COASTAL CURRENT PREDICTION USING DIFFERENTIAL EVOLUTION

Prédiction du courant côtier en utilisant l'évolution différentielle

Andrea, G.B., Tettamanzi[1]
Università degli Studi di Milano, DTI, via Bramante 65, 26013 Crema (CR), Italy
andrea.tettamanzi@unimi.it

Christel Dartigues-Pallez, Denis Pallez, Célia da Costa Pereira
Université de Nice Sophia, Antipolis, I3S (CNRS UMR-6070), 2000, route des Lucioles, 06903 Sophia Antipolis, France
christel.dartigues-pallez@unice.fr, denis.pallez@unice.fr, celia.pereira@unice.fr

Philippe Gourbesville, Université de Nice Sophia, Antipolis, Polytech'Nice-Sophia, 930, route des Colles, 06903 Sophia Antipolis, France
Phone: +33 (0)4 92 38 85 12, Fax: +33 (0)4 92 96 50 55, E-Mail: gourbesv@unice.fr

KEY WORDS

Current prediction, Evolutionary Computation, Differential Evolution, Gaussian model

ABSTRACT

We propose a data-driven evolutionary approach for modeling marine currents in the Bay of Monaco. Differential Evolution, an evolutionary algorithm well suited for continuous problem, is used to optimize the parameters of a predictive model that may be used as a surrogate of expensive and time-consuming finite-element simulations. This algorithm is compared to previous experiments based on previous optimisation technique as Evolution Strategies Monte Carlo search and a popular quasi-Newton optimization method, namely BFGS. The model allows to predict with reasonable accuracy the curren tspeed and direction at any arbitrary point of the bay from the current speed and direction in the (fixed) locations of the sensors, without having to perform an expensive and time-consuming simulation. The results show that Differential Evolution outperforms Evolution Strategies for this problem.

1. INTRODUCTION

During the last years, the monitoring of continental and marine environments has known a tremendous improvement with the combined evolution of the monitoring technologies and the numerical modeling tools. The evolution of technology, with new types of sensors such as acoustic Doppler current profilers (ADCPs), multibeam sonars, light detection and ranging (LIDAR), and pushbroom cameras, has deeply modified the quality and the quantity of data available for modeling tools [5]. Even if ADCPs offer a better precision, there still remain some problems to solve. One of them concerns the complexity of the marine currents, which is the result of a complex mix of regional and local currents.

This paper describes a data-driven, evolutionary approach to the modeling of the hydrodynamics of a specific, largescale marine environment in the framework of several study campaigns carried over in the Bay of Monaco. Since 2006, the Principality of Monaco has promoted a sea extension project, with the idea to extend the 2 km^2 urban domain, following various approaches which envisage the construction of floating platforms secured to the land.

To avoid the water stagnation problems and eutrophication processes encountered in similar projects like in Qatar and Dubai, to preserve the exceptional quality of the coast line which is still covered with red coral and protected sea grass species, the Monaco government has initiated an indepth investigation of the marine environment in order to have an exhaustive and detailed knowledge of currents patterns taking place along the Monaco coastline and around the future extension of the Principality.

[1] Corresponding author

Detailed historical series of current data have been acquired during several campaigns carried out in the sea section in front of Monaco by means of eight ADCPs deployed in strategic positions for catching the main characteristics of currents. The time series comprise measures of the direction and speed of current for each sensor at different depths, as well as data on the water temperature. These data could be used to calibrate a 3D hydrodynamic model describing the fluid dynamics of a body of sea water, with boundary conditions given by the specific bathymetry of the bay.

The purpose of the research work described in this paper is to approach this problem by using the available data to learn a predictive model of water circulation which captures the specificities of the Bay of Monaco in a succint and actionable representation. The model should allow to predict with reasonable accuracy the current speed and direction at any arbitrary point of the bay from the current speed and direction in the (fixed) locations of the sensors, without having to perform an expensive and time-consuming simulation.

