Evolutionary Algorithms for Roughness Coefficient Estimation in River Flow Analyses

Antonio Agresta[1][0000−0003−4037−7403], Marco Baioletti[2][0000−0001−5630−7173], Chiara Biscarini[1][0000−0003−2279−7244], Alfredo Milani[2][0000−0003−4534−1805], and Valentino Santucci[1][0000−0003−1483−7998]

[1] University for Foreigners of Perugia, Perugia, Italy
{antonio.agresta,chiara.biscarini,valentino.santucci}@unistrapg.it

[2] Department of Mathematics and Computer Science
University of Perugia, Perugia, Italy
{marco.baioletti,alfredo.milani}@unipg.it

Abstract. Management and analyses of water resources is of paramount importance in the implementation of water related sustainable development goals. Hydraulic models are key in flood forecasting and simulation applied to a river flood analysis and risk prediction and an accurate estimation of the roughness is one of the main factors in predicting the discharge in a stream. In practical implementation roughness can be represented by the prediction of the well known Manning's coefficient necessary for discharge calculation. In this paper we design an objective function that measures the quality of a given configuration of the Manning's coefficient. Such an objective function is optimised through several evolutionary approaches, namely: (1+1)-ES, CMA-ES, Differential Evolution, Particle Swarm Optimization and Bayesian Optimization. As case of study, a river in the central Italy was considered. The results indicate that the model, consistent with the classical techniques adopted in the hydraulic engineering field, is applicable to natural rivers and is able to provide an estimation of the roughness coefficients with a satisfactory accuracy. A comparison of the performances of the five evolutionary algorithms is also proposed.

Keywords: Evolutionary algorithms · River flow analysis · Estimating Manning's coefficient.

1 Introduction

Hydraulic sciences plays a relevant role in supporting the implementation of the sustainable development goals launched by United Nation Organisation, since they are crucial in assessing the impacts of climate changes and the sustainable management of water sources and environment.

In fact, hydraulic models are fundamental to properly predict floods and more in general to set up water management [17, 30], to organize river regulation [9, 21, 27], sediment transport [10], to help developing flood protection systems [16], to generate simulations in order to construct flooding maps and many others purposes [29, 11, 24]. Hydraulic computations of flow involve roughness coefficients, which represent the resistance to flood flows in channels and flood plains. The Mannings's equation (cite Manning in 1889) is an empirical equation that applies to uniform flow in open channels and is a function of the roughness coefficient (n) selected from standard reference, tables [20] or calculated from field measurements, the estimation of a Manning's coefficient can affect computational results [8].

With the aid of computational and numerical techniques, a precise assessment of the Manning's coefficient in open channels represents an important achievement in terms of modelling and study of possible solutions to prevent and fix flood generated disasters.

The parameters typically used in hydraulic engineering models are divided into physical parameters and empirical parameters. Physical parameters describe the physical properties of materials and they are usually constants, while empirical parameters, due to the complexity and the variability of specific elements characterizing the hydraulic engineering (e.g. the roughness of channel surface, the bed material, vegetation, channel alignment and irregularities, channel shape and size, stage and discharge, suspended sediment load and bed sediment loads) need to be calculated through mathematical models [2]. Among these, the Manning's roughness coefficient, usually denoted by n, is often uncertain and this led researchers to find empirical formulas to estimate as correctly as possible the value of n [4].

Determining the n value is a critical and complex task in the hydraulics of open channel flows: this value changes in time and space, and depends on the multiple factors such as the geometric, geomorphological, and hydraulic parameters of water current and river beds [5].

With these premises, in the present work we propose an evolutionary approach to the estimation of the Manning's roughness coefficients for different cross-sections of a river. We design a black-box optimization problem by introducing an objective function that, by encapsulating all the hydraulic methodologies, takes in input a vector of Manning's coefficients and returns a loss score.

Internally, the objective function runs an hydraulic simulation procedure by considering the inputted Manning's coefficients and returns the expected depth of the river in a particular station, then this depth is compared with the true depth observed at that station and their absolute difference is returned as the loss score of the Manning's coefficients in input.

Interestingly, this black-box formulation allows to estimate the roughness coefficients by adopting any evolutionary algorithm or, more in general, any meta-heuristic proposed in the literature for numerical optimization problems.

As case of study we selected data from a river reach in the central Italian region of Umbria, namely the river "Paglia", during the flood event of the year 2012. Then, experiments were held by considering five popular Evolutionary Algorithms (EAs) from the literature, namely: the evolution strategy (1+1)-ES [6], the Differential Evolution (DE) scheme [28], the Particle Swarm Optimization

(PSO) algorithm [18], the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [14], and the Bayesian Optimization (BO) algorithm [12].

The rest of the paper is organized as follows. Section 2 describes the hydrologic context, while Section 3 introduces the objective function. Section 4 briefly recalls the five EAs considered. Experiments are described in discussed in Section 5. Finally, conclusions are drawn in Section 6 where future lines of research are also depicted.

2 The Hydrologic Problem

The hydrodynamic analysis of river flows was performed using the free software HEC-RAS, created by The U.S. Army Corps of Engineers [1].

The hydrographic framework chosen as a test case is the river Paglia with a basin of $1187km^2$ that includes three Italian regions, Toscana, Umbria and Lazio, and an average flow rate of $11.3m^3/s$.

