13

The Porting of Wargi-DSS to Grid Computing Environment for Drought Plans in Complex Water Systems

Andrea Sulis[1], Valeria Ardizzone[2], Emilio Giorgio[2] and Giovanni M. Sechi[1]
[1]Hydraulic Sector, Department of Land Engineering, University of Cagliari
[2]National Institute of Nuclear Physics, Sezione di Catania
Italy

1. Introduction

The increasing developments in computer technology have motivated the concurrent development of Decision Support Systems (DSSs) aimed at facilitating the planning and management of complex water systems (Assaf et al., 2008). Simulation and optimization models within DSSs provide the main tool for researchers and practitioners to analyze the behavior and performance of any proposed water resource system design or management policy alternative before it is implemented in real systems. Various strategies have been proposed to combine the adherence and flexibility of simulation models with the efficient exploration of mathematical optimization models (Loucks and van Beek, 2005).

AQUATOOL (Valencia Polytechnic University) (Andreu et al., 1996), MODSIM (Colorado State University) (Labadie et al., 2000), RIBASIM (DELTARES) (Delft Hydraulics, 2006), WARGI-SIM (University of Cagliari) (Sechi and Sulis, 2009a) and WEAP (Stockholm Environmental Institute) (SEI, 2005) are representative of DSSs used for preliminary analysis of alternative plans and policies. Those popular generic simulation models have been implemented world-wide in a large number of water systems and incorporate most of the desirable attributes of a simulation model. WARGI (WAter Resources Graphical Interface) is a generic DSS for planning and management complex water systems developed at the University of Cagliari (Italy). The DSS is specifically developed to meet the system management requirements to satisfy the growing demands in multi-reservoir systems under water scarcity conditions, as frequently, happen in the Mediterranean regions. Sechi and Sulis (2009a) have recently developed a full integration of the simulation module WARGI-SIM and the linear optimization module WARGI-OPT in the DSS. This mixed simulation-optimization approach was proposed with the aim of identifying and evaluating mitigation measures in a proactive approach that anticipates the trigger of these actions.

The processes that govern the behavior of multireservoir water systems are affected by uncertainty that increases with time and space investigation scales (Simonovic, 2000). Uncertainty is mainly associated with the value of hydrological exogenous inflows and users demand patterns. A common disadvantage of the traditional modeling approach is the large number of system simulations required to achieve acceptable levels of confidence treating data uncertainties in the model. In fact, the proactive approach to drought mitigation must be based on a large number of hydrological scenarios that the water system might experience. In the case of computationally intensive software implementation, they can result extremely time and resources consuming and the applicability of these models is sometimes limited to relatively simple applications.

There's an upper limit to how fast a computer can complete an operation or how much information it can store. Most computers are upgradeable, which means it's possible to add more power or capacity to a single computer, but that's still just an incremental increase in performance. Grid computing is an innovative approach that leverages existing Information Technology (IT) infrastructure to optimize compute resources and manage data and computing workloads in a secure and collaborative environment: it links computer resources together in a way that lets someone use one computer to access and leverage the collected power of all the computers in the system. To the individual user, it is as if the user's computer has transformed into a supercomputer. Grid is not a new concept but one that has gained recent renewed interest and activity for Computing problems in several scientific and industrial fields. Grid involves processing large volumes of data and/or performing repetitive computations to the extent that the workload requirements exceed existing server platform capabilities. A grid is both hardware and software services, in the field of the advanced calculation and of the nets data transmissions to high speed network that allows to various geographic sites to share the own resources in dynamics and intelligent way and allowing the transparent, secure, controlled access by multiple users for public or private scientific and technologic research. Grids is built "on top of" hardware (like computers and networks), which forms the physical infrastructure of a grid. Networks link the different computers that form part of a grid, allowing them to be handled as one huge computer. Above the network layer lies the resource layer: actual grid resources, such as computers, storage systems, electronic data catalogues, sensors and telescopes that are connected to the network. The middleware layer provides the tools that enable the various elements (servers, storage, networks, etc.) to participate in a grid. Middleware is made up of many software programs, containing hundreds of thousands of lines of computer code. Together, this code automates all the "machine to machine" interactions that create a single, seamless computational grid. The highest layer of the grid architecture is the application layer, which includes applications in science, engineering, business, finance and more, as well as portals and development toolkits to support the applications. This is the layer that grid users "see" and interact with. A Virtual Organization (VO) represents a fundamental concept of Grid Computing technology: it is a group of grid users with similar interests and requirements who are able to work collaboratively with other members of the group and/or share resources (data, software, expertise, CPU, storage space, etc.) regardless of geographical location. A user will need to be a member of a VO before to be allowed to submit a request (properly called job) to the grid. Exceptionally, if an existing VO is not appropriate a new VO can be created.

An application that ordinarily runs on a stand-alone PC must be "gridified" before it can run on a grid. "Gridification" means adapting applications to include new layers of grid-enabled software. Once gridified, thousands of people will be able to use the same application and run it trouble-free on interoperable grids. Grid computing is the IT technology enabling worldwide scientific projects, such as the Large Hadron Collider (LHC) at CERN, and powering global efforts to combat climate change, discover new medicines, map the skies, reconstruct the sound of ancient instruments, covers some aspects of the preservation and the fruition of cultural heritages and so on. The Enabling Grids for E-sciencE (EGEE) project (http://eu-egee.com/) began by working with two scientific groups, High Energy Physics (HEP) and Life Sciences, and has since grown to support formally astronomy, astrophysics, computational chemistry, earth sciences, fusion, and computer science. The full user community runs applications from research domains as diverse as multimedia, finance, archaeology, and civil protection (http://grid.ct.infn.it/egee_applications/).

The specific benefits that can be achieved from grid computing are dependent upon the grid-enabled applications and the degree of implementation. In general, grid computing can be a cost effective way to resolve IT issues in the areas of data, computing and collaboration, especially if they require enormous amounts of compute power, complex computer processing cycles or access to large data sources, because of its benefits include infrastructure optimization, more access to data and increased collaboration.

This paper describes a research project funded by the Italian Ministry of Education, University and Research (PON-CyberSar Project) and the Regione Autonoma della Sardegna (PO Sardegna FSE 2007-2013 - L.R. 7/2007) on the porting of the WARGI-DSS to the Italian National GRID. Specifically, this paper focuses on the possible use of the combined simulation and optimization approach in the WARGI-DSS within the GRID environment for the definition of drought mitigation measures based on a large number of possible future system evolutions. It is organized as follows. In Section 2, an overview of the development and features of WARGI-DSS is presented. Section 3 describes the implementation of the GRID approach to satisfy the requirements of massive simulation-optimization runs. Sections 4 and 5 show a practical application to a complex water system in the Mediterranean area, and conclusions and perspective of future works.

2. An overview of the WARGI-DSS

2.1 The WARGI-DSS structure

WARGI-DSS is a user-friendly tool specifically developed to help users understanding interrelationships between demands and resources for the management and planning of multi-reservoir water systems under water scarcity conditions, as frequently happen in the Mediterranean regions. The DSS makes it possible to take into account a large number of system components that typically characterize water resources models. The tool is flexible and generalized in the system configuration and data input, in the attribution of planning and operating policies and in processing output. Moreover, the software modularity allows easy coding changes and the addition of new objects and features in the system diagram. The WARGI-DSS modeling capability includes several interrelated macro-modules implemented in C++ and Tcl/Tk (Figure 1), the main ones being: the Graphical User Interface (GUI) module, the System initialization and Data Input module (SDI), the SIMulation module (WARGI-SIM), the deterministic OPTimization module (WARGI-OPT), the QUALity optimization module (WARGI-QUAL), the SCENario optimization module (WARGI-SCEN), the Solver modules and the Result Evaluation and Output plotting module (REO).

As illustrated by Manca et al. (2004), in the GUI module there are procedures that create and set the various graphic objects of WARGI: the canvas, the palette, the menu bar and the relative drop down menus, the scroll bars and the state bar. The SDI module handles the values definition of the main parameters and the creation and possible modification of system elements. This module processes data coming from the GUI module, transfers data required by the simulation module WARGI-SIM and implements the optimization algorithm WARGI-OPT. The construction by means of independent modules also makes it possible to use the DSS either for system optimization alone or for simulation alone.

