



Optimization via simulation of catchment basin management using a discrete-event approach

J F Santucci and L Capocchi

Abstract

This paper deals with discrete-event simulation-based optimization of a catchment basin management. The integration of optimization techniques into modeling and simulation relies on the evolution of the studied model using decisions based on previous simulation results. Three different categories of optimization via simulation methods can be found in the literature: ranking-and-selection methods, discrete and continuous optimization via simulation methods. In this paper, we explain how an iterative process allows us to integrate continuous optimization via simulation algorithms into a discrete-event simulation in order to optimize the management of a catchment basin. We have performed a comparison between optimization methods belonging to the three categories of optimization via simulation methods when they are applied to the management of a catchment basin. An analysis of the comparison allows us to point out the benefits of continuous optimization via simulation methods compared with rank-and-selection and discrete optimization via simulation methods in the case of the optimization of a catchment basin management.

Keywords

optimization methods, discrete-event simulation, discrete-time simulation, simulation software, DEVS, OvS, water management

1. Introduction

Modeling and simulation (M&S) are widely used in research and industry in order to study the behavior of complex systems. These studies include tasks such as: planning, prediction, forecast, computer-aided design, etc. When people want to improve the performance of a studied system, the M&S phases are integrated with an optimization technique. The integration of optimization methodologies into M&S relies on the evolution of the model using decisions based on previous simulation results.^{1–3} Usually, the evolution is performed manually by a specialist of the domain. Therefore, it is natural to try to find the set of parameters that optimizes the system performance. This process is called optimization via simulation (OvS). Depending on the format of the parameters (also called decision variables), three main categories of OvS methods may be highlighted: (i) rank-and-selection category (R&S);^{4,5} (ii) discrete OvS category (DOvS);^{6,7} (iii) continuous OvS category (COvS).^{8–10} R&S and DOvS deal with discrete/integer-valued decision variables while COvS optimize continuous decision variables. Furthermore, when the number of feasible solutions is

small, R&S is often used because it provides a strong optimality guarantee. When solution space is large, DOvS/COvS can be used. We propose to use a COvS method based on a discrete-event (DE) modeling approach applied on a problem dealing with a large solution space.

In this paper we are interested in proposing an optimized management of a catchment basin involving dams, power station, pump station, valves, conducts, etc. This management optimization is performed by an automatic integration of optimization techniques into a DE modeling using the DEVS (Discrete Event system Specification)¹¹ formalism. This paper presents in detail how the behavior of the management of such a complex catchment basin is realized using the DEVS modeling leading to a DE simulation. It describes then how an integration of a COvS optimization technique into the modeling scheme allows

SPE UMR CNRS 6134 Laboratory, University of Corsica, Corte, France

Corresponding author:

J F Santucci, SPE UMR CNRS 6134 Laboratory, University of Corsica, Quartier Grimaldi, 20250 Corte, France.
Email: santucci@univ-corse.fr

to automatically obtain an optimized management of the basin. Three different kinds of optimization methods have been implemented based on the three OvS categories pointed out previously. The R&S method optimization has already been presented by Santucci and Capocchi,¹² while the DOvS optimization method has described by Capocchi et al.,¹³ Farges et al.,¹⁴ and Capocchi and Santucci.¹⁵

In this paper we detail the optimization of the catchment basin management using a COvS method based on a DE simulation approach. The optimized system has been defined with the DEVS formalism and implemented using DEVSIMPy framework.¹⁶

The rest of the paper is organized as follows. The next section presents in detail the background of this study. The OvS domain is introduced and the main commercial simulation tools including an optimization module are presented. Section 2 also introduces the DEVS formalism and the DEVSIMPy framework and why the DEVS formalism has been chosen. Section 3 introduces the case study: the management of a catchment basin. The problem associated with the management of a catchment basin is clearly stated. Then the DEVS-based DE modeling of the case study is detailed. In Section 4 we describe how we have realized the optimization of the catchment basin management by developing a COvS approach. The DEVSIMPy implementation and the results based on the proposed optimization method are presented. The convergence of the proposed method is also illustrated through the obtained results. Section 5 is dedicated to the analysis and discussion of the proposed approach. We compare the results introduced in Section 4 with results obtained when applying R&S and DOvS approaches. Finally, conclusions are given in the last section.

2. Background

In this section we introduce the OvS domain. We point out the main categories of methods which can be found under the name of OvS and we introduce a set of existing simulation tools which include an optimization technique. We also introduce the DEVS formalism which has been widely used within M&S to design, verify, and implement complex reactive systems.

2.1. Optimization via simulation

OvS is a structured approach to determine optimal settings for input parameters (i.e. the best system design), where optimality is measured by a function of output variables associated to a simulation model.¹⁷ One of the main features of simulation is that one can change the parameters of a simulation model easily and try to observe the system performance under different sets of parameters. Therefore, it is natural to try and to find a set of parameters that optimizes the system. This set is determined using an optimization model after evaluating an objective function.

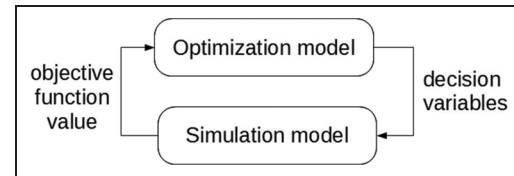


Figure 1. Coordination between the optimization and simulation models.

Figure 1 depicts the coordination between the optimization model and the simulation model. Recent comprehensive reviews of the research and practice of OvS are available in the literature.^{18,19} Depending on the format of the decision variables vector x , we may divide OvS problems into three categories.²⁰

- R&S category: the numerical decision variable set has a small number of potential solutions. Then, we may simulate all solutions and select the best among them. This problem is known as the ranking-and-selection (R&S) problem.²¹ R&S procedures evaluate exhaustively all members from a given (fixed and finite) set of alternatives.⁵
- DOvS category: the set of decision variables is discrete and integer ordered. This problem is known as the DOvS problem. A set of algorithms based on the principles of branch and bound (BB)⁷ approach can be exploited when performing DOvS approach. One of the best known is the dichotomy algorithm. BB algorithms are based on a systematic enumeration of all candidate solutions, where large subsets of fruitless candidates are discarded, by using upper and lower estimated bounds of the quantity being optimized. However, even if it is known that locally convergence DOvS algorithms tend to perform better than globally convergent algorithms, new algorithms combining local and global convergences have been proposed in the recent past.^{22–24}
- COvS category: the set of decision variables is a vector of continuous decision variables. This problem is known as the COvS problem. In this category the decision variables are continuous and the definition set of these variables is uncountable and infinite. This optimization category has been widely studied and several methods can be pointed out: (i) stochastic approximation (SA);⁹ (ii) metamodel-based optimization algorithm.²⁵

A set of commercial simulation tools have integrated OvS software modules.

- Arena simulation²⁶ is the most widely used tools for simulation of DE systems. It is an integrated graphical simulation environment including

resources for modeling, design, process visualization, statistical analysis. It integrates the OptQuest for Arena package,²⁷ which is an optimization tool that is customized for analyzing and optimizing the results of simulation runs, conducted in Arena.

- Witness²⁸ is a process simulation software commonly used for both educational and business purposes, and it is one of the most suitable for simulating plant layouts. The Witness simulation package offers an optimization suite within its experimental framework.
- Anylogic²⁹ is a simulation tool that supports all of the most common simulation methodologies: system dynamics, DE, and agent-based modeling. AnyLogic optimization is built on top of the OptQuest Optimization Engine.
- Simio³⁰ is a powerful DE simulation platform. The modeling aspects are easy to use thanks to a library of intelligent 3D objects. With Simio, complex systems can intuitively be modeled. OptQuest for Simio is available as an add-on to the standard Simio products.
- AutoMod³¹ is used to create DE simulation models of manufacturing systems, assembly lines, warehouses, distribution centers, airports, and equipments. It is used to: (i) run “what-if” scenarios in a simulation environment to predict results; (ii) analyze and optimize alternate system design concepts based on the AutoStat optimization package.

A review of the optimization packages found in commercial simulation software (AutoStat, OptQuest and the Witness optimizer) reveals that these packages are based on multi-objective optimization routine. AutoStat approaches the multi-objective optimization problem using the classical method in which multiple objectives are aggregated to form a single objective using a weight-vector. The drawback of this method is that for different decision scenarios, different weight-vectors have to be used and the same problem has to be solved repetitively. OptQuest also employed a similar approach. In a same way, Witness optimizer uses simulated annealing (SA) and elements of Tabu search are incorporated into the SA process. We can point out that most of the commercial OvS packages implement robust meta-heuristics. For instance, AutoStat uses evolution strategies; OptQuest uses scatter search, Tabu search and artificial neural networks and Witness optimizer uses SA and Tabu search. These meta-heuristics are often robust and perform well on deterministic problems. However they may face problems when random aspects are present.

