IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 8, 2018, accepted October 24, 2018, date of publication November 5, 2018, date of current version December 3, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2879571

Optimization of Water Reservoir Operation to Minimize the Economic Losses Caused by Pollution

SORIN N. CIOLOFAN[1], GHEORGHE MILITARU[2], AURELIAN DRAGHIA[3], RADU DROBOT[3], AND MONICA DRĂGOICEA[4], (Member, IEEE)

[1]Department of Computer Science, University Politehnica of Bucharest, 060042 Bucharest, Romania
[2]Department of Entrepreneurship and Management, University Politehnica of Bucharest, 060042 Bucharest, Romania
[3]Faculty of Hydrotechnics, Technical University of Civil Engineering Bucharest, 020396 Bucharest, Romania
[4]Department of Automatic Control and Systems Engineering, University Politehnica of Bucharest, 060042 Bucharest, Romania

Corresponding author: Monica Drăgoicea (monica.dragoicea@upb.ro)

ABSTRACT The pollution of river water as a result of an accident can cause different types of serious damages to people, aquatic life, agricultural land, plants, and animal ecosystems. Apart from tragic events such as deaths or injuries, the accidental water pollution causes important economic losses. One effective measure is to reduce the concentration of the pollutant in the river water by adding clean water released from reservoirs placed on tributaries of the river. In this paper, we focus on finding an optimal operation of the water reservoirs (such as opening/closing time of the gates) with the goal of minimizing the total cost of the economic damages. Two components of the total cost are defined and quantified: the cost of the water used to achieve dilution and the losses caused by the existence of the pollutant in a certain concentration in the water. This problem is formalized as a non-linear multi-objective simulation–optimization model, subject to constraints. The solution is evaluated for a real use case (Jijia River, respectively, Drăcşani and Hălceni reservoirs in North–East Romania). The results are presented for two experiments. They show that by using the proposed optimization solution, the total cost of economic losses may be reduced by almost six times as compared with an empirical operation of the reservoirs and by almost 10× if no dilution takes place.

INDEX TERMS Economic losses, optimization, simulation, water pollution control, water resources management.

I. INTRODUCTION

Accidental water pollution mitigation is a very difficult problem to solve from a cost and efficiency perspective. Depending on the pollutant types, in some cases it is possible to do a partial control of the pollution event, by releasing water from the reservoirs located on the tributaries of the main river. In other cases, it is not possible to take actions. In the literature, when discussing reservoir operation in the context of water quality, the authors are generally more concerned about the in-reservoir water quality and less about the quality of water outside the reservoir as a result of an accidental pollution.

Amirkhani *et al.* [1] discuss multi-objective operation of reservoirs from the perspective of water quality optimization. During floods, highly condensed total dissolved solids (TDS) can go into reservoirs. Later, these inorganic salts can be released with the water from the reservoir and create water quality issues such as salty taste, corrosiveness or elevated hardness. The optimal operation of the reservoir is studied with respect to two objectives: minimization of TDS and minimization of the difference of the temperature of input and output water in the reservoir. As a case study, the authors have implemented the method on Karaj two outlets reservoir in Iran. To implement the proposed simulation-optimization method, two models have been implemented: the Water quality and 2-D hydrodynamic models. These models are developed by combining the CE-QUAL-W2 model with the non-dominated sorting genetic algorithm-II (NSGA-II). Four scenarios corresponding to the annual seasons were defined. The results have shown that better operation is achieved when using the two outlets compared with one outlet utilization.

Shirangi *et al.* [2] discuss the optimal reservoir operation considering two objectives: a) the allocated water quantity and b) the quality of water. The problem is decomposed

for long term and annual models. For the long term model, the objective is to provide water supplies, whereas the annual model has two sub-objectives: the allocated quantity and the quality.

The work of Chaves and Kojiri [3] is placed in the same area of research. Here, the main concern is to improve the quality of in-reservoir water by applying optimal operational reservoir strategies. To achieve this goal, the authors use a fuzzy neural network trained by a genetic algorithm.

Regarding the management of reservoirs for downstream water quality, Jaworski *et al.* [4] proposed a general mathematical model for an optimal release sequence in a multi-reservoir system which targets either the best quality of water or the minimum storage costs. Dhar and Datta [5] considered the operation of a single reservoir, having a single objective function subject to optimization. The solution is verified on a hypothetical reservoir, using for simulation the CE-QUAL-W2 model, and for optimization a Genetic Algorithm is applied.

Wang *et al.* [6] study the impact of inter basin water diversion projects and cascade reservoirs on the quality and quantity of water. The study area for validation of their model is the middle-lower Hanjiang River in China. The basin is divided into 18 connected land units consisting of tributaries and reservoirs. A 1-D hydrodynamic model coupled with a water quality model is proposed. This model is compared with the historical time series data spread over on 42 years recording (1956-1998). The relative errors of simulated values related to observed values are less than 5% for water quantity and less than 20% for water quality. Next step was to asses the impact of the water projects (SNWD-South To North Water Diversion and YHWD-Yangtze-Hanjiang Water Diversion) on the water quality and quantity. The results show that SNWD leads to a decrease of water level (decrease rate 0.48%-2.68%) and to a decrease of water quality below the target. The solution yielded by YHWD project must partially compensate by increasing the flow. However, this solution introduces water quality degradation.

Cervellera *et al.* [7] introduce a numerical solution for an optimization problem of a water reservoir network consisting of 10 basins. Stochastic dynamic programming is used to release a certain amount of water from each reservoir to minimize a nonlinear cost. Voudouris *et al.* [8] developed a Decision Support System that assesses the groundwater pollution risks due to agricultural land use. The tool provides an optimal land use to prevent groundwater contamination.

In [9], a solution to reduce the concentration of pollutant on a river following an accidental pollution was proposed. Dilution of the pollutant is achieved by operating (open and close) the bottom gates of the reservoirs. The gates of the reservoirs located on the tributaries are left open for a certain amount of time, to release a significant volume of clean water in the river and to control the pollution level. This approach is applicable to any river having m reservoirs on the tributaries downstream the place where the accidental pollution occurs (Fig. 1).

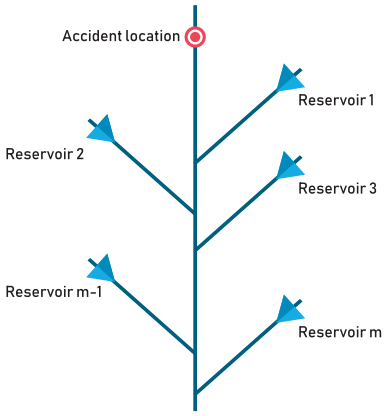


FIGURE 1. Accidental pollution control using dilution.

FIGURE 2. Drăcşani dam and artificial lake © http://www.visitbotosani.ro/.

The proposed solution has been tested on a 205 km stretch of the Jijia River in Romania (Fig. 4). The dilution on Jijia River is realized using the water released from two reservoirs, Drăcşani (Fig. 2) and Hălceni, that are placed on the Sitna respectively Miletin tributaries. It implies identifying an operation strategy for the bottom gates of the two dams. Drăcşani reservoir (Fig. 2) is made of an earth dam with a height of 5.85 m, while Hălceni reservoir has a dam with a height of 10.5 m.

The heuristic solution to operate the gates of the reservoirs from [9] has the disadvantage that it may not be optimal, as it may allow the release of more water from the reservoirs than necessary. The surplus of water means an extra cost, considering that the raw water from the reservoirs has a specific cost per cubic meter (it may be further transformed in energy or may be used for water supply). Since the accidental pollution is an extremely harmful phenomenon, even a non-optimal solution is preferred to ''taking no action'' scenario. By using dilution we are able to reduce the concentration of pollutant below the threshold that put people at risk and, generally, to reduce the concentration up to 90%.

This paper further elaborates on finding an optimal solution for the dilution method. While real-time data collection for the transport of the pollutant in the river water is not always accessible, a hydraulic model of the river stretch including Advection-Dispersion module is needed a priori. This model is created for this paper using a numerical modeling and simulation tool, MIKE11 [10].

The paper is organized as follows. Section II positions this work within the scope of the water pollution control with dilution, identifies the major components of the pollution scenarios to be evaluated, and describes the problem in terms of a a non-linear multi-objective simulation - optimization model subject to constraints. Several available methods for solving optimization problems are evaluated. Arguments for the selection of a derivative-free optimization method, specifically here the Nelder-Mead optimization with penalties, are presented. Appendix presents the algorithmic form of the Nelder-Mead algorithm and the corresponding updates of the penalties for the specific optimization problem for Jijia river described in Section III.

In Section III, the problem of evaluating the cost of damages of the river water for a number of reservoirs is mathematically formalized in terms of a multi-objective constrained optimization problem. The goal of the optimization process is to minimize the total cost (in EUR monetary units) of the damages produced by the accidental pollution and the cost of water used for dilution. Formulas for computing both costs are provided. The multi-objective function is transformed into a single-objective function by means of scalarization. It is described how this problem can be solved using a derivative-free method (such as Nelder-Mead algorithm) adapted to allow parameter constraints and bounds.

Section IV describes the application of the general multi-objective optimization method introduced in Section III for the specific study case of Jijia River that was detailed in [9]. The results obtained in two experiments performed on Jijia River are described in Section V. These results indicate the superiority of the optimal solution providing a six times lower total cost than the heuristic solution. These positive results are achieved by choosing the starting points as close as possible to the heuristic solution computed according to the Eqs. (1) and (2).

Section VI concludes the paper with a detailed discussion on the obtained results. Specific limitations of the proposed method and a description of possible further improvements and research directions are also addressed.

II. BACKGROUND AND RELATED WORK

As stated before, to apply the dilution method described in [9], it is critical to decide when to open/close the reservoirs. In our initial solution, the timestamps for opening/closing the gates of the *j-th* reservoir ($t_{open}[j], t_{close}[j]$) are computed in an empiric fashion, observing $\tau_1[j]$ (the moment when the pollutant concentration exceeds a predefined threshold value c^*) and $\tau_2[j]$ (the moment when concentration decreases below this threshold). A time margin Δt is added to these timestamps, making sure that the measure is effective. The time T_j needed for the water to travel from reservoir j placed on the tributary to the main river is also important, such as:

$$t_{open}[j] = \tau_1[j] - T_j - \Delta t; 1 \leq j \leq m \quad (1)$$

$$t_{close}[j] = \tau_2[j] - T_j + \Delta t; 1 \leq j \leq m \quad (2)$$

The actual values for c^* are chosen according to environmental regulations whereas the values for Δt are determined through expertise. The decision makers receive a recommendation regarding the moment to open/close the bottom gates of the two reservoirs, an estimation of the disposed water volume, and a graph depicting a comparison of the pollutant concentration in a dilution scenario vs. no action scenario. By performing the recommended actions, the discharge on Jijia River is increased and dilution takes place. The immediate effect is the decrease of pollutant concentration below the critical threshold that represents a danger for population or the environment.

For the heuristic approach, we proposed the simplifying assumption on the bottom gates being completely opened or totally closed. For this work, for the optimization process we will consider that the gates have a degree of opening (level) that ranges from 0% (totally closed) to 100% (completely opened).