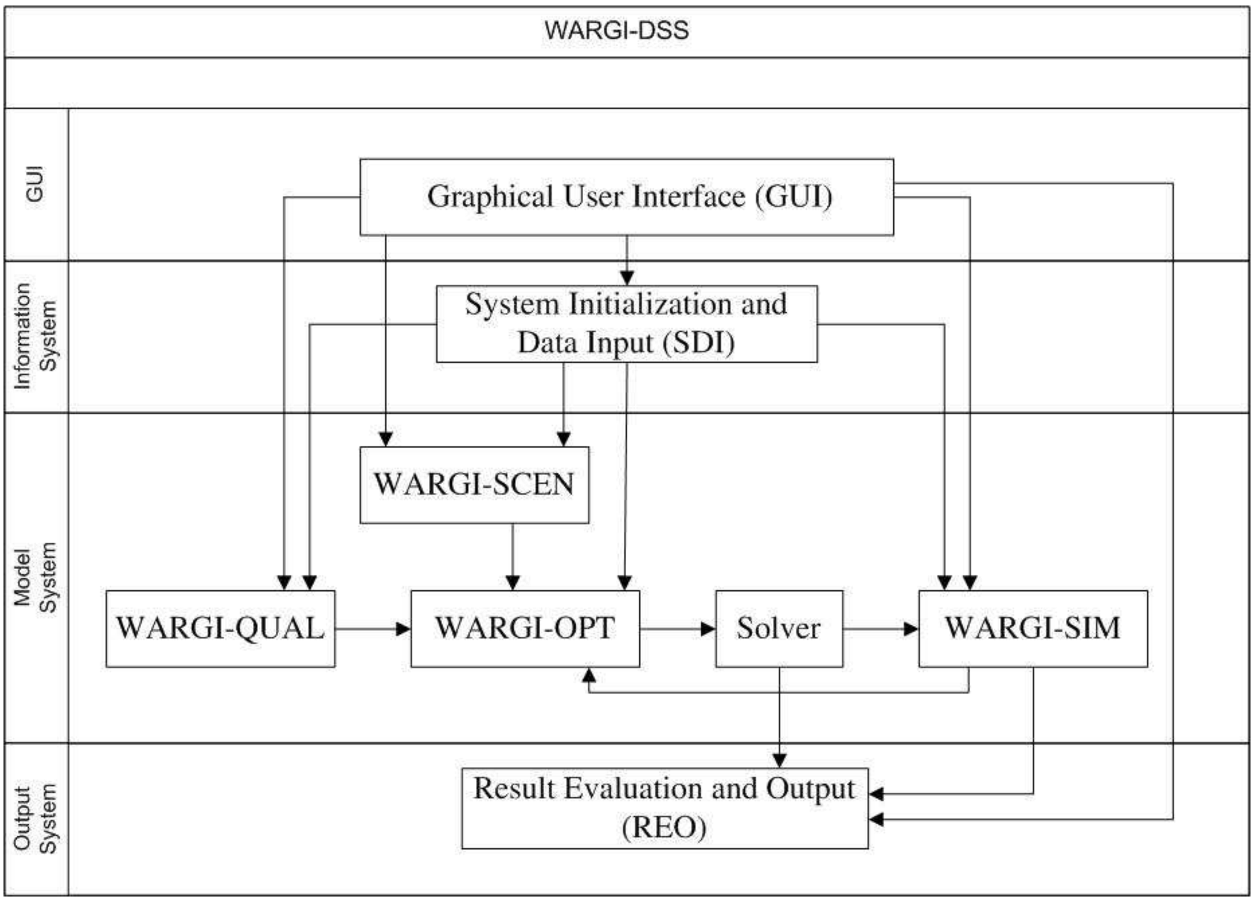

WARGI-DSS
GUI
Information System
Model System
Output System
Graphical User Interface (GUI)
System Initialization and Data Input (SDI)
WARGI-SCEN
WARGI-QUAL
WARGI-OPT
Solver
WARGI-SIM
Result Evaluation and Output (REO)

Fig. 1. General structure of WARGI-DSS.

When considering a single scenario in a deterministic analysis, WARGI-OPT models the problem by Linear Programming (LP) or Quadratic Programming (QP). The optimization model is supported by a multi-period dynamic graph (Sechi and Zuddas, 2007) and determines the construction of a file in the MPS (Mathematical Programming Standard) format to be submitted to a solver. The MPS is supported by efficient commercial and non-commercial mathematical programming computer codes. WARGI has been particularly tested using CPLEX (2006) and Lp_Solve (http://lpsolve.sourceforge.net/). Specific procedures in WARGI manage the dynamic link with the solver and allow to select the solving code.

If analysis with scenario optimization is required, WARGI-SCEN module passes to the WARGI-OPT module (Pallottino et al., 2005) parameters for model construction to consider different system evolutions. Moreover, water quality optimization considering synthetic quality indexes for water sources and demands is implemented in the WARGI-QUAL module (Sechi and Sulis, 2007b, 2009b).

WARGI-SIM does not require the input of specific operating rules (Loucks and Sigvaldason, 1982), but the definition of "preferences" and "priorities" by the user (Sechi and Sulis, 2009a). As usual, the preliminary requirement of WARGI-SIM is to represent in the model all the features the user thinks are important with respect to the objectives of the study. This enables WARGI-SIM to identify the technical and economic constraints of the system being modelled.

While the modular architecture allows the user to analyse multiple alternatives with the simulation or optimization alone, it is often advantageous to use both WARGI-SIM and WARGI-OPT. Sechi and Sulis (2009a) have recently proposed a combined optimization-simulation technique in WARGI-DSS. In the combined approach, WARGI-SIM processes the flows configuration provided by WARGI-OPT when foreseeing for many different hydrologic scenarios. The optimization results can be seen as a reference target for the simulation phase and WARGI-SIM can define drought mitigation measures that anticipate the occurrence of water scarcity. These measures are pre-emptive measures developed in a proactive approach to face drought events (Yevjevich et al., 1983; Rossi, 2000; Sechi and Sulis, 2007a).

This mixed optimization-simulation approach has to manage a large number of generated hydrologic scenarios and operating rules that might be tested in the simulation model and the GRID computing is very attractive to face this complex computation problem.

Previous papers have shown the wide applicability of GRIDs to solve large-scale computations and data-intensive computing problems in environmental systems. GRIDs appear to be promising in the optimization and simulation of complex systems but few applications have focused on the analysis of multi-reservoir systems (Sulis, 2009).

2.2 The data structure

WARGI-DSS represents a water resource system using a direct network (graph) that consists of nodes, both storage (water source) and non-storage (water demand, and hydropower, treatment, pumping plants), and arcs (pipeline, canals or natural rivers, and special arcs) (Figure 2). Nodes and arcs represent these physical and hydrologic features, but also symbolize artificial elements for modeling events (i.e. water scarcity conditions). These artificial elements are added automatically by WARGI-DSS. Each graphic element of the graph is denominated GOB (Graphical OBject) and is identified by both the type of GOB and the “identifying value” called Tk-id. Each element is inserted in a list of active GOBs.

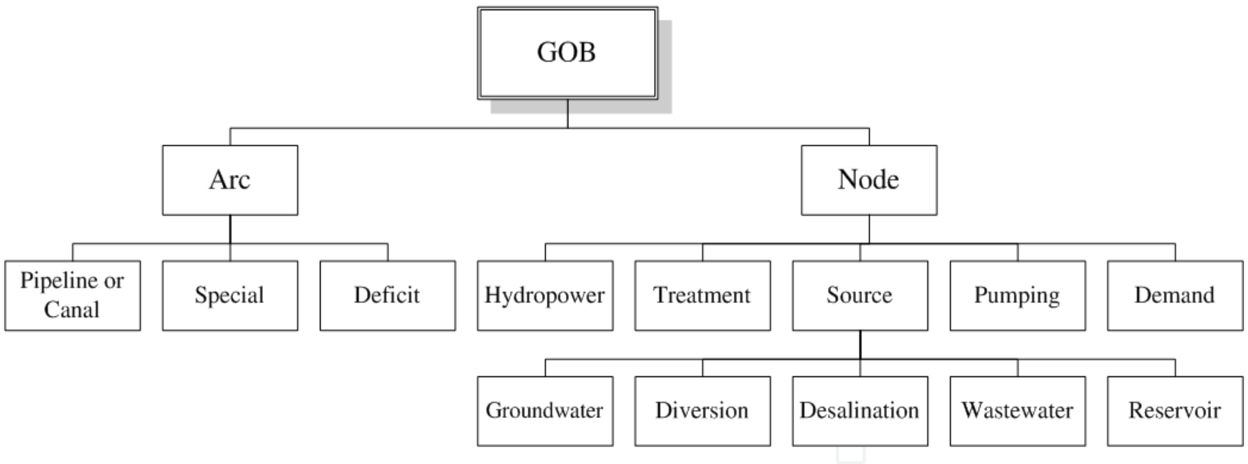


Fig. 2. GOB types.

