Evolutionary Optimization of Wetlands Design

Marco Gaudesi
Politecnico di Torino
Corso Duca degli Abruzzi, 24
10129 Torino, Italy
marco.gaudesi@polito.it

Andrea Marion
University of Padova
Via Marzolo, 9
35131 Padova, Italy
andrea.marion@unipd.it

Tommaso Musner
University of Padova
Via Marzolo, 9
35131 Padova, Italy
tommaso.musner@studenti.unipd.it

Giovanni Squillero
Politecnico di Torino
Corso Duca degli Abruzzi, 24
10129 Torino, Italy
giovanni.squillero@polito.it

Alberto Tonda
UMR 782 GMPA, INRA
1 Avenue Lucien Brétignières
78850 Thiverval-Grignon,
France
alberto.tonda@grignon.inra.fr

ABSTRACT

Wetlands are artificial ponds, designed to filter and purify running water through the contact with plant stems and roots. Wetland layouts are traditionally designed by experts through a laborious and time-consuming procedure: in principle, small patches of vegetation with purifying properties are tentatively placed, then the resulting water flow is verified by fluid dynamics simulators and when a satisfying outcome is reached, the wetland final layout is decided. This paper proposes to automate wetland design exploiting an evolutionary algorithm: a population of candidate solutions is cultivated by the evolutionary core, and their efficiency is evaluated using a state-of-the-art fluid-dynamics simulation framework. Experimental results show that the results obtained by the proposed approach are qualitatively comparable with those provided by experts, despite the complete absence of human intervention during the optimization process.

Keywords

Evolutionary Algorithms, Optimization, Wetlands Design

1. INTRODUCTION

Pollution control in natural bodies of water is one of the most important tasks of our time. High concentrations of different dissolved organic chemicals, such as carbon, nitrogen or phosphorus, can stress ecosystems, decrease overall environmental quality and change the characteristics of the water for human uses. Over the past 50 years, a great effort has been made to collect, control and process polluted water with treatment plants specifically designed for wastewater. While such an approach is effective with *point sources* (sources characterized by high concentrations and relatively small volumes of fluid), its cost may be excessive in the presence of *diffused sources* (sources characterized by low concentrations and big volumes of fluid). To treat wastewater in the latter cases, researchers proposed to exploit the biogeochemical processes present in natural environments, for example adopting *free surface constructed wetlands*. Wetlands are small artificial basins, partially covered by water, used to purify and filtrate polluted water using vegetation. Patches made of plant species that exploit dissolved organic matter present in water to support their vital functions (e.g., *Phragmites Australis, Typha Latifolia*), are distributed over the wetland area in order to obtain a considerable breakdown efficiency.

Designing an effective wetland, however, is a difficult task. While it is possible to determine the effect of a certain configuration of vegetation patches using simulation tools, the underlying dynamics are too complex to derive an inverse function. The only viable approach is the classical *trial and error*, deeply relying on human sensibility and experience. Vegetated areas are tentatively placed by an expert, and the effect evaluated using simulation tools. Then, the expert needs to manually tweak the characteristics and position of each vegetation patch until a satisfactory result is attained.

Over the past decade, evolutionary algorithms (EA) have been successfully employed as optimization tools in many real-world applications [26] [20]. EAs provide an effective methodology for tackling difficult problems, when no preconceived idea about the optimal solution is available. While it is not usually possible to mathematically guarantee that the optimal solution will be found in a finite amount of time, EAs have been demonstrated able to perform much better than traditional optimization techniques in several practical NP-hard problems.

This paper proposes an automatic approach to wetland design. Candidate layouts are generated by an EA that internally uses a state-of-the-art fluid-dynamics simulator for evaluating water purifications and water flow alteration. The process is completely automatized, and it is not based on human experience or sensibility. Nevertheless, experimental results clearly show that the best solution evolved is comparable to a solution devised by an expert starting from the same premises.

The paper is organized as follows: Section 2 recalls the necessary background concepts for the scope of this work.

SAC'13 March 18-22, 2013, Coimbra, Portugal.

The proposed approach is detailed in Section 3. Section 4 presents the experimental results, while Section 5 sums up the conclusions drawn from the whole experience.