2.1 The mathematical formulation and programming settings

HEC-RAS is used to calculate water surface profiles and energy grade lines in 1D / 2D, steady and unsteady state, and gradually varied flow analysis.

In the present study, HEC-RAS has been employed to perform one dimensional, steady, hydraulic calculation for a river flow.

The model employed to perform calculations is based on the steady 1D case which is described by the simple energy equation (1):

$$z_1 + h_1 + \alpha_1 \frac{v_1^2}{2g} = z_2 + h_2 + \alpha_2 \frac{v_2^2}{2g} + h_e, \tag{1}$$

where: z is the bottom elevation, h is the depth, v is the mean velocity in the channel cross-section, α is called the St. Venant coefficient and plays the role of the correction factor, including the effect of velocity profile non-uniformity, while g is the well known acceleration of gravity.

It is worthwhile to note that Equation (1) is provided for gradually-varying flow, when the assumption of hydrostatic pressure distribution may be suitable.

As for the specific program implementation, HEC-RAS, version 5.0.7 (the latest at time of writing) allows to set the geometric data of the river directly from geo-referenced files. Once the geometry is set, upstream and downstream boundary conditions are to be properly implemented.

Cross section coordinates were defined by entering the river station and elevation points from left to right bank in sequence along the river.

Twenty cross sections were allocated over the river reach as represented in Figure 1.

After the geometry is defined, data of discharge were defined for the calculation process finalizing the model creation.

The actual 1D, steady model in HEC-RAS uses empirical Manning's equation, in the form of equation (2), to supply the relationship between the river

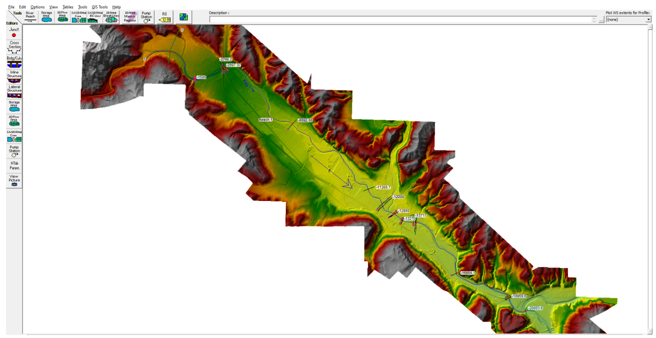

Fig. 1. HEC-RAS: Paglia's DEM file with the cross sections selected

discharge, hydraulic resistance, river geometry, and the friction energy loss. In the case of a change in channel geometry, energy losses are assessed using coefficients of contraction or expansion multiplied by the change in velocity head. Head loss between two sections is computed from equation (3), while the water surface is calculated from the energy equation (4) [1].

$$Q = K S_f^{1/2}, \tag{2}$$

$$h_e = L S_f + C \left(\frac{\alpha_1 v_1^2}{2g} + \frac{\alpha_2 v_2^2}{2g} \right), \tag{3}$$

$$h = Z + y + \frac{\alpha v^2}{2g}. \tag{4}$$

In equations (2) Q is the flow rate or discharge and K is the conveyance of the channel. Moreover, in equation (3) S_f is the energy slope, g is the acceleration due to gravity, h_e is the energy head loss, C is the expansion or contraction coefficient, α_1 and α_2 are the velocity weighting coefficients, while v_1 and v_2 the average velocities. Finally, in equation (4) h is the water surface level above a specified datum, Z is the bed elevation, y is the depth of flow, α is the kinetic energy correlation coefficient, and v is the average velocity. The subscripts 1 and 2 denote two different cross-sections in the same channel reach. The basic assumption is that the cross-section number 1 is located upstream of the cross-section number 2. L is defined as the distance weighted reach length and it is calculated as follows:

$$L = \frac{L_{lob}\bar{Q}_{lob} + L_{ch}\bar{Q}_{ch} + L_{rob}\bar{Q}_{rob}}{\bar{Q}_{lob} + \bar{Q}_{ch} + \bar{Q}_{rob}}, \tag{5}$$

where the subscripts lob, ch, rob stand respectively for left overbank, main channel, and right overbank. The quantity $(\bar{Q}_{lob} + \bar{Q}_{ch} + \bar{Q}_{rob})$ represents the arithmetic average of the flows between sections for the left overbank, the main channel and the right overbank.

Variable h_e, as before mentioned, describes friction losses due to bed and banks influence on flowing water and plays a fundamental role, as it includes effects of channel contraction and extension. If the depth is known in one cross-section, on this basis, the depth may be also determined in the second cross-section. To calculate the distribution of the depth along the channel, its value must be known in one cross-section immediately upstream. So it becomes clear that this cross-section is the inlet or outlet boundary of the channel reach. Hence, the condition is frequently called "boundary condition", but it is rather a hydraulic term than strict mathematical language.

The kinematic energy terms, as well as friction losses, depend on the magnitude of the flow, exactly the discharge Q. The influence of floodplains is included in the calculation of the St. Venant coefficients and calculation of weighted distance between cross-sections.

In order to determine the total conveyance and the velocity coefficient for a cross section, HEC-RAS approach requires the flow to be subdivided into sections whose velocities can be considered uniformly distributed.