Each pollution scenario is defined by a set of pollution variables supplied by the end user (such as accident location, volume of pollutant accidentally released, concentration of pollutant). For each pollution scenario, the transport of pollutant along the river is simulated using a specialized software, MIKE11 [10], a professional, license based, modeling software package for rivers and channels developed by DHI (Danish Hydraulic Institute). It is used extensively in environmental sciences [11]–[13]. MIKE11 allows the simulation of different water related scenarios, such as water level (flood forecasting), pollutant and sediment transport in rivers, irrigation canals or estuaries. The model of the river is 1-D.

MIKE11 is composed of a number of modules: the Hydrodynamic (HD) module, the Advection Dispersion (AD) module, the Non-cohesive Sediment Transport (NST) module, and the Rainfall-Runoff (RR) module. The core HD module is based on the 1-D Saint Venant equations. It is typically used for floods forecast, operation of drainage systems, and irrigation. The AD module is based on the equation of conservation of mass of the dissolved substance. It is used to model spreading of pollutants. The NST module is used for sediments that are not associated with contaminants, such as sands that creates dunes or ripples. The RR module computes the conversion of rainfall into runoff.

The input for MIKE11 simulation tool consists of the description of the physical structure of the system (cross-sections, roughness coefficients), boundary conditions (hydrographs and rating curve), and the pollutograph representing the variation of the pollutant concentration.

As output it generates TAB delimited text files representing a matrix of pollutant concentrations/discharges, each concentration/discharge being estimated for a specific simulated timestamp at a given location on the river. In this paper, we consider MIKE11 as a computational procedure repeatedly called.

Instead of using Equations (1) and (2), in this article we focus on finding an optimal solution for the triple $t_{open}[j]$, $t_{close}[j]$ and $level[j]$ (the level of gates opening) for each of the m reservoirs involved in dilution. The volume of water released from reservoir j depends directly on the above mentioned triple.

In Mathematics, Optimization refers to finding a solution that minimize/maximize an objective function. If the objective function is a cost function (for example cost of accidental pollution expressed in EUR currency) then the goal is to find the specific values of the function variables (also called ''design parameters'') that minimize the objective function.

In this specific case referring to the pollution control using dilution, the cost function has two components:

- the economic loss caused by the river being polluted (such as fisheries damage, domestic water consumption), and
- the cost of the reservoir water released for dilution.

The first component of the cost depends directly on the values of the pollutant concentration along the river. These values, as well as the volumes of released water from the reservoirs into the river, are obtained here using the specific site model created in MIKE11 [9].

Therefore, the values of the objective function are obtained by numerical computation, without the evaluation of the derivatives. This type of optimization is called ''derivative-free optimization''. Usually equality and/or inequality constraints are imposed on the input variables, the problem being named ''constrained optimization''. In this category enroll a large number of scientific applications lacking knowledge on the internal implementation of a given executable, such as chemical reactions or numerical code that involves integrals or partial derivative equations (PDE's).

Formally the problem can be described as follows:

Given a non-linear multi-variable function

$$f : R^n \rightarrow R \tag{3}$$

analytically unknown, but the values of $f(x)$ can be individually computed, find

$$\min_x f(x) \tag{4}$$

considering the l inequality constraints,

$$g_{j(x)} \leq 0; \quad j = 1..l \tag{5}$$

the m equality constraints,

$$h_{k(x)} = 0; \quad k = 1..m \tag{6}$$

and n individual bounds constraints,

$$x_i \in [l_i, u_i] \subseteq (-\infty, \infty); \quad i = 1..n \tag{7}$$

where l_i and respectively u_i are the corresponding *lower* and *upper* bounds. The functions g_j and h_k can be linear or non-linear functions.

The function f is said to be ''cheap'' if a large number of evaluations (for example, 1000) may be suitably fulfilled in terms of computational resources or running time. The function is said to be ''costly'' if using the same computational resources, for the same amount of time, the function can be evaluated only for a smaller number of times (for example, less than 200).

The optimization methods can be classified as *direct* or *model based* [14], [15]. In the first case the function is evaluated whereas in the second case a surrogate function that mimics the original function is used considering that the surrogate has the advantage of being a cheaper function. If the objective function f and the constraints functions are convex functions, then there is only one global optimal solution. For non-convex optimization problems, multiple locally optimal points can exist. The non-convex optimization problems are much harder to solve than the convex ones, because of the ''local optimum trap''.

In their survey of derivative-free optimization algorithms, Rios and Sahinidis [16] provide a historical overview and a state-of-the-art, comparing performance of 22 algorithms given identical initial conditions (same starting point, constraints, and bounds). The criteria used in the comparison refers to several aspects, such as: a) the precision of the determined solution, given a predefined maximum number of function evaluations; b) the ability of the algorithm to find a near-global solution for the non-convex problems; and c) the behavior for various starting points (non-favorable chosen or near-optimal chosen).

The authors classify the methods as *local search* methods (searching in the proximity of the starting point with the potential risk to be trapped in a local optimum) and *global search* methods (search realized in the entire solution space). Further, each category can be sub-divided into *direct search* and *model based search*. Examples of direct local search algorithms are the Nelder-Mead simplex [17] and the generalized pattern search (GPS) [18]–[20]. GPS uses a set of vectors (pattern) at each iteration that indicates which points to be searched for on the current iteration.

The local search model based methods make the assumption that a surrogate model of the objective function [21], [22] or of its gradient [23] can be developed and used further to conduct the search in an intelligent way.

Global search algorithms are categorized into deterministic methods (branch-and-bound [24], multilevel coordinate search [25]), model based (kriging [26], radial basis functions [27], [28]) and stochastic (such as Hit-and-run, Genetic algorithms and Particle swarm algorithms). The stochastic global search algorithms have non-deterministic steps.

The *Hit-and-Run* algorithms [29]–[31] compare the solution found at current iteration with a random candidate. This algorithm updates the solution only if the candidate is an improved point.

Genetic algorithms were used to represent points in the search space as individuals that are assigned a fitness function, as described in the works of Genlin [32] or Deb *et al.* [33]. The disadvantage of these methods is represented by the fact that they can require a large number of generations to obtain an acceptable solution.

Particle swarm algorithms are a type of bio-inspired methods that use principles of swarm intelligence in Computer Science [34]–[36]. A set of particles (representing the points in the search space) are maintained at each iteration and have a position and a velocity vector. Recently hybrid algorithms took the advantage of the global search scope offered by swarm optimization with the fast local convergence of Nelder-Mead method [37] or Generating Set Search Algorithm [38].

Mladenović *et al.* [39] proposed to use variable neighborhood search to solve (unconstrained and constrained) continuous optimization problems. The neighborhoods are changed within the search both for descent to local minima and for escaping from the local traps. In the same area of global minimization of functions depending on continuous variables, Chelouah and Siarry [40] discussed an adaptation of Tabu Search (TS) metaheuristic. The authors enhanced the basic TS described in [41] by introducing *diversification* (detection of promising areas) and *intensifiation* (search inside the most promising area). They pointed out that the method yields good performance on functions having more than 10 variables.

Tenne [42] addresses the problem of optimizing expensive black-box functions by means of machine learning algorithms. Specifically, the optimization process integrates classifiers to predict whether an input vector leads to a simulation tool failure. In these cases, the search will favor inputs for which the simulation succeeds.

Within the scope of this work, we chose the Nelder-Mead algorithm [17], adapted by introducing penalties to implement constraints. The algorithm is used here to solve our optimization problem, described in Section III. The advantage of this algorithm is that it requires only a few function evaluations in each iteration (as opposed to previous optimization algorithms). This is extremely important for costly functions (such as our proposed objective function whose evaluation depends on running the MIKE11 model simulation). This objective function is introduced in Section III, Equation (17). The method rapidly produces accurate results.

The algorithmic form of the Nelder-Mead algorithm and the corresponding updates of the penalties for the specific optimization problem for Jijia river that is addressed in Section III are presented in Appendix.

III. DILUTION-BASED OPTIMIZATION OF WATER RESERVOIRS OPERATION FOR WATER POLLUTION CONTROL

This section elaborates on a specific case study, considering as a starting point the generic river with m water reservoirs that is presented in Fig. 1.

The optimization problem associated to this generic representation is defined as follows: determine an optimal operation of these reservoirs that minimizes both the cost of released water used for dilution (denoted by C_1) and the cost induced by the accidental pollution (C_2). Hence, we define a multi-objective optimization problem:

$$\min_x \{C_1(x), C_2(x)\} \tag{8}$$

where C_1 and C_2 are two objective functions.

In general, there is not a single solution x that minimize all the objective functions. Instead, the so called ''Pareto optimal solutions'' are preferred. They represent a trade-off that is globally optimum, meaning that the Pareto solution can't be improved with respect to one objective function without increasing the other objective function(s).

A typical way of converting a multi-objective optimization problem to a single-objective optimization problem is through scalarizing (weighted-sum method):

$$f(x) = w_1 \times C_1 + w_2 \times C_2; \quad w_1, w_2 > 0 \tag{9}$$

where $w_1, w_2 > 0$ are called weights.

Mathematically, it can be proved that any solution x of the new single-objective optimization problem

$$\min_x f(x)$$

is a *strict* Pareto optimum of the original multi-objective problem from Eq. (8), as pointed out by Caramia and Dell'Olmo [43]. In other words, no other possible candidate solution x from the objective space can lead to smaller values of all objective functions (with at least one strict inequality).

For our specific reservoir optimization, we use MIKE11, the simulation numerical tool [10] to determine the *hydrograph* (i.e *Q(t)* - discharge as a function of time) and the *pollutograph* (i.e *c(t)* - concentration of pollutant as a function of time) at a specified location on the river, assuming a given pollution scenario. As already mentioned in Section II, this simulation software is considered further as a Black Box component, as there is no knowledge on its internal implementation.

With MIKE11, a *pollution scenario* is defined by a set of values for each of the pollution characteristics (such as accident location, pollutant volume, pollutant concentration). The volume of water that is released by each reservoir j depends on three parameters: $t_{open}[j]$, $t_{close}[j]$, and level of gate opening, $level[j]$ (j = 1... m). The first two parameters represent the moments when the reservoir gates are open and, respectively, closed. It is assumed that the gates are completely closed initially, then they are opened partially as indicated by the $level[j]$ value. For each gate, level $l_0[j]$ corresponds to gates completely closed, and $l_1[j]$ corresponds to gates completely open.

Thus, the optimization problem have $n = 3xm$ design parameters. In this particular case, the $\mathbf{x}$ vector is defined as:

$$\mathbf{x} = [t_{open}[1], t_{close}[1], level[1], \ldots, t_{open}[m], t_{close}[m], level[m]].$$

The function f defined in Eq. (3) becomes:

$$f(t_{open}[1], t_{close}[1], level[1], \ldots, t_{open}[m], t_{close}[m], level[m])$$

and it represents the cost of both the economical damages produced by the accidental pollution event and the cost of water that was used for dilution. The challenge further is to find a method to quantify these costs (in monetary units).