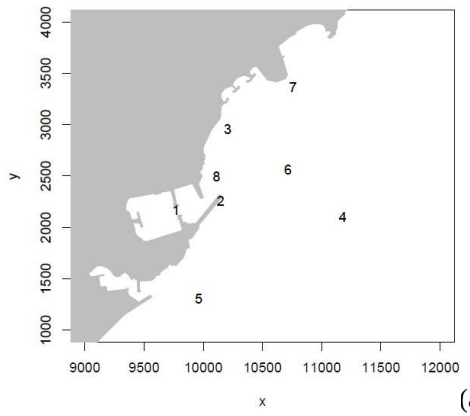
Recently, successful attempts at doing something similar have been made by using artificial neural networks (ANNs) in the somehow related hydrological settings of river stage forecasting in Bangladesh [8] and of tsunami forecasting [10]. ANNs have also been used, with promising results, in the same setting of the Bay of Monaco, to predict a profile of the current based on measurement of the wind at a number of nearby metheorological stations [1]. Although we do not use ANNs in this work, our proposal draws inspiration from such data-driven approaches to hydrological forecasting. The main advantage of our approach with respect to the ANN-based approaches is that we obtain more intuitive models that are easely interpreted by the experts.

The paper is organized as follows: Section 2 defines the modeling problem and Section 3 describes the data available for approaching it. Section 4 proposes a method whereby the modeling problem is reformulated as a parameter optimization problem that can be solved by means of an evolution strategy. Experimental results are discussed in Section 5 and some conclusions are drawn in Section 6.

2. PROBLEM DESCRIPTION

Two measurement campaigns were organized in 2006 in summer and winter conditions in order to obtain an accurate view of the currents along the cost during those periods. For this purpose, eight ADCP's were moored from the 7th of July to the 7th of August and from the 16th of November and the 17th of December. The location of the various ADCP's were selected in order to provide an optimal picture of the complex current patterns generated by the regional and local circulation.

The problem is to find a function that, for any time t, given the measurements of the current magnitude and direction in the fixed locations $x_1; \ldots x_8$ of the sensors, $v(t;\, x_1), \ldots, v(t;\, x_8)$, computes (i.e., predicts) the unknown magnitude and direction $v(t;\, x)$ at an arbitrary point x of the sea section of interest.



(a)

ADCP 1		…	ADCP 8	
u_1	v_1	…	u_8	v_8
$u_1(0)$	$v_1(0)$	…	$u_8(0)$	$v_8(0)$
$u_1(1)$	$v_1(1)$	…	$u_8(1)$	$v_8(1)$
.	.		.	.
.	.		.	.
.	.		.	.
$u_1(t)$	$v_1(t)$	…	$u_8(t)$	$v_8(t)$
.	.		.	.
.	.		.	.
.	.		.	.
$u_1(N)$	$v_1(N)$	…	$u_8(N)$	$v_8(N)$

(b)

Figure 1: (a) Relative positions of the eight ADCPs; (b) Structure of the datasets.

Here, we are just interested in modeling the current at a constant depth of 20 m below the sea level. In other words, of the whole 3D hydrodynamics of the bay, we choose to study a 2D section that is relevant to the problem of characterizing the circulation of water near the surface. Of course, this choice is justified by the specific application (understanding the environment the floating platforms would be placed in). However, the method we describe in the following sections could be generalized to approach a full 3D modeling problem.

The current magnitude and direction, in this 2D section, is represented as a vector $v = (u, v)$, where u is the longitudinal component and v is the latitudinal component of the speed, measured in m/s, of a hypothetical

water particle. A point of the sea section of interest is represented, as well, by a vector $x = (x, y)$ in the Lambert III France metric coordinate system of the Lambert conformal conic projection. Notice that, in this system, both the longitudinal component x and the latitudinal component y are expressed in m, which makes it easy to compute distances between points. The positions of the eight ADCPs deployed in the Bay of Monaco are shown in Figure 1(a).