The approach followed is to subdivide the flow in the overbank areas using the input cross section value break points (locations where n-values change) as the basis for subdivision. Conveyance is calculated within each subdivision from the Manning's equation 2.

The empirical correlation between the Manning's equation and the Manning's coefficient is given by the conveyance coefficient for subdivision, K which can be expressed by:

$$K = \frac{1.486}{n} A R^{2/3}, \tag{6}$$

where n is the Manning's roughness coefficient, A is the flow area and R is the hydraulic radius (area / wetted perimeter), all to be intended for each subdivision.

The program sums up all the incremental conveyances in the overbanks to obtain a conveyance for the left overbank and the right overbank. The main channel conveyance is normally computed as a single conveyance element. The total conveyance for the cross section is obtained by summing the three subdivision conveyances (left, channel, and right).

One can opt for a single Manning's n for each section (*lob*, *ch* or *rob*) or for multiple selection for a single section. In this work, as anticipated, 20 cross sections have been identified and a single n has been set for each segment, hence $3n$ values for each cross section, for a total of 60 Manning's roughness coefficients.

In order to evaluate the Manning's n values to be inserted in HEC-RAS, the standard tabulated values from the classic work of Chow [20], as in Figure 2.

2.2 The selected test case

Without any lack of generality for the scope of our study, as already introduced, a one-dimensional steady flow hydraulics computation was performed and a spe-

Type of Channel and Description	Minimum	Normal	Maximum
Natural streams - minor streams (top width at floodstage < 100 ft)			
1. Main Channels			
a. clean, straight, full stage, no rifts or deep pools	0.025	0.030	0.033
b. same as above, but more stones and weeds	0.030	0.035	0.040
c. clean, winding, some pools and shoals	0.033	0.040	0.045
d. same as above, but some weeds and stones	0.035	0.045	0.050
e. same as above, lower stages, more ineffective slopes and sections	0.040	0.048	0.055
f. same as "d" with more stones	0.045	0.050	0.060
g. sluggish reaches, weedy, deep pools	0.050	0.070	0.080
h. very weedy reaches, deep pools, or floodways with heavy stand of timber and underbrush	0.075	0.100	0.150
2. Mountain streams, no vegetation in channel, banks usually steep, trees and brush along banks submerged at high stages			
a. bottom: gravels, cobbles, and few boulders	0.030	0.040	0.050
b. bottom: cobbles with large boulders	0.040	0.050	0.070
3. Floodplains			
a. Pasture, no brush			
1.short grass	0.025	0.030	0.035
2. high grass	0.030	0.035	0.050
b. Cultivated areas			
1. no crop	0.020	0.030	0.040
2. mature row crops	0.025	0.035	0.045
3. mature field crops	0.030	0.040	0.050
c. Brush			
1. scattered brush, heavy weeds	0.035	0.050	0.070
2. light brush and trees, in winter	0.035	0.050	0.060
3. light brush and trees, in summer	0.040	0.060	0.080
4. medium to dense brush, in winter	0.045	0.070	0.110
5. medium to dense brush, in summer	0.070	0.100	0.160
d. Trees			
1. dense willows, summer, straight	0.110	0.150	0.200
2. cleared land with tree stumps, no sprouts	0.030	0.040	0.050
3. same as above, but with heavy growth of sprouts	0.050	0.060	0.080
4. heavy stand of timber, a few down trees, little undergrowth, flood stage below branches	0.080	0.100	0.120
5. same as 4. with flood stage reaching branches	0.100	0.120	0.160

Fig. 2. Manning's n for Channels

cific reach, taken as testing geographical framework, was selected. Data were collected from the hydrographic service of the Umbria Region[3].

The river Paglia originates from the Amiata mountain, an extinct volcano at 1738 meters above sea level and, proceeding from upstream to downstream, the river enters in the Orvieto plain.

The characteristics of the materials present along the riverbed are mainly graveled and rocky throughout its course, with short grassy banks depending on the seasons. The entire basin is generally characterized by materials with low permeability, about 75%, and its slope is averaged at 4.8‰ up to its confluence with the canal Subbisone, its first stretch, then a steeper one of 7‰ between this confluence and the Mount Rubiaglio, and finally, the stretch of our interest, sloping at about 3.3‰.

In order to perform a validation for the optimization procedure, data have been collected from a specific event that happened in the selected area. During the night between November 11st and 12th 2012, a massive rain fall has been registered by pluviometers, with a peak discharge of 307 mm registered at the gauging station positioned at the location of Orvieto Scalo.

The gauging spot of Orvieto Scalo corresponds with the cross section labeled with "12850", as in figure 4.

The event under analysis produced a considerable increase in the hydrometric levels, generating overflows that caused damages in many towns and villages in the surrounding area. A peak discharge was registered of $Q = 2200\ \mathrm{m}^3/\mathrm{s}$, a value that was employed, in our work, as the upstream boundary condition.

3 The Objective Function

The objective function is designed as a loss function where the variables to optimize are the Manning's coefficients at the different cross-sections of the considered river reach. As described in Section 2.2, we have 20 cross-sections and, for each cross-section, three Manning's coefficients are considered (left bank,

[3] Available at `https://www.regione.umbria.it/ambiente/servizio-idrografico`.

