CADSWES University of Colorado

Center for Advanced Decision Support for Water and Environmental Systems

RiverWare's Water Rights Allocation Solver

RiverWare User Group Meeting
August 13-14, 2008

Status in RiverWare 5.0

Purpose

- Allocate water from an **allocatable flow or natural flow** stream to water rights in such a way that in times of shortage, the oldest rights (seniors) are satisfied before younger rights (juniors) are satisfied.

Kinds of Water Rights

- Divert and use (and possibly return to stream)
- Divert and store (in on-stream or off-stream storage)
- Keep water in river for instream flow uses at a specific location (forcing juniors to leave water in the stream)

Rights have

- **Location** and **means** (of diversion or measurement)
- **Priority Date** (when first appropriated)
- **Quantity** (total amount taken in a given period /year)
- **Flow** (rate at which water may be taken or kept in the river)
- **Times** when water may be taken (seasonal fluctuations)
- **Legal limits** imposed by courts or contracts
- **Physical limits** imposed by the diversion or storage structures

Limits Imposed

- Usually depend on the state of the stream after senior rights have been satisfied
- Consider only the water that would naturally be in the stream

Concept	Model
Right	**Account on Simulation Object**
Diversion right	Diversion Account on Water User object
Storage right	Storage Account on Reservoir Object
In-stream Flow right	Instream Flow Account on ControlPoint object

Concept	Model
Right	**Account on Simulation Object**
Location, means of diversion	Location of containing simulation object and its linkage to other objects
Quantity, Flow	Account slots: **Initial Request, Accrual, Max Accrual** *
Seasonal fluctuations	Beginning-of-run disaggregation of **Annual Request** account slot, based on series of multipliers

Concept

Right

Limits imposed by courts or contracts

Model

Account on Simulation Object

Minimum bypass criteria at a control point on the stream can restrict net request for allocation

Subordination relationships: Senior gives up allocated water to junior(s)

Concept

Right

Limits imposed by physical structures

Model

Account on Simulation Object

Diversion capacity on a Reach object and

Conservation Pool capacity on a Reservoir object can restrict net request for allocation

