

## **DISCRETE OPTIMIZATION VIA SIMULATION OF CATCHMENT BASIN MANAGEMENT WITHIN THE DEVSIMPY FRAMEWORK**

Laurent Capocchi  
Jean-François Santucci

UMR CNRS 6134 SPE  
Université de Corse  
Corte, FRANCE

### **ABSTRACT**

This paper deals with Optimization via Simulation (OvS) of the management of a catchment basin involving dams, electrical power station, pumping station, valves, etc. We explain how an iterative process allows us to integrate optimization algorithms into a discrete-event simulation using a DOvS (Discrete Optimization via Simulation) methodology. This process has been implemented using the DEVSImPy (DEVS simulator in Python language) environment and the obtained results point out the feasibility of the proposed approach. Furthermore this software is actually used by the Corsican Water Agency in order to efficiently manage the South East water network.

### **1 INTRODUCTION**

When people want to improve the performance of a studied system the modeling and simulation phases are integrated with an optimization technique. The integration of optimization techniques into modeling and simulation relies on the evolution of the model using decisions based on previous simulation results. Usually a specialist of the domain performs the evolution manually. Therefore, it is natural to try to find the set of parameters that optimizes the system performance using an automatic procedure. This is what we call Optimization via Simulation (Fu 2000; Fu 2002; Fu et al 2005; Swisher 2000; Hong and Nelson 2009). Depending on the format of the parameters (also called decision variables), four main categories of OvS methods may be pointed out: (i) Rank and Selection category (Kim 2003; Kim 2007; Swisher et al. 2003); (ii) Discrete OvS category (Nelson 2010); (iii) Continuous OvS category (Box 1951; Dellino 2010; Angùn 2011); Nature analogue category (also called global approaches) which integrates genetic algorithms, or evolutionary strategies (Deb 2008; Azadivar and Tompkins 1999; Ding et al. 2006; Hagedorf and Pawletta 2008).

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 of the catchment basin is performed by an automatic integration of optimization techniques into an discrete-event modeling using the DEVS (Discrete EVent system Specification) (Zeigler et al. 2000) 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 discrete-event simulation and how an integration of an DOvS optimization technique into the modeling scheme allows to automatically obtain an optimized management of the basin. The validation of the DEVS modeling has been performed using the DEVSImPy framework (Capocchi et al. 2011; Capocchi 2012).

The rest of the paper is as follows. The next section presents in detail the context of the study. Section 3 introduced the DEVS formalism; DEVSImPy framework is also briefly presented. The modeling phase

of the behavioral management of a catchment basin is described in section 4. The integration of the DOvS optimization technique into a DEVSimPy modeling scheme is carefully described in section 5. Section 6 gives obtained results. Finally conclusions and perspectives are given in the last section.

## 2 CONTEXT OF THE STUDY

Water is not evenly distributed over the Corsica Island (island of the Mediterranean area) 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. The Corsican Water Agency (CWA) is a governmental institution in charge of distribution and quality of water resources and water bodies, including manmade water infrastructures. CWA manages a complex network of dams, storages 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 network 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 south-east of Corsica Island 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 to send the water from the Ospedale dam to the Figari dam or the consumers. The Ospedale dam (resp. Figari dam) has been built across the *Asinao* (resp. *Orgone*) river. Furthermore another river called the *Oso* river is also used to bring water to consumers.

We have to point out that the south-east region of Corsica has the following features:

- A lot of rainfall in winter while a drought period in summer.
- The consumer population is multiplied by 10 in summer season (due to tourism) according to the winter season.

Because of this dichotomy between summer and winter, two different behaviors for the network management have been defined:

- The winter behavior: since the rainfalls are heavy and the population is low, only Ospedale dam is used to 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 more 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 consumers.

The hydrology experts belonging to the CWA have defined a management strategy based on the definition of two dates (let us called  $d1$  and  $d2$ ) allowing to perform the winter behavior from date  $d1$  to date  $d2$  and then switch to the summer behavior from  $d2$  to  $d1$ . They also have to define the quantity of water which has to be sent to the power station associated with the Ospedale dam.

However the difficulty for the experts is to find the best dates  $d1$  and  $d2$  and the quantity of water (let us called *turbine*) which has to be sent to the hydroelectric power station allowing: (1) to respond to the water demand of the consumers all along the year; (2) to use as much as possible the power station associated with the Ospedale dam; (3) to use as less as possible the pump station associated with the Figari dam.