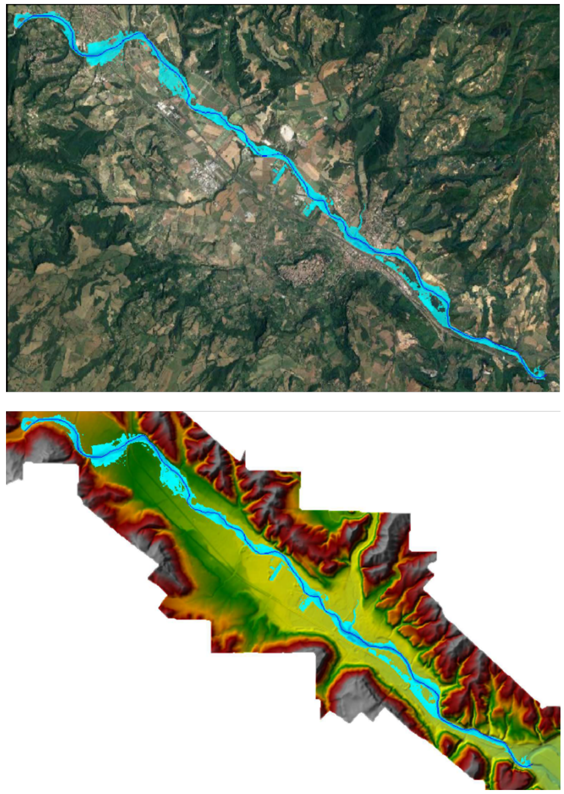

Fig. 3. Satellite and DEM representation of the simulated overflow event

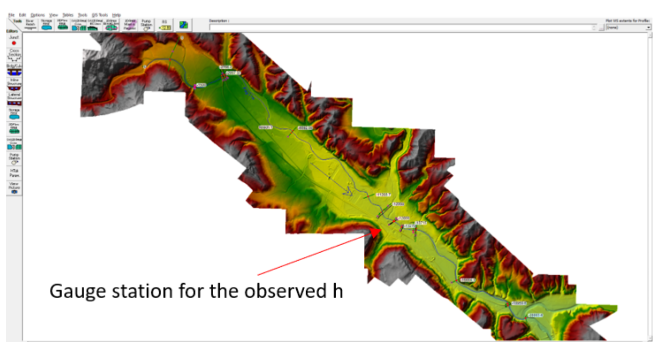

Fig. 4. HEC-RAS: Particular of the cross section corresponding to the gauge station

central channel, right bank). Hence, a solution to our optimization problem is real vector $x \in \mathbb{R}^{60}$. Furthermore, every vector's component is bounded to the suitable range of $[0.025, 0.2]$.

The objective function proceeds as follows: (i) uses the vector x of Manning's coefficients in order to setup an HEC-RAS simulation, (ii) runs the HEC-RAS simulation $\mathrm{HR}(x)$ which returns the expected depth of the water at the gauge station deployed along the river, and (iii) computes and returns the absolute difference between the simulated depth $\mathrm{HR}(x)$ and the true depth h_{obs} observed at the gauge station.

Formally, the goal is to find the vector of Manning's coefficients $x \in \mathbb{R}^{60}$ that minimizes the objective function

$$f(x) = |\mathrm{HR}(x) - h_{\mathrm{obs}}|, \tag{7}$$

with the constraints that $x_i \in [0.025, 0.2]$ for $1 \leq i \leq 60$.

The ideally optimal vector of Manning's coefficients is the one with a loss score of 0, thus 0 is a lower bound for $f(x)$.

4 Evolutionary Algorithms

In this section we describe the meta-heuristics employed for the optimization problem described in Section 3.

4.1 Differential Evolution

Differential Evolution (DE) is a population based evolutionary metaheuristic designed for continuous optimization, originally proposed in [28].

The DE population is composed by N d-dimensional vectors $\{x_1, \ldots, x_N\}$ and can be initialized using different strategies, the most common way is to sample the search space. In this work we chose to use the Scrambled Hammersley procedure which produces a low discrepancy sample of vectors.

The key operator of DE is the differential mutation which produces a mutant vector v_i for each population element x_i. The mutant vector is computed as a linear combination of some population elements. One the most popular strategy for the differential mutation operator is "current-to-best" scheme which is defined as

$$v_i = x_i + F_1 \cdot (x_{best} - x_i) + F_2 \cdot (x_{r_1} - x_{r_2}),$$

where x_{best} is the best population individual, F_1 and F_2 are two scale factors, and x_{r_1},x_{r_2} are two random population individuals which are different from each other and from x_i.

A target vector y_i is produced with the crossover operator applied to each population element x_i and the corresponding mutant vector v_i. The most used crossover operator is the uniform crossover: the j-th component of y_i, for $j = 1, \ldots, d$, is computed as

$$y_{i,j} = \begin{cases} v_{i,j} \text{ if } r_j < CR \text{ or } j = \bar{j} \\ x_{i,j} \text{ otherwise,} \end{cases}$$

where CR is the crossover ratio, r_j is a random number in $[0, 1]$ and $\bar{j}$ is a random index in $\{1, \ldots, d\}$.

At the end of each iteration, the fitter between y_i and x_i is kept in the population for the next iteration.

For further details see [7, 3, 25]. In our experimentation, the parameters have been set as follows: $N = 30$, $F_1 = F_2 = 0.8$, while CR is randomly chosen in $[0, 1]$ before every crossover application.