The constraints corresponding to Eq. (5) and Eq. (7) are the following:

$$t_{close}[j] \geq t_{open}[j]$$
$$level[j] \in [l_0[j]; l_1[j]]$$

A good practice for representation of the results is to normalize all the $3 \times m$ design parameters and bring them to belong to the same interval (0,1). Normalization of the time in (0,1) can be achieved considering the start of the simulation of the hydrodynamic process as being 0 and the stop simulation time as being 1. Then, $t_{open}[j]$ and $t_{close}[j]$ are scaled accordingly to fit the (0,1) interval.

A similar approach can be used for the gates level, considering $l_0[j] = 0$ and $l_1[j] = 1$. As stated before the total cost can be expressed as:

$$C = f(\mathbf{x}) = w_1 \times C_1 + w_2 \times C_2 \tag{10}$$

where C_1 is the cost of the raw water and C_2 is the cost of the damages produced by the pollutant on the river. The scalarization weights are w_1 and w_2.

Regarding the cost of the raw water, this is computed considering the volumes of water released from each of the m reservoirs:

$$C_1 = U_p \times \sum_{i=1}^{m} V_i \tag{11}$$

where U_p represents the unit price (for example, price of a cubic meter of raw water) and V_i is the volume of water released by *Reservoir i*.

The first step is to determine for each reservoir the discharge for the diluted/undiluted scenarios, $Q_i^{diluted}(t)$ and $Q_i^{undiluted}(t)$, immediately downstream the *Reservoir i*. The "diluted scenario" refers to the case when raw water is released from the reservoirs to decrease the pollutant concentration, whereas in the "undiluted scenario" no action is taken to mitigate the accidental pollution.

Then the volume V_i can be computed as:

$$V_i = \int_{t_0}^{t_1} Q_i^{diluted}(t)dt - \int_{t_0}^{t_1} Q_i^{undiluted}(t)dt \tag{12}$$

Introducing Eq. (12) into Eq. (11) yields:

$$C_1 = U_p \times \sum_{i=1}^{m} \left(\int_{t_0}^{t_1} Q_i^{diluted}(t)dt - \int_{t_0}^{t_1} Q_i^{undiluted}(t)dt \right) \tag{13}$$

where t_0 and t_1 represents the start and end timestamps of the simulation time.

The size of the economical loss depends on the concentration of the pollutant on the studied stretch of the river. Let us assume there are p points downstream *Reservoir 1* (see Fig. 1) for which we can compute the maximum concentration of pollutant at these points, respectively $c_1^{max}, c_2^{max}, \ldots, c_p^{max}$. In these locations, dilution positive effects in diminishing the pollutant concentration can be measured. Let's consider $\tilde{c}$ being the maximum of these p values,

$$\tilde{c} = max\{c_1^{max}, c_2^{max}, \ldots, c_p^{max}\} \tag{14}$$

This measure gives an indication of the degree of pollution in the part of the river where dilution takes place. The maximum values can be obtained from the pollutographs (max $c(t)$) given by the numerical simulation tool execution.

When analyzing the impact of pollution on the economy, we must take into account that damages are produced in a variety of sectors such as agricultural irrigation, fisheries, environment, drinking water, tourism. As previously established in the works of Zhang [44] or Faqing *et al.* [45], the economical loss function, $S(c)$, depends on concentration of pollutant and it is best modeled by a sigmoid function.

As the concentration of pollutant c increases, the losses function S increases too, till it reaches a maximum value S_{max} that corresponds to a concentration c_M. For concentrations higher than c_M there is no more significant increase in the losses, as displayed in Fig. 3.

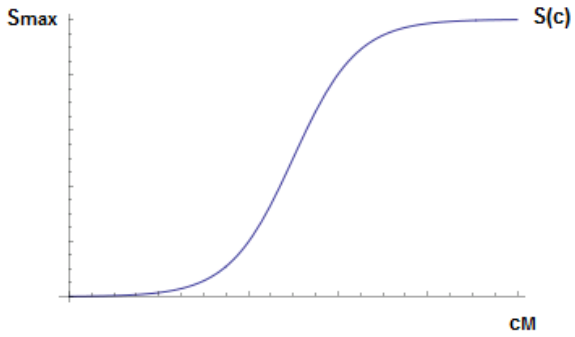


FIGURE 3. Economic losses curve.

Considering the fact that zero damages would correspond to no pollution (c = 0), we propose the following form of the loss function:

$$S(c) = \frac{S_{max}}{1 + e^{-\alpha \times (2 \times \frac{c}{c_M} - 1)}} \tag{15}$$

where S_{max} is the maximum value of the damages that are produced when the pollutant concentration is c_M; α is a tuning parameter that is chosen depending on the desired precision of the formula.

To calculate the cost C_2, we introduce Eq. (14) into Eq. (15):

$$C_2 = S(\tilde{c}) = \frac{S_{max}}{1 + e^{-\alpha \times (2 \times \frac{\tilde{c}}{c_M} - 1)}} \tag{16}$$

From Eq. (13), Eq. (16), and Eq. (9) the cost function formula is obtained:

$$C = f(x) = w_1 \times U_p \times \sum_{i=1}^{m} \left(\int_{t_0}^{t_1} Q_i^{diluted}(t)dt - \int_{t_0}^{t_1} Q_i^{undiluted}(t)dt \right) + w_2 \times \frac{S_{max}}{1+e^{-\alpha \times (2 \times \frac{\tilde{c}}{c_M} - 1)}} \quad (17)$$

where

- $w_1, w_2 > 0$ are the scalarizing weights;
- U_p is the unit price for raw water;
- m is the number of water reservoirs
- $[t_0, t_1]$ is the simulated time;
- S_{max} is the maximum value of the economical losses produced by pollution (expressed in the same unit as U_p);
- α is a precision parameter;
- c_M is the pollutant concentration to which the maximum damages S_{max} are registered;
- $\tilde{c}$ is the maximum concentration along the river. This value is computed based on the numerical simulation tool (MIKE11);
- $Q_i^{diluted}(t)$, $Q_i^{undiluted}(t)$ are the discharges in various points on the river downstream *Reservoir 1*. The values are obtained from the numerical simulation tool (MIKE11).

Eq. (17) represents in a unified form a single cost function that integrates both types of costs. As well, it can be used directly in the optimization process. This working procedure is presented in the following case study.

IV. OPTIMIZATION OF OPERATION OF DRĂCŞANI AND HĂLCENI RESERVOIRS TO MINIMIZE THE ECONOMIC LOSSES OF ACCIDENTAL WATER POLLUTION ON JIJIA RIVER

In this section we apply the general multi-objective optimization method that we proposed in Section III to the specific study case of Jijia River described in details in [9]. In this case, there are *m=2* reservoirs, namely Drăcşani and Hălceni. They must be operated in an optimal way to minimize both the economic losses produced by pollution on Jijia and the cost of water released to realize the dilution process.

Drăcşani reservoir (Fig. 2) was built in 1976 on Sitna River. Its water surface at the Normal Retention Level is 574 ha. It is the second largest reservoir in Jijia River basin and has a total volume of 26.73 millions m^3 of water. Hălceni artificial lake, located on the Miletin River, has an area of 385 ha and a total volume of 49.5 millions m^3 of water.

We consider further the notations introduced in Section III. In this specific case, the optimization problem depends on the following set of variables:

$$t_{open}[1], t_{close}[1], level[1], t_{open}[2], t_{close}[2], level[2] \quad (18)$$

representing the times for opening/closing the Drăcşani/ Hălceni bottom gates, and the opening level of the gates.

- For Drăcşani reservoir, the margins of the levels are: $l_0[1] = 75.6$ (completely closed) and $l_1[1] = 76.52$ (completely open).
- For Hălceni reservoir, the margins of the levels are: $l_0[2] = 47.4$ (completely closed) and $l_1[2] = 49.15$ (completely open).

The simulation of the pollutant transport is implemented in MIKE11. The t_0 and t_1 margins of the simulated time are "2100 1 8 8 0 0" respectively "2100 1 18 8 0 0" (considering the "YYYY MM DD HH MM SS" format).

There are 4 optimization constraints to be considered:

$$t_{close}[1] \geq t_{open}[1]$$
$$level[1] \in [75.6; 76.52]$$
$$t_{close}[2] \geq t_{open}[2]$$
$$level[2] \in [47.4; 49.15]$$

The ranges mentioned above, [75.6;76.52] and [47.4;49.15], refer to the position of the two dam's gates, expressed in MAMSL (meters above mean sea level), considering the Black Sea as zero reference point. To have all the variables belonging to the range [0,1], the following formula was used for normalization:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where x' denotes the normalized value and x_{min}/x_{max} the minimum/maximum values that the non-normalized variable can take. For date/time normalization, the timestamp can be first converted to seconds and then treated further as a long integer value.

The optimization problem is defined in the context of a specified pollution scenario, as described in more details in [9]. We consider further the pollution scenario described by the following four values: Discharge at S.G Vlădeni = 5 [m^3/s], Accident location = 140 [km], Pollutant volume = 20 [m^3] and Pollutant concentration = 800 [mg/l].

To run a MIKE11 simulation, the parameters of the pollution scenario like the ones mentioned above are required. In Eq. (9) we choose the weights equal to 1, $w_1 = w_2 = 1$.

Eq. (11) becomes:

$$C_1 = U_p \times (V_1 + V_2) \quad (19)$$

where V_1 represents the volume of water (expressed in m^3) that is released from Drăcşani reservoir and V_2 the volume released from Hălceni reservoirs. The price of 1 cubic meter of raw water, U_p, is approx. 0.19 EUR[1] in Romania. The cost function proposed in Eq. (17) is expressed in EUR currency.

The goal of the optimization process is to minimize the total cost in EUR that represents the damages produced by the accidental pollution plus the cost of water used for dilution.

For the function that evaluates the economic losses of the pollution (Eq. (16)), we consider for Jijia: $c_M = 0.15$ [mg/l], $\alpha = 20$, $S_{max} = 1500000$ [EUR]. The value $\alpha = 20$ is chosen

[1] http://www.apanovabucuresti.ro/

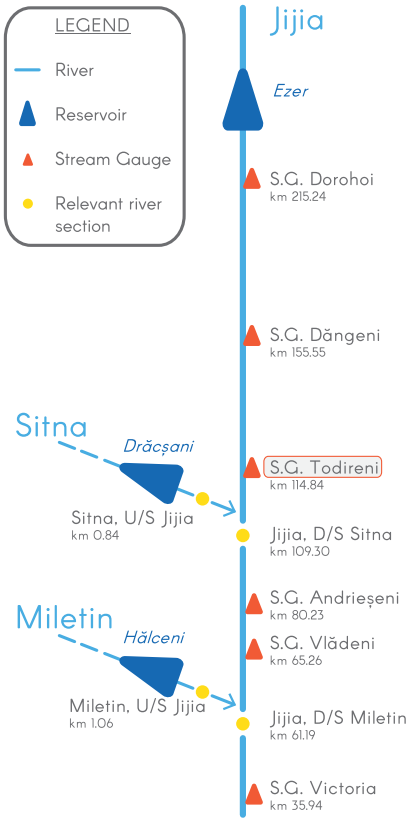


FIGURE 4. Simplified topological model of Jijia River, from [9].

to assure the desired precision, because in this case $\frac{1}{1+e^{20}} \cong 0.0000000020$ and $\frac{1}{1+e^{-20}} \cong 0.9999999979$, where:

$$C_2 = S(\tilde{c}) = \frac{1500000}{1 + e^{-20\times(2\times\frac{\tilde{c}}{0.15}-1)}}[EUR] \tag{20}$$

The concentration $\tilde{c}$ is computed as the maximum of the 5 concentrations at the following locations downstream Jijia confluence with Sitna (Fig. 4): Jijia D/S Sitna, S.G Andrieseni, S.G Vladeni, Jijia D/S Miletin, S.G Victoria. The concentrations in the above mentioned locations are obtained as a result of MIKE11 simulation: the input consists of the 4 known values defining the pollution scenario and the 6 decision variables that define the operation of the two reservoirs (Eq. (18)). To evaluate the costs C_1 and C_2 it is necessary to run at each step one MIKE11 simulation, as described in Section II.