2. BACKGROUND

2.1 Wetlands

Cowardin [5] defines a wetland as an ecosystem transitional between aquatic and terrestrial ecosystems, in which the water table is usually at or near the surface or the land is covered by shallow water [4]. Before the extensive land reclamation through the last century, wetlands were common along the coasts, where they functioned as a natural buffer between inner agricultural zones and coastal areas. Today there is a pressing necessity to restore these areas and their role, defining optimal design criteria to obtain, at reasonable costs, the best removal efficiency.

The removal efficiency of natural and constructed free-surface wetlands is controlled by the time spent by contaminants into vegetated zones [18]. The role of vegetation in wetlands is important for two main reasons: water passing through vegetated zones decreases its local velocity, favoring the sedimentation of suspended solids; and biochemical processes determine a transformation of the dissolved substances. In combination with bathymetry, distribution of vegetation can produce preferential pathways of water (hydraulic shortcuts) that can substantially decrease the overall efficiency of a wetland. Removal efficiency is also affected by other hydrodynamic characteristics, as water depth and discharge, both dependent on vegetation distribution and density [1] [14]. Wetlands constructed for waste water treatment are often designed considering an average water residence time [14], even though these methods cannot adequately describe spatial configurations of vegetation in real wetlands [15]. These models, usually called *zero-dimensional*, are often used because they require few data and are easy to manage. Nevertheless, zero-dimensional models produce significant inaccuracies in the prediction of the efficiency of contaminant removal. Other *one-dimensional* models with transient storage were recently used [17] to assess the contaminant removal in a constructed wetland, giving in most cases a good approximation of breakthrough curves.

These models, however, fail to describe different flow paths across the vegetation and through main channels. The evidence of different flow pathways results in a clear bimodality of the solute breakthrough curves, that account for the different characteristic time scales of water residence time. Since spatial heterogeneity of the variables assumes a prominent role in determining the removal efficiency, the use of a more detailed *two-dimensional* approach becomes necessary to obtain reliable predictions.

2.2 Evolutionary Algorithms

Natural evolution is not a random process: while it is based upon random variations, their preservation or dismissal is determined by objective evaluations. Darwinian *natural selection* is the process where only changes that are beneficial to the individuals are likely to spread into subsequent generations, and sometimes it strikingly resembles an optimization process. Unlike most optimization processes, however, it does not require the ability to design intelligent modifications, but only the assessment of the effect of random modifications.

Several researchers, independently, tried to replicate such a characteristic to solve difficult problems more efficiently. Evolutionary computation does not have a single recognizable origin, but most scholars agree on identifying four macro areas: genetic algorithms [13], evolution strategies [21], evolutionary programming [8], and genetic programming [16].

The different paradigms share some key concepts, and can be cumulatively called evolutionary algorithms. An EA starts by generating an initial set of usually random candidate solutions for the given problem. These solutions, called *individuals*, are evaluated using problem-dependent metrics. The result of the evaluation, that is, the *goodness* of the solution, is termed *fitness*. The set of candidate solutions, also known as *population*, is then sorted on its fitness values. Subsequently, offspring is produced by altering the existing solutions: often the best solutions have a higher probability of being selected for reproduction. Offspring might be added to the existing population, or replace it entirely; in any case, some of the worst solutions are deleted before iterating the process, starting from reproduction. When a given stop condition is met, the iterations end and the best solutions are returned to the user.

Being based on a population, EAs are more robust than pure hill climbing. Both small and large modifications are possible, but with different probabilities. Sexual recombination makes it possible to merge useful characteristics from different solutions, exploring efficiently the search space. Furthermore, EAs are quite simple to set up, and require no human intervention when running. They are inherently parallel, and a nearly-linear speed-up may be easily achieved on multiple instruction/multiple data (MIMD) architectures. Finally, it's easy to trade-off between computational resources and quality of the results.

3. PROPOSED APPROACH

The proposed approach exploits an evolutionary algorithm to create candidate solutions to the wetland design problem, each one representing a set of patches of vegetation to be placed inside the area at specific points. Candidate solutions are evaluated by simulating the water flow inside the wetland, keeping track of the quantity of fluid being purified as well as several related metrics.