4.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is one of the most famous metaheuristic based on swarm intelligence principle, firstly proposed in [18]. PSO maintains a population of N elements, each of them has a position x_i in the search space and a velocity v_i, which is also a d-dimensional vector.

Population individuals are connected by undirected links, which are used to propagate information among the population.

As in DE, the population is evolved during a given number of iterations. At each iteration, for $j = 1, \ldots, d$ and for $i = 1, \ldots, N$, the component j of the velocity of the element i is updated by means of the following formula

$$v_{i,j} \leftarrow v_{i,j} + c_1 r_{1,j}(lb_{i,j} - x_{i,j}) + c_2 r_{2,j}(pb_{i,j} - x_{i,j}),$$

where c_1 and c_2 are the *social* and the *cognitive coefficients*, respectively, and $r_{1,j}$ and $r_{2,j}$ are random number in $[0, 1]$.

The vector lb_i is the best element among those connected to x_i (the *local best*), while pb_i is the best position ever reached by x_i in all the previous iterations (the *personal best*).

After having computed the new velocity vectors, each position is simply updated as $x_i \leftarrow x_i + v_i$.

The most common choice for the connection schema among the population elements is the complete graph: in this setting, lb_i is the best population element (called *global best*).

Other details about PSO can be found in [7, 26]. In our experimentation: the population size N was set to 30, while all the other parameters were set as their default values in Nevergrad library [22].

4.3 (1+1)-ES

Evolution strategies [6] are evolutionary algorithms which evolve a population of μ individuals by producing a set of λ children by means of genetic operators recombination, mutation and selection.

The simplest form of evolutionary strategy is (1+1)-ES in which the population contains one element x and, at each generation, one child y is produced by means of a mutation operator based on a Gaussian distribution with mean 0 and a given variance σ. Formally, $y_j = x_j + N(0, \sigma_j^2)$, for $j = 1, \ldots, d$. If y is fitter than x, then x will be replaced by y in the next iteration.

In our experiments, the algorithm starts with the vector x^0 located at the center of the search space, i.e. $x_j^0 = L_j + \frac{U_j - L_j}{2}$, for each $j = 1, \ldots, d$, where L_j and U_j are, respectively, the lower and the upper bounds for dimension j. Moreover, we set $\sigma_j = \frac{U_j - L_j}{5}$.

4.4 CMA-ES

One of the most successful form of evolution strategies is the Covariance Matrix Adaptation evolution strategy (CMA-ES) [14].

CMA-ES is an iterative process which updates a mean vector m, the step size σ, and a covariance matrix C. At each iteration g, λ samples $x_1^{(g)}, \ldots, x_\lambda^{(g)}$ are generated with the multivariate normal distribution $N(m, \sigma^2 C)$.

m is then updated as the weighted average of μ fittest samples $x_{1:\lambda}^{(g)}, \ldots, x_{\mu:\lambda}^{(g)}$

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{(g)}$$

One of the most used method to update C is the Rank-μ-update, where the new value of C is computed as

$$C^{(g+1)} = {C^{(g)}}^{1/2} [I + c_\mu \sum_{i=1}^{\lambda} w_i (z_{i:\lambda}^{(g)} {z_{i:\lambda}^{(g)}}')] {C^{(g)}}^{1/2},$$

where c_μ is a smoothing coefficient, $z_{i:\lambda}^{(g)} = {C^{(g)}}^{-1/2} y_{i:\lambda}^{(g)}$ and $y_{i:\lambda}^{(g)} = \frac{x_{i:\lambda}^{(g)} - m^{(g)}}{\sigma^{(g)}}$, for $i = 1, \ldots, \lambda$.

Another method is Rank-one-update, where an evolution path p_C is updated at each iteration with the formula

$$p_C^{(g+1)} = (1 - c_C) p_C^{(g)} + \sqrt{c_C(2 - c_C)} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}}$$

and then C is updated as

$$C^{(g+1)} = (1 - c_1) C^{(g)} + c_1 p_C^{(g+1)} {p_C^{(g+1)}}'$$

Finally, σ is updated using a process called cumulative step-size adaptation (CSA). To accomplish this task, another evolution path p_σ is used. p_σ is updated with the following formula

$$p_\sigma^{(g+1)} = (1 - c_\sigma) p_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)} {C^{(g)}}^{-1/2} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}}.$$

Then σ is updated as

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{||p_\sigma^{(g+1)}||}{E||N(0, I)||} - 1 \right) \right).$$

In our experimentation, λ was set to 30, while all the other parameters were set to their default value in the Nevergrad library [22].

4.5 Bayesian Optimization

Bayesian Optimization (BO) [12] is used on optimization problems where the computation of the fitness function is time demanding, as the case described in this paper. This kind of approach is called "surrogate methods".

In BO, a probabilistic model, usually a Gaussian Process, is used to forecast the fitness function of any point of the search space and then to sample the most promising element.

More precisely, a Gaussian process (GP) is defined with mean function $\mu(x)$ and a covariance function (or kernel) $\Sigma(x, x')$ and it is built up by choosing an initial set of points $x_1, \ldots, x_n$ and their corresponding fitness values $f(x_1), \ldots, f(x_n)$.