3. MATERIAL

For the purpose of the study decribed in this paper, eight ADCPs have been placed throughout the coastal in order to collect data from differents positions and different depths. Two datasets of vectors which represents the current at the positions of the ADCPs at given moments in time have been extracted. The first dataset corresponds to the data collected during the summer campaign of 2006, while the second data set corresponds to the data collected during the autumn campaign of the same year. The structure of the dataset is summarized in Figure 1(b). Each row contains the current vectors recorded for all eight ADCPs at a given time. Actually, these are average values over the same 3-, 6-, 12-, or 24-hour period. Each current vector is represented by its longitudinal (i.e., eastward) component, u, and its latitudinal (i.e., northward) component, v. This gives a total of 16 columns (two for each ADCP). The times at which these readings were taken are not important, as long as they coincide for all vectors in a row. Therefore, they are not included in the dataset.

The datasets used for this study are the ones containing the average current vectors, over 3 hour periods, at a depth of 20 meters below the sea level, recorded, as said previously, during the summer and the autumn 2006 campaigns. Both datasets comprise 248 records, which have been randomly divided into a training and a test set. The training set contains 217 records (87.5% of the data), while the test set contains 31 records (12,5% of the data). Some current vectors are missing due to technical reasons, precisely 85 vectors in the summer training set and 12 vectors in the summer test set, besides the u_5 and v_5 columns in the autumn sets.

Note that, in fact, each record is equivalent to 8 training (or test) examples, as the current measured by each sensor may be used as the target for prediction based on the currents measured by the remaining 7 sensors. Therefore, 248 records produce 1984 training examples. These data are representative of current patterns in the Bay of Monaco over the whole year.

4. METHOD

To approach the problem described above, the idea is to adopt an ad hoc representation of a predictive model for marine currents, suggested by the nature of the phenomenon under study. A prediction by interpolation of current v at an arbitrary position x could be obtained by means of a weighted average of the currents measured by the eight

$$v(x) = \frac{\sum_{i=1}^{C} \frac{v(x_i)}{d(x,x_i)}}{\sum_{i=1}^{C} \frac{1}{d(x,x_i)}}, \qquad (1)$$

where $x_i \mathrm{x_i}$ is the position of the ith sensor, C = 8 is the number of sensors, and $\mathrm{d}(.,.)$ is a norm of the difference of its arguments.

In general, an interpolative model like the one in Equation 1 may be a good starting point to make predictions when nothing else is known about the context in which the current is measured. However, it will fail to take into account any specific knowledge of how currents behave in the particular environment that is being studied.

The idea, then, is to replace the generic weighting factor $1/\mathrm{d}(\mathrm{x}, \mathrm{x_i})$ in Equation 1 with several weighting functions $f_i(\mathrm{x})$, one for each available sensor, whose definition captures the influence the measures read by the ith sensors have on the prediction of the current at position x. The predictor, thus, becomes

$$v(x) = \frac{\sum_{i=1}^{C} \mathrm{v(x_i)} f_i(\mathrm{x})}{\sum_{i=1}^{C} f_i(\mathrm{x})}. \qquad (2)$$

According to such representation, the specific knowledge of the bathymetry of the bay would be represented by the shapes of functions f_i.

These weighting functions must have a large enough granularity in order to avoid the risk of overfitting the data. Furthermore, it is reasonable to assume that the weight of an ADCP on the prediction of current at a given position decreases as its distance from that position increases. This naturally suggests to model the

influence of each ADCP's readings as a two-dimensional bell-shaped function, such as the general two-dimensional elliptical Gaussian function centered in x_0, the position of the sensor,

$$f(x) = h.e^{-r^T C r} \quad (3)$$

where h is the height, i.e., the maximum of $\mathrm{f}(.), \mathrm{r} = x - x_0$, and the matrix

$$C = \begin{bmatrix} \mathrm{a} & \mathrm{b} \\ \mathrm{b} & \mathrm{c} \end{bmatrix}$$

is positive-definite. Equation 3 may thus be rewritten as

$$f(x,y) = h.\,\mathrm{e}^{-\mathrm{a}r_x^2 - 2\mathrm{b}r_x r_y - \mathrm{c}r_y^2}\ , \quad (4)$$

where $\mathrm{r_x} = x - x_0$, and $\mathrm{r_y} = \mathrm{y} - \mathrm{y_0}$.