A hash table contains the information associated to the GOB. The hash table is represented as a structure C of Tcl_HashTable type. The hash table associated to a GOB is generated as the GOB is created. The WARGI-DSS modules use scripts and procedures written in TCL/TK or C++ to communicate with hash tables getting or storing GOB data. Scripts and procedures are provided for the research and modification of old entries and the creation of new entries.

WARGI uses keys of the char* type to quickly localize its entry in the hash table. Such a key is associated to a value which is a pointer to the initial position of the hash table containing the information on the GOB attributes. The GOB attributes can be of a scalar, cyclic or vector type and are stored inside the structure defined in C++, initialized by the information contained in the TCL variables. The TCL variables of a GOB contain the values inserted directly by the user with the graphic interface or through linked data files.

The access to the hash table is through the TCL library procedure, written inside the procedures C++; these allow the possibility of accessing information on the attributes of the GOB directly by the execution of script TCL, or directly through the calling of function C or C++. The communication between variables TCL and C/C++ is guaranteed by the mechanism called "variable linking" which allows the value of the TCL variables to be associated to the C/C++ variables and vice versa. Such a mechanism allows different variables to share the same values.

In the graph of the water system the correspondence node-arcs is represented by a matrix of adjacency. Every node maintains a memory of a dynamic list which points at the first outgoing arc and a dynamic list which points at the first incoming arc. Such a list is updated every time a new arc is created or destroyed.

The information on the global variables (year-length simulation and optimization horizons, number of periods per year, optimization-simulation linking period) is stored in the TCL and C/C++ global variables. Such variables and the GOB attributes set up the case to be resolved. Decision variables are TCL and C/C++ local variables. These variables are an array of a number of dimensions based on the temporal scale chosen by the user.

The information of the optimization model can be saved in a file with idr extension. When the idr file is opened, all the GOBs are created and all the GOBs attributes in the data structure are uploaded. In such a case, the graph is showed in the main window of the graphical user interface (Figure 3). The TCL/TK interpreter and library procedures manage the data regarding the graphic layout of elements such as line, color, masks, graphs and zoom effects.

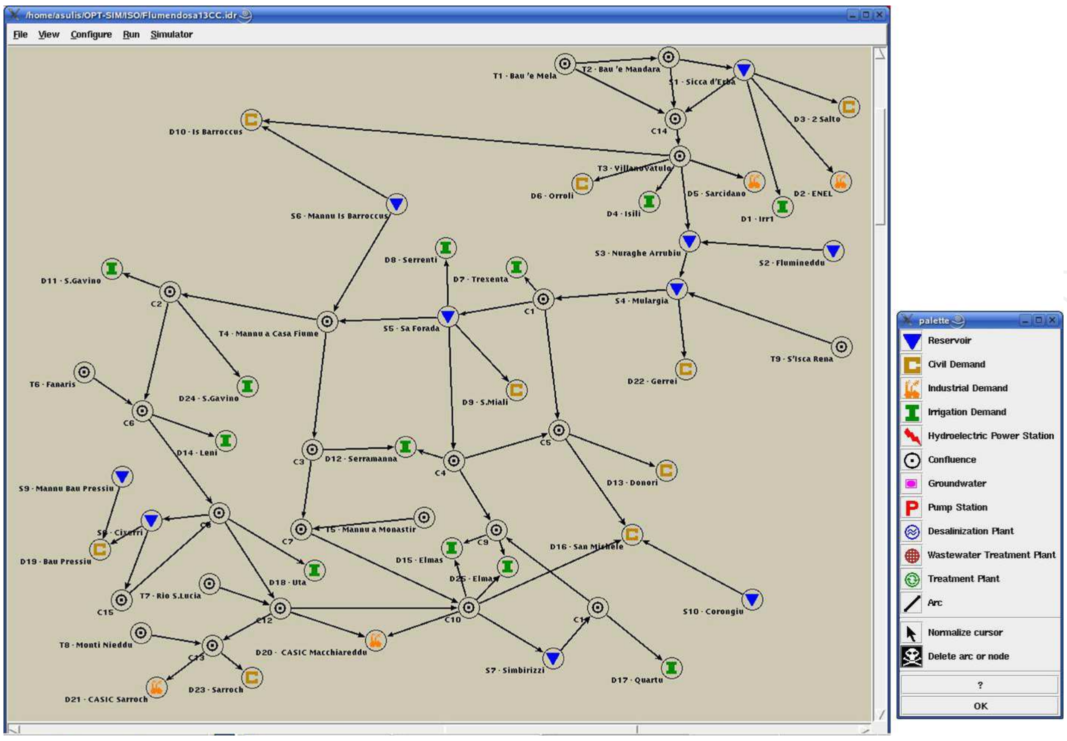
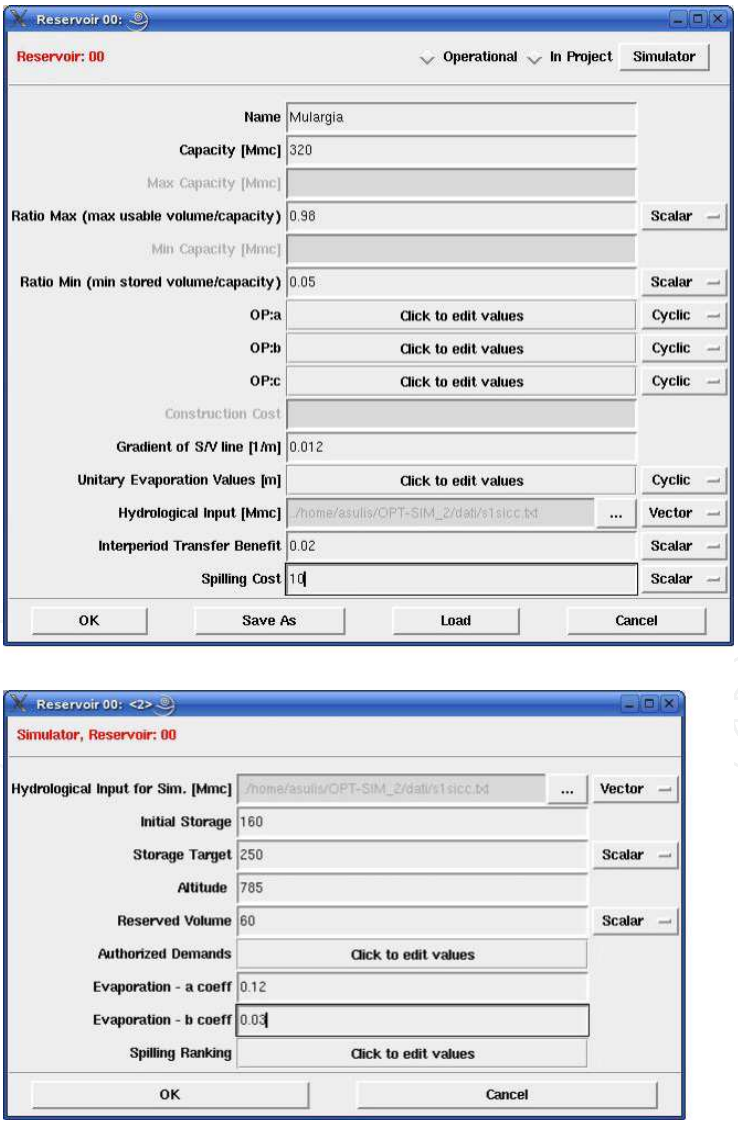


Fig. 3. Main windows in the GUI.