The GP can be used to compute the distribution of $f(x)$ given the values $f(x_1), \ldots, f(x_n)$, for all $x \in \mathbb{R}^d$.

An acquisition function is used to find the point x_{n+1} whose fitness value can be maximal. One of the most frequently employed acquisition function is based on the expected improvement, which correspond to select the point $x_{n+1} \in \mathbb{R}^d$ such that $E([f(x_{n+1}) - f^*]^+)$ is maximal, where $f^* = \max\{f(x_1), \ldots, f(x_n)\}$

Then, the fitness value $f(x_{n+1})$ is evaluated and x_{n+1} and $f(x_{n+1})$ are used to update the parameters of the GP.

BO proceeds in this way until a given number of fitness evaluation is reached. In our experimentation, the number of initial samples considered was set to 30, while all the other parameters were set to their default values in the Nevergrad library [22].

5 Experiments

In order to analyze the proposed methodology, we have held an experimental comparison among the five Evolutionary Algorithms (EAs) described in Section 4 by considering the objective function introduced in Section 3. Moreover, the effectiveness of any single algorithm is also compared with that of the standard approach described in Section 2.1.

The implementations of the selected algorithms available in the recently proposed Python's library Nevergrad [22, 23] (version 0.4.2, the latest one at time of writing) were adopted. All the experiments have been carried out on a machine equipped with an Intel Xeon E5-2650v4 clocking at 2.20 GHz, 128 GB of RAM and running Windows 10.

It is worthwhile to note that: (i) the Windows operating system was necessary in order to run the HEC-RAS simulation required by the objective function evaluation, and (ii) the communication between the Python's code and the HEC-RAS simulation was realized through the well known COM interface for inter-process communications [13].

Each one of the five EAs was executed 25 times on the case study described in Section 2 and every execution terminates when the computational budget of 600 seconds is exhausted. In terms of number objective evaluations, considering that a single evaluation lasts around two seconds on the available machine, every algorithm performed around 300 objective evaluations.

Moreover, in order to analyze the experimental results from different point-of-views, for every single execution we registered both the final objective value achieved and the full optimization trajectory.

In Table 1, we provide all the statistics about the best solutions obtained by the five EAs in all their executions. The algorithms are ordered by average objective value. Moreover, in the last line of the table we provide the objective value achieved by the standard methodology (a clearly deterministic method, so only one value is provided) described in Section 2.1. Best results are in bold.

Algorithm	**Final Objective Values**			
	Average	**Minimum**	**Maximum**	**St.Dev.**
(1+1)-ES	$\mathbf{< 10^{-5}}$	**0**	**0.00005**	$\mathbf{< 10^{-5}}$
DE	0.00147	0.00004	0.00665	0.00211
CMA-ES	0.00167	0.00013	0.00355	0.00139
BO	0.01364	0.00538	0.02898	0.00807
PSO	0.03719	0.00199	0.08304	0.03013
Std Method	1.63000			

Table 1. Statistics about the final objective values observed in the experiments.

From Table 1 it is possible to see that all the five EAs largely outperform the standard approach in any single execution. Since the objective value is the estimation error in meters, the accuracy gain obtained by the proposed automatic methodology is of more than 1.5 meters with respect to the standard approach, while BO, the worst average error of all tested evolutionary algorithms, obtains an error of less than 3.8 centimeters.

Regarding the comparison among the five EAs, it is interesting to note that (1+1)-ES clearly outperforms all the other competitors. This is possibly motivated by the small budget of evaluations allowed (due to the time complexity of the objective function) and the fact that (1+1)-ES evolves a single solution and not a population as the other methods. Moreover, the worst objective value obtained by (1+1)-ES differs from the theoretically optimal value of 0 by only 10^{-5}.

Moreover, we statistically validated the comparisons among the EAs by running a Kolmogorov-Smirnov statistical test [15] for each pair of EAs. By considering a significance threshold of 0.01, in Figure 5 we provide an heatmap to be interpreted as follows: an entry is red if the algorithm indicated in the column is significantly better than that indicated in the row, an entry is blue if the row's algorithm is significantly better than that column's algorithm, while the gray color indicates no significant difference.

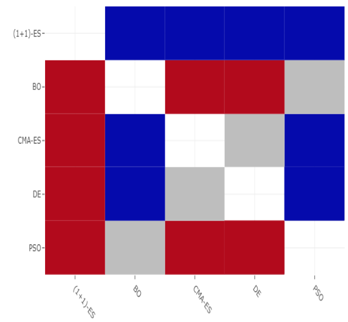

Fig. 5. Results of the Kolmogorov-Smirnov tests.

Observing Figure 5, it is clear that (1+1)-ES significantly outperforms all the other competitors. Moreover, DE and CMA-ES do not show significant differences in terms of effectiveness and both outperform the PSO and BO.

The empirical probability density functions of the final objective values obtained by the executions of any single algorithm are depicted in the violin-plot of Figure 6. This plot clearly confirms the main observation: (1+1)-ES is by a large amount the algorithm to go for the problem at hand under the considered budget. Furthermore, no execution of BO was competitive with respect to (1+1)-ES, CMA-ES and DE, probably because the dimension of the search space (60) is too large for BO. Regarding the robustness: (1+1)-ES is the most robust algorithm, while PSO shows the largest variance. Since PSO is known to suffer of premature convergence to local optima [19], an high variance may indicate the presence of a considerable amount of local optima in the fitness landscape associated to the objective function.

