Chapter 6
Symbolic Modeling of River Basin Systems

This chapter documents the basic architecture of the *Aquarius* software and discusses the advantages of using an object-oriented programming framework for modeling the hydraulic and mathematical connectivity of the flow network components. Comprehension of the material in this section is not essential for using *Aquarius*, and is helped by reading Chapter 7.

Object-Oriented Programming Framework

Earlier computer models for solving water resource problems have used algorithmic computer languages such as FORTRAN. Although these languages are well suited for numerical or algorithm-oriented models, they lack the flexibility to allow for alterations, additions, or deletions of flow network components. Recent research has overcome this problem by using an object-oriented programming (OOP) language[1], specifically C++. Objects are the building blocks of an OOP. An object contains properties that communicate with other objects. In turn, object behavior is controlled by methods, which are the rules and algorithms that tell an object how to act on the data it receives in its input slots. Objects may inherit both data (properties) and behavior (methods) from other higher-level objects.

Water systems are ideal candidates for modeling through an object-oriented framework. A water system may include different types of water components, including reservoirs, powerplants, diversions or junction points, irrigation areas, environmentally-sensitive river reaches, etc, which can be interpreted as objects of a flow network in which they interact. *Aquarius* models each component or structure of the water system as an equivalent node or object in the programming

---

The object-oriented terminology and formats for class diagrams used in this document are based on Booch's notation (Booch 1994). The following terms are used in this chapter:

Object-Oriented Programming (OOP): a programming method in which programs are organized as cooperative collections of objects that represent an instance of some class, and whose classes are all members of a hierarchy of classes united via inheritance relationships.
Class: a set of objects that share a common structure and a common behavior.
Object: an instance of a class.
Instantiation: a new object created from a class.
Hierarchy: a ranking or ordering of abstractions.
Inheritance: a mechanism of hierarchy in which one class shares the structure or behavior defined in one or more classes; there is single or multiple inheritance.
Aggregation: a mechanism of hierarchy wherein a class mimics the behavior of one or more classes by embedding their instances.
Polymorphism: the property of an object, achieved through either inheritance or aggregation, through which it represents objects of many different classes.
Persistence: the property of an object through which its existence transcends time and/or space; the object becomes capable of existing past the lifetime and address space of its creator (e.g., hard disk).
Runtime Type Identification: the property of an object through which it stores the identity of its class so that it is capable of identifying its class type when queried.

environment. In modeling terms, a physical link (e.g., a river reach) connecting two system components becomes an outflow slot of the upstream object connected to an inflow slot of the downstream object.

The user interacts with the model through a graphical user interface (GUI) that allows the analyst to readily create the river basin network of interest. This is a simple task due to the inherent capability of the object oriented paradigm for graphical representation. During the creation of the flow network, each system component (object) corresponds to a graphical network node. These nodes are represented by icons, which are a pictorial representation of the object. By dragging one of these icons from the menu, the model creates an instance of the object on the screen. This procedure also allows the user to connect graphically the input slots of this object with the output slots of one or more objects. By clicking on the icon, the object displays data slots for input and output and also allows the user to visually inspect for incorrect or missing data. Details on the use of the GUI are in Chapter 7.

Software Architecture

The overall design of *Aquarius* is depicted by the top-level class diagram in figure 6.1. As indicated in the figure, all classes implemented in *Aquarius* are organized into three basic class categories:

- Network Worksheet (NWS)
- Water System Components (WSC)
- Water System Links (WSL)