To deal with missing values, the summations in Equation 2 are carried out over the sensors whose data are available; if data for a sensor is missing, the relevant term in the summations is skipped.

A predictive model of the form of Equation 2, where the $f_i(.)$ are defined as in Equation 3, is completely determined by 4C real-valued parameters:

$$m = \left(h_1, a_1, b_1, c_1, \ldots, h_C, \mathrm{a}_C, b_C, c_C\right). \quad (5)$$

Given the current vectors read by the ADCPs at a certain moment, the predictor may be used to compute an estimate of the current in any arbitrary position. To calibrate the model, we may use a dataset of past readings of the current vectors.

The problem of finding a good current predictor of this form may thus be formulated as an unconstrained continuous minimization problem, whose objective function computes the root mean square deviation (RMSD) of the predicted current vectors with respect to the historical vectors recorded in the training set.

The RMSD of the predictions made by Equation 2 with parameters given by vector m of Equation 5 is computed as

$$RMSD(m) = \sqrt{\frac{1}{\mathrm{c}_n}\sum_{\mathrm{i}=1}^{n}\sum_{\mathrm{j}=1}^{\mathrm{C}} \parallel \mathrm{v_{ij}} - \mathrm{predict}\left(\mathrm{x_j}, \mathrm{i}; \mathrm{m}\right) \parallel^2} \quad (6)$$

where n is the number of records in the dataset considered, x_j is the position of the j[th] sensor, v_{ij} is the reading of the j[th] sensor recorded in the i[th] row of the dataset, and

$$\mathrm{predict}\left(\mathrm{x_j}, \mathrm{i}; \mathrm{m}\right) = \frac{\sum_{\mathrm{k}\neq\mathrm{j}} \mathrm{v_{ik}} f_{\mathrm{k}}(\mathrm{x_j};\mathrm{m})}{\sum_{\mathrm{k}\neq\mathrm{j}} f_{\mathrm{k}}(\mathrm{x_j};\mathrm{m})}. \quad (7)$$

4.1 Representation

Instead of encoding parameters a, b, and c directly in the genotype, we encode the angle θ and the standard deviations σ_x and σ_y along the two axes (longitude and latitude), from which parameters a, b, and c may be computed based on the following equations:

$$a = \frac{\cos^2\theta}{2\sigma_x^2} + \frac{\sin^2\theta}{2\sigma_y^2}, \quad (8)$$

$$b = \frac{\sin 2\theta}{4\sigma_y^2} - \frac{\sin 2\theta}{4\sigma_x^2}, \quad (9)$$

$$c = \frac{\sin^2\theta}{2\sigma_x^2} + \frac{\cos^2\theta}{2\sigma_y^2}, \quad (10)$$

This choice has the advantage that all values for θ, σ_x and σ_y, even negative ones, yield a positive-definite matrix C. As a consequence, a candidate solution to the modeling problem may be expressed as a vector

$$\left(h^{(1)}, \theta^{(1)}, \sigma^{(1)}, \sigma^{(1)}, \ldots, h^{(C)}, \theta^{(C)}, \sigma^{(C)}, \sigma^{(C)}\right) \in \mathbb{R}^{\mathrm{C}}. \quad (11)$$

4.2 The Differential Evolution Strategy

Differential Evolution (DE) is a search heuristic introduced by [11]. Its remarkable performance as a global optimization algorithm on continuous numerical minimization problems has been extensively explored [12]. DE belongs to the class of evolutionary algorithms [2, 3] which use biology-inspired operations of crossover, mutation, and selection on a population of candidate solutions in order to minimize an objective function over the course of successive generations. DE uses floating-point instead of bit-string encoding of population members, and arithmetic operations instead of logical operations in mutation. DE is particularly well-suited to find the global optimum of a real-valued function of real-valued parameters as CMA-ES [13], and is known to be able to handle non-differentiable, nonlinear, and multimodal objective functions and to be easy to use.