2.2. DEVS formalism and tools

The DEVS formalism introduced by Zeigler et al.¹¹ provides a mean of specifying a mathematical object called a

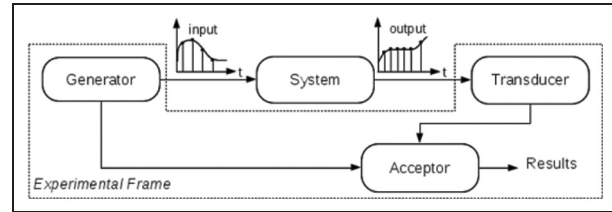


Figure 2. The DEVS experimental frame.

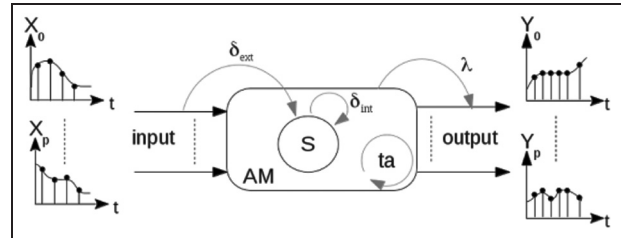


Figure 3. The DEVS atomic model.

system. Basically, a system has a time base, inputs, states, outputs, and functions for determining next states and outputs given current states and inputs. DEVS provides a formal representation of DE systems capable of mathematical manipulation just as differential equations serve this role. Furthermore, by allowing an explicit separation between the modeling phase and simulation phase, the DEVS formalism is an efficient way to perform simulation of complex systems using a computer.

The DEVS formalism procures an Experimental Frame (EF)³² that specifies conditions under which the system is experimented and observed.

As illustrated in Figure 2, an EF has three components: a Generator in charge of generating a set of inputs segments for the system; a Transducer in charge of observing and analyzing the output segments of the system; and an Acceptor in charge of selecting the desired data of the system and verifying whether the experimental conditions are fulfilled.

In the DEVS formalism, in order to model a system, one must specify: (i) basic models from which larger ones are built and (ii) how these models are coupled together in hierarchical order. An atomic model allows specifying the behavior of a basic element of a given system. Coupling between different atomic models can be defined using a coupled model (CM). An atomic DEVS (AM in Figure 3) model can be considered as an automaton with a set of states S and transition functions allowing the state change when an event occurs or not.

When no events occur, the state of the atomic model can be changed by an internal transition function (δ_{int}). When an external event X_i occurs, the atomic model can intercept it and change its state by applying an external transition

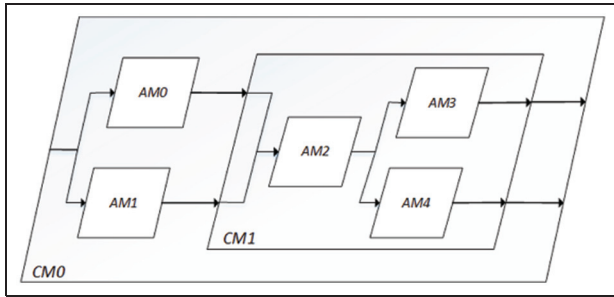


Figure 4. Hierarchical modeling using DEVS coupled models.

function (δ_{ext}). The lifetime of a state is determined by a time advance function (t_a). Each state change can produce output event Y_i via an output function (λ).

A coupled model tells us how to couple (connect) several component models together to form a new model. This latter model can itself be employed as a component in a larger coupled model, thus giving rise to hierarchical construction.

Figure 4 depicts an example of the hierarchy between sub-models of a system. The CM0 coupled model represents the system at the highest possible hierarchical level. It has two output ports OUT0 and OUT1, one input port IN and it contains two atomic models AM0 and AM1 but also one coupled model CM1. The two atomic models AM0 and AM1 are coupled with an input external coupling relation with CM0 and with an internal coupling relation with CM1. The output external coupling relation occurs between the two output ports of CM1 and the two output ports OUT0 and OUT1.

A simulator is associated with the DEVS formalism in order to exercise instructions of the atomic model to actually generate its behavior. The architecture of a DEVS simulation system is derived from the abstract simulator concepts¹¹ associated with the hierarchical and modular DEVS formalism. One of the advantages of the DEVS formalism is that the simulator is generated automatically from the model.

There are many tools which provide a user interface dedicated to help the user to define DEVS models and to perform simulations. A non-exhaustive list can be provided as follows: PowerDEVS,³³ DEVSIM++,³⁴ DEVSJAVA,³⁵ VLE,³⁶ DEVSIMPy,¹⁶ CD++Builder,³⁷ ATOM3,³⁸ MS4Me,³⁹ etc. Special attention will be given to DEVSIMPy (stands for DEVS simulator in Python language) which is a collaborative M&S software. It is used in order to model and simulate complex systems based on the DEVS formalism with the Python programming language. In particular, DEVSIMPy¹⁶ is an open-source project with the aim of providing a GUI for the M&S of PyDEVS³⁸ models. The basic idea behind DEVSIMPy is to wrap the PyDEVS API with a GUI allowing significant

simplification of handling PyDEVS models (such as the coupling between models or their storage).

Figure 5 gives an example of the window interface. We can pointed out that the window is split into two parts: (i) the left part (circle 1 in Figure 5) allows the user to visualize the classes of DEVS models which can be instantiated using a drag-and-drop; (ii) the right part (circle 2 in Figure 5) is dedicated to the design of DEVS coupled models (also called diagrams in the following) with a simple drag-and-drop of classes belonging to the left part of the window. The created coupled model can be simulated and saved under a specific format allowing it to be easily reused and shared.

2.3. Why DEVS formalism for OvS approach?

A set of different formalisms can be used when modeling complex systems. We can mention DAE, bond graphs, Forrester system dynamics, Petri nets, finite state automata, DEVS, state charts, queueing networks, etc. The choice of appropriate formalisms depends on criteria such as the application domain, the background of the modeler, the goals, and the available computational resources, etc.⁴⁰ The DEVS formalism has been chosen in the framework of this optimization work for at least two reasons: (1) due to the DE-based specification given by hydrologic experts of the CAW agency, the choice of a DE formalism is obvious even though a traditional approach would have been based on a discrete-time (DT) formalism; (2) it has been highlighted⁴⁰ that it is possible to map all formalisms, including approximations of continuous ones, onto DEVS. The DEVS formalism framework of M&S proposed by Zeigler et al.¹¹ offers concepts that are important for the integration of optimization methods into a DE simulation tool: (i) there is an explicit separation between the model and the context under which it is experimented with; (ii) there is a separation between the simulator and the simulation model. This DEVS formalism framework is therefore a good basis for the detailed formalization of the structural relationships between simulation and optimization techniques.

Moreover, we chose to provide a DE OvS approach using the DEVSIMPy tool for solving the problem of catchment basin management optimization. We have already pointed out the interest in choosing the DEVS formalism and we have decided to use the DEVSIMPy environment in order to integrate a COVS method for the following reasons: (i) the simulation model had already been realized using the DEVSIMPy framework, the basic model which is used as a basis for optimization has been realized using the DEVSIMPy environment in collaboration with the hydrologic experts of the CAW agency; (ii) DEVSIMPy is supported by the team of the SPE UMR CNRS 6134 laboratory.

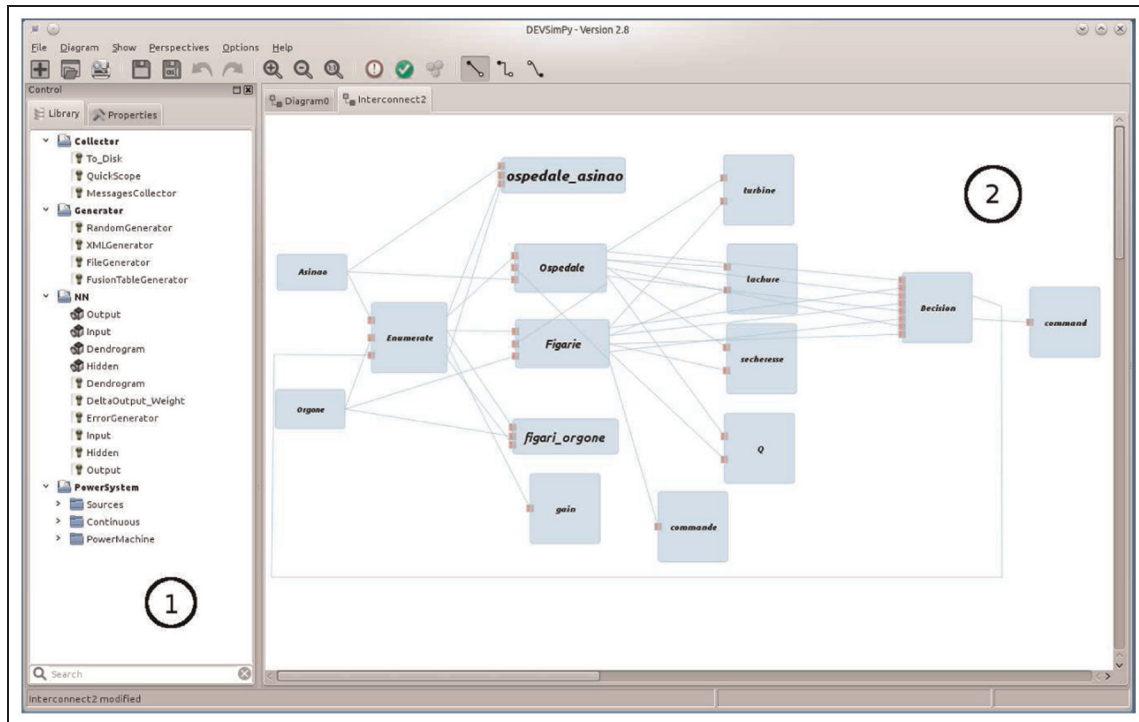


Figure 5. The DEVSimPy general interface.