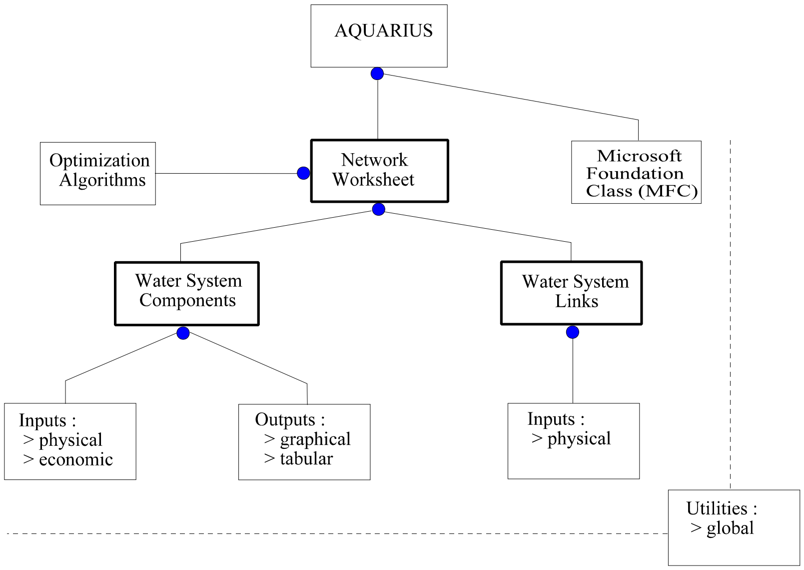


Figure 6.1 *Aquarius* top-level class diagram.

Aquarius is built upon Microsoft Foundation Classes (MFC), which are a set of reusable classes that provide, with minimal overhead, numerous important functions to software applications written for Microsoft Windows.

Network Worksheet

The Network Worksheet (NWS), one of the basic classes of *Aquarius*, is composed of two classes –*document* and *view*– that are responsible for storing and rendering to the screen, respectively, data associated with any flow network. The *document* is a data object that the user interacts with during editing sessions; for instance, during the creation or alteration of a flow network. *View* is the user's window to the data, which specifies how the user sees the document's data and interacts with it. Data pertaining to the network, which is also an object, are stored in the *document*, and the objects themselves are rendered onto the *view* window. Most of the important functions of the software, such as user interaction, data persistence, and optimization algorithms, are routed to the individual objects via the message handling facility of the *view* window.

The *document* portion of the NWS stores the other two basic class categories of *Aquarius*, the Water System Components (WSC) and the Water System Links (WSL).

Water System Components

The Water System Components (WSCs) are defined by several classes, depending on the component of the flow network that requires representation, although all WSCs are derived from a single basic class, *CNode*. In addition to the data corresponding to the individual network components, the *document* class also stores global data corresponding to the whole network (for instance, parameters controlling the optimization algorithm, the selected period of analysis, output data related to the optimal solution, etc.).

As illustrated in figure 6.2, the data structure object can be an instance of any of the following classes:

- *CDataJunDiv* (junctions and diversions)
- *CDataResv* (storage capacities)
- *CDataPower* (hydroelectric plants)
- *CDataOffstream* (offstream demand areas)
- *CDataInstream* (instream demand areas)
- *CDataWatershed* (Surface water sources)
- *CDataGWpumping* (Groundwater pumping)
- *CDataFloods* (flood control reaches)

Figure 6.2 also shows the association relation between the *CNode* Object and the respective WSC data structures. The WSC classes use other classes for user interaction, both in terms of inputs and outputs.