Differential Evolution approach function optimization problems in the n-dimensional real space by exploiting a real encoding of the objective function parameters. Candidate solutions are n-dimensional vectors. Essentially, for each individual of the population (target vector x_i(t)), a mutant vector m_i(t) is first

generated by adding the weighted difference (difference vector) between two randomly chosen vectors (parameter vectors $p_{i_1}(t)$ and $p_{i_2}(t)$) to a third chosen vector (base vector $b_{i_3}(t)$) as follows:

$$m_i(t) = b_{i_3}(t) + F.\left(p_{i_1}(t) - p_{i_2}(t)\right) \quad (12)$$

where $i \neq i_1 \neq i_2 \neq i_3$; i_1, i_2 are randomly and uniformly chosen between 1 and the population size and $F \in IR^+$ is scaling factor, controlling the amplification of the differential variation. Secondly, one child, called the trial vector, is obtained by crossover of the mutant vector and the target vector. Finally, the target vector is replaced in the population by the best of either the trial or target vector. There are some variations in how to determine the base vector and the difference vector; see [12] for a more detailed explanation. Although the algorithm has proved to be very effective when used to solve real-world problems, its design can make its application to high dimensional optimization problems very expensive.

5. EXPERIMENTS

We applied DE to the optimization of the predictor parameters, represented as explained in Section 4.1. In particular, we used the implementation in R provided by the "DEOptim" package [14]. We compare the results with previous results we have conducted [13].

5.1 Experimental Protocol

The R package for using DE allows us to use 6 different strategies for DE. After some experiments, DE/current-to-p-best/1 strategy is the most suited for solving our problem. In this DE variant, the top (100 *p) percent best solutions are used in the mutation, where p equals to 0.3 in our case. The value of F was set to 0.6, the value of CR (Crossover Rate) to 0.8 and the population size set to the value of C in equation 11.

We stop each run after 200 generations, which take between 1h40 and 3h to complete on the machines we used. This results in exactly 4,000 objective function evaluations per run.
The same number of objective function evaluations is used for the Monte Carlo method as well. The probability distribution used to randomly sample the parameter space is

$$h^{(i)} \sim \varepsilon(1),$$
$$\theta^{(i)} \sim \mathcal{U}(0,2\pi),$$
$$\sigma_x^{(i)} \sim \sigma_y^{(i)} \sim \varepsilon(10^{-3}),$$

where $\varepsilon(\lambda)$ stands for an exponential distribution with mean $\frac{1}{\lambda}$ and $\mathcal{U}(a,b)$ stands for a uniform distribution over the interval [a; b).
The BFGS is run with the default parameters of its implementation under the R system [9], which include stopping after 100 iterations. Each iteration of the BFGS method performs one evaluation of the objective function and one evaluation of its gradient. Since a closed form for the gradient of the RMSD of Equation 6 could not be given, BFGS must estimate each partial derivative numerically, by sampling two points along the relevant dimension; therefore, each gradient evaluation requires 64 evaluations of the objective function, yielding a total of 6,500 objective function evaluations per run, 62.5% more that a CMA-ES or Monte Carlo run. This "bonus" is compensated, in our opinion, by the handicap of BFGS not being a global optimization method like CMA-ES and the Monte Carlo method.
We have applied the methods to two modeling tasks:

- Summer—using the training dataset from the July– August 2006 campaign, learn a predictive model of the currents typical of the hot season;
- Autumn—using the training dataset from the November– December 2006 campaign, learn a predictive model of the currents typical of the cold season.

Since we know that the current patterns are very different for the hot and the cold season, the last task is considerably harder. Furthermore, since more data have to be taken into account, the Summer + Autumn task is also more demanding in terms of computing resources.