3. Case study: catchment basin management

3.1. Problematic

Water is the vital resource to support life on the Island of Corsica (an island in the Mediterranean area). Unfortunately, it is not evenly distributed over the island by season or location. Some parts of the island are prone to drought making water a scarce and precious commodity, while in other parts of the island it appears in raging torrents causing floods and loss of life and property. The Corsican Water Agency (CWA) is a governmental institution in charge of the distribution and quality of water resources and water bodies, including manmade water infrastructures. CWA manages a complex network of dams, storage and pipelines to provide an adequate supply of water to its customers. Furthermore, in addition to water supply the benefits of this management include also hydropower production. The difficulty when managing a water network^{41–43} involving dams, power stations, pump stations, pipelines, valves is to ensure the water distribution to consumers while the production of electricity using power stations is the highest possible and the use of pump stations is as low as possible. It is the case actually in the southeast of Corsica where a water network involves two interconnected dams (Ospedale and Figari), a power station (upstream to the Ospedale dam), a pump station (used in order to send the water taken from Figari dam to the

consumers) and a set of valves allowing the water to be sent from the Ospedale dam to the Figari dam or the consumers (Figure 1). We have to point out that the south-east region of Corsica has the following features: (i) a lot of rainfall in winter while a drought period in summer; (ii) the consumer population increases by a factor of 10 in summer season (due to tourism) compared to the winter season. Because of this dichotomy between summer and winter, two different behaviors for the management of the network have been defined by the CWA experts.

- The winter behavior: since the rainfalls are heavy and the population is low, only the Ospedale dam is used fulfill the water supply for the consumers (since the use of the Figari dam requires electricity consumption due to the pump station associated with the dam) and the power station is able to produce electricity when the Ospedale dam is quite full (since the water used by the power station is lost for water supply).
- The summer behavior: the power station is no longer used; the Ospedale dam is first used to fulfill the consumer water supply; when it is quite empty the pump station is then used in order to send water taken from the Figari dam to the consumer.

The hydrology experts belonging to the CWA have defined a management strategy based on the definition of two dates

(let us call them $d1$ and $d2$) allowing to perform the winter behavior from date $d2$ to date $d1$ and then switch to the summer behavior from $d1$ to $d2$. However, the difficulty for the experts is to find the best dates ($d1$ and $d2$) as well as the best amount of turbine water allowing the following three constraints to be respected: (i) to respond to the water demand of the consumers all along the year; (ii) to use as much as possible the power station associated with the Ospedale dam; (iii) to use as less as possible the pump station associated with the Figari dam.

In this paper, we are interested in the management of the hydrologic behavior of a catchment basin. The management consists in proposing a software approach based on DE simulation in order to find the optimized dates allowing to respond to the three main constraints: (i) ensuring water supply all along the year; (ii) optimal use of the power station; (iii) minimal use of the pump station.

Three different kinds of optimization methods have been implemented based on the three OvS categories pointed out in Section 2: R&S, DOvS and COvS. The two first categories have been presented respectively by Santucci and Capocchi¹² for R&S and by Capocchi et al.¹³ for the DOvS method with a BB algorithm based on the dichotomy algorithm. Each one of these three optimization methods leans on three different models of a catchment basin behavior that have been defined and implemented using DEVSimPy. In the next section we introduced the DE modeling scheme that have been defined in order to model the management of a catchment basin and that will be used by a COvS method presented in Section 4. An analysis of results obtained using the three kinds of optimization is given in Section 5.2.

3.2. DEVS-based DE modeling

In adopting a DE modeling approach, the study case can be modeled and managed differently from the DT modeling scheme introduced by Santucci and Capocchi¹² and Capocchi et al.¹³ Indeed, with the DT modeling, the system behavior is overall considered for the definition of a fixed simulation time step equal to 1 week. With DE modeling, the system is seen as a set of components (atomic or coupled DEVS models) having a set of states with finite lifetime. In such a case the simulation time step is not fixed and depends on the activity of the models. For the simulation model, a dam model has been defined and it is very important as it will play a primordial role for the simulation step definition. Furthermore, the model design depends on the DE or DT modeling scheme. The DE modeling relies only on the definition of one atomic model (dam model) although the DT modeling involves the definition of several atomic models (generators, dams, power station, managers models). Another interesting approach is to implement the dam model using ordinary differential equations (ODEs) with the quantized state systems

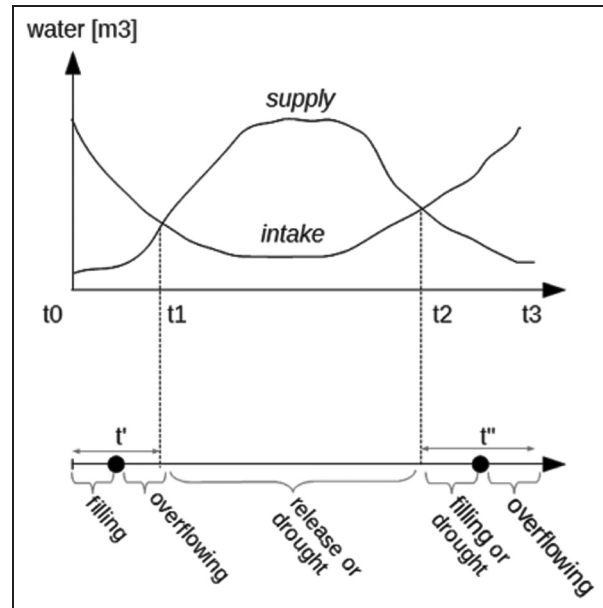


Figure 6. DAM water profile for the supply and the intake.

(QSS)⁴⁴ method. The QSS method allows the ODE-based modeling in the DEVS framework by proposing an integrator DEVS model. The implementation of the integrator model requires the definition of a quantum and an order for the algorithm of the integration (QSS{1,2,3} for the first, second or third order). Consequently, it is possible to implement a dam model described by ODEs using a QSS integrator which is setup correctly. In this paper, we propose to use a different approach consisting of implementing a simple model of the dam just by knowing the textual specifications given by the CWA experts.⁴⁵

Assumptions Before defining the dam atomic model with the help of the DEVS formalism, some assumptions have been raised. In this paper, we are not dealing with traditional approaches based on waterflows which should have been expressed in cubic meters/{some unit of time}. Instead, our approach relies on the use of quantities of water expressed in cubic meters. In the following, the water quantities are expressed in cubic meters in both figures and tables.

Figure 6 represents the curves of the water supply and the water intake for a dam over the 52 weeks. If the system is analyzed between a time period going from $t0$ to $t3$, the two dates $t1$ and $t2$ are identified by the two intersections between the curves (intake and supply in Figure 6). The approach presented in this paper depends on these two last dates. If the curves are not crossing, the approach cannot be applied. However, in our case, it is most unlikely that the two curves do not cross because of the nature of the two rivers Asinao and Orgone.

Due to the previous assumptions and Figure 6, the behavior of the dam has been translated in this way.

- Depending on its initial level of the water, the dam will be in a filling phase until t' . Then, from t' to t_1 , the dam may be in an overflowing phase. It is possible that t' is equal to t_1 and then the overflowing phase does not exist. In the same way, if t is close to t_0 , the filling phase could not occur.
- Between t_1 and t_2 , the dam is generally in a discharge phase (also called release). Of course, it can be in a drought phase if the supply curve is too high according to the intake curve.
- After t_2 , the dam is in a filling or drought phase until t'' . Then, from t'' to t_3 , the dam goes in an overflowing phase. The same previous remarks for t' can be stated for t'' .

DEVS modeling Due to the adopted DEVS-based DE modeling and previous assumptions, the dam model can be summarized as in Figure 7.