3.1 Mathematical Models

A wetland is modeled here using a two-dimensional depth averaged model that solves hydrodynamics coupled with a two-dimensional solute transport equation with a first order decay term. Under the assumption of hydrostatic pressure, stationary flow, and negligible wind and Coriolis forces, the depth-averaged velocity field and water depth can be described by the following equations [25]:

$$\frac{\partial(hU)}{\partial x} + \frac{\partial(hV)}{\partial y} = 0 \tag{1}$$

$$\begin{aligned} &\frac{\partial(hU^2)}{\partial x} + \frac{\partial(hUV)}{\partial y} = \\ &= -gh\frac{\partial z_s}{\partial x} + \frac{1}{\rho}\frac{\partial(hT_{xx})}{\partial x} + \frac{1}{\rho}\frac{\partial(hT_{xy})}{\partial y} - \frac{\tau_{bx}}{\rho} \end{aligned} \tag{2}$$

$$\frac{\partial(hUV)}{\partial x} + \frac{\partial(hV^2)}{\partial y} = = -gh\frac{\partial z_s}{\partial y} + \frac{1}{\rho}\frac{\partial(hT_{yx})}{\partial x} + \frac{1}{\rho}\frac{\partial(hT_{yy})}{\partial y} - \frac{\tau_{by}}{\rho} \tag{3}$$

The quantities U and V represent the depth-averaged velocities $[m\,s^{-1}]$ along the x and y direction respectively, h is the water depth $[m]$, z_s is the water surface elevation $[m]$, and ρ the water density $[kg\,m^{-3}]$. The bed shear stresses τ_{bx} and τ_{by} $[N\,m^{-2}]$ in the x and y direction respectively are calculated using the following relationships:

$$\tau_{bx} = \rho c_f m_b U\sqrt{U^2+V^2} \tag{4}$$

$$\tau_{by} = \rho c_f m_b V\sqrt{U^2+V^2} \tag{5}$$

In the case modeled here, the bed slope is set to zero and the investigated velocity range makes it possible to consider the friction coefficient as a constant. This assumption generally holds where the velocity is sufficiently fast to assume turbulent flow. For a flat bathymetry, the bed slope coefficient m_b is unitary and the coefficient of friction c_f can be rewritten using Manning equation as $c_f = gn^2h^{-1/3}$. The effect of different vegetation densities is modeled here using different values of Manning roughness coefficient. This choice is confirmed by many studies that relate vegetation density, stem diameter and flow conditions to an equivalent roughness coefficient [3] [10] [23]. Fluid shear stresses $T_{ij}(i,j = x,y)$ associated to viscous and turbulent effects, are determined using the Boussinesq assumption:

$$T_{xx} = 2\rho(\nu+\nu_t)\frac{\partial U}{\partial x} \tag{6}$$

$$T_{xy} = T_{yx} = \rho(\nu+\nu_t)(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}) \tag{7}$$

$$T_{yy} = 2\rho(\nu+\nu_t)\frac{\partial V}{\partial x} \tag{8}$$

where ν, ν_t, are the kinematic and eddy viscosities $[m^2\,s^{-1}]$. Since the kinematic viscosity has a lower value than the eddy viscosity, it can be neglected in most cases. For a turbulent flow regime, as it was assumed in this preliminary study, ν_t can be expressed using Elder depth-averaged parabolic model [6] as $\nu_t = \alpha U_* h$, where the term α is an empirical coefficient $[-]$ and U_* is the shear velocity $[m\,s^{-1}]$. For longitudinal dispersion Elder proposed a value of the coefficient α of about 5.9 [6], for transverse dispersion, Fischer found that α varies between 0.3-1.0 in irregular waterways with weak meanders [7]. In accordance with [2] [25], a value of α of 6.0 and 0.6 was chosen for the longitudinal and transversal dispersion coefficient respectively.