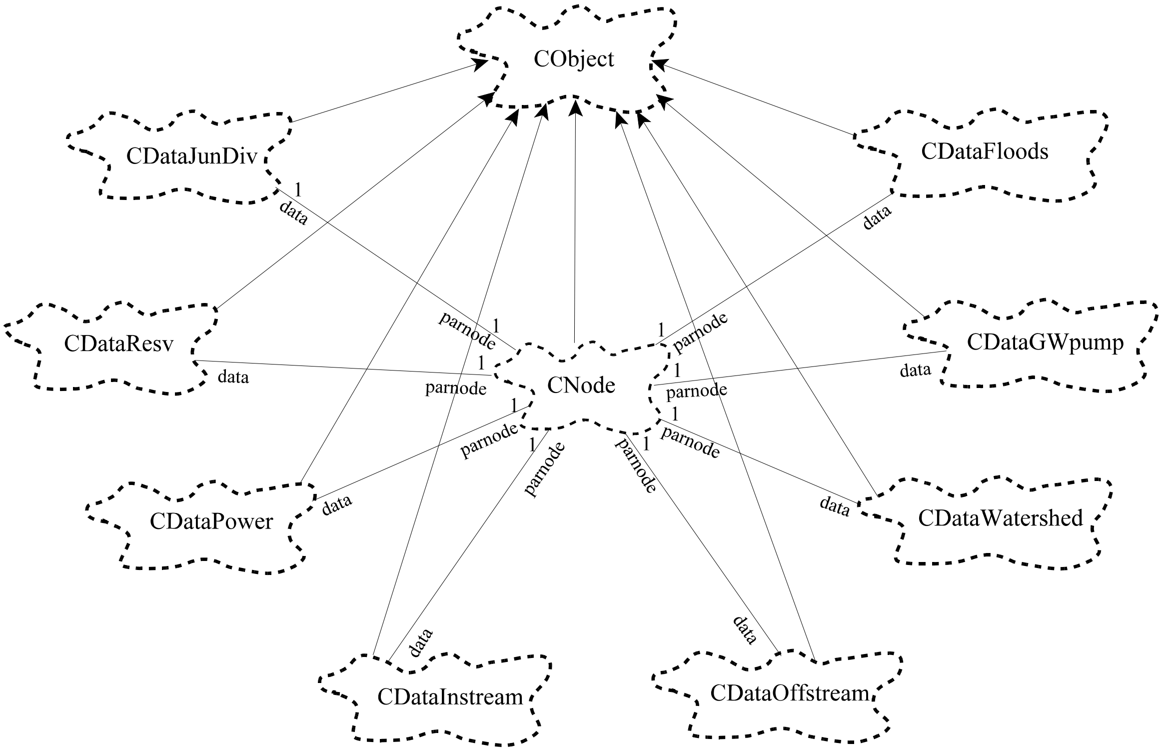


Figure 6.2 Associations between *CNode* object and the system objects.

Water System Links

This class category is comprised of the following two subclasses:

- River Reach (*CDataRiv)*
- Conveyance Structures, canals and pipelines (*CDataConvey*)

During “validation” of the network (discussed in Chapter 7) WSLs are classified according to the role they play and the place they occupy in the flow network. The classification of links is performed automatically by the model, which names the links according to the following definitions:

natflow	arc conveying surface water from a basin (SWS).
decision	arc conveying water released from a reservoir, diversion node or pumping well under the following types: (i) a controlled release from a reservoir serving an offstream water user (HPW, IRR or M&I), directly connected to the reservoir of reference; (ii) a controlled release from a reservoir serving an instream water user (IRA or FCA), directly connected to the reservoir of reference; (iii) a controlled release from a reservoir directly to an RDD object, which in turn serves demand area/s located farther downstream (and mathematically disconnected) from the reservoir of reference; (iv) a controlled release from a reservoir directly into a downstream reservoir (no water user in between); (v) a controlled water diversion through the offstream link of a diversion node; (vi) a controlled withdrawn (pumping) of water from an aquifer (GWS);
return	arc carrying return flows from an offstream demand area (IRR and M&I) back to a river reach or reservoir.
spill	arc evacuating spillages from a reservoir (always the central link).
dec&spill	arc acting as a *decision* and as a *spill* arc at the same time (dual classification).

Mathematical Connectivity of System Components

Solving the water allocation problem in Chapter 5 for any user-defined network requires an automated procedure to handle the formulation mathematics. In *Aquarius* the mathematical connectivity of the system components is derived automatically from the linkage of the objects comprising the network, which in turn reflects the direction of flow from one structure to the next (i.e., their hydraulic connectivity).

The requirements for the mathematical connectivity of the system components may vary depending on the characteristics of the optimization technique being implemented. For the optimization technique used in this model (Sequential Quadratic Programing) the mathematical connectivity of a river network serves to:

- Assemble the gradient vector and Hessian matrix of the second order approximation of the total objective function (see Chapter 5, Solution Method).
- Build up the set of operational constraints (see Chapter 5, Operational Restrictions).

The tasks indicated above required the development of an algorithm capable of automatically gathering information from the network about controlled and uncontrolled flows occurring upstream and downstream from a given system component. The way water sources, storage reservoirs, and demand zones are arranged in a river basin (i.e., network topology) determines the hydraulic and mathematical dependence among them. Controlled and uncontrolled flows occurring upstream from a given node influence the decisions at that node.

Figure 6.3 represents a river system that illustrates the intricacies of mathematical connectivity. The system has two headwater reservoirs (B and C) with hydropower facilities (the powerplant connected to reservoir C is a run-of-the-river type). Releases from the powerplants linked to the headwater reservoirs plus additional releases from Reservoir C to supplement downstream demands become regulated inflows to downstream reservoir A, which in turns supplies water to most downstream demand areas including hydropower, urban supply and a fish habitat protection area. An irrigation demand zone and an instream recreation area are in the middle of the system.