The management consists in proposing an optimization software approach based on DOvS in order to find the optimized dates ( $d1$  and  $d2$ ) and the optimized quantity of water *turbine* to respond to the 3 main con-

straints: 1) water supply all along the year, 2) optimal use of the power station, 3) minimal use of the pump station.

### 3 DISCRETE EVENT MODELING AND SIMULATION

Discrete-event simulation can be considered as a powerful computing technique which helps the modeler to understand the behavior of a system. It consists in designing a model of real systems and in driving the experiments in order to understand the behavior of the studied system and improve its performances. We briefly introduce in this section the DEVS formalism and the DEVSImPy framework which have been used in order to design and implement the DOvS optimization of the catchment basin management.

#### 3.1 DEVS Formalism

The DEVS formalism has been introduced by B.P Zeigler in the early 70's for modeling discrete-event systems in a hierarchical and modular way (Zeigler 1976). With DEVS, a model of a large system can be decomposed into smaller component models with coupling specification between them. DEVS formalism defines two kinds of models: (i) *atomic models* that represent the basic models providing specifications for the dynamics of a sub-system using function transitions: (ii) *Coupled models* that describe how to couple several component models together to form a new model. DEVS provide an automatic simulation based on time synchronization and message propagation.

An atomic model allows specifying the behavior of a basic element of a given system.

Basic models (called Atomic Models) are defined by the following structure:

$$AM = \langle X, S, Y, C, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$$

where,

- $X$  is the set of input values,
- $S$  is the set of sequential states,
- $Y$  is the set of output values,
- $\delta_{int}$  is the internal transition function dictating state transitions due to internal events,
- $\delta_{ext}$  the external transition function dictating state transitions due to external input events,
- $\lambda$  is the output function generating external events at the output,
- $ta$  is the time-advance function which allows to associate a life time to a given state.

An atomic model allows to specify the behavior of a basic element of a given system. Connections between different atomic models can be performed by a Coupled Model. A coupled model, tells 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.

A simulator is associated with the DEVS formalism in order to exercise coupled model's instructions to actually generate its behavior. The architecture of a DEVS simulation system is derived from the abstract simulator concepts (Zeigler 1990) associated with the hierarchical and modular DEVS formalism.

The behavior of an atomic model is illustrated as follows: the external transition function describes how the system changes state in response to an input. When an input is applied to the system, it is said that an external event has occurred. The next state  $s'$  is then calculated according to the current state  $s$ . The internal transition function describes the autonomous (or internal) behavior of the system. When the system changes state autonomously, an internal event is said to have occurred. The next state  $s'$  is calculated only according to the current state  $s$ . The output function generates the outputs of the system when an internal transition occurs. The time advance function determines the amount of time that must elapse before the next internal event will occur, assuming that no input arrives in the interim. In order to implement DEVS modeling and simulation we use the DEVSImPy framework presented in sub-section 2.3.

### 3.2 DEVSimPy Software

The DEVSimPy (Capocchi et al. 2011; Capocchi 2012) project is open-source software (under GPL V.3 license) dedicated to the DEVS modeling and simulation of complex systems. It has been implemented in Python language (Sanner 1999) with the wxPython graphic library (Rappin and Dunn 2006) and no other dependencies. The aim is to provide a Graphical User Interface (GUI) for the modeling and simulation of PyDEVS models (Li et al. 2002). PyDEVS is an Application Programming Interface (API) allowing the implementation of the DEVS formalism in Python language. The Scipy (Jones et al. 2001) and Numpy (Oliphant 2009) Python scientific libraries can be imported to extend some functionality like for signal processing or continuous system analysis. DEVSimPy inherits all the concepts of the DEVS formalism and adds exciting new capabilities like:

- Viewing atomic and coupled models in simple way in order to simplify their handling.
- Defining the coupling and the priority of models with a graphical interface using drag-and-drop.
- Editing models with code editor during a simulation.
- Handling models stored in local or web libraries.
- Loading and saving models into compressed file including additional information to separate the behavior and the view of models.
- Loading several plugins in order to add functionalities for debugging, handling GIS data, simulation monitoring and much more.
- Allowing dynamic modeling and simulation.
- Stopping the simulation in giving a final time or by models activity scanning.
- and much more.

The DEVSimPy GUI can be employed as a collaborative modeling framework. In fact, the specialist of a domain can submit its specifications to the DEVSimPy engineer in charge of the model implementation. Thus, a model library is built according to the modeling and simulation specification.