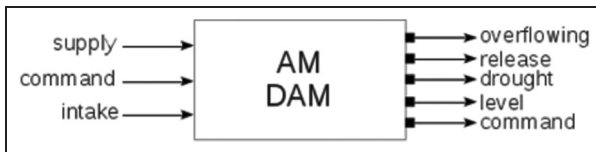


Figure 7. The DAM atomic model.

The DAM atomic model has three input ports: *supply*, *command* and *intake*. The port denoted *supply* allows to receive the curve (according to COVs approach) of the water consumption of the population. The port denoted *command* can receive a positive (respectively negative) value corresponding to a water quantity for the filling (respectively the discharge) of the dam. Finally, the port *intake* allows to receive the curve corresponding to the water intake of the dam.

Five output ports are planned in order to: (1) send the overflow value for the *overflowing* port, (2) send the discharge value for the *release* port, (3) send the Boolean value that informs if the dam is drought for the *drought* port, (4) send the water level value of the dam for the *level* port and, finally, (5) send the command value (positive or negative) for the *command* port. Concerning the output *command* port, positive value is used to inform that a quantity of water is available and negative value to inform that the dam needs a quantity of water.

Figure 8 depicts a simplified state graph of the DAM model. This representation is useful to understand the behavior of the dam without getting into the textual DEVS specifications. The dam behavior can be represented using eight states (circles in the Figure 8). A state is a combination of a status which can be *Filling*, *Overflowing*, *Release*

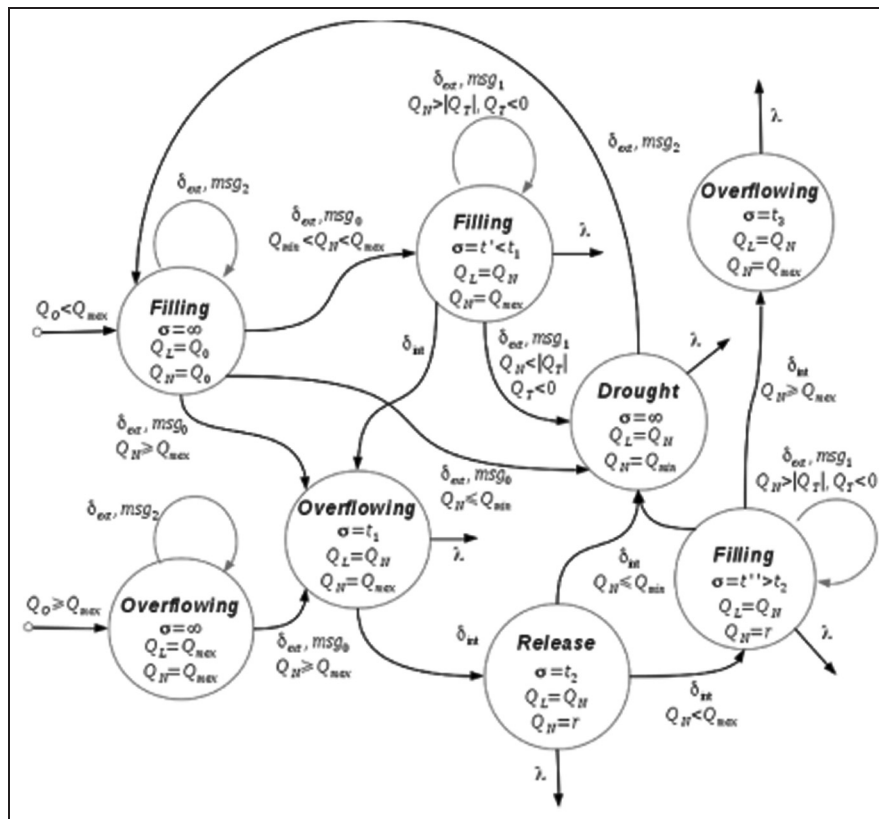


Figure 8. The state graph of the DAM model.

or *Drought* with the lifetime of status σ and with the last level (Q_L) and the next level (Q_N) of the dam water level. Here Q_0 is the initial value of the level, Q_T is the target quantity of water occurring on the command port (positive or negative), Q_{max} and Q_{min} respectively the minimum and the maximum capacity of the dam. The messages occurring on the ports are denoted as msg_i with i the port number.

Initially, the DAM model may be in the *Filling* or *Overflowing* state depending on its water level (Q_0). Then, its state may change only by receiving external messages depending on the next level Q_N computed according to the water supply curve included in msg_0 . Whatever the initial state, if $Q_N \geq Q_{max}$, the new state of the DAM model is *Overflowing* until t_1 and messages are sent on the output ports. Then, the internal transition computes the new state which becomes first *Release* until t_2 , second becomes *Filling* or *Drought* depending on the Q_N value. If the dam reaches its minimum water level ($Q_N < Q_{min}$), the drought is stated and the DAM model goes back to the initial *Filling* state. Otherwise, if the dam does not reach its maximum water level, the *Filling* state is set until t'' followed by the *Overflowing* state until t_3 . If a command occurs on port 1 (msg_1) and the target value required ($Q_T < 0$) is not possible ($Q_N > |Q_T|$), the state remains the same. Based on the previous description of the DAM atomic model, the model to be optimized (simulation model) can be considered as an interconnection of two kinds of these dams: Figari and Ospedale.

Figure 9 depicts the DEVS coupled model corresponding to the simulation model under study. The coupling allows the communication of the Figari dam with the Ospedale dam. When the Figari dam needs water, it asks (by sending a negative value on its command output port) to the Ospedale dam. Then, if the Ospedale dam has the capacity to answer, it sends on its command output message the desired positive value asked by Figari.

4. Optimization of catchment basin management model

4.1. Optimization problem formalization

An optimization problem concerns the minimization or maximization of an objective function $f(x)$ where x is a vector $x = \{x_1, \dots, x_n\}$ of n decision variables. A solution of the optimization problem is an assignment of specific values to x . All solutions form a set X of solutions ($x \in X$). When decision variables are continuous, $x \in \mathbb{R}^n$.

There are a number of important subclasses of optimization problems depending on the ability to define constraints on the decision variables. When there is no constraints on the decision variables, the unconstrained maximization formulation can be used as follows:

$$\begin{aligned} & \text{maximize } f(x) \\ & x \in \mathbb{R}^n \end{aligned}$$

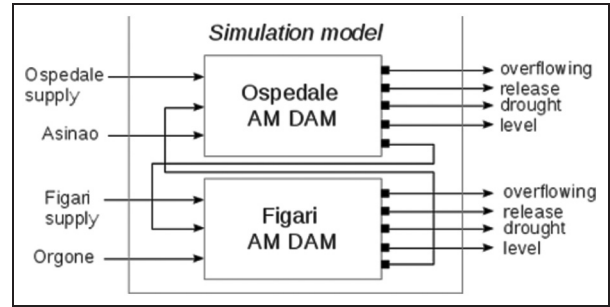


Figure 9. DEVS simulation coupled model of the catchment basin.

with the objective function $f: \mathbb{R}^n \rightarrow \mathbb{R}$. The decision variables are used to construct object criteria as a mathematical function. By formulating them and depending on the problem description (see in Section 3.1), we can get a mathematical model relating the variables and the objective function:

$$f(x) = (y_1(x_1, x_2) - y_2(x_1, x_2))y_3(x_1, x_2)y_4(x_1, x_2) \quad (1)$$

where:

- $x = \{x_1, x_2\} \in \mathbb{R}^2$ with x_1 and x_2 are two decision variables for the water consumptions of the population insured by the Ospedale dam (x_1) and the Figari dam (x_2); these decision variables (annotated by $Ospedale_{supply}$ and $Figari_{supply}$ in Figure 9) are represented by two curves and the problem can be considered as a COvS problem;
- $y_1 = Overflowing_{Ospedale}$ is the quantity of water turbined by the Ospedale dam between t_0 and t_3 (see Figure 6);
- $y_2 = Release_{Figari}$ is the quantity of water discharged by the Figari dam using the pump between t_0 and t_3 (see Figure 6);
- $y_3 = Drought_{Ospedale}$ and $y_4 = Drought_{Figari}$ respectively the drought flag (Boolean value) for the Ospedale and Figari dam.