Solute transport of a reactive tracer through the wetland is simulated with a depth-averaged solute transport model accounting for the effect of advection, turbulent diffusion, dispersion and decay. In the simulations, the tracer is assumed to interact with vegetation and the chemical breakdown due to the permanence in the vegetated zones is modeled with a first order decay relationship. The equation governing the transport of a reactive tracer in the wetland can be modeled as:

$$\frac{\partial(hUC)}{\partial x} + \frac{\partial(hVC)}{\partial y} = = \frac{\partial}{\partial x}(hE_x\frac{\partial C}{\partial x}) + \frac{\partial}{\partial y}(hE_y\frac{\partial C}{\partial y}) - h\lambda C \tag{9}$$

where C is the depth-averaged solute concentration $[kg\,m^{-3}]$, U, V are the vertically integrated velocity components under steady flow conditions $[m\,s^{-1}]$ in the x, y directions respectively. Coefficient E_x, E_x $[m^2\,s^{-1}]$, account for both turbulent diffusion and dispersion. For simplicity, a constant homogeneous value of E_x, E_y has been used $(10^{-5}\,m^2\,s^{-1})$ throughout the entire domain.

3.2 Evolutionary Core

μGP[19], the EA chosen for the experience, is a versatile toolkit developed at Politecnico di Torino in the early 2000s and available under the GNU Public License from Sourceforge[1]. μGP original use was to assist microprocessors' designers in the generation of programs for test and verification, hence, the Greek letter mu in its name. Over the years, the algorithm has been used as an optimizer in a much wider spectrum of problems, including numerical optimization and structure learning.

The algorithm initially creates a set of random candidate solutions to the given problem, that are evaluated and sorted by their fitness value (see Subsection 3.3). Offspring is subsequently created, favoring the fittest individuals and trying to maintain diversity among the population. New candidate solutions are then evaluated and added to the initial population. Solutions are sorted again, and the worst ones are removed until the population returns to its original size. The process is then iterated, starting from offspring generation, until a stop condition is reached.

Two categories of genetic operators are used to generate the offspring: *mutations*, or single-parent operators, and *crossovers*, or recombination operators. Mutation operators create new candidate solutions by altering one single parent solution; crossover operators mix the information contained in two or more parents solutions to create offspring. The most common operators are available inside μGP, but the toolkit also implements *differential evolution*, and other operators specially calibrated for real parameters.

Since individuals are internally represented as multigraphs, μGP relies on a external configuration file to constraints the multigraphs to sensible structure, and to map the internal individuals to valid solutions of the problem. In the specific context, each individual encodes a candidate configurations of the wetland, detailing the features of the patches of vegetation, with a variable number of occurrences that are going to be placed in the water; the order in which the patches are described within the individual is irrelevant. All patches are assumed to be of circular shape: since they can overlap, however, they can create more complex shapes. A vegetation patch is characterized by its position (x, y coordinates expressed in real values) in the wetland, its radius, and the friction value in the center. Patch position is constrained by the size of the wetland; its radius is constrained following the minimum and maximum size of actual vegetation patches used in real wetlands; and the friction value is selected among several values associated to different kinds of vegetation.

Intuitively, vegetation patches tend to be denser in the middle and sparser near their outer bounds. Thus, vegetation in an individual present two discontinuities, at $radius/2$ and $3*radius/4$ respectively, where the friction value in the center is lowered. If two islands overlap, the friction value is increased in the common parts.

[1] http://ugp3.sourceforge.net/

A sample individual, along with its representation, is presented in Figure 1.

3.3 Fitness Function

The definition of an appropriate fitness function is a key aspect in the use of an EA. The process of evolution is based on *differential survival*, that is, different individuals must have a different chance to spread their offspring in future generations. In the artificial environment modeled by an EA, it is essential that different individuals get different fitness values. It is a common practice to include some heuristic knowledge in the fitness, in order to help the EA explore the most promising regions of the search space.

In μGP, the fitness is not a single value but a vector of positive coefficients. Fitness values are compared as in a lexicographical ordering.

In order to evaluate the goodness of a candidate wetland layout, a simulation of the hydrodynamic field is performed extracting computed values of discharge $Q\,[m^3\,s^{-1}]$ and water depth h at the inlet and at the outlet sections of the wetland. During the simulation, a *reactive tracer* with a known concentration is injected at the inlet. Thanks to the presence of vegetation, the tracer is gradually degraded and reaches the outlet section. Mass flux $\hat{M}\,[kg\,s^{-1}]$ passing through these sections is measured, and the difference between the two values represent the first parameter of the fitness function. In order to obtain the optimal vegetation distribution, this difference must be maximized.