Due to its simplicity of use and by the number of functionalities, DEVSimPy can be used by the engineer in charge of the translation of the system specifications provided by the specialist of a given domain. The goal of the engineer is to build libraries of DEVSimPy models knowing their behavioral specifications. Then, the simulation would be invoked and analyzed by the specialist in order to validate the model. These actions must be iterated until the output of the simulation and the output of the system are identical.

The use of DEVSimPy provides a powerful framework to enable computer sciences engineers and specialists of the application domain to simulate and analyze the behavior of catchment basin management. Due to the need of using optimization algorithms for modeling this kind of system, the modelers (working together with specialist of optimization) can build these algorithms by using atomic models in DEVSimPy.

## 4 CATCHMENT BASIN MANAGEMENT MODELLING AND SIMULATION

The proposed model has been defined from the textual and verbal descriptions given by the CWA experts. No unambiguous mathematical equations representing the relationships between inputs and outputs are available.

The catchment basin management model is composed with several DEVS atomic models. Figure 1 shows the architecture of the model called *Ospedale\_Figari*.

We can first point out the twelve inputs:

- *Turbine* which represents the quantity of water to be sent to the hydroelectric powerStation.
- *Asinao*, *Oso* and *Orgone* for the water inputs.

- $s_i$  with  $i$  from 0 to 3 representing the high and low thresholds for the Ospedale and Figari dams.
- $c1$  and  $c2$  for the water consumptions (caused by consumers in the two main town of South East of Corsica island).
- $d1$  and  $d2$  for the dates which control the dams behavior depending on the seasons.

On the bottom of Figure 1 we can find the outputs of the model:

- $Oport0$  (resp.  $Oport1$ ) represents the quantity of water in the Ospedale dam (resp. the quantity of water sent to the consumers).
- $Oport2$  represents the power station activity.
- $Oport3$  represents the first consumption profiling called  $c1$ .
- $Oport4$  consumption the second consumption profiling called  $c2$ .
- $Oport5$  (resp.  $Oport6$ ) represents the quantity of water in the Figari dam (resp. the quantity of water sent to the consumers).
- $Oport7$ ,  $Oport8$  and  $Oport9$  represent the management process by indicating respectively the quantity of water sent to the PowerStation, the quantity of water which is sent from Ospedale dam to Figari dam, the quantity of water which has to be pumped in the Figari dam.

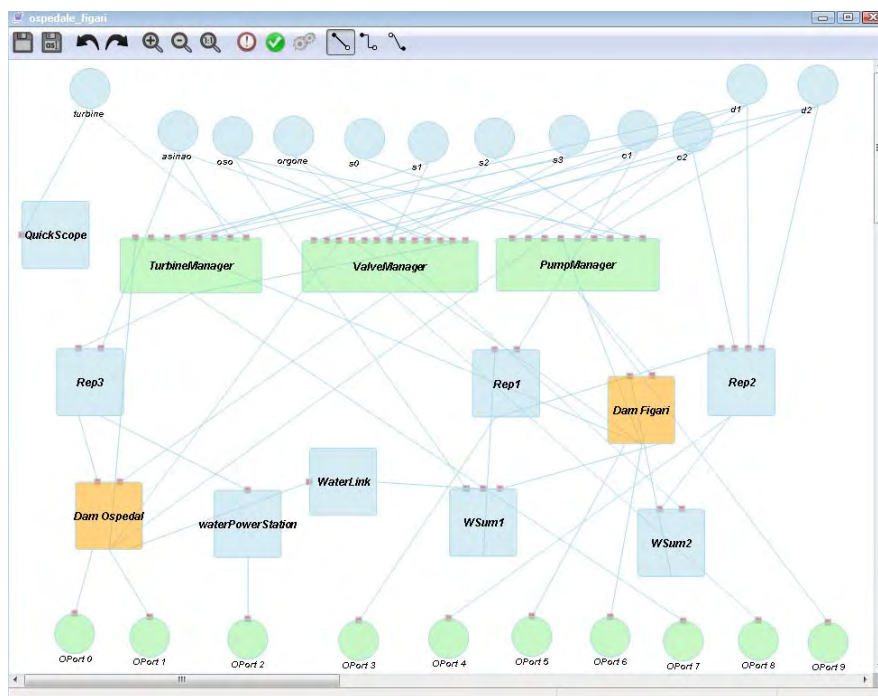


Figure 1: DEVSIMPy Ospedale\_Figari model

The middle of Figure 1 presents the three management models in charge of: (i) the water turbine (TurbineManager), (ii) the water allocation using valves (ValveManager) and (iii) the water pumping (PumpManager). The TurbineManager behavior is very simple: it instantly transfers the value of the turbine input to its output. The ValveManager model is in charge of the water distribution depending of the Ospedale dam level and of the water flow coming from *Asinao*, *Orgone* and *Oso*. It decides of the filling up of Figari dam and water releases of Ospedale dam for water supply purposes. The PumpManager im-