We can note in the formula 1 that the values of the y_i ($i \in 1..4$) functions are obtained by simulations of the simulation model (Figure 9). Indeed, optimizing the simulation model consists in finding the best couple $\{x_1, x_2\}$ of curves among a theoretical infinite set which maximizes the objective function value. In summary, the optimization problem can be formulated as: using DE simulation through the DEVS formalization, to find the best combination of two curves (decision variables) which maximizes the objective function value. Usually, the algorithms used to solve unconstrained optimization problem are:⁴⁶ descent methods, line search, gradient descent method, steepest descent method, Newton's method, conjugate gradient method or quasi-Newton's methods. In this paper, the

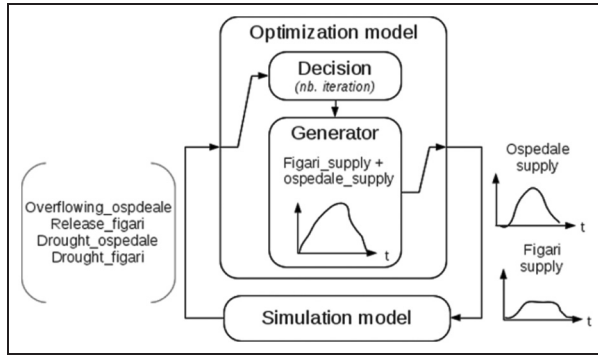


Figure 10. Ospedale-Figari DE simulation and optimization model.

proposed algorithm is inspired by the descent methods and starting from a particular point x_0 , a strategy-based algorithm gives a set of value $(x_k)_{k \in \mathbb{N}}$ such that:

$$f(x_{k+1}) \geq f(x_k) \quad \text{for all } k \in \mathbb{N}$$

It is therefore for each iteration k , to find a point x_{k+1} toward a direction defined by strategies based on: (i) a gain defined by $|y_1 - y_2|$; (ii) a Boolean condition defined by y_3, y_4 .

4.2. Proposed simulation-based optimization algorithm

Figure 10 depicts the optimization for the simulation of the Ospedale-Figari simulation model in the context of COvS.

The aim of the *Generator* model is to generate couples of curves $Ospedale_{supply}$ and $Figari_{supply}$ from the total water consumption of the population. The model *Decision* analyzes the system performance (objective function value) and decides if an additional simulation is possible.

Due to the infinite number of possible solution X , the proposed approach cannot guarantee to find the optimal solution. However, we are sure to find the best solution included into a subset of X corresponding to the number of iterations. The proposed algorithm allows to drive towards the best solution by analyzing the shape of the curves ($Ospedale_{supply}$ and $Figari_{supply}$) according to the engineers requirements.

Engineers of dam management are only considering the water consumption of the population. This means that, they only give the sum of the $Ospedale_{supply}$ and $Figari_{supply}$ quantities. Therefore, only this sum must be taken into account by the optimization model and it must be divided to obtain two curves representing the two decision variables. Moreover, the curves represent the water consumption of the population and they can be considered as a trapezoidal profiles (see Figure 11). The trapezoidal profiles are often used to model natural phenomena presenting an evolution with three stages (growth, stability and

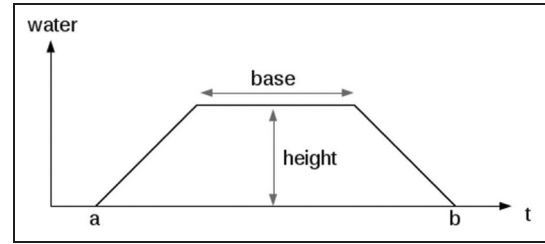


Figure 11. Trapezoidal profile template for the water consumption.

declination) characterized by a duration and a specific form.⁴⁷ This kind of profile is a good candidate to reflect the population's water consumption behavior and is especially suitable for the presented modeling approach. Figure 11 shows that a water consumption curve depends on four parameters: (i) *base* which allows the control of the duration of maximum water consumption, (ii) *height* that represents the maximum level of water consumption, (iii) *a* and (iv) *b* the two dates controlling the water consumption period. Modifying a consumption curve leads to change these parameters.

The model in charge of finding a couple of decision variables is called *Generator* and its behavior has been resumed in Algorithm 1. The strategies used by the *Generator* model to effectively guide the modification of the curves during the optimization process represents a key of the proposed method. The model in charge of activating the simulation model through the *Generator* model is the *Decision* model (see Figure 10). It receives a message from the simulation model and analyzes the objective function value.

This paper presents strategies for the *Generator* model which depends on a parameter ϵ defined by the user as follows.

- Strategy $s1$ or $s2$: increasing or decreasing the *height* parameter ($height = height \pm \epsilon$).
- Strategy $s3$ or $s4$: moving on the left or on the right the base of the curve, i.e. changing the middle m of the trapeze without changing the base ($m = m \mp \epsilon$).
- Strategy $s5$ or $s6$: increasing or decreasing the *base* parameter ($base = base \pm \epsilon$).

There are two ways to combine strategies in order to guide the modification of the curves. If the $Figari_{supply}$ curve is chosen, then the repeated sequence $s2 \rightarrow s4 \rightarrow s6$ has to be employed. Otherwise, if the $Ospedale_{supply}$ curve is chosen, the repeated sequence $s1 \rightarrow s3 \rightarrow s5$ has to be considered. The two scenario are equivalent since the two curves are linked by the total consumption that is an initial data given by the engineers of dam management as explained previously.

Algorithm 1. Generator model algorithm.

Constants:

- C is the curve of total consumption given by the engineers of dam management;
- $Nb = 5000$ is the number of iterations defined by the modeler.

Variables:

- k is the current iteration;
- s is the current strategy;

Output:

- $x^* = (Figari_{supply}^*, Ospedale_{supply}^*)$ is the best solution.
 1. $k = 0$; $s = s2$; $Figari_{supply}$ is generated randomly from a trapezoidal shaped (see Figure 11) and it fits into the total curve C .
 2. As long as k is not equal to Nb :
 - (a) Deduct the $Ospedale_{supply}$ curve by performing the difference between C and $Figari_{supply}$;
 - (b) Send the two curves to the simulation model for new simulation;
 - (c) Analyze the results of the previous simulation (input message from the model *Decision*) and compute the objective function f (see Equation (1) according to the simulation results y_1, y_2, y_3 and y_4). If f decreases or is null, the current strategy s is changed in an order determined by the repeated sequence $s2 \rightarrow s4 \rightarrow s6$. If f increases, s does not change. In any case, a new $Figari_{supply}$ curve is proposed for a new simulation according to the current strategy s and the previous $Figari_{supply}$ curve.
 - (d) $k = k + 1$;
 3. Return the best solution $x^* = \{Figari_{supply}^*, Ospedale_{supply}^*\}$ which is deduced among the solutions computed during the Nb iterations.

Algorithm 1 is based on the choice of the $Figari_{supply}$ curve and allows the cyclical switching between the ordered strategies $s2, s4$ and $s6$ depending on the objective function value. In this algorithm the following process is proposed: to increase the turbine of overflowed water for the Ospedale dam, change the parameters for the $Figari_{supply}$ curve as follows: decreasing the *height* parameter (strategy $s2$) or moving to the right the *base* (strategy $s4$) or decreasing the *base* (strategy $s6$). If the user want to interact with the $Ospedale_{supply}$ curve instead of the $Figari_{supply}$ curve, it must decrease the pumping of the water within the Figari dam by increasing the *height* parameter (strategy $s1$) or to move to the left (strategy $s3$) or to increase the *base* parameter (strategy $s5$) of the $Ospedale_{supply}$ curve.

In practice, a stopping test should be selected to ensure that the algorithm always stops after a finite number of iterations and the last iteration guarantees a value of x close to x^* . Consider the factor of precision $\alpha > 0$, this test consists of a combination of criteria which can be based on:⁴⁶

- stagnation of the solution, $\|x_{k+1} - x_k\| > \alpha\|x_k\|$;
- stagnation of the current value, $|f(x_{k+1}) - f(x_k)| > \alpha|f(x_k)|$;
- number of iterations that exceed a threshold established in advance, $k < IterMax$.

In this paper, a stopping step based on the total number of iteration is proposed (see step 2 in Algorithm 1). However, considering the previous criteria, it is possible to implement a new condition to stop the algorithm before reaching the final iteration.