V. CASE STUDY

The general workflow of the application that was implemented for the operation optimization of Drăcşani and Hălceni reservoirs is presented in Fig. 5. The loop contains two main software components:

- a Java application that is responsible for evaluating the objective function and running the derivative-free optimization algorithm (Nelder-Mead in our case), and
- MIKE11 modeling and simulation tool, that executes the Hydrodynamic (HD) and Advection/Dispersion (AD) modules for Jijia, Sitna and Miletin in order to simulate the pollutant transport. The executable MIKE11 is launched as a process from the Java application.

The loop ends when the solution x found at the current iteration meets the termination criteria. The two programs communicate through a set of text files that represents the input/output for MIKE11 tool. The input is represented by a vector $\mathbf{x}$ consisting of 6 variables (Eq.(18)). At the beginning of the execution, an initial estimation $\mathbf{x_0}$ for the input must be provided.

It is worth mentioning here that the MIKE11 model for Jijia pollution is a complex one (composed of three separate models *Jijia D/S*, *Sitna* and *Miletin*). It is computationally intensive. The server where MIKE11 runs is an Intel Xeon with 2 CPU's at 2.1GHz and 16GB RAM, 885GB HDD, running Windows 8. One simulation of MIKE11 (i.e one objective function evaluation) takes approximately 10 minutes to complete on the above mentioned server. The output of MIKE11 tool is a set of text files containing data corresponding to the discharge $Q(t)$ and concentration of pollutant $c(t)$ at locations of interest along the river. Discharge values are used to calculate the water volumes V_1 and V_2 that are part of the Eq. (19). Using the concentration data, $\tilde{c}$ is computed and then the C_2 cost is evaluated according to Eq. (20). The actual optimization using the Nelder-Mead simplex Algorithm adapted with penalties is described in Appendix. The corresponding experiments are described below. The obtained results are also discussed.

A. EXPERIMENT 1

In this experiment, the starting point $\mathbf{x_0}$ in the Nelder-Mead simplex algorithm is considered as being the heuristic solution for the pollution scenario described above. The heuristic solution has been derived in [9] and it is in agreement with Eq. (1) and Eq. (2) in Section II. The values for the input vector, normalized to (0,1) range, in the format described in Eq. (18), are the following:

$$\mathbf{x_0} = [0.224, 0.304, 1, 0.354, 0.491, 1] \tag{21}$$

De-normalized, the vector presented in Eq. (18) becomes: $t_{open}[1]$ = "2100 01 10 13 45 36"; $t_{close}[1]$ = "2100 01 11 08 57 36"; $level[1] = 76.52$; $t_{open}[2]$ = "2100 01 11 20 57 36"; $t_{close}[2]$ = "2100 01 13 05 50 24"; $level[2] = 49.15$.

This input corresponds to a total volume of released water of 4632009 m^3 and a concentration $\tilde{c} = 0.051$ mg/l. In this case, the total cost computed according to Eqs. (19) and (20) is $Cost(\mathbf{x_0}) = 882569.9$ EUR.

In case no action will be taken (no operation of reservoirs), the concentration $\tilde{c} = 0.258$ corresponds to a total cost *Cost(no-action)* = 1500000 EUR. The steps vector for the calculation of the initial simplex was taken $\mathbf{h}$ = [0.1,0.1,0.1,0.1,0.1,0.1].

The evolution of the total cost during the optimization process, at each iteration, is presented in Fig. 6.

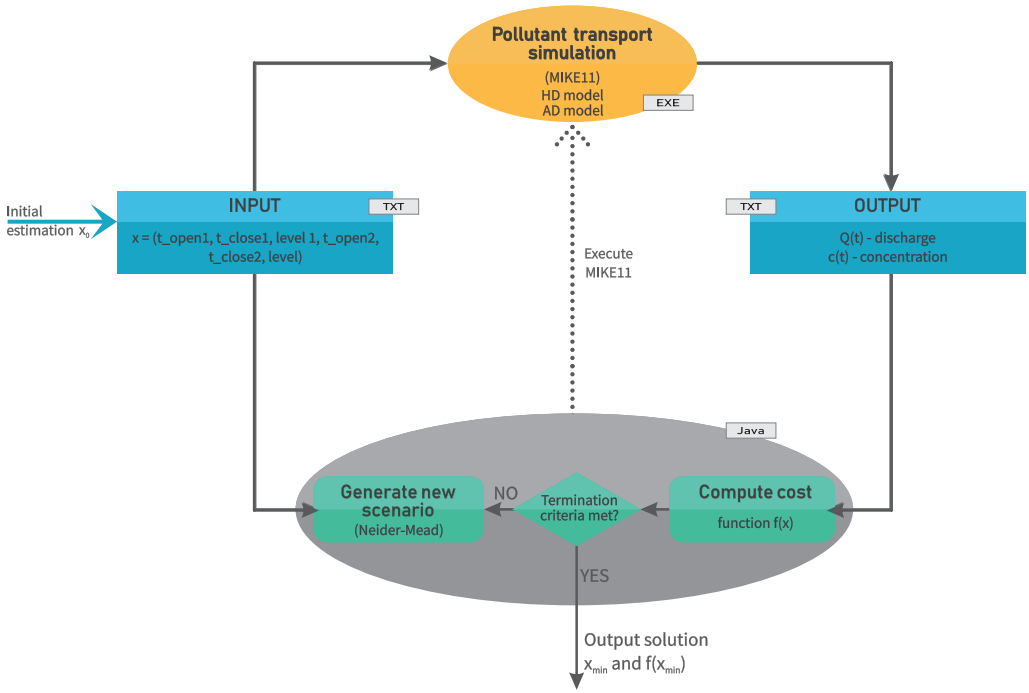


FIGURE 5. Workflow of the optimization application.

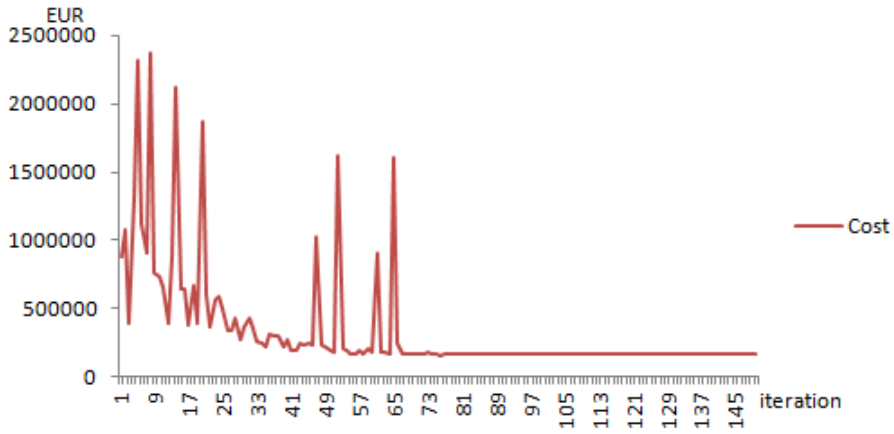


FIGURE 6. Total cost optimization in Experiment 1.

It can be observed that after 70 iterations the cost stabilizes. The algorithm is executed for a maximum number of 150 iterations. The minimum is obtained starting with iteration 74 for

$$\mathbf{x_{min}} = [0.216, 0.275, 0.995, 0.500, 0.501, 0.994] \quad (22)$$

and it corresponds to a minimum cost $Cost(\mathbf{x_{min}}) = 158871.9$ EUR (released volume of water of 813871 m^3 and a concentration $\tilde{c} = 0.053$ mg/l).

Denormalized, the vector presented in Eq. (22) becomes: $t_{open}[1] =$ "2100 01 10 11 50 24"; $t_{close}[1] =$ "2100 01 11 02 00 00"; $level[1] = 76.51$; $t_{open}[2] =$ "2100 01 13 08 00 00"; $t_{close}[2] =$ "2100 01 13 08 14 24"; $level[2] = 49.13$

The optimization process strongly reduced the operation time of Hălceni reservoir, from 33 hours to only 14 minutes. For Drăcşani reservoir, it was reduced to 14 hours from 19 hours.

The cost was diminished by decreasing the volume of water to be released for dilution, but maintaining approximately the same final effect of dilution upon concentration of pollutant (0.053 compared to 0.051). The gates were left almost completely open on the operation period. The same conclusions can be drawn from Fig. 7, where the operation time (in hours) of both reservoirs is presented.

The optimization process highly decreases the operation time of Hălceni reservoir whereas for Drăcşani the reduction

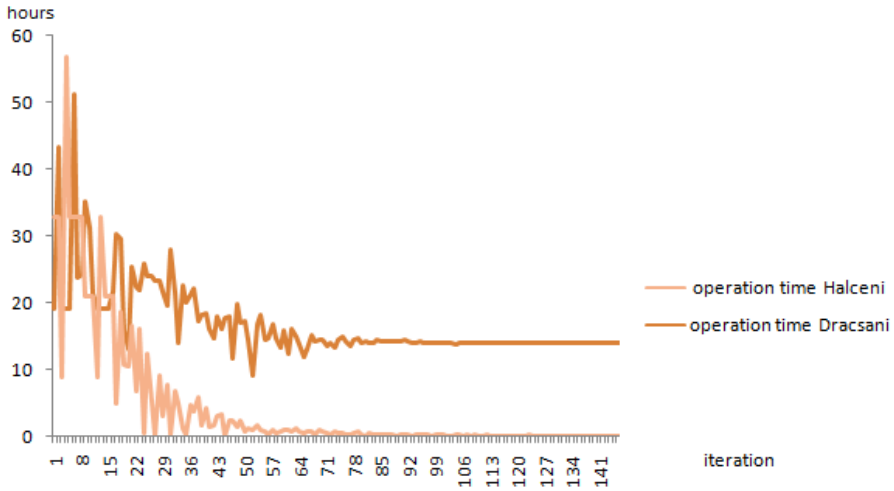


FIGURE 7. Reservoir operation time in Experiment 1.