2.3 The graphical interface

WARGI-DSS employs an interactive graphical user interface that permits easy data entry and display, and control of model operations. A specific procedure creates and sets the graphic object of the GUI: the canvas, the palette, the menu bar and the relative drop down menus, the scroll and the state bars. The canvas allows user to create, locate and connect the components of the water system as shown in Figure 3. The core of the canvas is the graph of nodes and arcs (GOBs). The graph reflects the spatial relationships between the elements of the water system. A GIS layer can be added to provide clarity and impact. A central feature is the "drug-and-drop" that allows user to easy create and link GOBs. Each GOB must be associated with a hash table consisting of as many fields as the number of attribute in the GOB. The user can activate the GOB, open a window associated with that GOB and populate the hash table with the attributes of the GOB. The window of the reservoir GOB is shown in Figure 4 and the corresponding hash table must contain numerical information on the physical, hydrologic and management attributes of the reservoir. These attributes can be mandatory or optional; the latter may be included to further describe the reservoir.

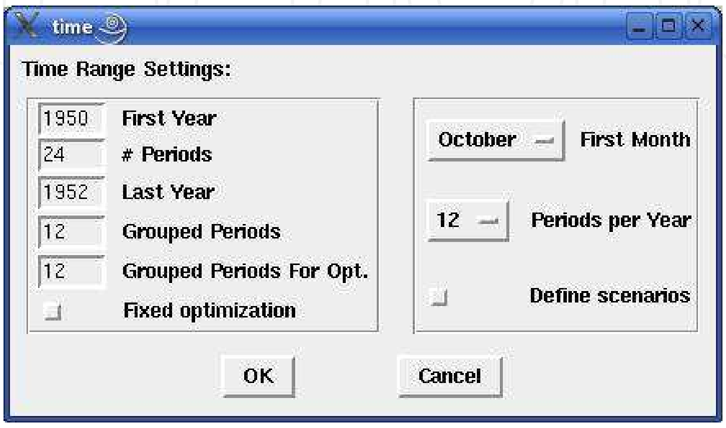


Name	Mulargia	
Capacity [Mmc]	320	
Max Capacity [Mmc]		
Ratio Max (max usable volume/capacity)	0.98	Scalar
Min Capacity [Mmc]		
Ratio Min (min stored volume/capacity)	0.05	Scalar
OP:a	Click to edit values	Cyclic
OP:b	Click to edit values	Cyclic
OP:c	Click to edit values	Cyclic
Construction Cost		
Gradient of S/V line [1/m]	0.012	
Unitary Evaporation Values [m]	Click to edit values	Cyclic
Hydrological Input [Mmc]	./home/asulis/OPT-SIM_2/dati/s1sicc.txt ...	Vector
Interperiod Transfer Benefit	0.02	Scalar
Spilling Cost	10	Scalar

Hydrological Input for Sim. [Mmc]	./home/asulis/OPT-SIM_2/dati/s1sicc.txt ...	Vector
Initial Storage	160	
Storage Target	250	Scalar
Altitude	785	
Reserved Volume	60	Scalar
Authorized Demands	Click to edit values	
Evaporation - a coeff	0.12	
Evaporation - b coeff	0.03	
Spilling Ranking	Click to edit values	

Fig. 4. Reservoir window in the GUI.

The menu bar provides access to the most important features of the DSS. The right-button mouse click on links opens the File (structure of the files), View (options to graphics), Configure (time settings and choice of optimization solver), Run (solver launch) and Simulation (simulation alone or mixed simulation-optimization) pull-down menus. Under the Configure menu, the Time menu allows the user to set the global variables (Figure 5) and the Solver menu to choose the solver. The Run and Simulation menus allow to use the DSS for system optimization alone (WARGI-OPT) or for simulation (WARGI-SIM), respectively.



Time Range Settings:

1950	First Year
24	# Periods
1952	Last Year
12	Grouped Periods
12	Grouped Periods For Opt.
□	Fixed optimization
October	First Month
12	Periods per Year
□	Define scenarios

Fig. 5. Time window setting in the GUI.

2.4 The methodology

To perform the analysis of a water system, the user has to input the values of the global variables in the Time Menu (Figure 5):

T: time horizon of simulation for overall system analysis;
Δ: time horizon of optimization;
t=1,T: time step of simulation;
∂=1, Δ: time step of optimization;
$\tau = \tau_1, \tau_n$: period of simulation-optimization synchronization.

The WARGI-OPT module forecasts the system evolution on the time horizon Δ at each synchronization period τ_i based on the state indicators of the system [I] and a user-selected future hydrological synthetic scenario [b_g] (Figure 6). The state indicators [I] are used to trigger reactive measures; they are decision variables of the source GOBs. In multi-reservoir systems, the state indicators [I] are usually the reservoir storages. The hydrological scenarios [b_g] are attributes of the reservoir GOBs.

The linear optimization model can be therefore written in the following form:

$$\min_{t=(\tau,\tau+\Delta)} \quad (c_Y Y + c_i x_i + c_j x_j) \tag{1}$$

$$\text{s.t.} \qquad A[x] = b_g \tag{2}$$

$$F(Y,x) \geq 0 \tag{3}$$

$$l \leq x \leq u \tag{4}$$

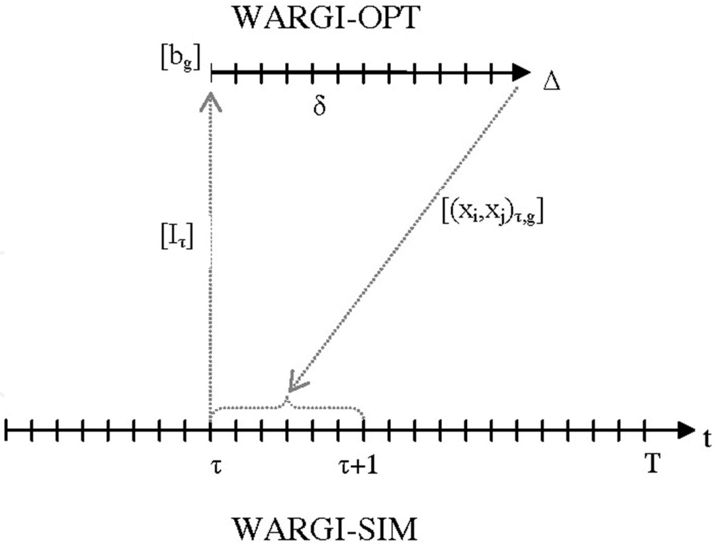


Fig. 6. Link between WARGI-OPT and WARGI-SIM (adapted from Sechi and Sulis, 2009a).

The GOBs attributes [c_Y] are costs related to the decision variables of GOBs in project [Y], [c_i] represents operative, maintenance, and replace costs (OMR) or user-defined costs of arc GOBs, and [c_j] represents deficit costs based on priority ranking of demand GOBs. The decision variables [x_i ,x_j] are the subsets of the variables [x] related to flows of pipeline or canal GOBs and to deficit of special arc GOBs, respectively. The GOBs attributes l and u are the lower and upper bounds on decision variables [x_i ,x_j].

In the WARGI-SIM module, preferences and priorities [v] are attributes of sources and demands GOBs, respectively. Decision variables [x_i ,x_j], and preferences and priorities [v] defines proactive mitigation measures [z_τ] that are attributes of sources and demands GOBs:

$$z_\tau = f_1\left(\left[x_i, x_j\right]\ , \nu\right)_\tau \qquad \tau = \tau_1, \tau_n \tag{5}$$

The decision variables of water allocation [X_t] in the system are the solution of a minimum cost flow problem between source and demand GOBs. These preemptive measures [z_τ] can modify the water allocations [X_t] from those previously defined using the attributes of allocation rules [r] of arc GOBs and user-defined preferences and priorities [v]. Consequently, during the subsequent periods until τ_{i+1}:

$$X_t = f_2\left(z_\tau, \nu\ , r\right) \qquad t = (\tau_i, \tau_{i+1}) \tag{6}$$

In the case of water scarcities more severe than those forecasted by WARGI-OPT, the preemptive measures [z_τ] do not make it possible to overcome the water scarcity, and WARGI-SIM introduces further restriction measures [s_t] in a reactive approach. [s_t] are attributes of source GOBs. These reactive actions are defined following the state indicators of the system [I_t], the user-defined preferences and priorities [v], and the pre-defined water allocations [X_t]:

$$s_t = f_3(I_t, X_t, \nu) \qquad t = (\tau_i, \tau_{i+1}) \tag{7}$$

The goal of this mixed optimization-simulation approach is to define the best combination of drought mitigation measures that minimizes the economic impact of drought in the water supply system. The economic response function R is the sum of the costs associated with the construction of new works in the system ([C_Y]), OMR costs ([C_{OMR}]), and costs related to mitigation measures ([C_{PD}] and [C_{NPD}]):

$$R = C_Y\, Y + \sum_{\tau_i} C_{PD} z_\tau + \sum_{t=1}^{T} C_{NPD} s_t + \sum_{t=1}^{T} \min\left(C_{OMR} X_t\right) \tag{8}$$

[CP_D] and [C_{NPD}] are associated with the drought mitigation measures in the proactive and reactive approaches, respectively. [CP_D] and [C_{NPD}] include the OMR costs for drought measures, agency income lost from reduced water sales, and reduced consumer surplus due to these measures.