On the other hand, a candidate layout must still let the water flow, avoiding configurations where the vegetation is so dense to excessively reduce the flow. The energy requested by the water to flow can be represented by the difference between the water depth at the inlet and outlet section. This difference represents the second parameter of the fitness function. This parameter is minimized by the algorithm: solutions that completely block the water flow are then heavily penalized.

The third and last fitness parameter measures the difference of discharge between the inlet and the outlet sections of the wetland. This value assures that the stationary flow conditions are reached and that the mass fluxes are finely computed. This discharge difference is strongly minimized.

4. EXPERIMENTAL EVALUATION

4.1 Setup

The flow domain is given here by a 200m-long-by-100m-wide rectangular wetland. The length of the wetland allows the solute to spread throughout the cross section and make sure that the whole vegetated area can act on the breakdown process. The elevation of the bed is assumed to be constant, as in a large set of natural wetlands the bed topography does not vary significantly in space and the effect of bed slope can be discarded [24] [25].

Inlet and outlet sections (each 10 m wide) are located symmetrically in the middle of the shorter sides of the wetland domain. A constant discharge of $0.2m^3s^{-1}$ is imposed at the inlet section and a constant water depth of $0.5\,m$ acts as the downstream boundary condition at the outlet section. The remaining boundary is treated as impermeable (no flux condition) and no friction is applied to the lateral walls. Reactive solute with a constant concentration of $1\,kg m^{-3}$ is injected at the inlet section and, once the steady state is reached, the average value of concentration at the outlet section is calculated in order to define the value of the fitness function.

An adaptive triangular mesh is used to ensure numerical stability and resolution in case of steep gradients of the hydrodynamic and solute transport solutions. A value of the Manning roughness coefficient and a particular decay value are assigned to each node of the grid, according to the particular generated individual. The value of decay coefficient λ $[s^{-1}]$ is assigned only to those zones in which vegetation is present, assuming higher values in zones with higher roughness coefficient. Decay coefficients are conveniently scaled compared to natural ones in order to obtain a measurable breakdown (not affected by numerical errors) at the outlet sections. Manning roughness coefficients vary from $0.02\,s m^{-\frac{1}{3}}$ to $0.20\,s m^{-\frac{1}{3}}$ and decay coefficients vary from $10^{-6}\,s^{-1}$ to $10^{-5}\,s^{-1}$. No decay coefficient is assigned to zones without vegetation.

The evolutionary core exploited is μGP version 3.2.0 (revision 198). Fitness of each individual is evaluated solving equations (1-3) (9) by a free, open source code called *TELEMAC2D*, part of the wider set of programs *openTELEMAC* [9] [12]. The code has been specifically modified in order to meet the requirements of the performed simulations.

Each individual evolved by μGP is converted to the *TELEMAC2D* format: a map of the nodes in the basin is created and each node covered by a vegetation patch is assigned the correct laws of friction imposed to the natural water flow and the laws of reduction of pollutants. In order to reduce the computation time required to simulate an entire population, individual processing is distributed on two machines; through this approach and thanks to EAs' parallelism characteristics, it is possible to process different individuals at the same time. *GNU Parallel* [22], a shell tool able to execute jobs in parallel on distributed machines collect the results, is used to distribute the computing effort. In the experiments detailed in the following, the optimization process is run on two distributed machines, configuring the system to concurrently simulate up to 4 individuals on each one. The first machine is equipped with an *Intel Core i5-2500* CPU running at 3.3 GHz, while the second is equipped with an *Intel Core i7-950* CPU running at 3.06 GHz. With this configuration, it was possible to run the simulation of a maximum of 8 individuals at the same time, requiring an average computation time of 80 minutes for each individual.

The experiment, whose results are detailed in the next section, used a population size of 20 individual ($\mu = 20$), and 12 genetic operators applied at each step of evolution ($\lambda = 12$)[2]. The number of patches in each individual was constrained in the range 20–35.

μGP utilizes a pool of genetic operators, and constantly adjusts their activation probabilities in order to enhance the evolution process. In this work, 11 different genetic operators were chosen from the 20 available. Active genetic operators are differentiated in two types: standard crossover operators with one and two cut points; nine different mutation operators that insert, remove or exchange small parts in the genome. Further details can be found in [19].