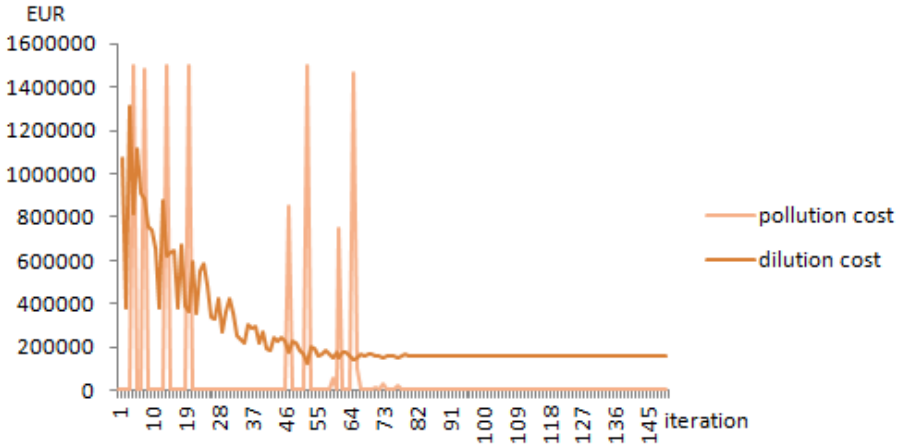


FIGURE 8. Cost components in Experiment 1.

is not so steep (it is only about 25%). This is consistent with the fact that Hălceni reservoir has a higher discharge than Drăcşani and the volume evacuated in the same time period is much higher.

Fig. 8 shows the separate graphs for the two components of the total cost, the water cost and the pollution cost. The best result is obtained when we get zero pollution costs with the minimum released water. Thus, the dilution cost is reduced with 75%.

Finally, Fig. 9 shows a comparative graph of costs when no action is taken, for non-optimized dilution and for optimized dilution. The savings introduced by optimization are 723698 EUR (5.86 times smaller cost).

Compared with the non-action cost, the cost of pollution is 10 times smaller when the operation of reservoirs is optimized (and the maximum concentration of pollutant is reduced from 0.258mg/l to 0.053mg/l).

B. EXPERIMENT 2

In this experiment the starting point $\mathbf{x_0}$ is chosen such that halves the operation times of both reservoirs in comparison with $\mathbf{x_0}$ from *Experiment 1* (Eq. (21)) and it opens the gates only to half of their maximum capacity. In normalized form:

$$\mathbf{x_0} = [0.224, 0.264, 0.5, 0.354, 0.422, 0.5] \quad (23)$$

which de-normalized means $t_{open}[1]$ = "2100 01 10 13 45 36"; $t_{close}[1]$ = "2100 01 10 23 21 36"; $level[1] = 76.06$; $t_{open}[2]$ = "2100 01 11 20 57 36"; $t_{close}[2]$ = "2100 01 12 13 16 48"; $level[2] = 48.27$. Running the MIKE11 simulation with the input parameters as above leads to a water volume of 1369375 m^3 and a concentration $\tilde{c} = 0.091$mg/l.

The total cost $Cost(\mathbf{x_0}) = 1739429$ EUR is almost double than the total cost for $\text{Cost}(\mathbf{x_0^{Experiment1}}) = 882569$ EUR and is even higher than the cost when no action is taken (1500000 EUR).

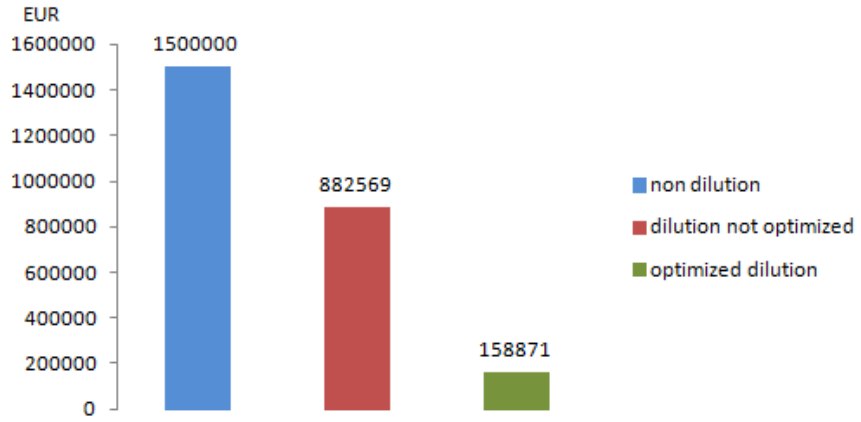


FIGURE 9. Pollution costs for various scenarios in Experiment 1.

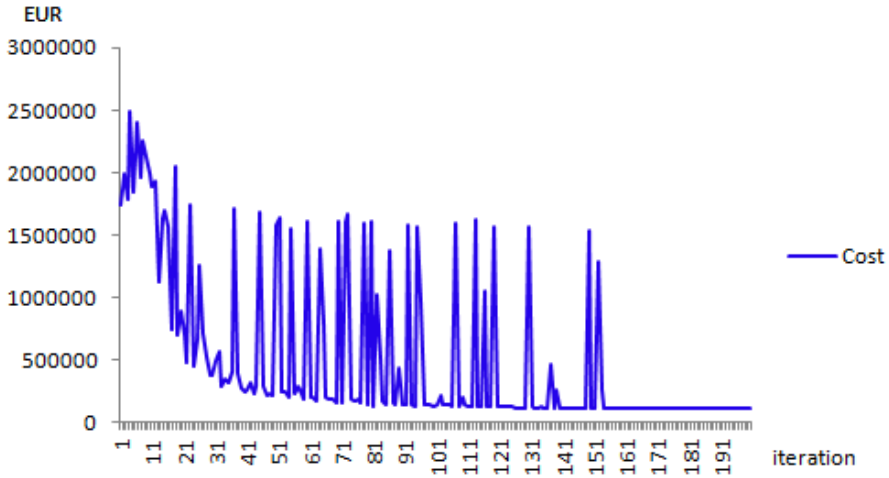


FIGURE 10. Total cost optimization in Experiment 2.

To increase the search space of the initial simplex, we choose $\mathbf{h} = [0.3, 0.3, 0.3, 0.3, 0.3, 0.3]$. As expected, the optimization algorithm requires more iterations to converge than previously (160 iterations compared with 70 iterations). The graphic presented in Fig. 10, describing the evolution of the total cost at each iteration, displays more oscillations compared with the analogous previous graph from Fig. 6. However, the minimum cost that is found (109599) outperforms the value in *Experiment 1* (158871).

The minimum found in *Experiment 2* at iteration 193 is with 31% lower than the minimum found in *Experiment 1*. Even the starting point $\mathbf{x_0}$ did not seem initially so promising, the optimization algorithm found a better solution eventually (Fig. 11).

The minimum is found for

$$\mathbf{x_{min}} = [0.237, 0.276, 0.758, 0.356, 0.362, 0.414] \quad (24)$$

and corresponds to a minimum cost $Cost(\mathbf{x_{min}}) = 109599.47$ EUR (released volume of water of 512398 m^3 and a concentration $\tilde{c} = 0.057$ mg/l).

De-normalized, the vector presented in Eq. (24) becomes: $t_{open}[1] =$ "2100 01 10 16 52 48"; $t_{close}[1] =$ "2100 01 11 02 14 24"; $level[1] = 76.29736$; $t_{open}[2] =$ "2100 01 11 21 26 24"; $t_{close}[2] =$ "2100 01 11 22 52 48"; $level[2] = 48.1245$.

As Fig. 12 shows, the savings are realized by reducing the pollution costs to negligible values by decreasing the concentration $\tilde{c}$ on one hand, and on the other hand by reducing the cost of water used for dilution by minimizing the volume of water needed to achieve the concentration to a level of 0.057mg/l.

This is realized, as depicted in Fig. 13, by shortening the operation time of both reservoirs (for Hălceni from 16 hours to 1.5 hours, and for Drăcşani from 10 hours to 9 hours).

In this experiment, the optimization process reduced the costs in the dilution scenario by 15,8 times.

The experimental results are summarized in Table 1, including the values of the six decision variables obtained for the optimal solution, the steps vector $\mathbf{h}$ used in the Nelder-Mead algorithm, the total cost, the cost components, and the maximum concentration of the pollutant in the river water.

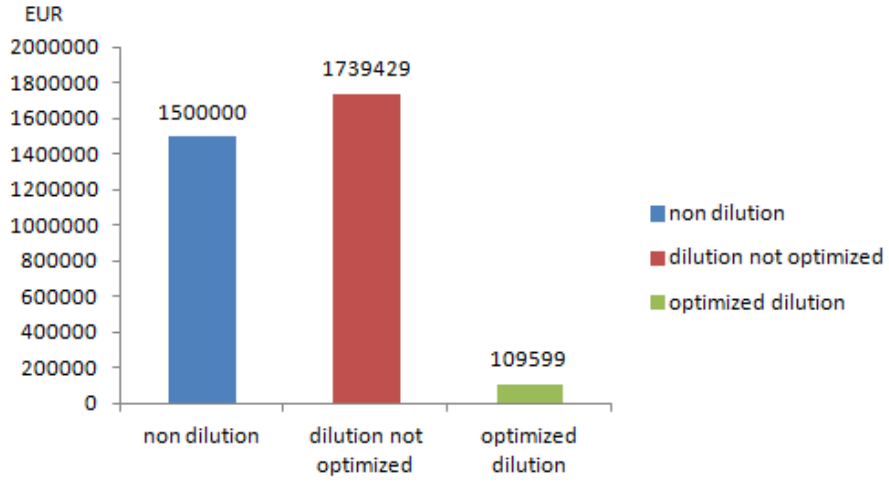


FIGURE 11. Pollution costs for various scenarios in Experiment 2.

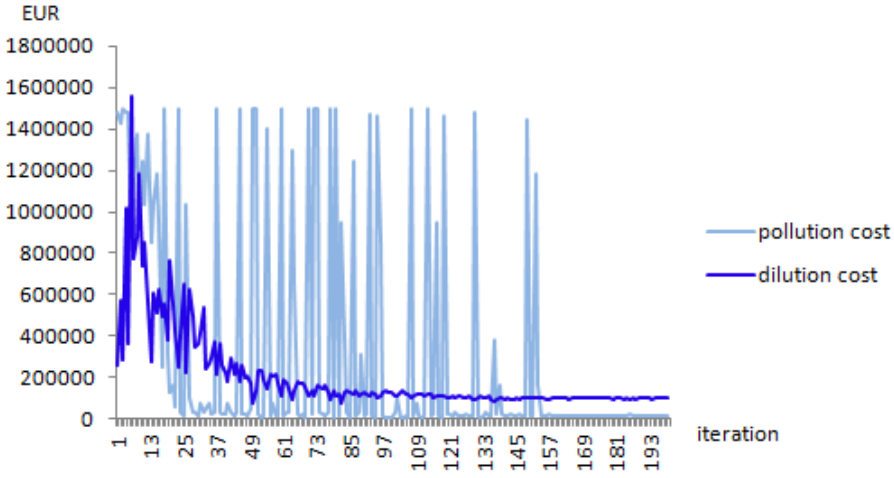


FIGURE 12. Cost components in Experiment 2.

The values of the decision variables in the heuristic in the optimization algorithm for Experiment 1. Half of these values represent the starting point used in Experiment 2.