5.2 Results

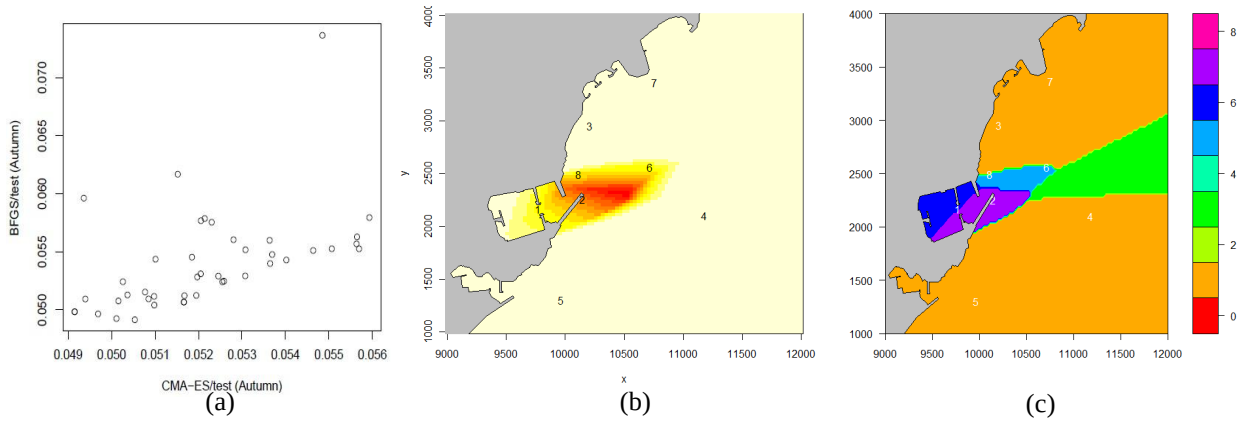
Table 1 shows statistics of the results obtained by the four methods considered over 30 independent runs for the Summer 2006 data, and 30 independent runs for the Autumn 2006 data. In [13], we have empirically

determined that no single model can be found that works equally well for all seasons. This is the reason why, here, we did not even try to fit models on the whole Summer + Autumn dataset.
We can notice that, both for Summer and Autumn, the average error of the results obtained by DE on the test set is lower than those obtained by the other three methods. This is made stronger by the fact that the standard deviation is very small, which means that every runs tends to provide very similar results. We can also notice that, on the Summer test set, the minimum error obtained by DE is lower than the minimum error obtained by the other three methods; instead, for Autumn, the minimum obtained by DE is lower than the minimum obtainend by the Monte Carlo method but slightly higher than those obtained by both CMA-ES and BFGS. Finally, we can also notice that, both on the Summer and Autumn test sets, the maximum error committed by DE is lower than those committed by the other three methods.

Method/Dataset	avg	stdev	min	max
CMA-ES/training (Summer)	0.056457	0.002260	0.052756	0.061005
CMA/ES/test (Summer)	0.056735	0.002455	0.053288	0.062262
CMA-ES/training (Autumn)	0.061989	0.002471	0.058034	0.069365
CMA-ES/test (Autumn)	0.052154	0.001902	0.049150	0.055932
BFGS/training (Summer)	0.057639	0.002174	0.053336	0.061856
BFGS/test (Summer)	0.057756	0.002388	0.053868	0.062925
BFGS/training (Autumn)	0.064292	0.005092	0.058413	0.084768
BFGS/test (Autumn)	0.053964	0.004281	0.049144	0.073654
Monte Carlo/training (Summer)	0.056667	0.000563	0.055254	0.057709
MonteCarlo/test (Summer)	0.056904	0.000915	0.054976	0.058496
Monte Carlo/training (Autumn)	0.062398	0.000683	0.061274	0.063883
Monte Carlo/test (Autumn)	0.052503	0.000733	0.051331	0.054474
DE/training (Summer)	0,0546042933	0,0005462391	0,0535045	0,0560547
DE/test (Summer)	0,0549436	0,0009477908	0,053171	0,0576263
DE/training (Autumn)	0,05909073	0,0004002111	0,0582641	0,0597088
DE/Test (Autumn)	0,0501928933	0,0004382537	0,0496309	0,0510463

Table 1: Summary of the comparison of CMA-ES, BFGS and Monte Carlo

The independent samples, heteroskedastic t-test performed on the experimental results on the test datasets indicates, with a confidence greater than 99.99%, the superiority of DE in comparison to the other three methods.