[2]Differently from the usual terminology, in μGP "λ" represents the number of genetic operators activated in each generation. Since each genetic operator may generate any number of individuals (even zero), the true offspring size cannot be defined.

```
# syntax: Island 'center x' 'center y' 'radius' 'friction law'
Island 48.1731011406175 22.297275427573 15 004
Island 27.43375329418 97.7155652634412 11 002
Island 138.667235389794 94.842476121962 17 004
Island 176.065211977289 66.7838987985231 12 002
Island 187.884214316532 56.1975518147804 12 004
Island 130.431501510587 69.7471832053967 17 004
Island 67.922702973285 27.1557276012409 7 002
Island 130.366914094233 20.9424435114695 19 002
Island 35.5953428389348 88.2080024141715 17 002
Island 46.3400587020331 39.9522313815215 19 004
Island 169.978918784022 27.327012844966 17 004
Island 54.3869499724167 46.5862835152972 12 003
Island 46.9849018483991 28.5297399287444 17 002
Island 176.549598401373 37.772453726328 6 003
Island 106.087630697217 6.25186033557057 16 004
Island 77.4953452001256 17.9570479710488 17 004
Island 72.0428459999546 98.0416110846903 13 002
Island 141.027156536735 80.7492376983664 19 003
Island 152.135305299105 50.2003339859435 9 002
Island 111.556264676085 73.4521127393105 12 002
Island 87.745792491544 97.6509297737394 10 002
Island 182.496731060392 27.1918444226689 17 002
Island 124.670737108581 27.2202285317697 12 003
Island 184.051992140687 79.9257982629538 7 004
Island 93.230943058888 60.5796999096526 19 002
Island 158.947412283074 86.7441557861856 6 002
Island 91.9863073435515 73.6411365464328 18 003
Island 15.4599043637802 49.3633404415345 20 004
Island 105.537528615347 36.8068661382026 16 002
Island 47.1686615742598 44.225776765936 15 003
Island 14.5567385912578 61.7258300640152 11 004
Island 169.604583162246 59.4294569689676 15 002
Island 113.263217256384 25.0068967753379 5 003
Island 70.2897186377996 2.54027768073293 14 003
```

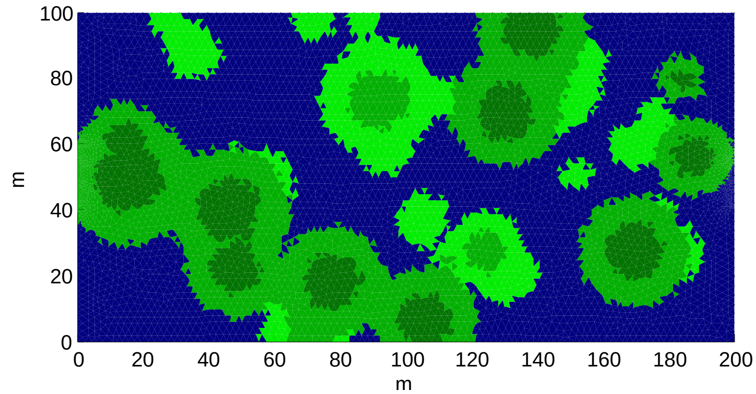

Figure 1: Sample individual. On the left, the genotype, represented by a list of circular patches with respective features. On the right, the phenotype, represented as a visualization of the vegetation distribution over the wetland surface. Darker shades of green indicate a higher coefficient of friction.

4.2 Results and Discussion

Results of the optimization are described in Figure 2, and were reached after approximately 100 generations and 1100 individuals analyzed. Three individuals at three different evolution stages are shown as a reference for the whole optimization process. It is interesting to note that each individual presents the same number of vegetated patches. Individual A, at the initial stage of the evolution, shows a poor vegetative covering and vegetation patches are characterized by both dense and sparse vegetation (brighter color for some patches compared to the others). Level of mass degradation is around 20%.

During the computation, the evolution promotes individuals with vegetation patches to spread over the wetland surface and with a thicker vegetation. This is clear in the case of the individual B, in which the percentage of superimposed vegetation patches decreases. Vegetation patches tend to reach the maximum allowed *radius* of 20 m: diameter of vegetation patches becomes indeed more homogeneous compared to individual A so that the remaining small vegetated areas does not impact on the degradation process. Mass degradation increases and reaches a value close to 32%.