4.3. DEVSimPy implementation and results

The DEVS EF insures the ideal conditions for the control of the OvS process. The DEVS EF can be exploited to guide the simulation with the callback managing of the simulation model depending on the activity of the optimization model. This last one analyzes the simulation model results (through the *Decision* model in Figure 10) and decides, depending on the number of iterations if a new activation for the simulation model is possible. In this latter case, the optimization model (through the *Generator* model in Figure 10) is in charge of the generation of a new solution which will be validated by the simulation model. Figure 12a shows the implementation of the optimization model in DEVSimPy coupled with the previously described simulation model. The *total_consumption*, *Asinao* and *Orgone* are three curve generators instantiated from the *curveGen* model in order to provide respectively the total water consumption by the population, the intake for the Ospedale and the Figari dam. The *Decision* model (in Figure 12b) has four input ports according to the

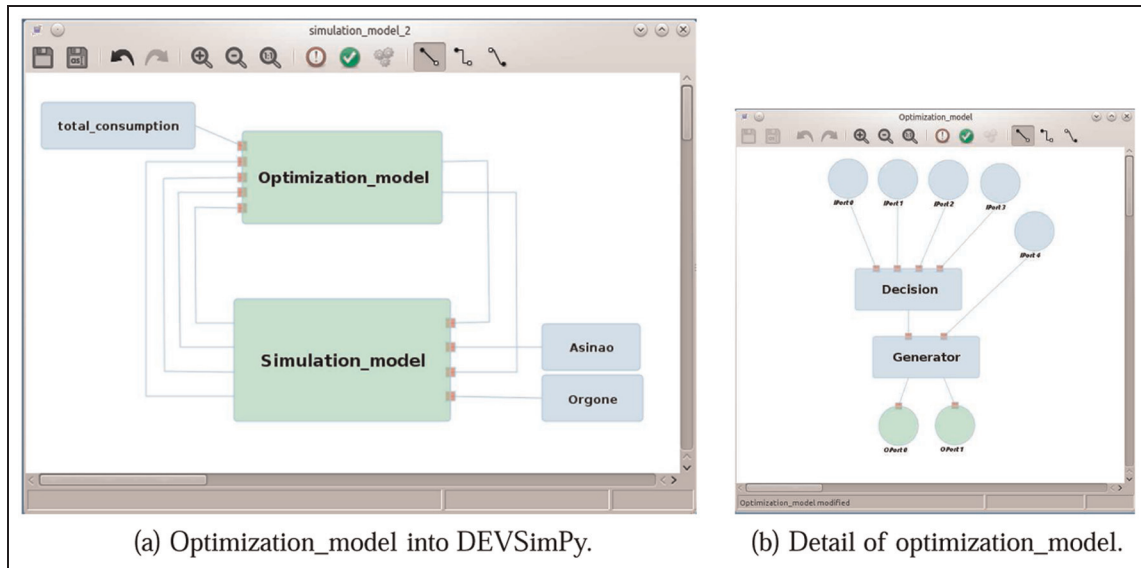


Figure 12. Modeling of the optimization_model coupled model with DEVSimPy.

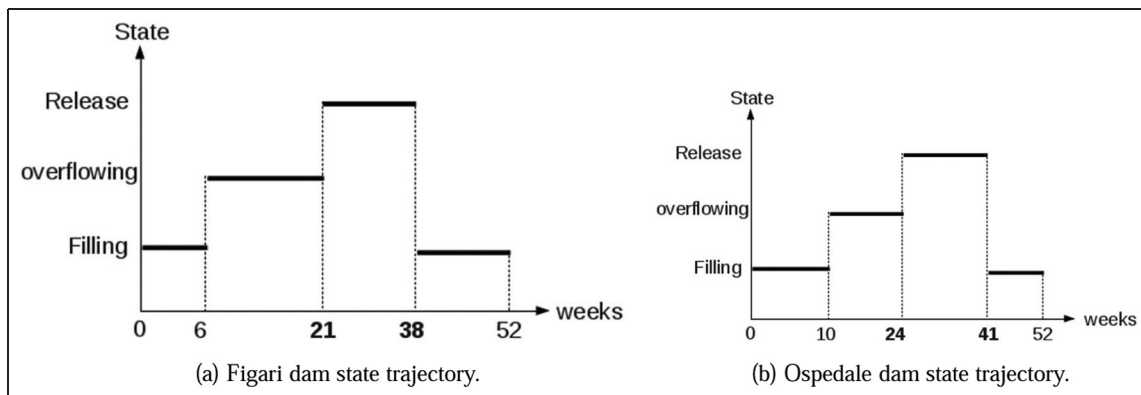


Figure 13. States trajectories for the dams.

previous description (see Figure 10) in order to receive the following signals: $Overflowing_{ospedale}$, $Release_{figari}$, $Drought_{ospedale}$ and $Drought_{figari}$. The *Generator* model (see Figure 12b) receives the decision message on port 0 and the total consumption on port 1. It is then able to generate the two curves $Ospedale_{supply}$ and $Figari_{supply}$.

Figure 13 depicts the state trajectories for the Figari and Ospedale dams obtained after a simulation with 5000 iterations. According to Figure 13a for the Figari dam, the transition of summer/winter occurs at week 21 (d1) and for winter/summer at week 38 (d2). In the same way according to Figure 13b, for the Ospedale dam, the transition of summer/winter (d1) occurs at week 24 and for winter/summer (d2) at 41.

The results presented in Figure 13 correspond to the best solution which is illustrated in Figure 14 (respectively Figure 15) by the $Figari_{supply}$ curve (respectively $Ospedale_{supply}$ curve).

Figure 14 shows three curves representing the two trapezoidal shapes the total consumption (the largest) and the $Figari_{supply}$ curve (the smallest) and for the third the *Orgone* river intake. In the same way Figure 15 depicts the curves for the total consumption (the largest trapezoidal curve) and the $Opedale_{supply}$ curve (the smallest trapezoidal curve) and, finally, the *Asinao* river intake.

Table 1 resumes the simulation results with the obtained best solution. These results have been obtained with 5000 iterations and 6.125 s system CPU time. In Table 1 Fig. (respectively Osp.) stands for Figari (respectively Ospedale) dam. Moreover, we can recall that the overflowing (respectively release) column resumes the turbine (respectively pump) value for the Ospedale (respectively Figari) dam.

It is suggested that readers consult the video <https://code.google.com/p/devsimpy/wiki/Videos> to see an example of the simulation run.

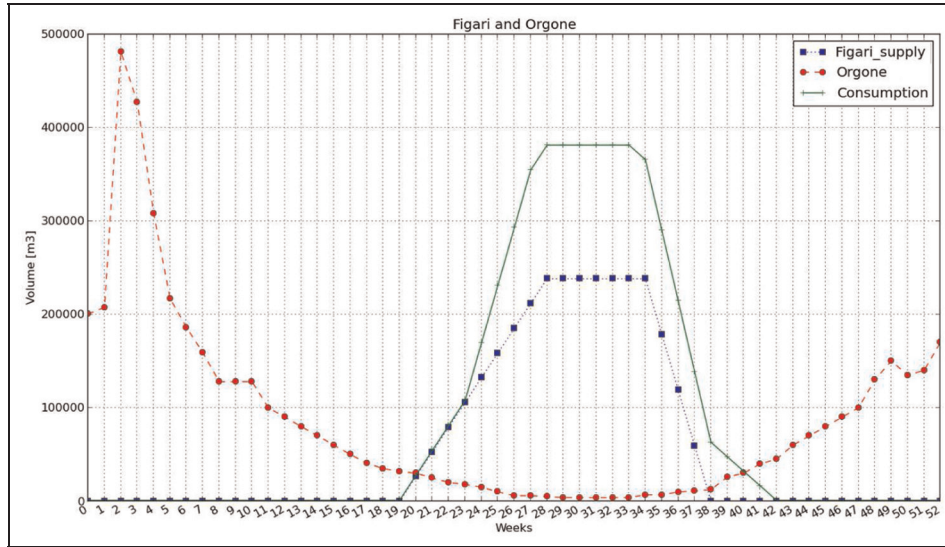


Figure 14. The best solution for the Figari dam.

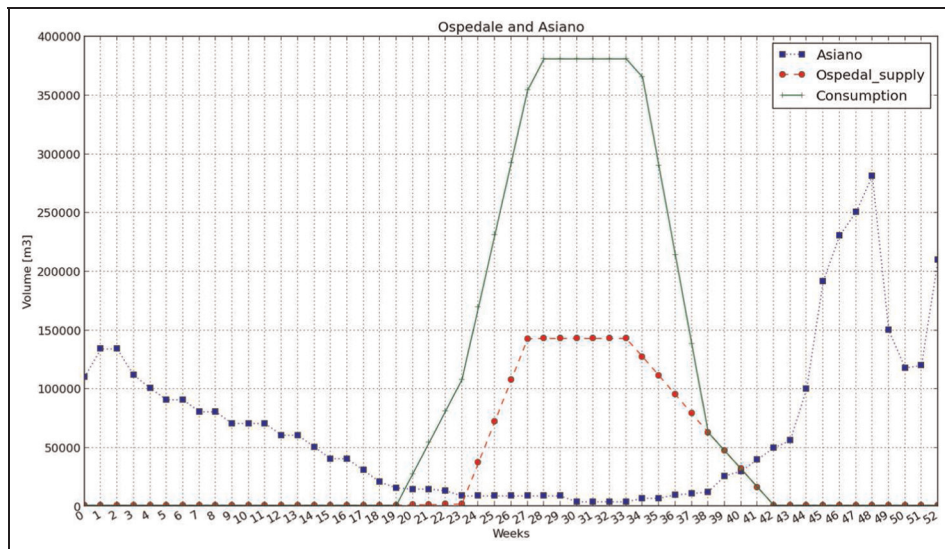


Figure 15. The best solution for the Ospedale dam.