Concept

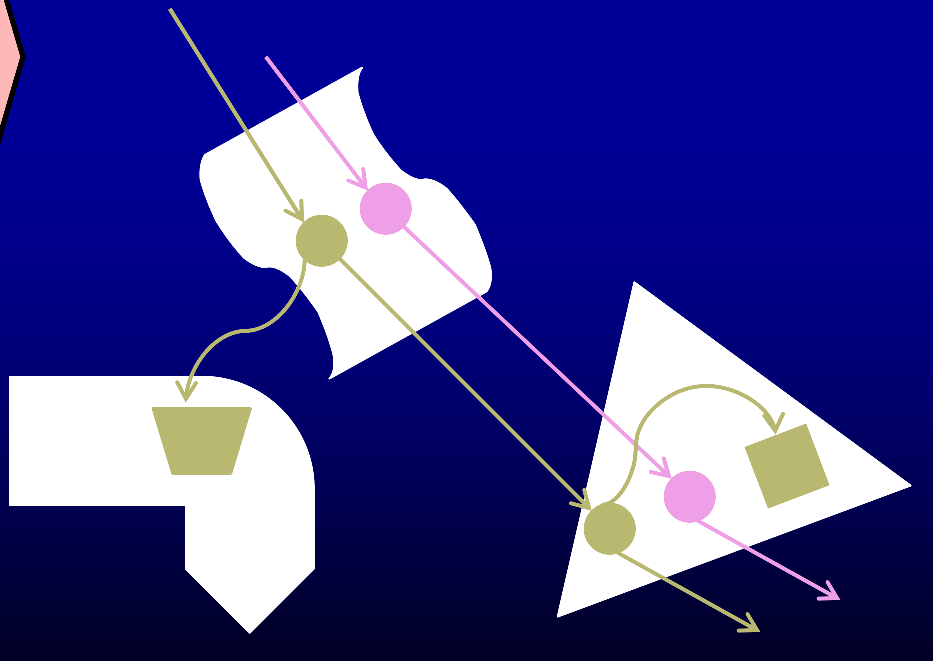
Model

Allocatable Flow

and

Non-allocatable (e.g., project) water

Supply chains with different Water Types



Rules

Initialize Accounts

Distribute Initial Request values

Call solver without respect to
 instream flow rights

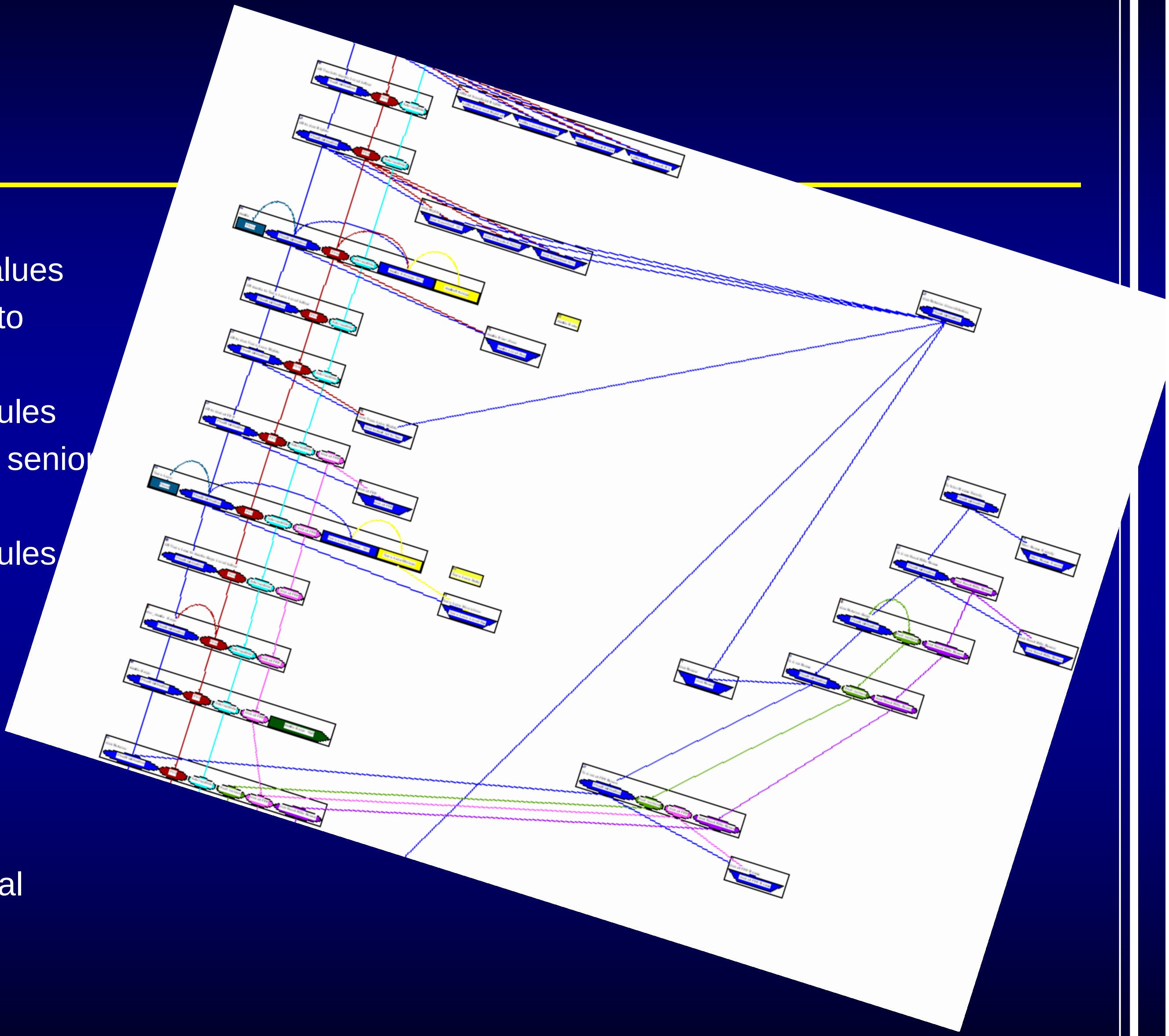
Execute reservoir release rules

Call solver respecting most senior
 instream flow right

Execute reservoir release rules

Call solver respecting
 next most senior
 instream flow right

… and so on



Transfer releases to physical system

Calling the Solver (no lags)

```
FOREACH(
   LIST pair IN
   SolveWaterRights(“Basin”, “Allocatable WT”,
                              @”24:00:00 January 1, 2001”)
) DO
        (GET SLOT @INDEX 0 FROM pair) [] =
                          GET NUMERIC @INDEX 1
    FROM pair
ENDFOREACH
```

Solution Algorithm (Acts on Clones)

Visit each right (account with priority date) in order:
most senior → most junior. For each, do:

1) compute Appropriation Request slot (net request), taking into account state of the system after seniors have been satisfied
2) allocate lesser of Appropriation Request and amount available in the stream at appropriation point
3) check for flow deficit @ downstream seniors
4) from most-downstream senior that has deficit, solve upstream to point of diversion (applying losses, taking larger of the deficit from downstream and the deficit at any senior's appropriation point encountered, to determine necessary reduction of allocation
5) **Supply** ← allocation

Solution with Subordination

- When the junior (“dominant”) right’s allocation is computed, if the junior is short, see if reducing a subordinate (senior)’s allocation will help the junior’s case.
- Senior subordinates must have allocated water to give up, and giving up that water must have an effect at the dominants junior’s appropriation point.
- A junior might not be able to take water from a subordinate due to the presence of a (non-subordinate) senior downstream that needs the water
- If the junior has two or more subordinated rights, the most junior subordinate gives up water first, then the next most junior, etc.