Finally, Figure 7 depicts the convergence behaviours of the five algorithms averaged over the different executions. By recalling that the population-based algorithms were set with a population size of 30, an interesting observation here is to note that (1+1)-ES, after the first 30 evaluations, is the most effective algorithm. Then, (1+1)-ES is able to keep the gap till the end of the execution. However, Figure 7 also shows that, with a larger budget, it is likely that both DE and CMA-ES will be able to match the same effectiveness of (1+1)-ES.

6 Conclusion and Future Work

Hydraulic computation of flow is largely involved in modeling and preventing flood disasters depending on the conformation and features of channels, river and flood plains. The accuracy of the flow computational results is strictly connected with the estimate of Manning's roughness coefficients for different cross-sections of the rivers. Although deterministic methods exist for estimating Manning's coefficient, they exhibit a low accuracy. In this work a black-box approach is proposed, which allows to formulate Manning's roughness coefficient estimation

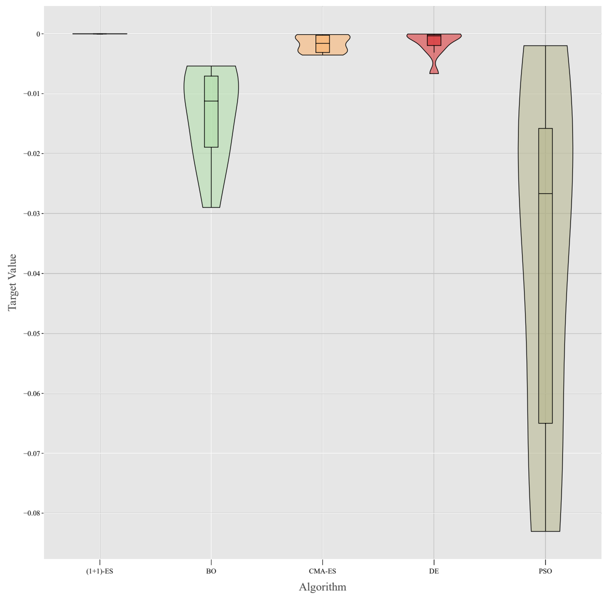

Fig. 6. Empirical probability density functions of the final objective values registered by the executions of each algorithm.

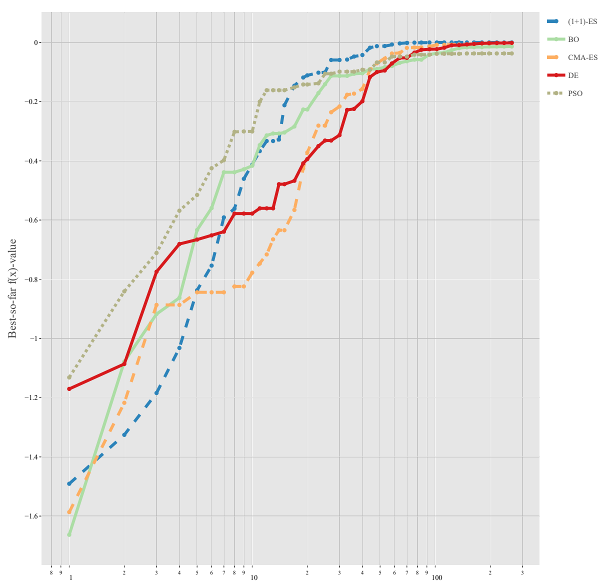

Fig. 7. Average convergence behaviours (abscissa is in log-scale).

problem as an optimization process, which tries to minimize the estimation error of a river depth by varying the coefficient value. Evolutionary algorithms, using the error estimation as a blackbox objective function, have been applied; they cover different classes of metaheuristics including DE, PSO, (1+1)-ES, CMA-ES and BO. The algorithms have been tested using a real dataset, from a case study of a river in central Italy, and compared with a standard deterministic methodology used in hydraulic computation literature. Results show that the application of evolutionary metaheuristics always outperforms by orders of magnitude the traditional methodology. The simple (1+1)-ES is the most performing metaheuristic, obtaining errors of less than 10^{-5} meters in river depth estimation with respect to more than 1.5 meter of the deterministic method.

The proposed approach is quite promising, future works will aim to confirm the performances by its application to other datasets, also considering multiple gauge stations deployed along the river. It must be pointed out that the proposed black-box formulation for Manning's coefficient estimation is quite general, since it allows to apply any meta-heuristic for numerical optimization problem to roughness coefficients estimation. Therefore, a systematic experimentation of other metaheuristic approaches is planned, as well as considering the extensions of the black box approach to the estimation of other hydraulic flow parameters.