As the evolution proceeds, the vegetative cover tends to increase (individual C) even though mass degradation values show an asymptotic behavior in generations 45 to 70. That means that, at this stage of evolution, processed individuals are very similar to each other and the population can be regarded as mature. Under the assumption of the model, mass degradation percentage of the best individual reaches a satisfying 43%, approaching values that are common in real constructed wetlands [11].

5. CONCLUSIONS

Wetlands are artificial ponds, extensively used to filtrate and purify water. Achieving an optimal design for this purpose is an extremely complex task, usually carried on by experts on the basis of fluid-dynamics simulations. In this paper, an evolutionary algorithm is applied to the wetlands design problem. Each candidate solution is evaluated by a state-of-the-art fluid-dynamics simulator, on the basis of several relevant metrics. Experimental results on the best solution provided by the algorithm show a performance comparable with human-devised designs, despite the absence of human intervention during the optimization process.

Future works will include a more complex individual representation, with patches of several different shapes and a more refined management of friction values. Managing larger populations, or different sub-population, might also prove beneficial to the quality of the final solutions: nevertheless, the computational-intensive simulations needed to evaluate a single candidate represent a severe bottleneck. For this reason, further developments will probably exploit the parallelism innate in evolutionary algorithms, using clusters or grids to speed up the process. Finally, the choice of decay coefficients has a predominant role in determination of the final breakdown efficiency: a more detailed analysis on a real case should be used to demonstrate the potential of the proposed approach, that shows promising results in this first experience.

6. REFERENCES

[1] C.S. Akratos and V.A. Tsihrintzis. Effect of temperature, HRT, vegetation and porous media on removal efficiency of pilot-scale horizontal subsurface flow constructed wetlands. *Ecological Engineering*, 29(2):173–191, 2007.

[2] Feleke Arega and Brett F Sanders. Dispersion model for tidal wetlands. *Journal of Hydraulic Engineering*, 130(8):739–754, August 2004.

[3] DCM Augustijn, F. Huthoff, and E.H. Velzen. Comparison of vegetation roughness descriptions. 2006.

[4] G. Bendoricchio and S. E. Jorgensen, editors. *Fundamentals of Ecological Modelling, Third Edition*. Elsevier Science, 3 edition, August 2001.

[5] Lewis M. Cowardin. *Classification of Wetlands and Deepwater Habitats of the United States*. DIANE Publishing, 1979.

[6] J.W. Elder. The dispersion of marked fluid in turbulent shear flow. *J. Fluid Mech*, 5(4):544–560,