proves the water pumping cost by sharing best practice between ValveManager and TurbineManager. We also have to point out the presence of the two atomic models representing Ospedale dam and Figari dam.

Finally, we highlight the use of models like: (i) PowerStation dedicated to compute electricity from water flow; (ii) WSum which compute the sum of its inputs; (iii) Rep for input distribution.

The model is reused into a complete DEVSIMPy model including the DOvS optimization algorithm in order to drive simulation efficiently. We have first validated the simulation model before integrating this one into a more complex model allowing to perform DOvS method. The validation phase has been performed in collaboration with CWA engineers according to a set of data they collected for a period more than twenty years. The Ospedale\_Figari model responded correctly for this set of data and is therefore ready for being used in an optimization step.

The approach we defined for an automatic optimization of management of water supply involves two steps:

- The modeling of the management of a catchment basin taking into account water supply devices, power stations and pump stations. With the model we are able to simulate the behavior of the resulting system when the values of the inputs  $d1$ ,  $d2$  and turbine are defined by the user.
- The definition of a model allowing the implementation of an optimization process in order to find the best values for  $d1$ ,  $d2$  and the best amount of turbine water according to answer the three constraints presented in sub-section 3.1.

This optimization model is presented in the next section.

## 5 AUTOMATIC MANAGEMENT OPTIMIZATION OF A CATCHMENT BASIN BASED ON DEVSIMPY SIMULATION

The optimization management of a catchment basin leans on the determination of the dates ( $d1$ ,  $d2$ ) and the quantity of water to be sent to the hydroelectric PowerStation (turbine) among a set of solutions that permit to:

- Reducing pumping within the Figari dam.
- Maximizing the production of electricity within the power station using the Ospedale dam.
- Detecting drought events.

The determination of the best couple of dates and the value turbine is accomplished by performing the simulation of the previous model (the Ospedale\_Figari model). The integration scheme presented in Figure 2 has been implemented using the DEVSIMPy framework.

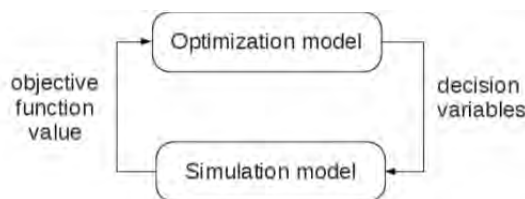


Figure 2: Optimization Via Simulation scheme

In order to optimize the simulation model using the DEVSimPy framework, the decision variables and the objective function value has to be first clearly defined. There are three decision variables:  $d1$ ,  $d2$  and  $turbine$ . The objective function value to be maximized can be defined as follows:

$$Max((turbine - pump) \times drought)$$

where:

- $turbine$  is the water quantity which has been sent to the hydroelectric PowerStation during 1 year (52 weeks),
- $pump$  represents the quantity of water which is sent from the Figari dam to the consumers during 1 year (52 weeks),
- $drought$  is a Boolean value which is set to 1 if the no drought events have been detected during 1 year (52 weeks) and will take a value of 0 otherwise.

In order to perform the simulations which will allow to find the values of the three decision variables which return the best objective function value, the simulation model will also require a set of inputs corresponding to the parameters which are set before the simulation: (i) the three rivers water distribution over the year (52 weeks); (ii) the water consumption over a year (52 weeks). In order to define an DOvS optimization, we need to propose a framework allowing to solve the six following tasks:

- Task 1: to perform the simulation of the model presented in section 4 for 52 weeks with values of the three selected decision variables :  $d1, d2$  and  $turbine$ .
- Task 2: to perform an evaluation of the results of the previous simulations according to the objective function value.
- Task 3: to perform the selection of values for three decision variables using an optimization algorithm.
- Task 4: to perform again the simulation of the model presented in section 4.2 for 52 weeks with the new selected values for the decision variables (new iteration).
- Task 5: to define a variable (called  $nbiter$ ) allowing to stop the iterations : this variable allows the user to set the number of attempts he wants to perform.
- Task 6: to print the set of admissible solutions with the associated information such as the produced electricity, the electricity required in order to activate the pump, the height of water staying in both dams after the last try.