(a) (b) (c)

Figure 2: (a) Scatter plot of the RMSDs obtained by the BFGS against those obtained by the CMA-ES on the Autumn test set, starting from the same seed solutions **(b)** Superposed Gaussian functions of the best model for the Summer dataset. **(c)** Zones of influence of the sensors for the model of Figure . Each color corresponds to the sensor whose number is given in the right bar, and shows the zone where the weigh of the sensor is greater than the weight of all the other sensors

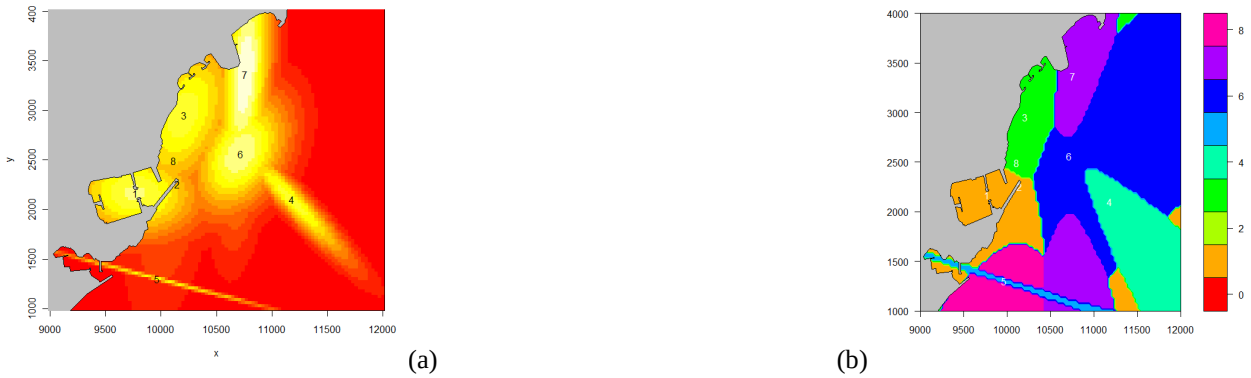
In particular,

- DE performs better than CMA-ES with p-value 5.81×10^{-5} on the Summer test set and with p-value 1.98×10^{-8} on the Autumn test set;
- DE performs better than BFGS with p-value 7.22×10^{-9} on the Summer test set and with p-value 4.36×10^{-7} based on the Autumn test;
- DE performs better than the Monte Carlo method with p-value 1.63×10^{-12} on the Summer test set and with p-value 1.66×10^{-26} on the Autumn test set.

5.3 Visualization of the Results

To allow the interpretation of the models thus obtained, we propose four types of diagrams. Figure (b) and Figure (a) show images of the superposition of the eight weighting functions: lighter areas correspond to higher weights and darker areas to lower weights. The weight of the superposition in a given position is the maximum of the weights of the weighting functions. This type of image gives an overall idea of the model structure. In particular, the model shown in Figure (b) is the best model found by the CMA-ES for the Summer dataset; the one shown in Figure (a) is the best model found by BFGS for the Autumn dataset—the best model found by the CMA-ES is almost identical.

Figure (c) and Figure (b) give an idea of how the eight sensors distribute their influence on the prediction of the current all over the area considered. The zones where the contributions of one sensor prevail over the others are filled with the color associated with that sensor. Zones not covered by any sensor would appear in red. However, all the models found by the three methods we have used cover the whole area.



(a) (b)

Figure 3: (a) Superposed Gaussian functions of the best model for the Autumn dataset. Each color corresponds to the sensor whose number is given in the right bar, and shows the zone where the weight of the sensor is greater than the weight of the other sensors