3. The porting of WARGI-DSS to GRID computing

Uncertainty due to inherent randomness of hydrological events cannot be eliminated. Some of this uncertainty can be incorporated into models. Stochastic models have been applied to numerous water resources planning and management problems. Clearly if the system being analysed is very complex, the stochastic models are not robust enough to be applied to describe all complexities in the real world problem. An alternative approach is to solve a set of deterministic optimization models for each fixed interval over which the uncertain values can be discretized. A simulation phase is then performed to refine the optimization results (Loucks and van Beek, 2005) and to obtain a management policy for the system. In Sechi and Sulis (2009a) the proposed mixed optimization-simulation approach includes the hydrological uncertainty issue considering a large number n of synthetic hydrological series (b_g in (2)). The model (1-8) was solved n-times considering one hydrological series at once to provide n sets of drought mitigation measures. The main disadvantages of that approach were:

1. Each set of drought mitigation measures was strictly related to the selected hydrological series. An ex-post sensitivity analysis was then carried out to estimate the economic impact of hydrological uncertainty on water system management;
2. The approach becomes rapidly time-consuming as the number of hydrological series increases.

In the proposed GRID approach (Figure 7), WARGI-SIM considers together n-sets of decision variables [x_i, x_j] (5) from the solution of n linear optimization models in the GRID environment. A weight is assigned to each set of decision variables [x_i, x_j]. A unique set of drought mitigation measures [z_τ] on the number n of synthetic hydrological series [b_g] is then tested over the time horizon between the two synchronization periods τ_i and τ_{i+1}.

Each model (1-4) is solved by the linear programming solver LpSolve downloaded on the Computing Elements that satisfy the job requirements. As expected, the user can get a significant reduction of computational time saving by applying the proposed GRID approach instead of the traditional local approach. Sulis (2009) discussed this computational advantage for the Implicit Stochastic Optimization (ISO) where the simulation and optimization models, not fully integrated, were used to define multireservoir operating rules, basically the optimum storage trajectories.

This article presents the porting of a DSS under which the simulation and optimization models have been implemented and fully integrated with the specific aim of defining drought mitigation measures. Details of the porting to the GRID environment of this integrated approach are presented in this section.

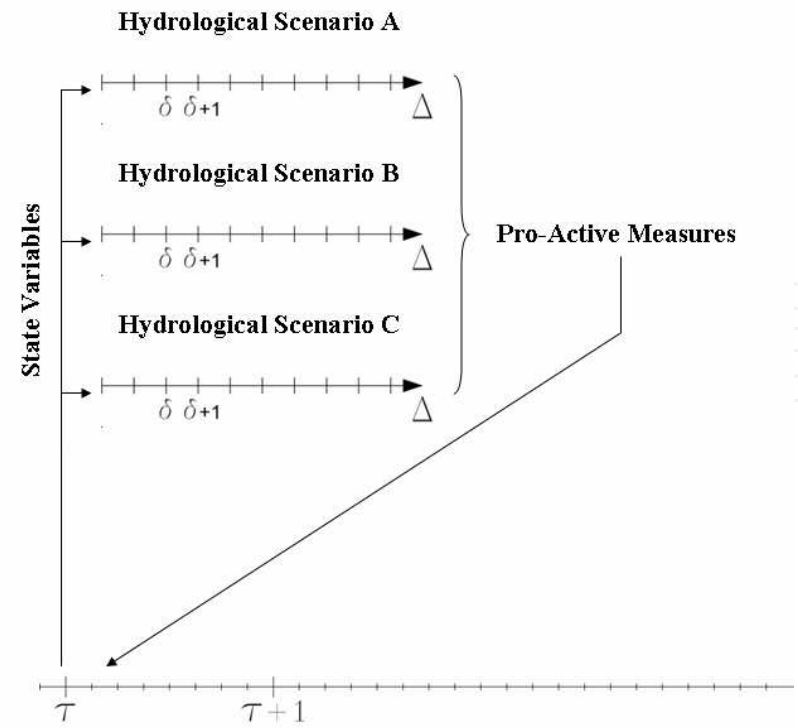


Fig. 7. Link between WARGI-OPT and WARGI-SIM in the proposed GRID environment.

3.1 Porting of the mixed optimization-simulation approach in WARGI-DSS

The middleware chosen for the implementation of the optimization-simulation approach is gLite. gLite is a full featured grid middleware stack, that provides to the user capabilities for interactions with grid elements at any level of complexity. In fact with gLite users can exploit directly resources, at low level, or let the so called central services the task of selecting, among those available, the most appropriate resource for the execution of the computational task. Clearly this second option is the most powerful, because it really enables the user to access the whole set of resources available.

In grid terminology, a computational task is defined as a job. In gLite, the WMS (Workload Management System) is the component deputy to accept users' job request, and dispatch them to the most appropriate resource. Users formalize the request through a declarative language, the JDL (Job Description Language), which allows them to specify job characteristics (as the executable name, additional files for the job execution, further requirements about resources, the output files to be retrieved…). The request for the job execution is the provision to the WMS of the file containing the job description; as outcome of the request, a job identifier is returned. This job identifier is eventually used for monitoring purposes, and finally for the job output retrieve, by the user.

Using the gLite middleware, a request for a job execution can trigger more than one computational task. Job of this type are called compound jobs, because they are actually made of sub-jobs, that can be still monitored and managed as if they were a single one, through a single job identifier. The advantage of this approach is that in case of big input files, they are transferred once to the resource, providing so an optimization of the bandwidth usage. Further, relationship among the sub-jobs can be expressed, allowing so the implementation of job workflows. For the WARGI-DSS porting, a simple compound job has been used, without relationship among the sub-jobs. This type of compound job is called

job collection, as they express a set of computational tasks having some common input. In this case, the job forming the collection have the common part made by the solver, while the sub-jobs differs each other for the different instance of the optimization to be solved.
There are two types of tools directly provided by gLite for the interaction with services: the Command Line Interface and the Application Programming Interface. The first one is a classic set of executables, in Unix style, invoked from the shell, while the second is a set of libraries, generally in C/C++ or Java. In the first case, users generally adapts their own applications such to make system calls invoking the CLI executable. Further, the output given from the CLI to the shell has to be parsed and adapted again for the application. While this approach is simple, and yet very powerful, allowing any kind of interactions, has the limit of the considerable computational overhead taken by the command line calls made from inside the application. For this reason, the approach with API is sometimes preferable, because it carries better performances and a cleaner design of the application interacting with the grid environment. Using the API approach, the users recompile their own application including the API libraries. Further interactions with grid services are then made using only structures and functions provided from the API, which is clearly cheaper, in computational terms, than invoking CLI through a system call. Further, there is no need of parsing executable outputs, because when services are invoked through API functions, their output are still API objects that can be easily dealt inside the native application. So, paying the price of a longer develop time, because application source code needs to be adapted, API's provide better performances, and a cleaner design of the grid application. The effort needed for the inclusion of API libraries is reduced by the choice of the embedding of only the needed functions, rather than the whole set of gLite API. For the WARGI-DSS porting, have been so exploited API functions for Job Submission, Job Status Control, and Job Output retrieve.
The Job Submission function, takes as input a Security Context, i.e. the location of user credential, the URL of the WMS service and the JDL which describes the submitting job. So, if they don't exist or they are unknown, all these parameters have to be generated before the execution of an instance of the application, but then can be then reused for further instances. If Submission is successful, the WMS register the job to the LB and gives back its job identifier, which is a string to be used for further queries about the job status. The Logging and Bookkeeping service (LB) is a central registry, where are logged all the steps of the job life cycle, for instance when it has been assigned to a resource, or when it's being executed or if some errors happens. LB so can be queried at any time in order to discover the job status and whether errors happened. Once the job collection is done, the resources where the jobs have been executed copy back the job outputs to the WMS; resources also register the Done status for the job. When performing the next query, the application will find the Done status, and, by means of Job Output retrieval, proceed to the download of the Output Sandbox, which contains the solutions as computed from the solver.