The proposed framework is presented in Figure 3: the integration of optimization techniques into DEVSimPy involves two additional atomic models called *EnumerateOpt* and *DecisionOpt*. These two atomic models are connected with the *Ospedale\_Figari* model. Figure 3 describes how the three models are interconnected. The approach presented in Figure 2 to process simulation based optimization methods has been easily implemented using the DEVSimPy framework as it is shown by the similarity between Figure 2 and Figure 3.

The *EnumerateOpt* atomic model has 2 inputs linked with the 2 outputs of the *DecisionOpt* atomic models: the first one (which we can call  $gain$ ) corresponds to the current value of the difference between the quantity of water which has been sent to the powerStation ( $turbine$ ) and the quantity of water which has been pumped from the Figari dam ( $pump$ ); the second one (called  $drought$ ) is a boolean value which is equal to 0 if there no drought and 1 if not. These two inputs allows to easily compute the objective function value used in order to select the best solution.

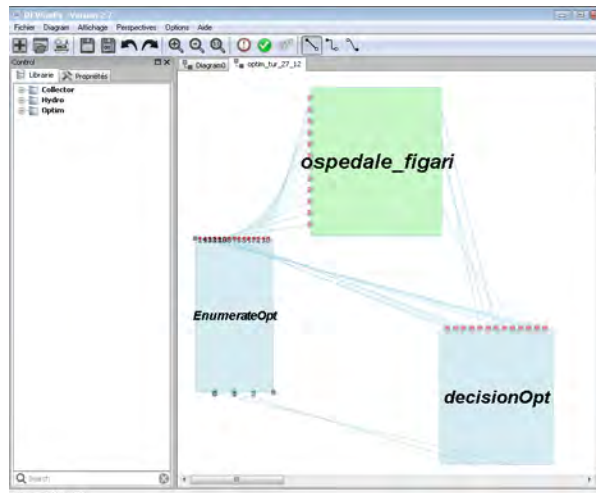


Figure 3: DEVSIMPy coupled model allowing to perform DOvS.

The goal of the *EnumerateOpt* atomic model is to:

- Propose a solution for task 1: the atomic model is a kind of generator atomic model allowing to generate the value of the three decision variables which are to be tried by simulating the Ospedale\_Figari model presented in section 4 for 52 weeks.
- Propose a solution for task 3: a set of algorithms based on the principle of branch and bound (BB) 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 *en masse*, by using upper and lower estimated bounds of the quantity being optimized. A BB algorithm requires two steps: (i) the branching step and (ii) the bounding step. The first one is based on a recursive procedure that, given a set  $S$  of candidates, returns two or more smaller sets ( $S_1, S_2, \dots, S_i$ ) whose union covers  $S$ ; the result is a tree structure (the *search tree*) whose *nodes* are the subsets of  $S$ . The second step leans on the computation of upper and lower bounds for the objective function value obtained by the simulation model within a given subset of  $S$ . The key idea of the BB algorithm is: if the *lower* bound for some tree node (set of candidates)  $A$  is greater than the *upper* bound for some other node  $B$ , then  $A$  may be safely discarded from the search. This step is called pruning, and is usually implemented by maintaining a global variable  $M_v$  (shared among all nodes of the tree) that records the minimum upper bound seen among all sub-regions examined so far. Any node whose lower bound is greater than  $M_v$  can be discarded. The recursion stops when the current candidate set  $S$  is reduced to a single element, or when the upper bound for set  $S$  matches the lower bound. Then any element of  $S$  will be a minimum of the function within  $S$ . In order to compute the different values of the turbine decision variable, we implement one of the processes used for the estimation of a maximum or minimum point of a function : the dichotomy. The dichotomy algorithm allowing to compute the values of the turbine variable which will be sent to the simulation model presented in section 3.1.2 is inserted in the  $\delta_{ext}$  function of the *EnumerateOpt* atomic model.
- Propose a solution for task 5: the user will be able to define the number of iteration for which it will be exercised the simulation model with the required inputs (the three selected decision variables). For each obtained couple  $(d1, d2)$  the simulation model will be exercised with a certain number of values of the turbine decision variable. This number is set by the user (at the beginning of the optimization using the variable called *nbiter*).