The series of physical links (river reaches, canals/pipelines) connecting surface water sources, storage capacities, and demand areas are used by the model to automatically formulate the mathematical structure of the water allocation problem. First, the model identifies the decision sets that control water allocation in the flow network. Six decision sets: d_A , d_B , d_C , d_D , d_E , d_F are identified and randomly numbered by the model. Links conveying controlled flows are distinguished by dashed lines in figure 6.3. According to the classification given above (Water System Links) decision sets d_A , d_B, d_C , d_E are type (i), decision set d_F is type (iii) and the set d_D is type (iv).

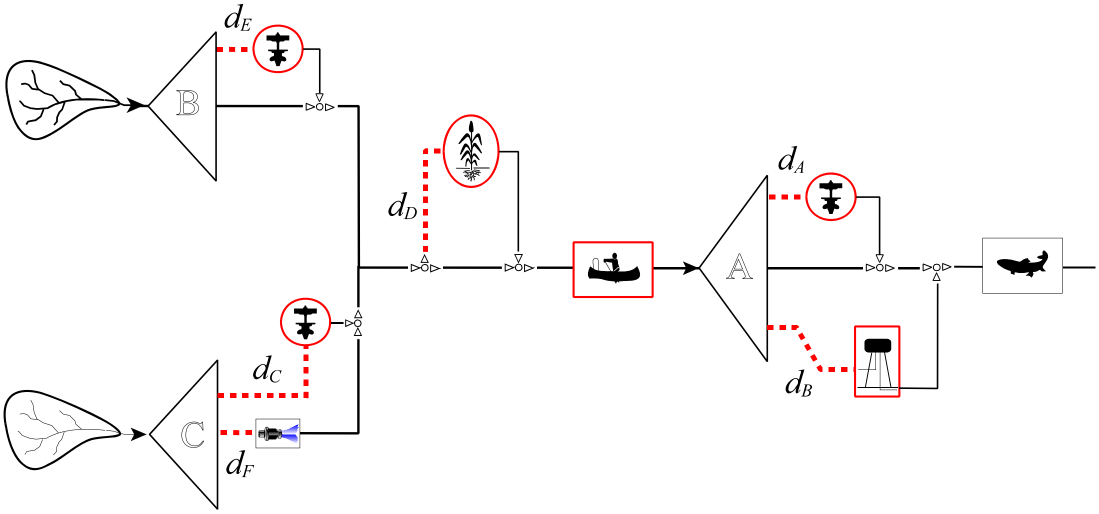


Figure 6.3 Network that demonstrates mathematical connectivity.

The model then collects information regarding all controlled and uncontrolled flows in the network using a recursive search algorithm. Some of the output generated by the search procedure is in table

6.1, which shows the coefficients of the six decision sets. For example, the first row corresponds to reservoir A, where the 1.00 values for decision sets d_C and d_E correspond to controlled releases from the two upstream powerplants. The 1.00 value for decision set d_F corresponds to additional controlled releases from reservoir C (central outlet work) to help satisfy the downstream demands. All three releases enter reservoir A as controlled inflows. Decision set d_D carries a coefficient equal to -0.7. The value and sign assigned to this coefficient indicates that 70 percent of the water diverted from the river into the irrigation area is consumptively used, with the remaining 30 percent ($r = 0.3$ for the irrigation area) reaching reservoir A via return flows. The remaining two coefficients, -1.00 values for decision sets d_A and d_B , represent controlled releases from the reservoir under consideration. The values in table 6.1 can be accessed in the model using the available software menus (see Chapter 7, Exploring the Network Worksheet Screen).

Figure 6.4 shows the coefficients listed in table 6.1 plus some additional information, also gathered by the search procedure, for all nodes composing the example network in figure 6.3. This information, which is stored as part of the objects data structure, is used by the model to automatically build the set of constraint equations and compute the gradient vector and Hessian matrix. The information collected is organized in four quadrants: controlled inflows *XI* at the upper-left corner, uncontrolled inflows *UI* at the lower-left corner, controlled releases *XR* at the upper-right corner (alternatively the return coefficient *r* for WSCs with consumptive use), and uncontrolled releases at the lower-right corner. The superscript for *UI* and *UR* indicates the originating reservoir. The reader can check the information in figure 6.4 with assistance from figure 6.3 and table 6.1.