VI. DISCUSSIONS, CONCLUSIONS, AND FURTHER DEVELOPMENTS

Water pollution control [46], [47] and water resources management based on digital services and information-based intelligence [48] have received important concerns both on research and practical endeavors related to public safety [49], [50]. In [48] tools and statistics algorithms that may be used for analyzing pollution-related information and its correlation to real events are presented. Based on this, specific process activities, services, and resources were identified to support the development of data processing components in information sub-systems of Enterprise Information Systems supporting the design of improved environmental monitoring smart services.

In this respect, this paper extends the research topic introduced in [9] concerning the water pollution control realized using pollutant dilution, defined as an optimization problem. In the previous work, it has been shown how the opening/closing times of the bottom gates with an empirical method may be computed. The gates may be open completely or partially. Even though using the above mentioned heuristic solution is definitely better than taking no action, the procedure is not optimal in terms of cost. Better results may be achieved with less released water.

The optimization problem that is proposed in this paper refers to the optimal operation of water reservoirs to minimize the cost produced by accidental pollution. Two components of the cost are identified, namely: a) the cost of the water that is released from reservoirs to achieve dilution, and b) the cost of river water pollution with a pollutant at a certain concentration. The target is the minimization of both costs.

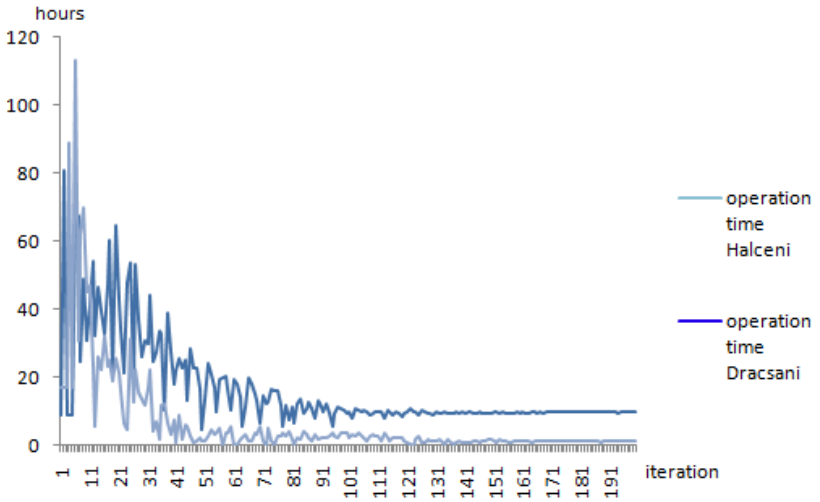


FIGURE 13. Reservoir operation time in Experiment 2.

TABLE 1. Experimental results: Summary of Experiment 1 and Experiment 2.

	Non dilution	Heuristic dilution	Optimized dilution in Experiment 1	Optimized dilution in Experiment 2	Optimized dilution with random starting point
$t_{open}[1]$	N/A	"2100-01-10 13:45:36"	"2100-01-10 11:50:24"	"2100-01-10 16:52:48"	"2100-01-09 23:10:33"
$t_{close}[1]$	N/A	"2100-01-11 08:57:36"	"2100-01-11 02:00:00"	"2100-01-11 02:14:24"	"2100-01-15 21:31:12"
$level[1]$	N/A	76.52	76.51	76.29	75.6
$t_{open}[2]$	N/A	"2100-01-11 20:57:36"	"2100-01-13 08:00:00"	"2100-01-11 21:26:24"	"2100-01-16 09:16:48"
$t_{close}[2]$	N/A	"2100-01-13 05:50:24"	"2100-01-13 08:14:24"	"2100-01-11 22:52:48"	"2100-01-16 09:41:16"
$level[2]$	N/A	49.15	49.13	48.12	48.02
h	N/A	N/A	[0.1,0.1,0.1,0.1,0.1,0.1]	[0.3,0.3,0.3,0.3,0.3,0.3]	[0.8,0.8,0.8,0.8,0.8,0.8]
Pollution cost (EUR)	1500000	2488,2	4236,5	12243,8	1500000
Dilution cost (EUR)	0	880081,7	154635,4	97355,6	6072,2
Total cost (EUR)	**1500000**	**882569.9**	**158871,9**	**109599.4**	**1506072,2**
$\tilde{c}(mg/l)$	**0.258**	**0.051**	**0.053**	**0.057**	**0.258**

The pollution scenario considered for the optimization problem: Discharge at S.G Vlădeni = 5 $[m^3/s]$, Accident location = 140 [km], Pollutant volume = 20 $[m^3]$, Pollutant concentration = 800 [mg/l]. The accident is assumed to happen on "2100-01-09 08:00:00"

A non-linear multi-objective simulation - optimization model subject to constraints it defined, where the parameters that the two costs depend on (i.e the volume of released water and the pollutant concentration after dilution) are computed through a numerical simulation tool for pollutant transport, MIKE11 (i.e. the Black Box component of the software architecture). The decision variables are subject to various constraints (for example, the closing time should be greater than the opening time; the gates level should be in the given operation range where the lower bound correspond to completely closed, and the upper bound corresponds to completely open).

The survey of the currently available methods for solving optimization problems that we performed within the scope of this work indicated the fact that to solve this problem we need to focus on the derivative-free optimization methods and specifically on the Nelder-Mead optimization. The further challenge has been to find an analytical formula to describe the cost of pollution. For the economical losses, the sigmoid function was customized to express the fact that when concentration of pollutant is approximately zero then there are no losses. As well, when the concentration reaches a threshold value c_M then the losses become maximal.

The two experiments presented in Section V strengthened the idea that the result of Nelder-Mead algorithm is influenced by the starting point and the steps used for computing the initial simplex. When the initial solution $\mathbf{x_0}$ is chosen randomly and far from the global optimum, the optimal solution can be worse than in the ''no dilution'' scenario. In such a case, the operation of the reservoirs is not correlated with the pollutant propagation timing and the dilution of pollutant does not happen. The concentration of pollutant remains at the same value as in the no-action scenario, despite the fact that water was released from reservoirs. This leads to an extra cost of released water besides the maximal cost induced by the accidental pollution. In this case the optimization algorithm will be attracted in a ''local trap'' trying to minimize the

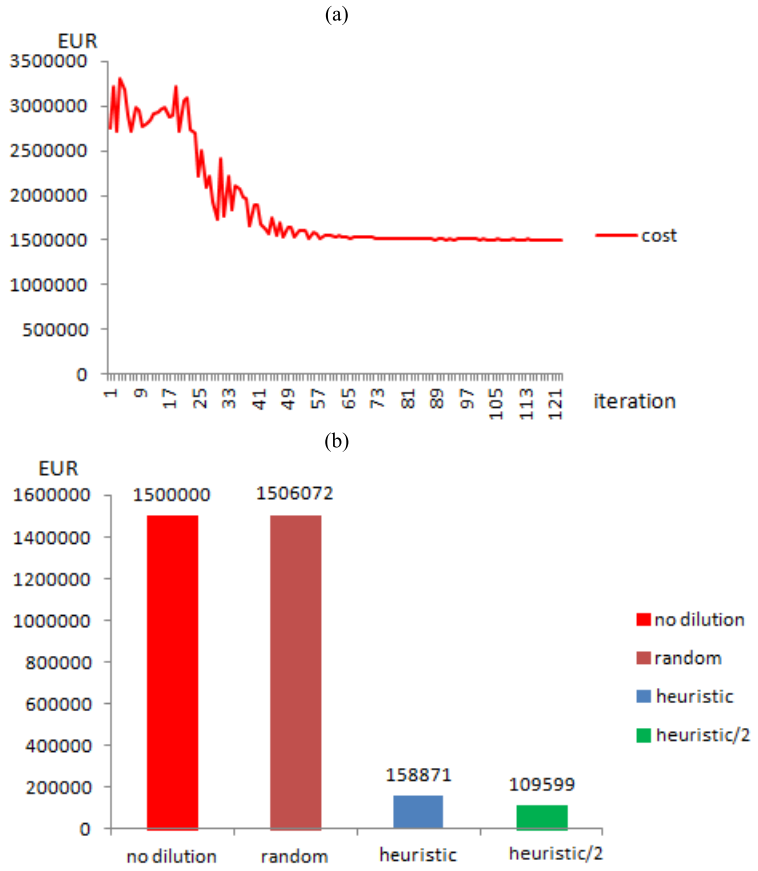


FIGURE 14. Influence of starting point on total cost. (a) Optimization cost for unfavorable chosen x0. (b) Influence of starting point on cost.

volume of water released (basically equivalent with the no-action scenario). The total cost converges (after a sufficient number of iterations) to the total cost when no dilution takes place (i.e. 1500000 EUR).

In Fig. 14-A) the optimization process when $\mathbf{x_0}$ was chosen randomly is depicted. Note that even with a large step of 0.8 the algorithm could not find a better estimation of the cost than the no-action case. In Fig. 14-B) the total cost of pollution in four different cases is represented: with no action, with random starting point, with the starting point being the heuristic solution, and respectively half of the heuristic solution. It can be observed that in the case of random selection there is no gain. In the last two cases there is a reduction of costs of about 10 to 13 times.

As a natural future development, we consider necessary to evaluate closely the impact of initial solution $\mathbf{x_0}$ and steps vector $\mathbf{h}$ on the quality of optimal solution found by the Nelder-Mead algorithm. As it is shown in this paper, for the reservoirs optimal operation, randomly generated starting points do not represent a suitable solution. The starting point must be chosen close to the heuristic solution presented in [9].

The most important, innovative results presented in this paper can be summarized as follows. The proposed method gives the possibility to quantify in monetary units (EUR) the economic damages produced by the accidental water pollution on Jijia River, defining a cost function that relies on MIKE11 simulation of pollutant transport. Further, the cost function is minimized by finding an optimal operation of the two reservoirs (Hălceni and Drăcşani) and releasing clean water used for dilution of the pollutant. Finally, it is possible to evaluate the cost in EUR of taking no action to mitigate pollution. As well, it is possible to evaluate the cost of dilution according to a heuristic, respectively the cost of making optimized dilution. By mitigating pollution with the optimized dilution, a total savings of about 1342000 EUR is obtained.