From these diagrams, it can be observed that the models obtained capture relevant apsects of the phenomenon under study and represent them in an intuitive way. For instance, Figure (a) shows that ADCPs 4 and 5 are well positioned to measure the coastal current during the summer, as their influence extends toward open sea, whereas ADCPs 1 and 3 capture turbulences caused by the coastline. The Gaussian weighting functions, which are mostly parellel to the coast in the summer model of Figure (b), have different orientations in the autumn model (Figure (a)), which determine a less regular partition of the zones of influence of the sensors, with ADCPs 4 and 5 still accounting for the coastal current.
Figure (a) shows an example of the current pattern predicted by the best Summer model based on the measurement of the eight sensors on July 8, 2006, at 0:00 AM.
Finally, Figure (b) shows a simulation of the trajectory that would have been followed by eight particles dropped at the positions of the sensors on July 11, 2006 at midnight. The trajectories have been integrated with an integration step of 10 minutes, by linear interpolation of the currents predicted by the model of Figure (b). This type of diagram gives an idea of the water circulation patterns under different conditions, which is the main objective of the study.

(a) (b)

Figure 4: (a) Current predicted by the model of Figure 4 (gray arrows), based on the reading of the eight captors (red arrows) on July 8, 2006, at 0:00 AM. The length of the arrows is equal to the distance traveled in one hour by a particle moving at the speed of the (predicted or actual) current. **(b)** Simulated trajectory of particles of water based on the currents predicted by the model of Figure 4 starting on July 11, 2006 at 0:00 AM and for 36 hours

6. CONCLUSION

We have shown that models of marine currents that may be used as surrogates of expensive and time-consuming finite element simulations can be obtained for specific environments by means of a powerful evolutionary optimization method. The models obtained are reasonably accurate and have good generalization properties. Furthermore, such models seem to capture relevant aspects of the phenomenon in an intuitive and easy to interpret representation. The results obtained indicate Differential Evolution as the top performer for this problem, ahead of Evolution Strategies (CMA-ES), BFGS, and Monte Carlo. Possible future work would consist of trying alternative parametric shapes for the weight functions, possibly asymmetric.

REFERENCES

[1] Chew C. Z., Gourbesville P., & Liong S.-Y. (2007). *3D current characteristics simulation with ANN*. In J. G. et al., editor, Ocean Science, volume 12 of Advances in Geosciences. World Scientific, Singapore.

[2] DeJong K. A. (2002). *Evolutionary Computation: A unifed approach*. MIT Press, Cambridge, MA.

[3] Eiben A. E. & Smith J. E. (2003). *Introduction to Evolutionary Computing*. Springer-Verlag, Berlin.

[4] Fletcher R. (1993). *An overview of unconstrained optimization*. Technical Report NA/149, Numerical Analysis, June.

[5] Gourbesville P. (2009). *Data and hydroinformatics: new possibilities and challenges*. Journal of Hydroinformatics, 11(3–4):330–343.

[6] Hansen N. (2006). *The CMA evolution strategy: a comparing review*. In J. A. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, Towards a new evolutionary computation. Advances on estimation of distribution algorithms, pages 75–102. Springer-Verlag, Berlin.

[7] Hansen N. (2009). *Benchmarking a bi-population CMA-ES on the BBOB-2009 function testbed*. In Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference, pages 2389–2395. ACM Press.

[8] Liong S.-Y., Lim W.-H., & Paudyal G. N. (2000). *River stage forecasting in Bangladesh: Neural network approach*. Journal of Computing in Civil Engineering, 14(1):1–8, January.

[9] R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

[10] Romano M., Liong S.-Y., Vu M. T., Zemskyy P., Doan C. D., Dao M. H., & Tkalich P. (2009). *Artificial neural network for tsunami forecasting*. Journal of Asian Earth Sciences, 36:29–37.

[11] Storn R, Price K (1997). Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, 11(4), 341–359.

[12] Price KV, Storn RM, Lampinen JA (2006). *Differential Evolution: A Practical Approach to Global Optimization*. Springer-Verlag, Berlin.

[13] Andrea G. B. Tettamanzi, Christel Dartigues, Célia da Costa Pereira, Denis Pallez, Philippe Gourbesville (2011). *Coastal Current Prediction Using CMA Evolution Strategies*. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'11), pages 1715-1722

[14] M. Mullen, D. Ardia, L. Gil, D. Windover, J. Cline. *DEoptim: An R Package for Global Optimization by Differential Evolution*.(2011). Journal of Statistical Software. Volume 40, Issue 6.