Table 6.1 Table of mathematical connectivity.

d_A	d_B	d_C	d_D	d_E	d_F	WSC
-1.00	-1.00	1.00	-0.70	1.00	1.00	Reservoir A
0.00	0.00	0.00	0.00	-1.00	0.00	Reservoir B
0.00	0.00	-1.00	0.00	0.00	-1.00	Reservoir C
1.00	0.00	0.00	0.00	0.00	0.00	Hydroplant A
0.00	0.00	0.00	0.00	1.00	0.00	Hydroplant B
0.00	0.00	1.00	0.00	0.00	0.00	Hydroplant C
0.00	0.00	1.00	-1.00	1.00	1.00	Diversion
0.00	0.00	0.00	1.00	0.00	0.00	Irrigation
0.00	0.00	1.00	-0.70	1.00	1.00	Boating
0.00	1.00	0.00	0.00	0.00	0.00	Urban Supply
1.00	0.80	0.00	0.00	0.00	0.00	Fish Habitat
0.00	0.00	0.00	0.00	0.00	1.00	RDD

Constraint Set Assemblage

The model is capable of attaching operational constraints to the following system components (see Chapter 5, Operational Restrictions, for more details):

- storage reservoir (RES)
- reservoir with lake recreation (RLR)
- hydropower plants (HPW)
- instream water demands: recreation (IRA) and flow protection (IFP)
- offstream demand areas: municipal water supply (M&I) and irrigation (IRR)
- flood control river reaches (FCA)
- groundwater source (GWP),

After the user specifies which operational constraints are included in the formulation of the water allocation problem, the NWS delegates the responsibility for building the set of restrictions to each of the components listed above.

The information required by the system components for the global task of computing the right- and left-hand-sides of the equality and inequality constraints is gathered from the network, using the recursive algorithms outlined previously. Once the NWS completes the constraint set computation, the information is passed to the optimization routine.