4. The application to Flumendosa-Campidano water system: results and discussion

The Flumendosa-Campidano water system (Figure 8) is located in the South Sardinia, Italy. The hydrologic regime is characterized by significant variation in both monthly and annual flows. The system is composed by 10 reservoirs that are expected to regulate the unstable streamflows and to assure sustainable supplies to nine urban, two industrial and eleven irrigation demands. Data of monthly streamflows into the reservoirs and at the diversion sections in the streams were provided by the Sardinia water authority (SISS, 1994) over a period 54 years in length. The urban, industrial and irrigation demands were also obtained from the authority and the average values per year are equal to 116, 19 and 224 millions of cubic meter, respectively. Table 1 shows the main statistical properties of the available hydrological series at the sites of interest.

Site	Min (10^6 m^3/yr)	Max (10^6 m^3/yr)	Mean (10^6 m^3/yr)	Standard Deviation (10^6 m^3/yr)
Sicca d'Erba	2.28	89.91	19.41	16.26
Flumineddu	4.32	177.86	46.19	37.18
Nuraghe Arrubiu	3.42	100.84	31.01	21.32
Mulargia	1.68	53.61	15.73	11.86
Sa Forada	0.00	0.26	0.08	0.07
Is Barroccus	1.11	32.74	12.28	8.85
Simbirizzi	21.7	22.79	22.06	0.25
Cixerri	2.59	102.03	32.88	23.58
Bau Pressiu	0.03	11.33	2.96	2.75
Corongiu	0.42	15.01	4.29	3.16
Bau Mela	5.84	92.33	24.84	17.22
Mau Mandara	1.24	26.23	5.68	4.44
Flumendosa	15.6	217.74	94.12	49.54
Casa Fiume	0.95	83.50	23.34	19.51
Mannu Monastir	1.72	74.75	24.06	19.08
Fanaris	0.52	20.42	5.90	4.25
Santa Lucia	0.40	15.89	5.47	3.61
Monti Nieddu	1.79	54.57	16.51	12.85
S'Isca Rena	4.24	141.60	51.70	36.39
System	134.80	1032.7	438.5	239.35

Table 1. Main statistics of inflows to reservoirs and diversion points in Flumendosa-Campidano water system.

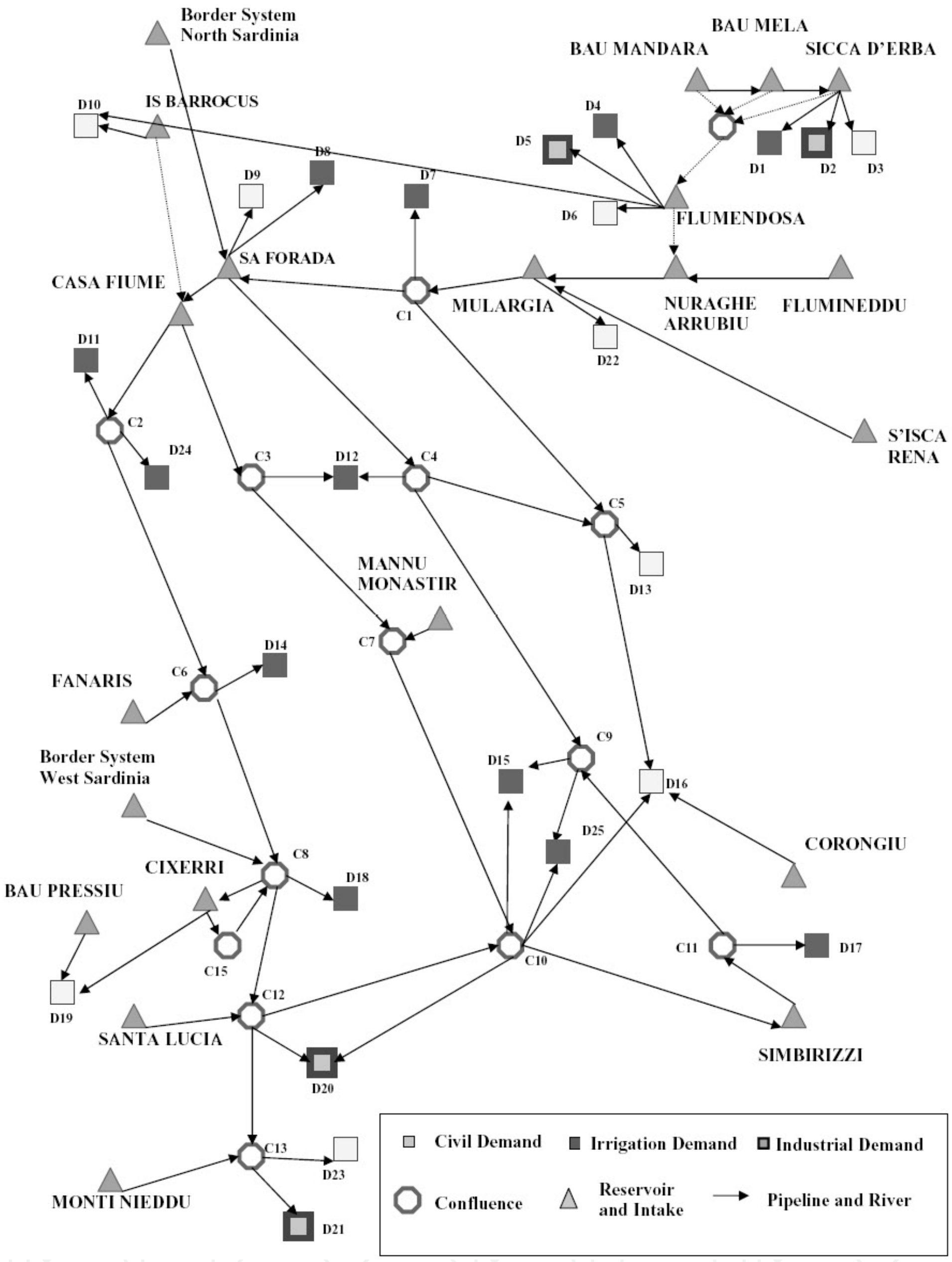


Fig. 8. Network representation of Flumendosa-Campidano water system.

Measure	*[z]*	*[s]*
Supply Increase	Increase of available resources	Use of additional sources
Demand Reduction	Pricing and Use Reallocation	Restrictions and Rationing
Impact Minimization	Plans, Warning Systems and Education Activities	Temporary Resource Reallocation

Table 2. General classification of drought mitigation measures currently applied or planned.

The increase in demand (particularly urban and agriculture) and the intense water scarcity events verified in recent years, require the water authority to revive drought mitigation plans to guarantee fulfilment of high priority demands during drought conditions. The success of these plans hinges on increasing the water system reliability, which is subject to hydrologic uncertainty probably related with long-term climate changes.

Table 2 shows proactive and reactive measures currently applied and planned in the Flumendosa-Campidano water system. The WARGI DSS has been applied to the system to provide the water authority of the Flumendosa-Campidano with a first estimate of the relationship between hydrologic uncertain data and drought mitigation measures.