- Propose a solution for task 4: the simulation model is simulated over 52 weeks until the number of iterations represented by the variable  $nbiter$  is reached. The  $\delta_{int}$  function of *EnumerateOpt* atomic model allows to count the current number of iteration. As soon as this number is equal to  $nbinter$ , the *EnumerateOpt* model is set to an “EndState” which means that the simulation process is over.

The goal of the *DecisionOpt* atomic model is twofold:

- Propose a solution to task 2: it has to sent to the *Enumeratopt* atomic model using the  $\lambda$  function of the *DecisionOpt* model the results of the performance obtained with the tried potential solution. The results are sent on two outputs. The first one (called *drought*) is the value of the Boolean output corresponding to the fact there it has been or not a drought period during the 52 simulated weeks; the value of the output (called *gain*) corresponding to the difference between the quantity of water which has been used by the PowerStation (*turbine* variable) and the quantity of water which has been sent by the Figari dam to consumers (*pump* variable); let us point out that these two values *drought* and *gain* are computed over a period of 52 weeks.
- Propose a solution to task 6: it has to compute at every step if the current couple ( $d1, d2$ ) and the *turbine* value under study can belong to the set of admissible solutions. This computation is performed using the  $\delta_{ext}$  function of the *OptimizationOpt* atomic model.

We can highlight the generic feature of the proposed solution:

- the optimization algorithm can be easily modified (just by modifying the algorithm implemented in the  $\delta_{ext}$  function of the *EnumerateOpt* atomic model).
- the computation of the objective function value is easily defined in the  $\lambda$  function of *DecisionOpt* atomic model.

## 6 OBTAINED RESULTS

The context of the experiment is given by the inputs of the simulation models involved in the DOvS method: the water flow of the three rivers, the threshold  $s1, s2, s3, s4$  and the water supply of the consumers (see Figure 2). Figure 4 shows the three profiles corresponding to the water flow of the three rivers (*Asinao, Oso* and *Orgone*) which have been used in order to perform the simulation.

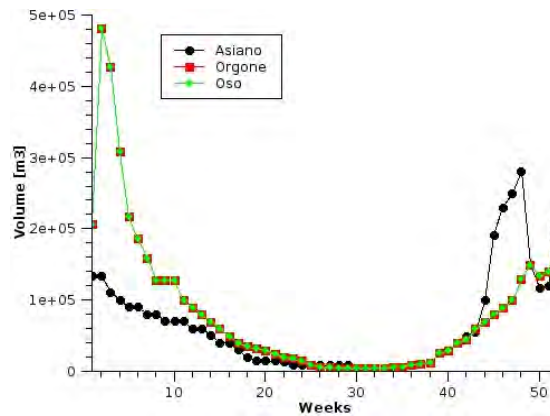


Figure 4: Water flow inputs of *Asinao, Oso* and *Orgone*

The data coming from the water consumption of the inhabitants of South-east of Corsica are given in Figure 5.

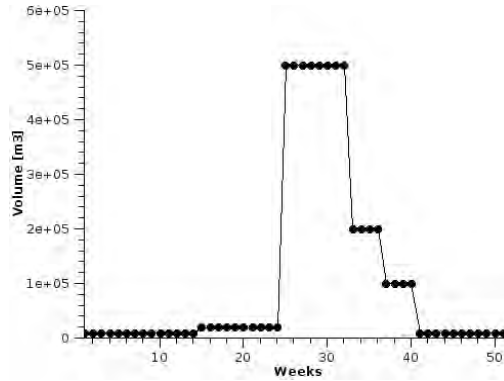


Figure 5: Water consumption of South-East Corsican zone

The final simulation results consist in a set of admissible couples and turbine value.

Nbiter	(d1, d2)	Turbine [m <sup>3</sup> ]	Pump [m <sup>3</sup> ]	HO [m <sup>3</sup> ]	HF [m <sup>3</sup> ]	CPU [second]
3	(39, 26)	3135494	1800228	3000000	5000000	1399,7560
5	(40, 23)	3328974	1800228	71278	5000000	2559,4651
100	(41, 24)	3562791	1800228	70143	5000000	9351512,1

Table 1: Set of admissible solutions

Table 1 presents the final results expressed in m<sup>3</sup> where:

- **Turbine** represents the quantity of water which has been sent to the Ospedale power station.
- **Pump** represents the quantity of water which has been pumped using the Figari pump station.
- **HO (HF)** represents respectively the water level of the Ospedale dam (Figari dam) at the end of the year (after 52 weeks).
- **d1** and **d2** represent the two dates defining the management of the catchment basin.
- **CPU** represents the computational time consumption when performing the optimization.