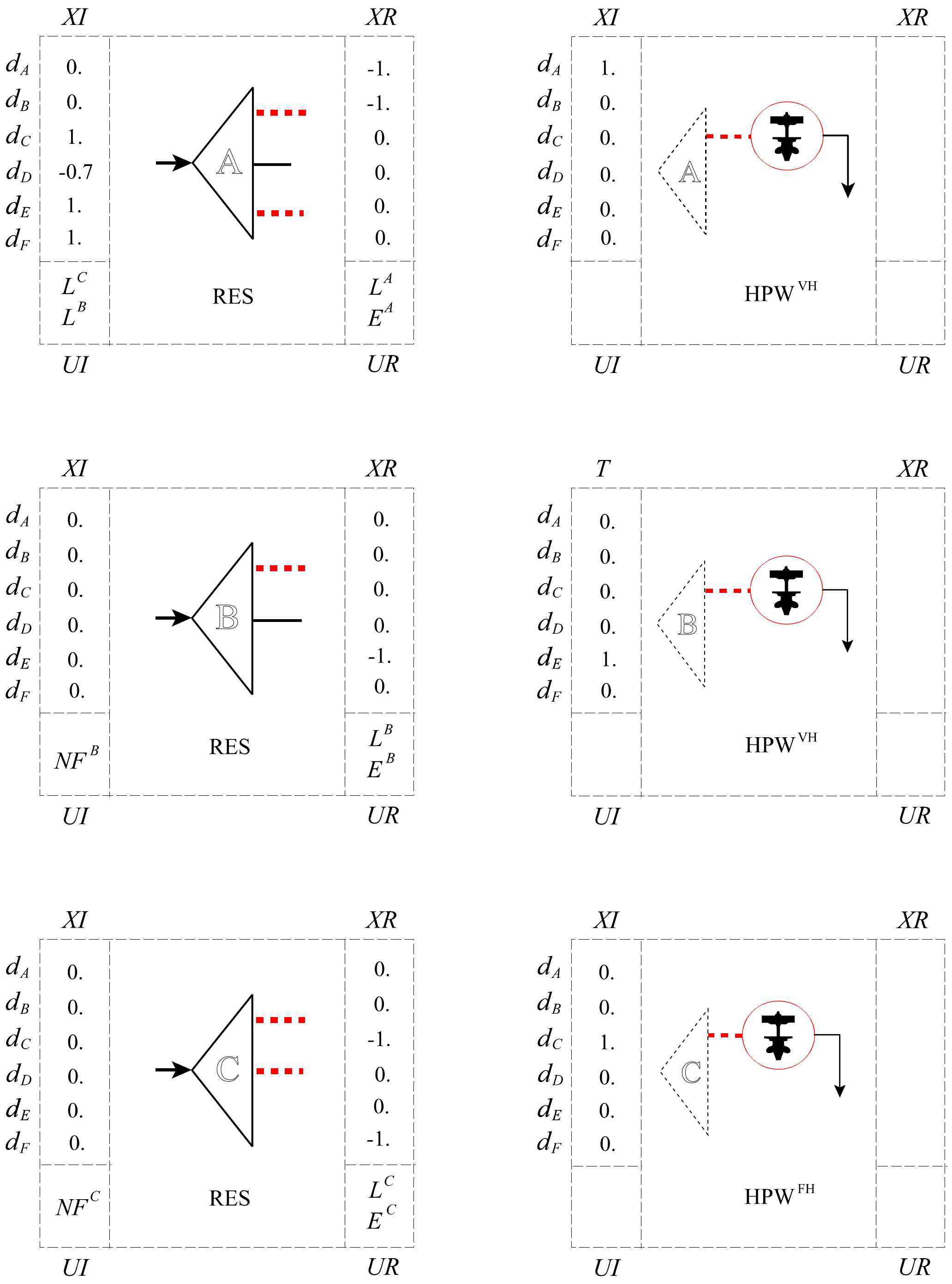
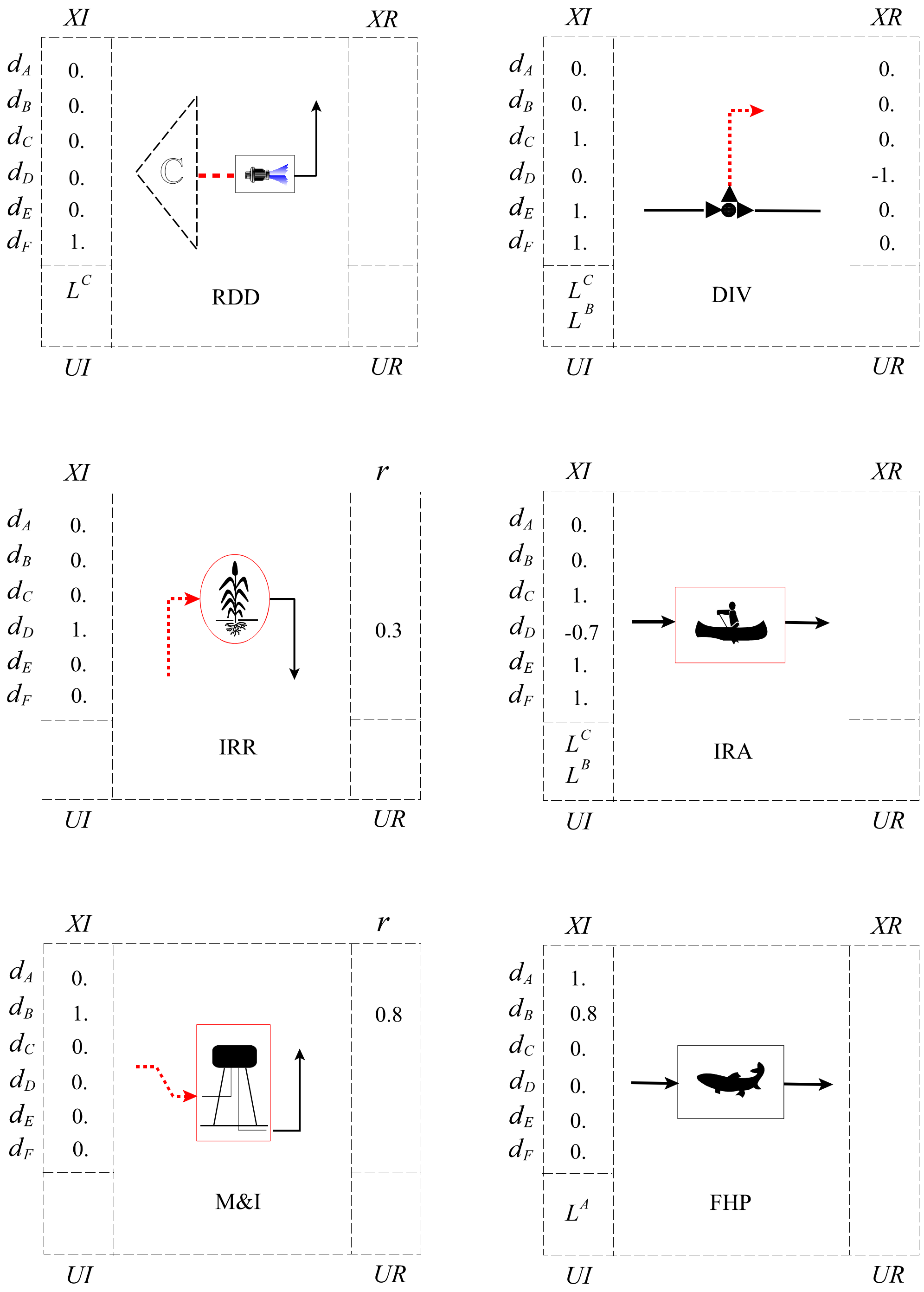


