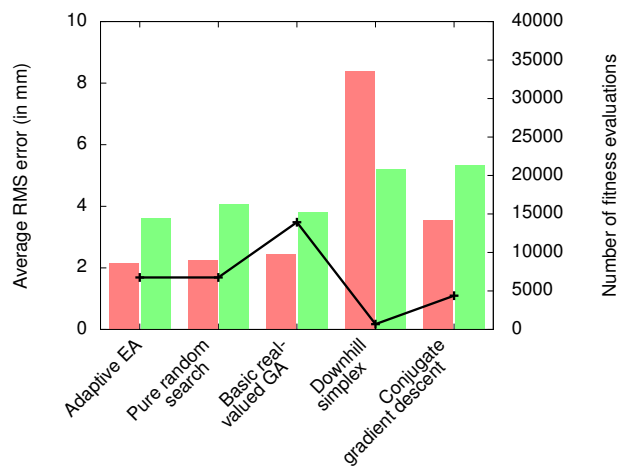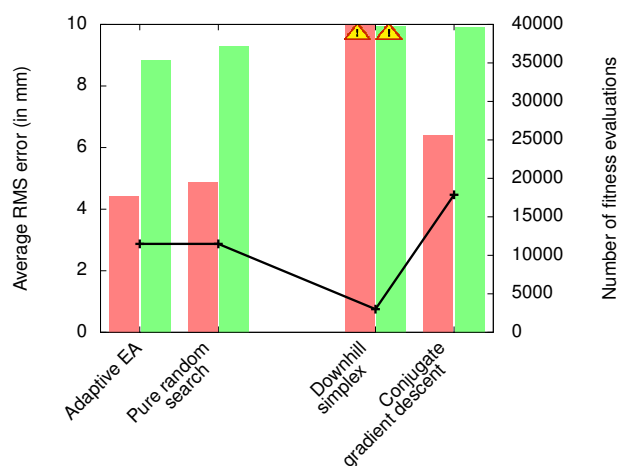

(a) Patient A.
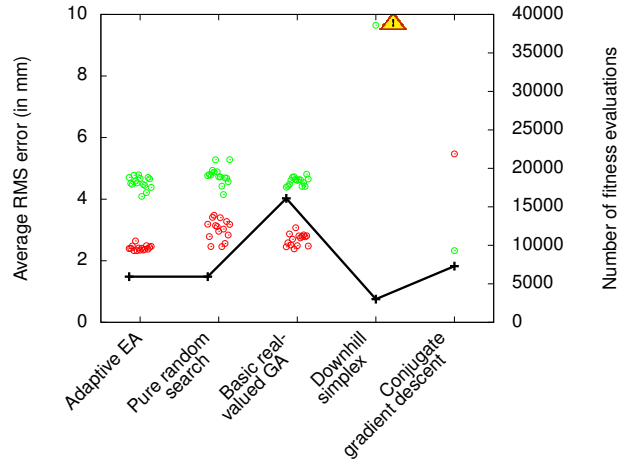
(b) Patient B.
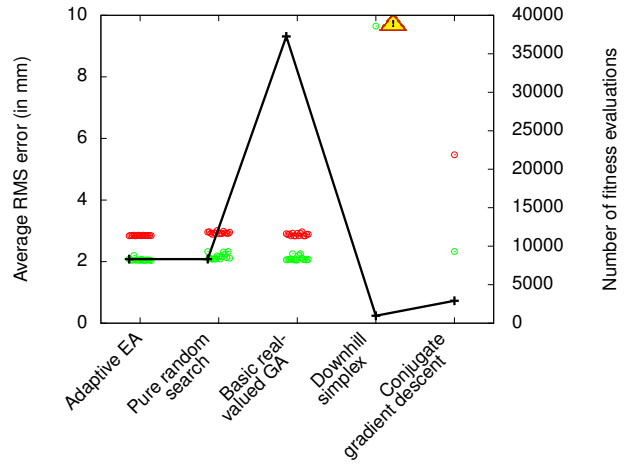
(c) Patient C.

Fig. 5. Results for the three patient datasets that have been acquired with the "breath hold" protocol.

Fig. 6(b)). Fig. 5 also shows that our adaptive EA provides stable results, which is not the case of the other stochastic methods. In addition, the error is lower with our method, which shows the improvement due to the random search oriented by

(a) Patient D.

(b) Patient E.

Fig. 6. Results for the two patient datasets that correspond to 4D CT scans. The values are averaged accross the ten timesteps. For the stochastic optimisation methods, the values are also averaged accross the fifteen runs per timestep.

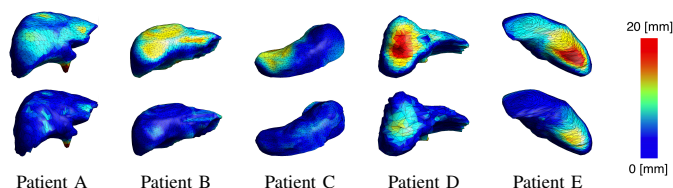Patient A   Patient B   Patient C   Patient D   Patient E

Fig. 7. 3D plots of surface meshes with localised errors. The first row shows the initial difference map between real inhale and real exhale states. The second row shows the difference between real exhale and simulated exhale with our genetic algorithm.

the genetic operators over a purely "blind" random search.

Finally, Fig. 7 shows 3D plots of surface meshes for all the patients. The printed colour depends on a lookup table (LUT) that corresponds to the localised error. Its range varies from blue for no error to red for the maximum error. The motion is fairly well compensated using our genetic algorithm.