Because the problem of accidental water pollution is addressed in this paper, we could not have a real pollution case study. Therefore, the transport of pollutant on the river could not be experimented in reality, and within the scope of this work the pollutant propagation along the river is simulated using MIKE11 software. In this way, the graphic of the concentration of the pollutant (i.e. the *pollutograph*) in a given point on the river, $\mathbf{c_i(t)}$, was estimated. Based on this, the maximum value of this function was determined. However, for future validations, measuring real data when

Algorithm 1 Nelder Mead Simplex Algorithm

function SIMPLEXNELDERMEAD($x_0 \in R^n$, h, l, u) ▷ input: initial vertex x_0, steps vector h, lower/upper bounds constraints vectors.
  ▷ compute the remaining n simplex vertexes
  for $j \leftarrow 1; j \leq$n; $j \leftarrow j+1$ **do**
    $x_j \leftarrow x_0 + h_j * e_j$ ▷ e_j is the unit vector
    if VIOLATESCONSTRAINTS(x_j, l, u) **then**
      $f(x_j) \leftarrow$ APPLYPENALTIES(x_j) ▷ apply penalties
    end if
  end for
  while TERMINATIONTEST( )=False **do**
    ▷ determine the indices of the worst, second worst and best vertex in the current simplex (w,s,b)
    $w \leftarrow indexOf(max\{f_j = f(x_j)\})$
    if VIOLATESCONSTRAINTS(x_w, l, u) **then**
      $f(x_w) \leftarrow$ APPLYPENALTIES(x_w)
    end if
    $s \leftarrow indexOf(max\{f_j = f(x_j) | j \neq w\})$
    if VIOLATESCONSTRAINTS(x_s, l, u) **then**
      $f(x_s) \leftarrow$ APPLYPENALTIES(x_s)
    end if
    $b \leftarrow indexOf(min\{f_j = f(x_j)\})$
    if VIOLATESCONSTRAINTS(x_b, l, u) **then**
      $f(x_b) \leftarrow$ APPLYPENALTIES(x_b)
    end if
    $f_w = f(x_w)$
    $f_s = f(x_s)$
    $f_b = f(x_b)$
    $c \leftarrow \frac{1}{n} \times \sum_{j \neq w} x_j$ ▷ compute the centroid of the best side
    $x_r \leftarrow$ REFLECT(α, c, x_w) ▷ Apply the reflection transformation
    if $f_b \leq f(x_r) < f_s$ **then**
      Replace x_w with x_r in the simplex and go to 8)
    end if
    $x_e \leftarrow$ EXPAND(γ, c, x_r) ▷ Apply the expansion transformation
    if $f_e < f(x_r)$ **then**
      Replace x_w with x_e in the simplex and go to 8)
    else
      Replace x_w with x_r in the simplex and go to 8)
    end if
    if $f_s \leq f(x_r)$ **then**
      $x_c \leftarrow$ CONTRACT(β, c, x_r, x_w) ▷ Apply the contraction transformation
      if $f(x_c) \leq f(x_r) \vee f(x_c) < f_w$ **then**
        Replace x_w with x_c in the simplex and go to 8)
      else ▷ Apply the shrink transformation
        Keep only the best vertex x_b and compute n new vertexes by shrink(δ, x_b)
        go to 8)
      end if
    end if
  end while
  return x_b and $f(x_b)$ ▷ output: the minimum value and its x
end function
function REFLECT(α, c, x_w)
  return $c + \alpha \times (c - x_w)$
end function
function EXPAND(γ, c, x_r)
  return $c + \gamma \times (x_r - c)$
end function

Algorithm 2 *(Continued.)* Nelder Mead Simplex Algorithm

```
function CONTRACT(β, c, x_r, x_w)
    if f_s ≤ f(x_r) < f_w then
        return c + β × (x_r − c)
    else
        return c + β × (x_w − c)
    end if
end function
function SHRINK(δ, x_b)
    for j = 0; j ≤ n ∧ j ≠ b; j ← j + 1 do
        x_j ← x_b + δ(x_j − x_b)
    end for
    return the vector composed of x_j values
end function
function APPLYPENALTIES(x_j)
    return f(x_j) + p × [∑_{i=0}^{n1} (attempted_i − u_i)^2 + ∑_{i=0}^{n2} (l_i − attempted_i)^2]
end function
function VIOLATESCONSTRAINTS(x_j, l, u)
    for k ← 1; k ≤n; k ← k + 1 do
        if x_j[k] ∉ [l[k]; u[k]] then
            return true;
        end if
    end for
    return false;
end function
```

a future accident will happen, to be eventually compared with the simulated data, must be considered.

Based on the application of the proposed working methodology, some metrics related to the algorithm execution time may be evaluated. The Nelder-Mead algorithm execution time is negligible compared with the simulation time. Therefore one iteration for the optimization process which is composed of one MIKE11 simulation and the processing steps specific to Nelder-Mead takes about 10 minutes. Because of the necessary time to run a MIKE11 simulation, while completing the optimization algorithm requires in average 175 iterations, a total of about 30 hours time is needed just for experimenting for only one initial condition $\mathbf{x_0}$. Thus, to test 10 different $\mathbf{x_0}$ starting points, it will require about 13 days of continuously running MIKE11 simulation tool. Therefore, for future developments, we intend to study how to parallelize this process, if possible, on a cluster of high computing nodes.

The future research refers to *meta-optimization*, i.e. using other optimization methods to determine $\mathbf{x_0}$ and the vector $\mathbf{h}$ to tune the Nelder-Mead optimization. In this respect, Burmen *et al.* [51] propose to use a high performance parallel computing cluster composed of 100 CPU's to solve the computationally intensive problem of optimizing the parameters of Nelder-Mead simplex algorithm. The above mentioned research can be extended and applied in the meta-optimization problem of reservoir operation.

Another topic of further investigation is to study the impact of choosing different scalarization weights w_i in Eq. (10), regarding the Pareto solutions of the multi-objective optimization problem. Even if the Pareto solutions are ''equally good'', some of them could be preferred over others according to some user defined criteria. A possible goal of the research is to analyze if such user preferences can be defined in the context of the reservoir operation optimization and to identify the weights w_i according to these preferences.

APPENDIX
NELDER-MEAD SIMPLEX ALGORITHM

The Nelder-Mead algorithm [17] is one of the best algorithms available today for optimization of multidimensional unconstrained functions. The algorithm uses the notion of a *simplex* in R^n which is a convex hull of (n+1) points in R^n (for example, in 2-D space a triangle, in 3-D a tetrahedron). The method starts with a simplex and for each vertex $x_i \in R^n$ keeps the values $f(x_i)$. The algorithm executes a series of geometrical transformation on the simplex to decrease the $f(x_i)$ values. The transformations are: reflection, expansion, contraction, shrink. Three possible termination criteria can be used: *domain convergence* (when the vertexes of the simplex are very close one to another, so the simplex becomes very small), *function value convergence* (the values of the function in the vertexes are close enough) or *max evaluations* (a predefined limit of function f evaluation was reached).

The original algorithm is adapted to be suitable for constrained optimization by introducing a penalty for the case when one computed vertex (let's name it x^{new}) of the simplex

violates a constraint:

$$f(x^{new}) = f(x^{old}) + p \times [\sum_{i=0}^{n_1}(attempted_i - u_i)^2 + \sum_{i=0}^{n_2}(l_i - attempted_i)^2] \quad (25)$$

The new value is replaced with its old value plus additional penalty that is p times the sum of the squares of differences between the attempted value and the bounds. In Equation (25), n_1 represents the number of parameters violating upper bound constraints and n_2 the number of parameters violating lower bound constraints. This can be applied to constraints involving only one parameter or to multi-parameters constraints (in this later case, the $attempted_i$ is the sum of the parameters that is subject to a bound constraint).

The algorithm requires at input an initial estimation x_0 for a vertex and a vector of steps that will be used in line 3 to compute the remaining n vertexes. Together with x_0 they define the initial simplex. The $\mathbf{e_j}$ is the unit vector in R^n.

The main loop in lines 8-44 is executed until the termination condition is evaluated to *true*. Its goal is to replace the worst vertex with a new vertex computed as a result of one of the geometric transformations. Each vertex is verified if violates the constraints (lines 4,10,14,18). If yes, penalties are added to the current value of the objective function. At each iteration a new simplex, smaller than the previous one is obtained. Four main vertexes are used in the computations (lines 11,13,17,24): the worst vertex (corresponding to the highest value of function *f*), the second worst vertex and the best vertex (corresponding to the lowest value), and the centroid of the side that is opposed to the worst vertex. Only one transformation from four attempts is applied (lines 25-43), in the following order: reflection, expansion, contraction and shrink. If one transformation can not be applied, then the algorithm tries the next one. If none of the first three transformations can be applied, then shrink transformation computes n new vertexes, keeping only the best vertex. After the main loop exits, the best vertex is returned. The pseudo code for computing the new vertex for each of the four transformations is presented in lines 47-65.

The transformation are parametrized by the α, β, γ and δ which obeys to following constraints:

$$\alpha > 0; \quad 0 < \beta < 1; \ \gamma > 1; \ \gamma > \alpha; \ 0 < \delta < 1$$

The values of these parameters that are often used in many implementations are: $\alpha = 1$, $\beta = 0.5$, $\gamma = 2$, $\delta = 0.5$.

REFERENCES

[1] M. Amirkhani, O. Bozorg-Haddad, E. Fallah-Mehdipour, and H. A. Loáiciga, "Multiobjective reservoir operation for water quality optimization," *J. Irrigation Drainage Eng.*, vol. 142, no. 12, p. 04016065, 2016.

[2] E. Shirangi, R. Kerachian, and M. S. Bajestan, "A simplified model for reservoir operation considering the water quality issues: Application of the Young conflict resolution theory," *Environ. Monitor. Assessment*, vol. 146, nos. 1–3, pp. 77–89, 2008.

[3] P. Chaves and T. Kojiri, "Deriving reservoir operational strategies considering water quantity and quality objectives by stochastic fuzzy neural networks," *Adv. Water Resour.*, vol. 30, no. 5, pp. 1329–1341, 2007.

[4] N. A. Jaworski, W. J. Weber, and R. A. Deininger, "Optimal reservoir releases for water quality control," *J. Sanitary Eng. Divis.*, vol. 96, no. 3, pp. 727–742, 1970.

[5] A. Dhar and B. Datta, "Optimal operation of reservoirs for downstream water quality control using linked simulation optimization," *Hydrol. Process. Int. J.*, vol. 22, no. 6, pp. 842–853, 2008.

[6] Y. Wang, W. Zhang, Y. Zhao, H. Peng, and Y. Shi, "Modelling water quality and quantity with the influence of inter-basin water diversion projects and cascade reservoirs in the middle-lower Hanjiang river," *J. Hydrol.*, vol. 541, pp. 1348–1362, Oct. 2016.

[7] C. Cervellera, V. C. Chen, and A. Wen, "Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization," *Eur. J. Oper. Res.*, vol. 171, no. 3, pp. 1139–1151, 2006.

[8] K. Voudouris, M. Polemio, N. Kazakis, and A. Sifaleras, "Modelling water quality and quantity with the influence of inter-basin water diversion projects and cascade reservoirs in the middle-lower Hanjiang river," in *Regional Development: Concepts, Methodologies, Tools and Applications*. Hershey, PA, USA: IGI Global, 2012, pp. 454–466.

[9] S. N. Ciolofan, A. Draghia, R. Drobot, M. Mocanu, and V. Cristea, "Decision support tool for accidental pollution management," *Environ. Sci. Pollut. Res.*, vol. 25, no. 7, pp. 7090–7097, 2018.

[10] *MIKE11 Reference Manual*. Accessed: Nov. 7, 2018. [Online]. Available: http://euroaquae.tu-cottbus.de/hydroweb/Platform/Notes/Mike11_Reference.pdf

[11] C. Xu, W. Li, L. Fen, S. Yao, and Z. Lei, "Application of MIKE11 model in water quality prediction of Hunhe river basin," *Water Resour. Power*, no. 6, p. 015, 2013.

[12] J. Liang, Q. Yang, T. Sun, J. Martin, H. Sun, and L. Li, "MIKE 11 model-based water quality model as a tool for the evaluation of water quality management plans," *J. Water Supply, Res. Technol.-Aqua*, vol. 64, no. 6, pp. 708–718, 2015.