Figure 6.4 Information collected by the search algorithms.



<u>Note</u>: *XR* for offstream water (IRR and M&I) is replaced by the return flow coefficient *r*.

Figure 6.4 Information collected by the search algorithms (continued).

Gradient Vector and Hessian Matrix Assemblage

How water storages and demands are arranged in a river basin determines the physical and mathematical interdependence among them. For instance, in figure 6.3, releases from the headwater reservoirs become regulated inflows to the downstream reservoir, which supply water to the remaining portion of the system. The series of links connecting the various components of a flow network defines the mathematical linkage among the sets of decision variables and consequently the structure of the gradient vector $\nabla \boldsymbol{f}$ and Hessian matrix $\mathbf{H}$ in equations (5.4) and (5.5), respectively. Diaz and Fontane (1989) demonstrated, in an earlier more exclusive study solely on hydropower, that it is possible to identify geometrical patterns in the mathematical structure of the gradient vector and global Hessian regarding the topology of the flow network. In this study we extended the aforementioned work to river basins with multiple water uses.

Aquarius first identifies all controlled and uncontrolled flows present in the network, then automatically builds the gradient vector and Hessian matrix. The assemblage of $\nabla \boldsymbol{f}$ and $\mathbf{H}$ is determined based on information provided by the network search procedures and the library of partial derivatives introduced in Appendix A. The algebraic expressions of the first and second order partial derivatives of the benefit functions are a property of each water user. For the example network shown in figure 6.3, there are six basic groups of first partial derivatives ($\partial f/\partial d_A$ $\partial f/\partial d_B$, $\partial f/\partial d_C$, $\partial f/\partial d_D$, $\partial f/\partial d_E$, $\partial f/\partial d_F$), one for each decision set, which contain derivatives for *np* time periods. The computation of the gradient vector results from the combination of the following equations:

$$\frac{\partial f}{\partial d_A} = \text{Eq.(A.1)} \qquad \textit{for } i=1, 2, ..., np$$

$$\frac{\partial f}{\partial d_B} = \text{Eq.(A.21)} + \text{Eq.(A.2)} \qquad \textit{for } i=1, 2, ..., np$$

$$\frac{\partial f}{\partial d_C} = \text{Eq.(A.1)} + \text{Eq.(A.24)} + \text{Eq.(A.3)} \qquad \textit{for } i=1, 2, ..., np$$

$$\frac{\partial f}{\partial d_D} = \text{Eq.(A.18)} + \text{Eq.(A.24)} + \text{Eq.(A.3)} \qquad \textit{for } i=1, 2, ..., np$$

$$\frac{\partial f}{\partial d_E} = \text{Eq.(A.1)} + \text{Eq.(A.24)} + \text{Eq.(A.3)} \qquad \textit{for } i=1, 2, ..., np$$

$$\frac{\partial f}{\partial d_F} = \text{Eq.(} \quad \text{)} \qquad \textit{for } i=1, 2, ..., np$$

Assembly of the global Hessian matrix (i.e., the Hessian for the entire network) follows rules similar to those used for assembling the gradient vector, although, because of the presence of second cross-partial derivatives, it may appear more involved. Again, the mathematical connectivity of the decision sets (d_A, d_B, d_C, d_D, d_E, d_F) within the global Hessian is based on the information provided by the search algorithm and the algebraic expressions of the partial derivatives contained as part of the data structure of the objects.

Note that the number of decision sets (red links) and the number of water uses competing for water (red framed WSCs) do not have to coincide. Just by chance the example network in figure 6.3 yields six decision sets and six users competing for water (IFP acts only as an operational constraint). Moreover, there can be WSCs which, due to their location in a flow network, have their incoming flows indirectly controlled by the release decisions made upstream from the WSC of reference; the Boating area (IRA) in figure 6.3 is an example of that. The model expresses controlled flows at the IRA as a linear combination of the four upstream decision sets (see figure 6.5 and Table 6.1).