VI. Conclusion and Future Work

We have presented an artificial evolution strategy to finely tune the parameters of a multidimensional model of respiration with soft tissue deformations. Further details can be found in our original paper [6]. The efficiency of the method has been validated using five datasets of real patients (that is 23 different optimisation problems in total). The advantage of artificial evolution over the downhill simplex, the conjugate gradient descent, the purely random search, and a black box basic real-valued genetic algorithm has also been demonstrated. Results obtained using our artificial evolution framework were both more accurate and more stable.

The proposed evolutionary optimisation is adaptive in two ways:

- The mutation variance is adapted using the fitness
- The weight the two-objective compound fitness is automatically balanced

The current solution that is to balance the different objectives in a single fitness function can be revisited. A cooperative-coevolutionary approach can be used as the problem we presented here includes most of the features that have been identified to be difficult to solve using single-population evolutionary algorithms [24]. Also a Classical multi-objective evolutionary approach like the famous *NSGA-II* [25] will be also considered for dealing with multiple objectives.

Another alternative will be the use of packages for automatic algorithm configuration, such as *irace* or *SPOT* [26], [27].

References

[1] F. Lamare *et al.*, "List-mode-based reconstruction for respiratory motion correction in PET using non-rigid body transformations," *Phys Med Biol*, vol. 52, no. 17, pp. 5187–5204, 2007.

[2] Q. Zhang *et al.*, "Correction of motion artifacts in cone-beam CT using a patient-specific respiratory motion model," *Med Phys*, vol. 37, no. 6, pp. 2901–2909, 2010.

[3] J. H. Lewis and S. B. Jiang, "A theoretical model for respiratory motion artifacts in free-breathing CT scans," *Phys Med Biol*, vol. 54, no. 3, pp. 745–755, 2009.

[4] P. J. Morgan *et al.*, "Applying theory to practice in undergraduate education using high fidelity simulation," *Med Teach*, vol. 28, no. 1, pp. e10–e15, 2006.

[5] P.-F. Villard *et al.*, "A method to compute respiration parameters for patient-based simulators," in *Medicine Meets Virtual Reality 19 - NextMed*, ser. Stud Health Technol Inform, vol. 173, 2012, pp. 529–533.

[6] F. P. Vidal, P. Villard, and E. Lutton, "Tuning of patient specific deformable models using an adaptive evolutionary optimization strategy," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 10, pp. 2942–2949, Oct. 2012.

[7] P.-F. Villard *et al.*, "A prototype percutaneous transhepatic cholangiography training simulator with real-time breathing motion," *Int J Comput Assist Radiol Surg*, vol. 4, no. 6, pp. 571–578, 2009.

[8] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 3rd ed. Wiley-IEEE Press, Dec. 2005, pp. 106–107.

[9] A. Al-Mayah *et al.*, "Effect of friction and material compressibility on deformable modeling of human lung," in *Proceedings of the 4th international symposium on Biomedical Simulation*, ser. ISBMS '08. Springer-Verlag, 2008, pp. 98–106.

[10] V. B. Zordan *et al.*, "Breathe easy: model and control of simulated respiration for animation," in *Proceedings of the 2004 ACM SIG-GRAPH/Eurographics symposium on Computer animation*, ser. SCA '04, 2004, pp. 29–37.

[11] A. Hostettler *et al.*, "A real-time predictive simulation of abdominal viscera positions during quiet free breathing," *Progress in Biophysics and Molecular Biology*, vol. 103, no. 2 - 3, pp. 169–184, 2010.

[12] K. Moore, A. F. Dalley, and A. M. R. Agur, *Clinically Oriented Anatomy*, 6th ed. Lippincott Williams & Wilkins, 2009.

[13] Y. Li and K. Brodlie, "Soft object modelling with generalised chainmail extending the boundaries of web-based graphics," *Computer Graphics Forum*, vol. 22, no. 4, pp. 717–727, 2003. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2003.00719.x

[14] S. F. Gibson, "3D chainmail: a fast algorithm for deforming volumetric objects," in *Proc Symp on Interactive 3D Graphics*, 1997, pp. 149–154.

[15] U. Meier *et al.*, "Real-time deformable models for surgery simulation: a survey," *Computer Methods and Programs in Biomedicine*, vol. 77, no. 3, pp. 183–197, 2005.

[16] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "MESH: Measuring Errors between Surfaces using the Hausdorff distance," in *Proc IEEE Int Conf on Multimedia and Expo*, vol. I, 2002, pp. 705–708.

[17] F. Herrera, M. Lozano, and A. M. Sánchez, "A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study," *Int J Intell Syst*, vol. 18, no. 3, pp. 309–338, 2003.

[18] T. Bäck, "Optimal mutation rates in genetic search," in *Proc 5th Int Conf on Genetic Algorithms*, 1993, pp. 2–8.

[19] M. N. Hersh *et al.*, "Adaptive mutation and amplification in Escherichia coli: two pathways of genome adaptation under stress." *Res Microbiol*, vol. 155, no. 5, pp. 352–9, 2004.

[20] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[21] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *Computer Journal*, vol. 7, no. 2, pp. 152–162, 1964.

[22] J. W. Wong *et al.*, "The use of active breathing control (ABC) to reduce margin for breathing motion," *Int. J. Radiation Oncology Biol. Phys.*, vol. 44, no. 4, pp. 911–919, 1999.

[23] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987.

[24] C. A. Pena-Reyes and M. Sipper, "Fuzzy CoCo: A cooperative-coevolutionary approach to fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 5, pp. 727–737, 2001.

[25] K. Deb *et al.*, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE T Evolut Comput*, vol. 6, no. 2, pp. 182–197, 2002.

[26] M. López-Ibáñez *et al.*, "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

[27] T. Bartz-Beielstein, *Experimental Research in Evolutionary Computation: The New Experimentalism*, ser. Natural Computing Series. Springer, 2006.