Table I. DEVSimPy DE simulation results with COvS method.

	Release (m ³)	Overflowing (m ³)	Level (m ³)	d1 (week)	d2 (week)
Osp.	2,902,127	3,844,223	2,353,038	24	41
Fig.	1,933,506	550,247	1,300,956	21	38

The convergence of the algorithm presented in Section 4.2 is illustrated by the curve in Figure 16. The abscissa expresses the number of iterations and the ordinate the value of the objective function which is defined by the difference between the amount of the quantity of turbine water minus the quantity of pumped water.

Algorithm 1 as described in Section 4.2 consists of applying alternatively the three strategies called s_2 , s_4 , s_6 . As soon as a strategy does not provide any improvement (increment) of the objective function value, the algorithm allows to switch to another strategy. In the considered example we can notice in Figure 16 that the strategy s_2 is

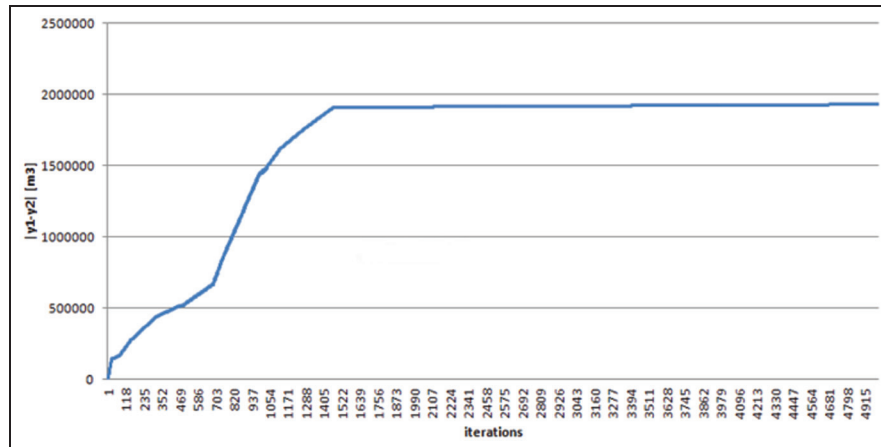


Figure 16. Evolution of the objective function based on the number of iterations.

applied from iteration 1 to iteration 158, then strategy *s4* is applied from 158 to 515, the strategy *s6* is then applied from iteration number 315 to iteration 474, etc. This process allows to convergence toward a good solution since at each step the objective function value is incremented.

5. Analysis and comparison

In this section we present an analysis of the DE modeling and simulation approach (presented in Section 4) used for implemented a COvS method presented in Section 4.3. This analysis is performed by comparing the approach described in this paper with the two approaches (respectively belonging to the R&S and DOvS categories) already developed and presented by Santucci and Capocchi¹² and Capocchi et al.¹³

5.1. Context of the experiments and results

The context of the experiment is given by the inputs of the simulation models (both involved in the three OvS methods): the *Orgone* and *Asinao* intakes water and the water supply of the consumers (respectively shown in Figures 14 and 15).

The R&S and DOvS optimization of the Figari–Ospedale model is presented in Figure 17. We can point out that two decision variables are used: *d1* and *d2* while two objectives functions values are highlighted in Figure 17: the *Gain* (the difference between the turbine and the pumping) and the *Drought* variable. In the case of DOvS optimization a third decision variable is added: the *Turbine* corresponding to the quantity of water which is going to be sent to the power station.

The solutions obtained by the three methods are summarized in Table 2. Table 2 presents the final results where: Method represents the name of the optimization method; *d1* (respectively *d2*) the date corresponding to the

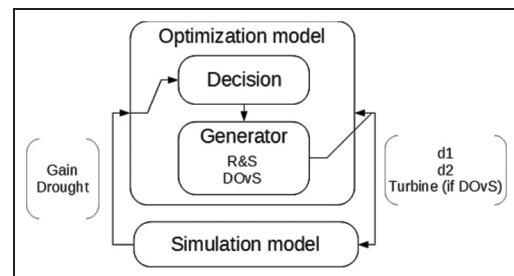


Figure 17. R&S and DOvS for the Ospedale–Figari simulation model.

switch winter/summer (respectively summer/winter); Turbine represents the quantity of water which has been turbined by the Ospedale power station; Pump represents the quantity of water which has been pumped using the Figari pump station; CPU represents the time spent by the processor when performing the corresponding method; Iteration represents the number of iterations involved in the method. Concerning the COvS method, the number of iterations is much greater due to the fact that the CPU time required during each iteration step is much lower than with a R&S or DOvS method. The stopping condition of the optimization algorithm (Algorithm 1) is the number of iterations. In Table 2, the R&S method requires 1763 iterations to obtain a best solution for *d1* and *d2*. This number of iterations is given by the number of possible combinations (52×52) less the non-eligible couples (*d1*, *d2*) as $d1 > d2$ and $d1 = d2$. For the DOvS method, the required number of iterations to reach a local convergence is smaller than 500. Concerning COvS method, we can see that in the given example the best solution is found after more than 2000 iterations (see Figure 16).

We can point out that using COvS we are able to disconnect the behavior of the two dams. Indeed, in the Table

Table 2. Comparison of solutions for the three optimization methods.

Method	$d1$ (week)		$d2$ (week)		Turbine (m ³)	Pump (m ³)	CPU (s)	Iteration
R&S	Osp. 21	Fig. 21	Osp. 35	Fig. 35	45,836	90,256	987.6	1763
DOvS	Osp. 24	Fig. 24	Osp. 41	Fig. 41	3,562,791	1,800,228	9,351,512	500
COvS	Osp. 24	Fig. 21	Osp. 41	Fig. 38	3,844,223	1,933,500	6.1	5000

2 the value for $d1$ (respectively $d2$) depends on the considered dam (see Table 2 where Osp. (respectively Fig.) for Ospedale (respectively Figari) dam).

5.2. Discussion

The results presented in Section 5.1 point out several features concerning the COvS optimization method presented in this paper.

1. The system CPU time involved by the COvS method based on DE simulation is very low in comparison with the system CPU time required when dealing with the simulation involved in the R&S and DOvS methods.
2. The best solution obtained using COvS is better than solutions obtained using R&S and DOvS. The quality of a solution is evaluated using the difference of the quantity of water turbine (Turbine column in Table 2) and the quantity of water which has been pumped (Pump column in Table 2). The reader can note that the highest difference (the best solution) can be found for the COvS method in Table 2.
3. The results obtained with the COvS method point out that due to the modeling approach based on DE modeling and simulation, we are able to propose to the user different dates for the two dams which have to be controlled while the same $d1$ and $d2$ are proposed for the R&S and DOvS methods.
4. The main difference between the COvS based on DE simulation and the two methods (R&S, DOvS) is highlighted by Figures 17 and 10 where we can observe that the decision variables of the COvS method are two curves describing how each of the dams are responding to the water consumption while the decisions variables of the R&S and DOvS simulation methods are integer values of the $d1$, $d2$ and turbine (for DOvS) variables.

From these features we are able to propose the following analysis.

1. Because the system CPU time is so low using the COvS method proposed in this paper, we can deduce that we have the possibility to enhance the

method by exploring several other strategies in order to try other configurations for the water supply curves associated with each of the dams.

2. The fact that we are able to dissociate the dates $d1$ and $d2$ for the two dams, we are able to propose the user a better management of the catchment basin since the user will be able to change the behavior of a dam independently from the other dam.
3. The fact that the COvS method handles continuous curves instead of discrete values as in the case of R&S and DOvS facilitates the optimization module. Therefore, the change of the repartition of amount of water to send to the consumers between the dams all along the year is realized just by modifying the decision variables (i.e. the two curves). In contrast, such modification using DOvS requires several simulations based on 52 weeks.

6. Conclusion

This paper has pointed out a case study requiring the use of OvS methods applied to a system whose no mathematical specification is available. The case study concerns the optimization of management of a catchment basin involving dams, electrical power station, pumping station, valves, etc. The DEVS formalism has been used to model the case study integrating three OvS methodologies (R&S, DOvS and COvS). The simulation results highlight that the proposed DEVS-based COvS provides a better performance than the R&S and DOvS in the specific case of the management of a catchment basin. However, improvements are to be introduced to the proposed optimization algorithm: (i) a better stopping test has to be implemented in order to reduce the number of iterations and the CPU time consumption; (ii) new strategies can be defined and tested in order to explore new search path in the solution space; (iii) new global approaches based on evolutionist algorithms will be investigated. Concerning the future work, we plan to implement a COvS method based on the use of ODEs for a new hybrid simulation model using DEVS QSS methods.

Funding

This work was supported by the RESMAR Marittimo European project (<http://www.maritimeit-fr.net/>).