Acknowledgement

This work was partially supported by the following research grants: (i) Università per Stranieri di Perugia – Progetto di ricerca *Artificial Intelligence for Education, Social and Human Sciences*; (ii) Università per Stranieri di Perugia – *Finanziamento per Progetti di Ricerca di Ateneo — PRA 2020*; (iii) PRIN project *PHRAME: Phraseological Complexity Measures in learner Italian.*

References

1. HEC-RAS, User Manual, Davis Version 4.0
2. Atanov, G.A., Evseeva, E.G., Meselhe, E.A.: Estimation of Roughness Profile in Trapezoidal Open Channels. J. of Hydraulic Engineering **125**(3), 309–312 (1999)
3. Baioletti, M., Milani, A., Santucci, V.: Variable neighborhood algebraic differential evolution: An application to the linear ordering problem with cumulative costs. Information Sciences **507**, 37–52 (2020)
4. Becker, L., Yeh, W.W.G.: Identification of parameters in unsteady open channel flows. Water Resources Research **8**(4), 956–965 (1972)
5. Becker, L., Yeh, W.W.G.: Identification of multiple reach channel parameters. Water Resources Research **9**(2), 326–335 (1973)
6. Beyer, H.G., Schwefel, H.P.: Evolution strategies – A comprehensive introduction. Natural computing **1**(1), 3–52 (2002)
7. Brabazon, A., O'Neill, M., McGarraghy, S.: Natural Computing Algorithms. Springer, 1st edn. (2015)
8. Di Francesco, S., Zarghami, A., Biscarini, C., Manciola, P.: Wall roughness effect in the lattice Boltzmann method. In: AIP Conference Proceedings. vol. 1558, pp. 1677–1680. American Institute of Physics (2013)

9. Di Francesco, S., Biscarini, C., Manciola, P.: Numerical simulation of water free-surface flows through a front-tracking lattice Boltzmann approach. Journal of Hydroinformatics **17**(1), 1–6 (2015)
10. Di Francesco, S., Biscarini, C., Manciola, P.: Characterization of a flood event through a sediment analysis: The Tescio River case study. Water **8**(7), 308 (2016)
11. Drake, J., Bradford, A., Joy, D.: Application of HEC-RAS 4.0 temperature model to estimate groundwater contributions to Swan Creek, Ontario, Canada. Journal of Hydrology **389**(3), 390–398 (2010)
12. Frazier, P.I.: A tutorial on bayesian optimization. Preprint arXiv:1807.02811 (2018)
13. Gray, D.N., Hotchkiss, J., LaForge, S., Shalit, A., Weinberg, T.: Modern languages and Microsoft's component object model. Comm. of the ACM **41**(5), 55–65 (1998)
14. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**(2), 159–195 (2001)
15. Hollander, M., Wolfe, D.A., Chicken, E.: Nonparametric statistical methods, vol. 751. John Wiley & Sons (2013)
16. Horritt, M., Bates, P.: Evaluation of 1D and 2D numerical models for predicting river flood inundation. Journal of Hydrology **268**(1), 87–99 (2002)
17. Kachiashvili, K.: Software realization problems of mathematical models of pollutants transport in rivers. Adv. in Engineering Software **40**(10), 1063–1073 (2009)
18. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. of IEEE International Conference on Neural Networks. vol. 4, pp. 1942–1948 (1995)
19. Larsen, R.B., Jouffroy, J., Lassen, B.: On the premature convergence of particle swarm optimization. In: Proc. of 2016 Eur. Control Conf. pp. 1922–1927 (2016)
20. Perry, B.: Open-Channel Hydraulics. Science **131**(3408), 1215–1215 (1960)
21. Pinar, E., Paydas, K., Seckin, G., Akilli, H., Sahin, B., Cobaner, M., Kocaman, S., Atakan Akar, M.: Artificial neural network approaches for prediction of backwater through arched bridge constrictions. Adv. in Eng. Software **41**(4), 627–635 (2010)
22. Rapin, J., Teytaud, O.: Nevergrad - A gradient-free optimization platform. https://GitHub.com/FacebookResearch/Nevergrad (2018)
23. Rapin, J., Bennet, P., Centeno, E., Haziza, D., Moreau, A., Teytaud, O.: Open source evolutionary structured optimization. In: Proc. of the 2020 Genetic and Evolutionary Computation Conference Companion. pp. 1599–1607 (2020)
24. Rodriguez, L.B., Cello, P.A., Vionnet, C.A., Goodrich, D.: Fully conservative coupling of HEC-RAS with MODFLOW to simulate stream–aquifer interactions in a drainage basin. Journal of Hydrology **353**(1), 129–142 (2008)
25. Santucci, V., Baioletti, M., Di Bari, G., Milani, A.: A binary algebraic differential evolution for the multidimensional two-way number partitioning problem. In: Proc. of 2019 Eur. Conf. on Evolutionary Comp. in Combinat. Optim. pp. 17–32 (2019)
26. Santucci, V., Milani, A., Caraffini, F.: An optimisation-driven prediction method for automated diagnosis and prognosis. Mathematics **7**(11), 1051 (2019)
27. Shen, D., Jia, Y., Altinakar, M., Bingner, R.L.: GIS-based channel flow and sediment transport simulation using CCHE1D coupled with AnnAGNPS. Journal of Hydraulic Research **54**(5), 567–574 (2016)
28. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization **11**(4), 341–359 (1997)
29. Violante, C., Biscarini, C., Esposito, E., Molisso, F., Porfido, S., Sacchi, M., et al.: The consequences of hydrological events on steep coastal watersheds: The Costa d'Amalfi, eastern Tyrrhenian Sea. IAHS publication **327**, 102 (2009)
30. Zischg, A.P., Mosimann, M., Bernet, D.B., Röthlisberger, V.: Validation of 2D flood models with insurance claims. Journal of Hydrology **557**, 350–361 (2018)