The water authority often need to identify the sensitivity of the system economic performance associated with changes in hydrology and to quantify the consequences of alternative hydrological assumptions about the future. Consequently, the analysis of the system should be done with a wider range than, as usual, the only "best guessing" hydrological series or the "worst-case" sets. In Sechi and Sulis (2009a), the 2 years hydrological series were counted and selected in descending order of total flow from the historical records until 7 hydrological series were reached. They applied 7-times the mixed optimization-simulation approach obtaining 7 sets of drought mitigation measures at each synchronization period τ_i. Each set contained both proactive and reactive measures: the proactive measures were selected based on the results of LpSolve under the selected hydrological series, 2 year length, whereas the proactive measures [z] were tested and reactive measures [s] eventually were implemented by WARGI-SIM under the historical record between the synchronization periods τ_i and τ_{i+1}. Each set of drought mitigation measures was strictly related to the selected hydrological series. The best set, that is the set that minimizes the response function R (8) over the period of 54 years, was found through an ex-post sensitivity analysis. Figure 9 summarizes the results of this ex-post economic analysis showing the values of response function R of the Flumendosa-Campidano system for each of the seven selected hydrological series. Results showed that the "best guessing" produces a potential expensive mitigation plan, whereas, the "worst-case" set is very conservative. In particular, when the hydrological series are correctly assumed, R equals 1.3 million Euros per year with significant economic savings in respect to series with the lowest (too much pessimistic) or highest (too much optimistic) values foreseen for future hydrology (1.89 and 1.57 million Euros per year, respectively).

In the GRID approach those 7 linear programming models related to the selected hydrological series 2 year length, are solved at each synchronization period τ_i by LpSolve running in different Computing Elements. Results are weighted using the information provided by the Flumendosa-Campidano water authority, a weight representing the "importance" assigned by the authority to the running model. A unique set of proactive measures [z_τ] based on these weighted results is then tested and a unique set of reactive measures is implemented by WARGI-SIM until the next synchronization period τ_{i+1}. Over the 54 time period of Flumendosa-Campidano system analysis, the value of the economic function R equals 1.39 million Euros per year. While the GRID approach provides a set of drought mitigation measures a little more expensive (+6.9%) than the best set in Sechi and Sulis (2009a), the presented approach does not need of an ex-post sensitivity analysis, quite subjective, and more important, it attempts to face the uncertainty issue by taking into account all different system evolutions corresponding to the selected hydrological series.

In the local approach of Sechi and Sulis (2009a), the software ran on an Intel Pentium with Single CPU clocked at 1.60 GHz having 1GB RAM. As shown in Figure 10, the computation time in the local approach vastly increased and the software simulated the system in about 21 min when the seven selected hydrological series were considered in the optimization phase. At that time of the presented research, there were 10 Intel Pentium Dual CPU running Scientific Linux SLC in the GRID authorized for a given job. The total Grid computation time is defined by the Grid latency (the time needed to submit a complete set of jobs, and collect a complete set of results), the waiting time in case a job is delayed in the Grid queues and the running time on the nodes. Figure 10 shows that the total GRID computation time at different hydrological series remains approximately stable (~ 8 min). The GRID approach seems promising for more than 2 hydrological series and the benefit improved as the number of hydrological series increases. The total computation time was reduced more than 60% in the case of seven hydrological series.

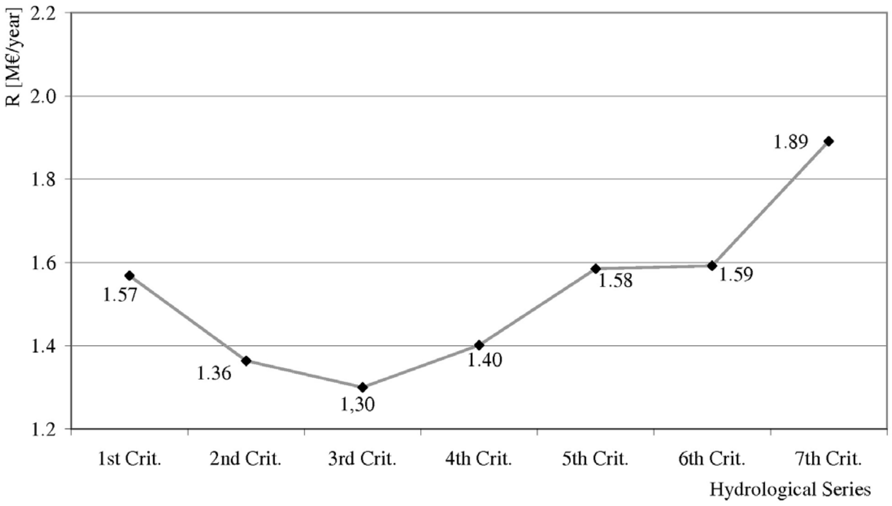


Fig. 9. Response function trend of the system for different hydrological series.

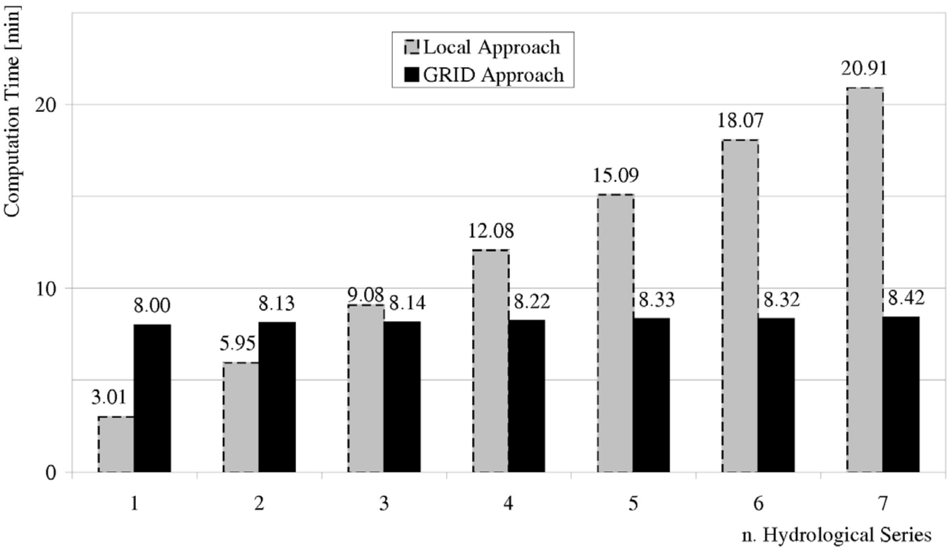


Fig. 10. Response function trend of the system for different hydrological series.

5. Final considerations on time-saving in a different GRID environment

In a more realistic GRID environment, where resources are shared with other VOs, their status is not known in advance as in the presented application, and there may be many factors that could affect the significant time saving of applying the proposed approach in a GRID environment. If we keep constant the number of operation totally executed by a collection, the global computation time decreases proportionally with the number of jobs executed in parallel in the GRID nodes (until a certain threshold is passed, as we will see). In addition, jobs could run simultaneously on nodes in the GRID with different hardware configurations, if jobs have no data or logic dependency. As the number of jobs increases, the heterogeneity of resources makes the jobs' running time different. Finally the latency time in a generic GRID environment would increase proportionally to the number of jobs to be submitted to the nodes. So it is rather expected that, as long as more jobs are added to the collection, the global execution time will be higher. However, this is not necessarily true, because the added jobs could be assigned to quick resources, thus not raising the global execution time.

The global computation time (T_C) for a collection of n jobs can be estimated as the sum of job latency times plus the time requested for the slowest job to be executed (in the meanwhile we expect other jobs to be done):

$$T_c = \sum_i T_{l,i} + max\left(T_{e,i}\right) \qquad i = 1,n \tag{9}$$

The above expression suggests that the addition of a small number of jobs to the collection, might not increase significantly (or could not increase at all) the global running time. Let's try to estimate, in average, how many jobs (each of them representing an hydrological series) is convenient to add, before the performance will certainly decrease, and will no longer convenient to push further with parallelism.