References

1. Azadivar F. Simulation optimization methodologies. In *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future - Volume 1 (WSC'99)*. New York: ACM Press, pp. 93–100.
2. Ferscha A and Tripathi SK. Parallel and Distributed Simulation of Discrete Event Systems. Technical report, College Park, MD, USA, 1994.
3. Bachelet B and Yon L. Model enhancement: Improving theoretical optimization with simulation. *Simulation Modelling Practice and Theory* 2007; 15(6): 703–715.
4. Kim SH and Nelson B. Selecting the best system: theory and methods. In *Proceedings of the 2003 Winter Simulation Conference*, vol. 1, pp. 101–112.
5. Kim SH and Nelson BL. Recent advances in ranking and selection. In *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come (WSC'07)*. Piscataway, NJ: IEEE Press, pp. 162–172.
6. Nelson BL. Optimization via simulation over discrete decision variables. *Tutorials in Operations Research*. Hanover, MD: INFORMS, 2010, pp. 193–207. DOI: 10.1287/educ.1100.0069.
7. Xu WL and Nelson BL. Empirical stochastic branch-and-bound for optimization via simulation. *IIE Transactions* 2013; 45(7): 685–698. <http://www.tandfonline.com/doi/pdf/10.1080/0740817X.2013.768783>.
8. Box GEP and Wilson K. On the experimental attainment of optimum conditions (with discussion). *Journal of the Royal Statistical Society Series B* 1951; 13(1): 1–45.
9. Dellino G, Kleijnen JP and Meloni C. Robust optimization in simulation: Taguchi and response surface methodology. *International Journal of Production Economics* 2010; 125(1): 52–59.
10. Angun E. A risk-averse approach to simulation optimization with multiple responses. *Simulation Modelling Practice and Theory* 2011; 19(3): 911–923.
11. Zeigler BP, Kim TG and Praehofer H. *Theory of Modeling and Simulation*, 2nd edn. Orlando, FL: Academic Press, 2000.
12. Santucci J and Capocchi L. Catchment basin optimized management using a simulation approach within DEVSimPy framework. In *Proceedings of the Summer Simulation Multiconference*, Genova, Italy, pp. 28–36.
13. Capocchi L, Santucci J, Farges J, et al. Biblioteca DEVS dei modelli ottimizzati. Technical report 5.3.1. Report European Project RESMAR, Action G, 2012.
14. Farges J, Amoretti R, Capocchi L, et al. Specificazione dei metodi di ottimizzazione basati sulla programmazione dinamica. Technical report 5.1.2. Report European Project RESMAR, Action G, 2011.
15. Capocchi L and Santucci JF. Discrete optimization via simulation of catchment basin management within the DEVSimPy framework. In *WSC 2013*, Washington, DC, pp. 205–216.
16. Capocchi L, Santucci JF, Poggi B, et al. DEVSimPy: A collaborative python software for modeling and simulation of DEVS systems. In *WETICE*. Los Alamitos, CA: IEEE Computer Society, pp. 170–175. <http://code.google.com/p/devsimpy/>.
17. Swisher J, Hyden P, Jacobson S, et al. A survey of simulation optimization techniques and procedures. In *Proceedings of the Winter Simulation Conference*, vol. 1, pp. 119–128.
18. Fu MC. Feature article: Optimization for simulation: Theory vs. practice. *INFORMS J Comput* 2002; 14(3): 192–215.
19. Fu MC, Glover FW and April J. Simulation optimization: a review, new developments, and applications. In *Proceedings of the 37th Winter Simulation Conference (WSC'05)*, pp. 83–95.
20. Hong L and Nelson B. A brief introduction to optimization via simulation. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*. IEEE, pp. 75–85.
21. Swisher JR, Jacobson SH and Yücesan E. Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey. *ACM Trans Model Comput Simul* 2003; 13(2): 134–154.
22. Hong L, Nelson B and Xu J. Speeding up COMPASS for high-dimensional discrete optimization via simulation. *Operat Res Lett* 2010; 38(6): 550–555.
23. Xu J, Nelson B and Hong J. Industrial strength COMPASS: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation* 2010; 20(1): 3.
24. Xu J, Nelson BL and Hong LJ. An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems. *INFORMS J Comput* 2013; 25(1): 133–146.
25. Barton RR and Meckesheimer M. Metamodel-based simulation optimization. In Henderson SG and Nelson BL (eds.), *Simulation (Handbooks in Operations Research and Management Science)*. Amsterdam: Elsevier, 2006, pp. 535–574.
26. Rossetti MD. *Simulation Modeling and Arena*. New York: Wiley, 2009.
27. Eskandari H, Mahmoodi E, Fallah H, et al. Performance analysis of commercial simulation-based optimization packages: Optquest and witness optimizer. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pp. 2358–2368.
28. Waller A. WITNESS simulation software. In *Proceedings of the Winter Simulation Conference (WSC'12)*, pp. 436:1–436:12.
29. Borshchev A. XJ technologies: AnyLogic 6. In *Proceedings of the 37th Winter Simulation Conference (WSC'05)*.
30. Pegden CD and Sturrock DT. Introduction to simio. In *Proceedings of the Winter Simulation Conference (WSC'11)*, pp. 29–38.
31. Muller D. Automod: providing simulation solutions for over 30 years. In *Proceedings of the Winter Simulation Conference (WSC'12)*, pp. 434:1–434:15.
32. Lee H, Zeigler B and Kim D. A DEVS-based framework for simulation optimization: Case study of link-11 gateway parameter tuning. In *IEEE Military Communications Conference*, pp. 1–7.
33. Bergero F and Kofman E. PowerDEVS: a tool for hybrid system modeling and real-time simulation. *Simulation* 2011; 87(1–2): 113–132.
34. Kim TG, Sung CH, Su-Youn H, et al. Devsim++ toolset for defense modeling and simulation and interoperation. *J Defense Model Sim* 2011; 8(3): 129–142.

35. Sarjoughian HS and Zeigler B. Devsjava: Basis for a DEVS-based collaborative M&S environment. *Simulation* 1998; 30: 29–36.
36. Quesnel G, Duboz R and Ramat É. The virtual laboratory environment—an operational framework for multi-modelling, simulation and analysis of complex dynamical systems. *Sim Modell Practice Theory* 2009; 17(4): 641–653.
37. Bonaventura M, Wainer GA and Castro R. Graphical modeling and simulation of discrete-event systems with cd++builder. *Simulation* 2013; 89(1): 4–27.
38. de Lara J and Vangheluwe H. ATOM3: A tool for multi-formalism and meta-modelling. In *Proceedings of FASE'02*. London: Springer-Verlag, pp. 174–188.
39. Chungman Seo RC, Zeigler B and Kim D. DEVS modeling and simulation methodology with MS4Me software. In *Theory of Modeling & Simulation Symposium, SpringSim Multi-Conference (SpringSim'13)*.
40. de Lara J and Vangheluwe H. Computer aided multi-paradigm modelling to process Petri-nets and statecharts. In *Proceedings of the First International Conference of Graph Transformation (ICGT)* (Lecture Notes in Computer Science, vol. 2505). New York: Springer-Verlag, pp. 239–253.
41. Martin A, Klamroth K, Lang J, et al. *Mathematical Optimization of Water Networks* (International Series of Numerical Mathematics, number 162). Basel: Birkhäuser, 2012.
42. Mays LW. *Water Distribution System Handbook*. New York: McGraw-Hill, 2000.
43. di Pierro F, Khu ST, Savić D, et al. Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms. *Environ Model Softw* 2009; 24(2): 202–213.
44. Kofman E and Junco S. Quantized-state systems: a DEVS approach for continuous system simulation. *Trans Soc Comput Simul Int* 2001; 18(3): 123–132.
45. Giacobbi J and Palazzi JM. Consolidamento di un modello di caratterizzazione e di gestione di un bacino idrico. Technical report 4.1.1. Report European Project RESMAR, Action G, 2010.
46. Csiszar S. Optimization algorithms (survey and analysis). In *Proceedings of the International Symposium Logistics and Industrial Informatics*, pp. 185–188.
47. Dorp JRv and Kotz S. Generalized trapezoidal distributions. *Metrika* 2003; 58(1): 85–97.

Author biographies

Jean-François Santucci has been a full professor in the Department of Computer Science at the University of Corsica since 1996. His main research interests are modeling and simulation of complex systems. He has been author or co-author of more than 150 papers published in international journals or conference proceedings. Furthermore, he has been the advisor of more than 20 PhD students and the principal investigator for four European research projects and more than ten national industrial contracts since 1990.

Laurent Capocchi is with the SPE (Sciences pour l'environnement) UMR CNRS 6134 laboratory at the University of Corsica. His main research concern the modeling and simulation of complex systems using concurrent or sequential discrete-event approaches. He is founder member of the DEVSImPy open source project, and he contributes actively to its development.