We can notice that the best solution is given by the last line while the two other lines represent admissible solutions according to the user requirements and the water flow (from *Asinao*, *Oso* and *Orgone*).

## 7 CONCLUSIONS AND PERSPECTIVES

This paper describes how we have defined an optimization based simulation process in order to propose a computer aided management of a catchment basin involving a set of dams, power hydroelectricity stations, pump stations, valves, water links, etc. The process has been developed using the DEVS formalism and implemented in the Python language within the DEVSImPy environment. The core of technique consists in the elaboration of two atomic models (called Enumerate and Decision) which are interconnected with a coupled model describing the behavior of the catchment basin management under study (to be optimized). This approach allows to compute the best solution in order to maximize the gain between the electricity production and water pumping according to the water flows of the river and the consumption of the inhabitants.

Furthermore because of the modularity of the presented approach allows us to choose different kinds of optimization algorithms and implement them using the Enumerate atomic model and decision atomic model. We have implemented two other kinds of local optimization methods: a Rank and Selection opti-

mization of the management of the catchment basin presented in (Santucci and Capocchi 2012) and more recently a COvS approach described in (Santucci and Capocchi 2013).

We are also currently developing a generic global optimization method based on Genetic Algorithm for optimization problems such as the management of a catchment basin. Furthermore we are also interested in the pluviometry prediction. We plan to develop a set of data chronological models based on data fitting which will be inserted into the DEVSimPy framework. We plan also to improve the CPU time involved by the simulation of complex optimization algorithms embedded in the Enumerate atomic model. DEVSimPy will offer different kinds of simulators. For example in addition to the Classical hierarchical DEVS simulator we are about to provide the user the possibility to improve the CPU simulation time using a direct-coupling simulator or a concurrent-based simulator reusing the DEVSimPy models.

## REFERENCES

- Angùn, E., 2011. "A risk-averse approach to simulation optimization with multiple responses." *Simulation Modelling Practice and Theory* 19 (3):911-923.
- Azadivar, F., and G. Tompkins. 1999. "Simulation Optimization with Qualitative Variables and Structural Model Changes: A Genetic Algorithm Approach." *European Journal of Operational Research* 113(1):169-182.
- Box, G. E. P., K. Wilson, 1951. "On the experimental attainment of optimum conditions (with discussion)." *Journal of the Royal Statistical Society Series B* 13 (1):1-45.
- Capocchi, L., Santucci, J.F., Poggi, B., Nicolai, C. 2011. "DEVSimPy: A Collaborative Python Software for Modeling and Simulation of DEVS Systems." In *Proceedings of the 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 170-175, Paris, France.
- Capocchi, L., 2012. DEVSimPy software. <http://code.google.com/p/devsimpy/>
- Deb, K. 2008. "Introduction to Evolutionary Multiobjective Optimization." In *Multiobjective Optimization*, Berlin, Heidelberg: Springer, 59-96.
- Dellino, G., J.P. Kleijnen, J. P., C. Meloni. 2010. "Robust optimization in simulation: Taguchi and response surface methodology" *International Journal of Production Economics* 125 (1), 52-59.
- Ding, H., L. Benyoucef, and X. Xie. 2006. "A Simulation-Based Multi-Objective Genetic Algorithm Approach for Networked Enterprises Optimization." *Engineering Applications of Artificial Intelligence* 19(6):609-623.
- Fu, M. C. 2002. "Feature article: Optimization for simulation: Theory vs. practice" *INFORMS J. on Computing* 14 (3), 192-215.
- Fu, M. C., F.W. Glover, J. 2005. "Simulation optimization: a review, new developments, and applications" In *Proceedings of the 2005 Winter Simulation Conference*, 83-95.
- Fu, M. C., S. Andradottir, J. S. Carson, F. Glover, C. R. Harrell, Y.-C. Ho, J. P. Kelly, and S. M. Robison. 2000. "Integrating Optimization and Simulation: Research and Practice." In *Proceedings of the 2000 Winter Simulation Conference*, 610-616.
- Hagendorf, O., and T.Pawletta, 2008, "An approach for simulation based structure optimization of discrete event systems," In *Proceedings of the 2008 Spring simulation multiconference*, 431-438.
- Halim, R.A. and M.D. Seck. 2011. "The Simulation-based multi-objective evolutionary Optimization (SIMEON) framework" In *Proceedings of the 2011 Winter Simulation Conference*, 2839-2851.
- Hong, L., B. Nelson. 2009. "A brief introduction to optimization via simulation" *Proceedings of the 2009 Winter Simulation Conference*, 75-85.
- Jones, E., Oliphant, T., Peterson, P. 2001. "Scipy: Open source scientific tools for python." [http://www.scipy.org/Citing\\_SciPy](http://www.scipy.org/Citing_SciPy)
- Kim, S.-H., Nelson, B. 2003. "Selecting the best system: theory and methods." In *Proceedings of the 2003 Winter Simulation Conference*, 101-112.