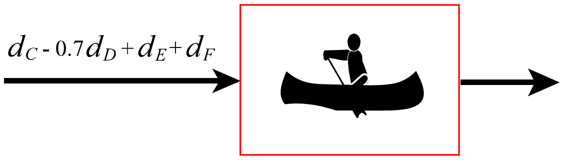


Figure 6.5 Decision variables entering the instream recreation area.

Figure 6.6 is the global Hessian matrix for the example network. The referenced numbers in the figure indicate the corresponding equations in Appendix A. Blank portions of the Hessian matrix are zero values.

Despite the generally complex dependence among decision variables, the global **H** matrix is square and symmetric. Furthermore, as the model randomly selects the order of the decision sets every time the network is validated, it is possible that associations will change if the network is altered. For instance, the decision set d_A, which in the example is associated with the releases from reservoir A to the powerplant, might be associated with some other controlled release if the network was validated after an alteration was introduced. The change in the order of the decision sets would alter the arrangement of the submatrices in the Hessian matrix and possibly yield a slightly different optimal solution of the water allocation problem due to numerical precision in the computations.

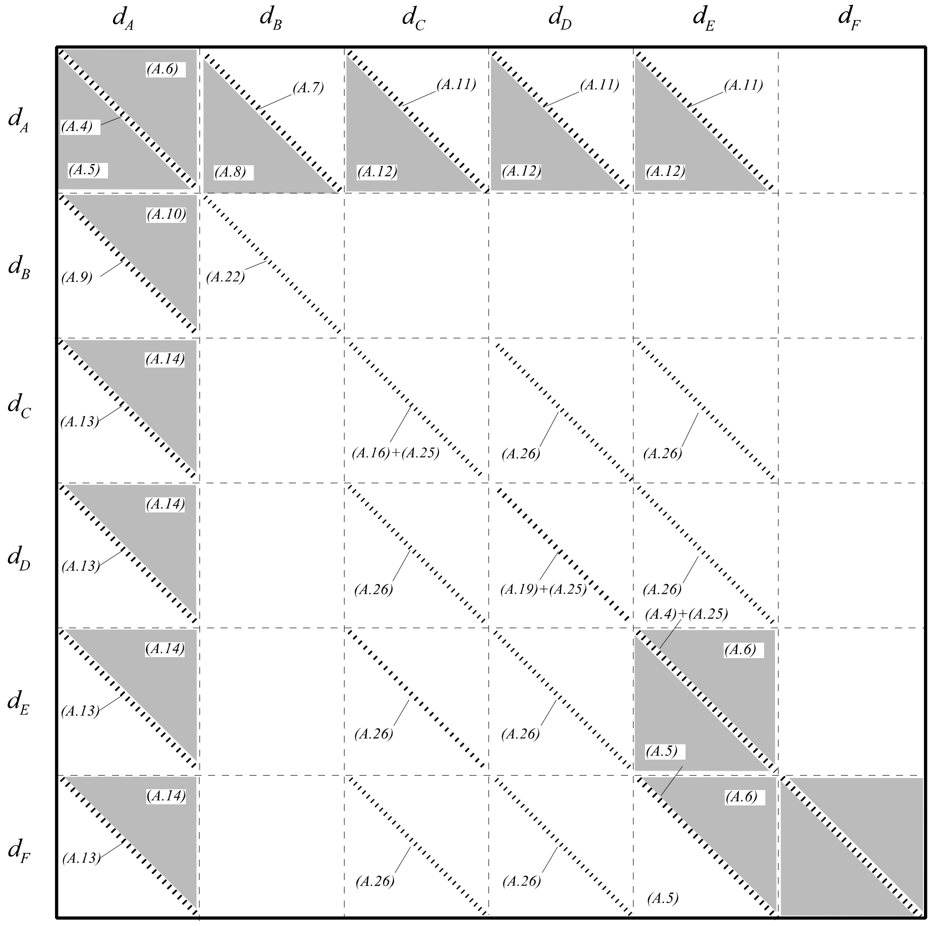


Figure 6.6 Global Hessian matrix for the example network in Figure 6.3 (equations from Appendix A)

FIGURE 6.6 IS UNDER PREPARATION