Solution with Lags

- Models with small timestep sizes need to take into account travel time of water.
- PassThrough accounts in a water rights model may have lags (integral # timesteps: approximations of the physical routing).
- The solution to the allocation problem must then take into account how today's allocation at junior J affects downstream senior S on the day that water taken for J would have reached S.

Solution with Lags

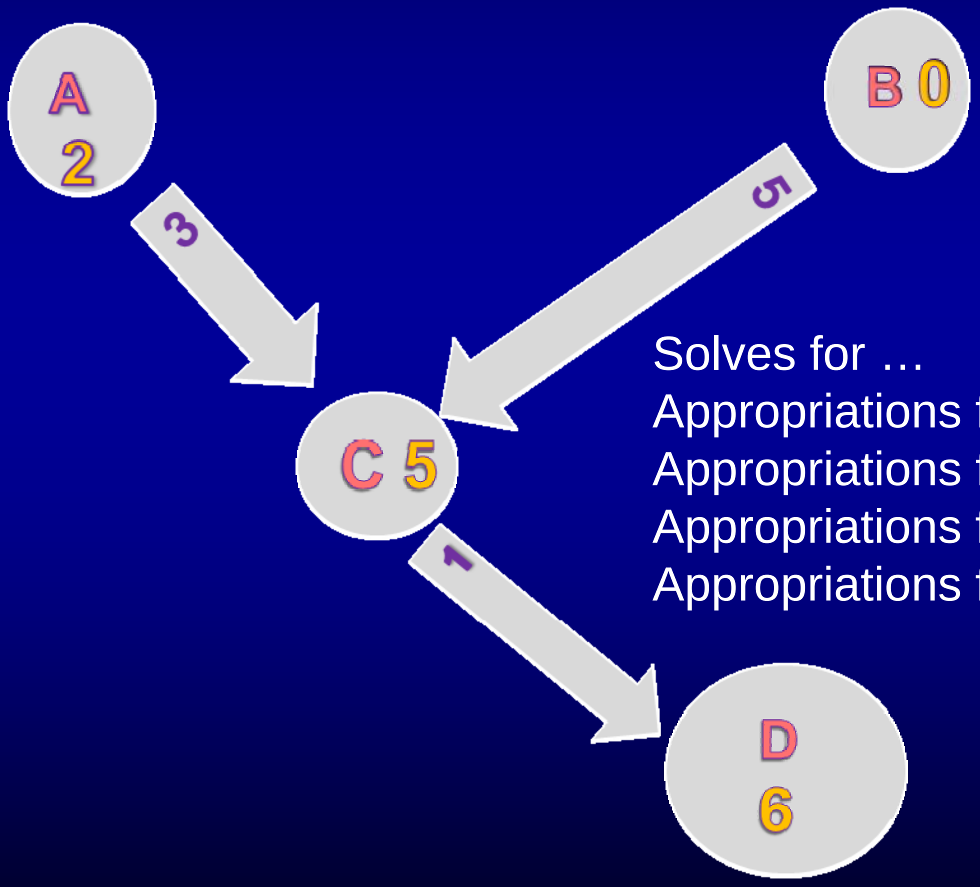
- For lags, use the function **SolveWaterRightsWithLags ()**

```
FOREACH(
   LIST triplet IN
   SolveWaterRightsWithLags(“Basin”, “Allocatable WT”,
                              @”24:00:00 January 1, 2001”)
) DO
     (GET SLOT @INDEX 0 FROM triplet) [
                  GET DATETIME @INDEX 1 FROM triplet] =
                           GET NUMERIC @INDEX 2 FROM triplet
ENDFOREACH
```

Solution with Lags

- Solver solves for allocation to each right at a **single** timestep, its **local timestep** (not necessarily the current rules controller timestep).
- Each right's local timestep is determined by the **local timestep offset** of its appropriation point, computed at beginning of run.

Local Timestep Offset



Solves for …
Appropriations from B @ current TS
Appropriations from A @ TS + 2
Appropriations from C @ TS + 5
Appropriations from D @ TS + 6

Status

- In RiverWare 5.0 *
- In use by LCRA