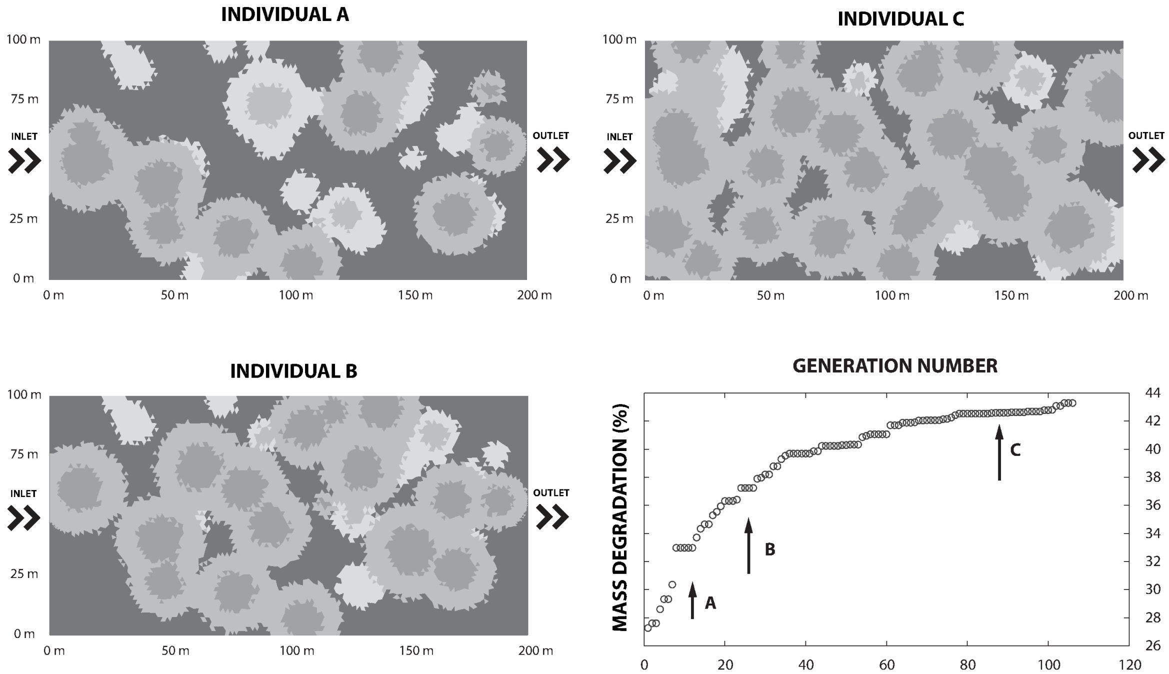

Figure 2: Progressive optimization of candidate solutions. On the left, the best layout in the population, for several generations. On the right, a graph showing the increase in the best fitness value as the EA proceeds.

1959.

[7] H.B. Fischer. *Mixing in inland and coastal waters*. Academic Pr, 1979.

[8] L. J. Fogel. Autonomous automata. *Industrial Research*, 4:14–19, 1962.

[9] J. C. Galland, N. Goutal, and J. M. Hervouet. TELEMAC: a new numerical model for solving shallow water equations. *Advances in Water Resources AWREDI*,, 14(3), 1991.

[10] J.E.P. Green and J.E. Garton. Vegetation lined channel design procedures. *Transactions of the American Society of Agricultural Engineers*, 26(2):437–439, 1983.

[11] R. Haberl, R. Perfler, and H. Mayer. Constructed wetlands in europe. *Water Science and Technology*, 32(3):305âĂŞ316, 1995.

[12] J. M. Hervouet, J. L. Hubert, J. M. Janin, F. Lepeintre, and E. Peltier. The computation of free surface flows with TELEMAC: an example of evolution towards hydroinformatics. *Journal of Hydraulic Research*, 32(S1):45âĂŞ64, 1994.

[13] John H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992.

[14] R.H. Kadlec and S. Wallace. *Treatment wetlands*. CRC, 2009.

[15] Robert H. Kadlec. The inadequacy of first-order treatment wetland models. *Ecological Engineering*, 15(1-2):105–119, June 2000.

[16] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[17] Christopher J Martinez and William R Wise. Analysis of constructed treatment wetland hydraulics with the transient storage model OTIS. *Ecological Engineering*, 20(3):211–222, July 2003.

[18] J. Persson, N.L.G. Somes, and T.H.F. Wong. Hydraulics efficiency of constructed wetlands and ponds. *Water Science & Technology*, 40(3):291–300, 1999.

[19] E. Sanchez, M. Schillaci, and G. Squillero. *Evolutionary Optimization: the µGP toolkit*. Springer, 1st edition, April 2011.

[20] E. Sanchez, G. Squillero, and A. Tonda. *Industrial Applications of Evolutionary Algorithms*, volume 34 of *Intelligent Systems Reference Library*. Springer, 2012.

[21] Hans-Paul Schwefel. *Cybernetic Evolution as Strategy for Experimental Research in Fluid Mechanics (Diploma Thesis in German)*. Hermann Föttinger-Institute for Fluid Mechanics, Technical University of Berlin, 1965.

[22] O. Tange. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, February 2011:42–47, 2011.

[23] B.L. White and H.M. Nepf. Scalar transport in random cylinder arrays at moderate reynolds number. *Journal of Fluid Mechanics*, 487(25):43–79, 2003.

[24] A. Worman and V. Kronnas. Effect of pond shape and vegetation heterogeneity on flow and treatment performance of constructed wetlands. *Journal of Hydrology*, 301(1-4):123–138, 2005.

[25] W. Wu. *Computational river dynamics*. CRC, 2007.

[26] Tina Yu, Lawrence Davis, Cem M. Baydar, and Rajkumar Roy, editors. *Evolutionary Computation in Practice*, volume 88 of *Studies in Computational Intelligence*. Springer, 2008.