To begin, we have to consider that for each job that is added to the collection, a small increase in latency time should be accounted. The latency time has been in fact defined before as the time needed to submit a collection, plus the time needed to collect its outputs. The times needed for the job submission and matchmaking don't need to be accounted, because in the job collection implementation, there is always one submission and one matchmaking, independently from the number of jobs actually forming the collection. The submission and matchmaking times can be so considered constant. The augment led by the addition of a job to the collection is given only by the increased time for the retrieve of this job's output. It is worth noting also that the "weight" of submission, matchmaking and job output retrieve is much minor respect to the execution time; it can be considered relevant only when dealing with a very large number of jobs (more than 200). So it's better to focus on the time that each job spends on resources. This quantity can be computed as the time that each job waits in the resource queue, plus its real running time. As said before, the addition of a single job to the collection, not necessarily increases the global execution time. However, the higher is the number of added jobs, the bigger is the probability that one of them is assigned to a slow resource. We can define slow resources those Computing Elements either non efficient (long running time), or busy, where jobs there scheduled have to wait a long time before they are actually ran (long queuing time). Is reasonable to expect that, as long as we add jobs to the collection, the 'quality' of them, either in terms of performance or busyness won't be improving. Although this is not always true, because for instance 'good' resources could have been released in the meanwhile by other jobs, we say that in general as long as we add job to the collection, its global execution time will be growing. Since it's not known the number of resources in advance, and their availability status, a rule of thumb is that collection size should not exceed half of the free CPUs available, at the submission, for the testbed. Otherwise, it's likely that one or more of the collection jobs will be assigned to a slow resource, degrading the global execution time. A ringing bell could be the fact that waiting time in the Computing Element queue has exceeded the running time, and this is indeed easy to spot, as we approximately know in advance how much time our jobs should be running.

6. Conclusions

Results demonstrate that GRID computing is a promising approach for scaled-up simulation analysis of complex water system with high uncertainty on hydrological series. However in the authors' view, there are still some hindrances for an extensive use in the water system field. A larger number of hydrological series used in the optimization phase to account for all the relevant statistics of drought events and different weights for the correspondent LP models should be of interest. Future research in the area of GRID approach for this combined optimization-simulation model also include its application in a more realistic GRID environment to estimate how many hydrological series is convenient to add before the performance of the GRID approach will certainly decrease.

7. Acknowledgements

The Authors gratefully acknowledge the support received from the Italian Ministry of Education, University and Research through grant under the PON-CyberSar Project (National Operation Programme Scientific Research, Technological Development, Higher Training – Activity 11.2: High Performance in Data Process and Simulation) and the Regione Autonoma della Sardegna that supports the final part of this research under the PO Sardegna FSE 2007-2013 (L.R.7/2007 – Promozione della Ricerca Scientifica e dell'Innovazione Tecnologica in Sardegna).

8. References

Andreu, J.; Capilla, J. & Sanchìs, E. (1996). AQUATOOL, a generalised decision-support system for water-resources planning and operational management. *Journal of Hydrology*, 177, 3-4, 269-291.

Assaf, H.; van Beek, E.; Borden, C.; Gijsbers, P.; Jolma, A.; Kaden, S.; Kaltofen M.; Labadie, J.W.; Loucks, D.P.; Quinn, N.W.T.; Sieber, J.; Sulis, A.; Werik, W.J. & Wood, D.M. (2008). Generic simulation models for facilitating stakeholder involvement in water resources planning and management. A comparison, evaluation, and identification of future needs, In: *Environmental Modelling, Software and Decision Support (3): The State of the Art and New Perspective*, Jakeman, Voinov, Rizzoli & Chen (Ed.), 229-246, Elsevier.

CPLEX Optimization, Inc. (2006). *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*. Incline Village, Nevada.

Delft Hydraulics (2006). River Basin Planning and Management Simulation Program. *Proceedings of the iEMSs Third Biennial Meeting: "Summit on Environmental Modelling* *and Software*", Voinov, Jakeman & Rizzoli (Ed.), International Environmental Modelling and Software Society, Burlington, Vermont.

Hashimoto, T.; Stedinger, J.R. & Loucks, D.P. (1982). Reliability, Resiliency and Vulnerability Criteria for Water Resource System Performance Evaluation. *Water Resources Research*, 18, 1, 14-20.

Labadie, J.W.; Baldo, M.L. & Larson, R. (2000). *MODSIM: Decision Support System for River Basin Management: Documentation and User Manual*, Colorado State University and U.S. Bureau of Reclamation, Ft Collins, Colorado.

Loucks, D. P. & Sigvaldason, O.T. (1982). Multiple-reservoirs Operation in North America. In: *The Operation of Multiple Reservoir Systems - IIASA Collaborative Proceedings Series CP-82-S3*, Kaczmarek & Kindler (Ed.), 1-104, Institute for Applied Systems Analysis, Laxenburg, Austria.

Loucks, D.P. & van Beek, E. (2005). *Water Resources Systems Planning and Management: An Introduction to Methods, Models and Applications*. UNESCO Press, Paris.

Manca, A.; Sechi, G.M.; Sulis, A. & Zuddas, P. (2004). Complex water resources system optimization tool aided by graphical interface. In: *Proceedings of 6th International Conference on Hydroinformatics.* LLiong Phoon & Babovic (Ed.), 1059-1066, World Scientific Publishing Company, Singapore.

Mishra, S. (2009). Uncertainty and sensitivity analysis techniques for hydrologic modeling. *Journal of HydroInformatics*, 11, 3-4, 282-296.

Pallottino, S.; Sechi, G.M. & Zuddas P. (2005). A DSS for water resources management under uncertainty by scenario analysis. *Environmental Modelling & Software*, 20, 8, 1031-1042.

Rossi, G. (2000). Drought mitigation measures: a comprehensive framework. In: *Drought and drought mitigation in Europe,* Vogt & Somma (Ed.), 233–246, Kluver.

Sechi, G.M. & Sulis, A. (2007a). Mixed simulation-optimization technique for complex water resources systems analysis under drought conditions, Chapter 11. In: *Methods and Tools for Drought Analysis and Management,* Rossi, Vega & Bonaccorso (Ed.), 217-237, Springer, Berlin.

Sechi, G.M. & Sulis, A. (2007b). Multi-reservoir system optimization using Chlorophyll-a trophic indexes. *Water Resources Management,* 21, 5, 849-860.

Sechi, G.M. & Sulis, A. (2009b). Dynamic attribution of water quality indexes in a multi-reservoir optimization model. *Desalination*, 237, 1, 99-107.

Sechi, G.M. & Sulis, A. (2009a). Water system management through a mixed optimization-simulation approach. *Journal of Water Resources Planning and Management*, 135, 3, 160-170.

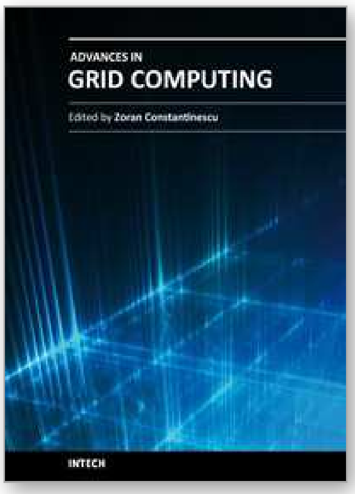
Sechi, G. M. & Zuddas, P. (2007). Multiperiod hypergraph models for water systems sptimization. *Water Resources Management*, 22, 3, 307-320.

SEI Stockholm Environment Institute (2005). *WEAP: Water Evaluation and Planning System, User Guide*, Somerville, Massachussets.

Simonovic, S.P. (2000). Tools for water management: A view of the future. *Water International*, 25, 1, 76-88.

Sulis, A. (2009). GRID computing approach for multireservoir operating rules with uncertainty. *Environmental Modelling & Software*, 24, 7, 859-864.

Yevjevich, V.; Hall, W. & Salas, J. (1983). *Coping with Drought*, Water Resources Publication, Littleton, Colorado.



ADVANCES IN
GRID COMPUTING
Edited by Zoran Constantinescu
INTECH

Advances in Grid Computing
Edited by Dr. Zoran Constantinescu

ISBN 978-953-307-301-9
Hard cover, 272 pages
Publisher InTech
Published online 28, February, 2011
Published in print edition February, 2011

This book approaches the grid computing with a perspective on the latest achievements in the field, providing an insight into the current research trends and advances, and presenting a large range of innovative research papers. The topics covered in this book include resource and data management, grid architectures and development, and grid-enabled applications. New ideas employing heuristic methods from swarm intelligence or genetic algorithm and quantum encryption are considered in order to explain two main aspects of grid computing: resource management and data management. The book addresses also some aspects of grid computing that regard architecture and development, and includes a diverse range of applications for grid computing, including possible human grid computing system, simulation of the fusion reaction, ubiquitous healthcare service provisioning and complex water systems.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821