[13] E. K. Kanda, J. R. Kosgei, and E. C. Kipkorir, "Simulation of organic carbon loading using MIKE 11 model: A case of River Nzoia, Kenya," *Water Pract. Technol.*, vol. 10, no. 2, pp. 298–304, 2015.

[14] J. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*, vol. 97. Boston, MA, USA: Springer, 2005.

[15] W. Forst and D. Hoffmann, *Optimization—Theory and Practice*. New York, NY, USA: Springer, 2010.

[16] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: A review of algorithms and comparison of software implementations," *J. Global Optim.*, vol. 56, no. 3, pp. 1247–1293, 2013.

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

[18] R. Hooke and T. A. Jeeves, "'Direct search' solution of numerical and statistical problems," *J. ACM*, vol. 8, no. 2, pp. 212–229, 1961.

[19] R. M. Lewis and V. Torczon, "Pattern search algorithms for bound constrained minimization," *SIAM J. Optim.*, vol. 9, no. 4, pp. 1082–1099, 1999.

[20] E. D. Dolan, R. M. Lewis, and V. Torczon, "On the local convergence of pattern search," *SIAM J. Optim.*, vol. 14, no. 2, pp. 567–583, 2003.

[21] M. J. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*. Dordrecht, The Netherlands: Springer, 1994, pp. 51–67.

[22] M. J. Powell, "The NEWUOA software for unconstrained optimization without derivatives," in *Large-Scale Nonlinear Optimization. Nonconvex Optimization and its Applications*. Boston, MA, USA: Springer, 2006, pp. 255–297.

[23] P. Gilmore and C. T. Kelley, "An implicit filtering algorithm for optimization of functions with many local minima," *SIAM J. Optim.*, vol. 5, no. 2, pp. 269–285, 1995.

[24] J. D. Pintér, "Global optimization: Software, test problems, and applications," in *Handbook of Global Optimization*. Springer, 2002, pp. 515–569.

[25] W. Huyer and A. Neumaier, "Global optimization by multilevel coordinate search," *J. Global Optim.*, vol. 14, no. 4, pp. 331–355, Jun. 1999.

[26] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Stat. Sci.*, vol. 4, no. 4, pp. 409–423, 1989.

[27] M. Powell, "Recent research at Cambridge on radial basis functions," in *Proc. 2nd Int. Dortmunt Meeting Approximation Theory*, 1998, pp. 215–232.
[28] H.-M. Gutmann, "A radial basis function method for global optimization," *J. Global Optim.*, vol. 19, no. 3, pp. 201–227, 2001.
[29] A. Boneh and A. Golan, "Constraints redundancy and feasible region boundedness by random feasible point generator (RFPG)," in *Proc. 3rd Eur. Congr. Oper. Res. (EURO)*, Amsterdam, The Netherlands, 1979, p. 22.
[30] R. L. Smith, "Efficient monte carlo procedures for generating points uniformly distributed over bounded regions," *Oper. Res.*, vol. 32, no. 6, pp. 1296–1308, 1984.
[31] C. J. Bélisle, H. E. Romeijn, and R. L. Smith, "Hit-and-run algorithms for generating multivariate distributions," *Math. Oper. Res.*, vol. 18, no. 2, pp. 255–266, 1993.
[32] J. Genlin, "Survey on genetic algorithm," *Comput. Appl. Softw.*, vol. 21, no. 2, pp. 69–73, 2004.
[33] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. Int. Conf. Parallel Problem Solving From Nature*. Berlin, Germany: Springer, 2000, pp. 849–858.
[34] D. Karaboga and B. Akay, "A survey: Algorithms simulating bee swarm intelligence," *Artif. Intell. Rev.*, vol. 31, nos. 1–4, pp. 61–85, 2009.
[35] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. Hoboken, NJ, USA: Wiley, 2006.
[36] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Boston, MA, USA: Springer, 2011, pp. 760–766.
[37] S.-K. S. Fan and E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *Eur. J. Oper. Res.*, vol. 181, no. 2, pp. 527–548, 2007.
[38] A. I. F. Vaz and L. N. Vicente, "A particle swarm pattern search method for bound constrained global optimization," *J. Global Optim.*, vol. 39, no. 2, pp. 197–219, Oct. 2007.
[39] N. Mladenović, M. Dražić, V. Kovačevic-Vujčić, and M. Čangalović, "General variable neighborhood search for the continuous optimization," *Eur. J. Oper. Res.*, vol. 191, no. 3, pp. 753–770, 2008.
[40] R. Chelouah and P. Siarry, "Tabu search applied to global optimization," *Eur. J. Oper. Res.*, vol. 123, no. 2, pp. 256–270, 2000.
[41] P. Siarry and G. Berthiau, "Fitting of Tabu search to optimize functions of continuous variables," *Int. J. Numer. Methods Eng.*, vol. 40, no. 13, pp. 2449–2457, 1997.
[42] Y. Tenne, "Machine-learning in optimization of expensive black-box functions," *Int. J. Appl. Math. Comput. Sci.*, vol. 27, no. 1, pp. 105–118, 2017.
[43] M. Caramia and P. Dell'Olmo, *Multi-Objective Management in Freight Logistics: Increasing Capacity, Service Level and Safety With Optimization Algorithms*. London, U.K.: Springer, 2008.
[44] X. Zhang, "Assessing the economic costs of water pollution in the Yangtze river, China," *J. Ocean Coastal Econ.*, vol. 2014, no. 1, p. 2, 2014.
[45] Z. Faqing, G. Guanmin, L. Guoti, L. Jinbin, and Q. Gongyi, "Economic loss from water pollution in Donghu Lake," *Acta Scientiae Circumstantiae*, vol. 13, no. 2, pp. P214–222, 1993.
[46] (Dec. 2018). *Advanced Sensor Technologies on Water Monitoring and Modeling*. [Online]. Available: http://ieeeaccess.ieee.org/special-sections/advanced-sensor-technologies-on-water-monitoring-and-modeling/
[47] *Sendai Framework for Disaster Risk Reduction 2015–2030*, United Nations, Third UN World Conf. Disaster Risk Reduction, Sendai, Japan, Mar. 2015.
[48] C. G. Chiru, M. I. Mocanu, M. Drăgoicea, and A. D. Ioniţă, "Digital services development using statistics tools to emphasize pollution phenomena," in *Exploring Service Science*, S. Za, M. Drăgoicea, and M. Cavallari, Eds. Cham, Switzerland: Springer, 2017, pp. 370–382.
[49] *Hyogo Framework for Action (HFA) 2005—2015*, United Nations Office for Disaster Risk Reduction, Geneva, Switzerland, 2005. [Online]. Available: https://www.unisdr.org/we/coordinate/hfa
[50] M. Drăgoicea, N. G. Badr, J. E. Falcão Cunha, and V. E. Oltean, "From data to service intelligence: Exploring public safety as a service," in *Exploring Service Science*, G. Satzger, L. Patrício, M. Zaki, N. Kühl, and P. Hottum, Eds. Cham, Switzerland: Springer, 2018, pp. 344–357.
[51] A. Burmen, T. Tuma, and I. Fajfar, "Meta-optimisation on a high-performance computing system," *Elektrotehniski Vestnik*, vol. 79, no. 5, pp. 231–236, 2012.

SORIN N. CIOLOFAN received the B.S., M.S., and Ph.D. degrees in computer science from the Faculty of Automatic Control and Computers, University Politehnica of Bucharest, in 2003, 2004, and 2017, respectively. From 2003 to 2012, he acted in different positions as a Software Engineer, a Research and Development Engineer, a Solution Architect, and a Senior Software Engineer with IBM Ireland–Dublin Software Laboratory, Foundation of Research and Technology Hellas–Institute of Computer Science, Heraklion, Greece, Forte Business Services, Bucharest, Romania, and Qualisoft, Bucharest.

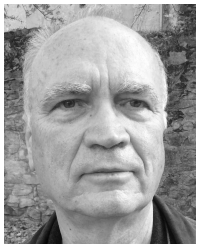
He is currently a Lecturer with the Department of Computer Science, University Politehnica of Bucharest. His main research interests are cloud computing, big data, information systems, cyber-physical systems, and modeling and simulation-based systems engineering.

GHEORGHE MILITARU received the B.S. degree in technological physics from the Faculty of Physics, University of Bucharest, in 1984, the Ph.D. degree in management and industrial engineering from the University Politehnica of Bucharest, Romania, in 1996, and the B.S. degree in finance from the Bucharest University of Economics Studies in 2005. He was a Certified Business Counselor in Small Business with Washington State University, USA, in 1997.

He is currently a Full Professor with the Faculty of Entrepreneurship, Business Engineering and Management, University Politehnica of Bucharest. His research interests include service management, financial management, production management and engineering, information systems for management, digital marketing, and technology entrepreneurship. He is an active member of Romanian and international professional associations in management and engineering.

AURELIAN DRAGHIA received the bachelor's degree and the M.S. degree in hydraulic engineering from the Faculty of Hydrotechnics, Technical University of Civil Engineering Bucharest, in 2010 and 2012, respectively. He is currently a Research Assistant with the Faculty of Hydrotechnics, Technical University of Civil Engineering Bucharest. He is an expert in hydrology and a Co-Founder of the ROUA SOFT S.R.L., a company specialized in developing specific information-based application in the domain of hydrology.

He is a MIKE11 Professional Engineer. He was involved in several projects, such as the international project "Danube Floodrisk" from 2010 to 2012 (statistical analyzing of computing maximum discharge and volumes for flood waves along the Danube and the main tributaries in Romania, hydraulic modeling and analyzing of different flood scenarios). At national level, he contributed actively in the developing of National Normative "Setting the maximum discharges and volumes of synthetic floods for designing the hydraulic structures with water retention."



RADU DROBOT received the B.S. degree in civil engineering from the Technical University of Civil Engineering Bucharest, Romania, in 1970, the B.S. degree in mathematics from the University of Bucharest, Romania, in 1975, and the Ph.D. degree in integrated management of water resources from the Technical University of Civil Engineering Bucharest in 1981. He is currently a Full Professor with the Faculty of Hydrotechnics, Technical University of Civil Engineering Bucharest.

His research interests include statistical processing of hydrological data, optimization procedures, hydrological and hydro-geological modeling, and integrated management of water resources with a special focus on flood risk management.

MONICA DRĂGOICEA (M'18) received the B.S. degree in automatic control from the Faculty of Automatic Control and Computers, University Politehnica of Bucharest, in 1993, the M.S. degree in engineering management from Technische Universität Wien, Austria, and Oakland University, Rochester, MI, USA, in 1999, and the Ph.D. degree in automatic control from the University Politehnica of Bucharest in 2000. She is currently a Full Professor with the Faculty of Automatic Control and Computers, University Politehnica of Bucharest.

She has been involved with theoretical and experimental work in software and systems engineering for the past 20 years. Her research interests include modeling and simulation-based systems engineering, real-time systems, service systems engineering, digital design of services, and computational intelligence. She is a member of the IEEE Systems, Man, and Cybernetics Society, the International Society of Service Innovation Professionals, and The Society for Modeling & Simulation International.