- Kim, S.-H., B.L.Nelson, 2007. "Recent advances in ranking and selection." In *Proceedings of the 2007 Winter Simulation Conference*, 162-172.
- Li, X., H. Vangheluwe, Y. Lei, H.Song, Wang, W. 2011. "A testing framework for devs formalism implementations." In *Proceedings on the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, 183-188.
- Nelson, B. L., 2010. "Optimization via Simulation Over Discrete Decision Variables." *Tutorials in Operations Research*, 193-207.
- Oliphant, T. E., 2007. "Python for scientific computing" *Computing in Science and Engineering* 9, 10-20.
- Rappin, N., and R. Dunn. 2006. *wxPython in Action*. Manning Edition.
- Sanner, M.F. 1999. "Python: a programming language for software integration and development." *J. Mol. Graphics Mod*, vol. 17, 57-61.
- Santucci, J.F., Capocchi, L. 2012. "Catchment basin optimized management using a simulation approach within devsimpy framework." In *Proceedings of the 2012 Summer Simulation Multiconference*, 28-36, Genova, Italie.
- Santucci, J.F., and Capocchi, L., 2013. "Continuous Optimization via Simulation of Discrete-Event System with DEVSimPy." Internal Report N°13, SPE CNRS 6134 Laboratory, Corsica, France.
- Swisher, J.R., and P.D.Hyden. 2000. "A Survey of Simulation Optimization Techniques and Procedures." In *Proceedings of the 2000 Winter Simulation Conference*, 119-128.
- Zeigler, B.P. 1976. *Theory of Modeling and Simulation* New York: Wiley.
- Zeigler, B.P. 1990. *Object-Oriented Simulation with Hierarchical, Modular Models* London: Academic Press.
- Zeigler, B.P., H. Praehofer, and T. G. Kim. 2000. *Theory of Modeling and Simulation*. 2nd ed. San Diego: Academic Press.

## AUTHOR BIOGRAPHIES

**LAURENT CAPOCCHI** was born in Bastia (Corse, France). He received the Master degree in electrical engineering from Ecole Supérieure d'Ingénieurs de Nice Sophia Antipolis, Nice (France) in 2001, the Phd degree in Computer Science from University of Corsica, Corsica (France) in 2005. He is SCS (Society for Modeling and Simulation International) member since 2001 and he is also part of the team of SPE (Sciences pour l'environnement) laboratory at the University of Corsica. His main research concern the modeling and simulation of complex systems (with a special focus on digital and electrical systems) using concurrent or sequential discrete-event approaches. His research goal is to propose a new approach for the modeling and the concurrent simulation of behavioral faults for complex systems using a discrete-event approach. He is founder member of the DEVSimPy open source project and he contributes actively to its development (<http://code.google.com/p/devsimpy/>). His email address is [capocchi@univ-corse.fr](mailto:capocchi@univ-corse.fr) ; his web page is <http://lcapocchi.free.fr>.

**JEAN FRANÇOIS SANTUCCI** is Full Professor in 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 or co-advisor of more than 20 PhD students and since 1998 he has been involved in the organization of ten international conferences. He is conducting newly interdisciplinary researches involving computer science, archaeology and anthropology: since 2006 he is working in interdisciplinary research topics: in the one hand he is performing researches in the archaeoastronomy field (investigating various aspects of cultural astronomy throughout Corsica and Algeria) and on the other hand he is applying computer science approaches such as GIS (Geographic Information Systems) or DEVS (Discrete Event system Specification) to anthropology. His email address is [santucci@univ-corse.fr](mailto:santucci@univ-corse.fr) ; his web page is <http://jeanfranoissantucci.cgpublisher.com/>.