PARETO

Release 0.3.0

The PARETO Project

Aug 09, 2022

CONTENTS

ONE

PROJECT GOALS

The goal of the project is to develop a modeling and optimization application, PARETO, that can help organizations better manage, better treat, and – where possible – beneficially reuse produced water from oil and gas operations. Specifically, PARETO will help decision-makers with:

- buildout of the produced water infrastructure

- management of produced water volumes

- selection of effective treatment technologies

- placement & sizing of treatment facilities

- identification of beneficial water reuse options

- distribution of treated produced water for reuse

The initiative is committed to viewing produced water management from a "systems" perspective and to building an inclusive framework that will unite stakeholders from across the produced water community. The vision is that PARETO will not only help oil & gas but also allow other industries (e.g., agriculture, mining) explore beneficial reuse opportunities for treated produced water. Figure 1 (below) illustrates the scope of "Project PARETO".

TWO

COLLABORATING INSTITUTIONS

The PARETO team is comprised of collaborators from the following institutions:

- National Energy Technology Laboratory (Lead)
- Sandia National Laboratory
- Lawrence Berkeley National Laboratory

PROJECT TIMELINE

Project PARETO is a 3-year initiative that has been split into three distinct phases; with each phase taking up exactly one year. In execution year 2021, PARETO will capture produced water management, i.e., capturing options for coordinating water deliveries in a given development area. By execution year 2022, the project will shift its attention towards produced water treatment. Finally, execution year 2023 will be dedicated to produced water beneficial reuse.

Key Project Phases
- Phase I: Water Management (2021)
- Phase II: Water Treatment (2022)
- Phase III: Beneficial Reuse (2023)

12/2023
Release of beneficial reuse-focused model

03/2023
Extended case studies completed

06/2022
Management best practices report

06/2024
Beneficial reuse best practices report

03/2022
Initial case studies completed

06/2023
Treatment best practices report

09/2021
Formation of stakeholder board

12/2022
Release of treatment-focused water model

03/2022
Stakeholder review; DOE Go/No-Go

12/2021
Release of initial water management model

04/2021
Project Kick-Off

Key Project Deliverables
- Executable software releases
- Industry-driven case studies
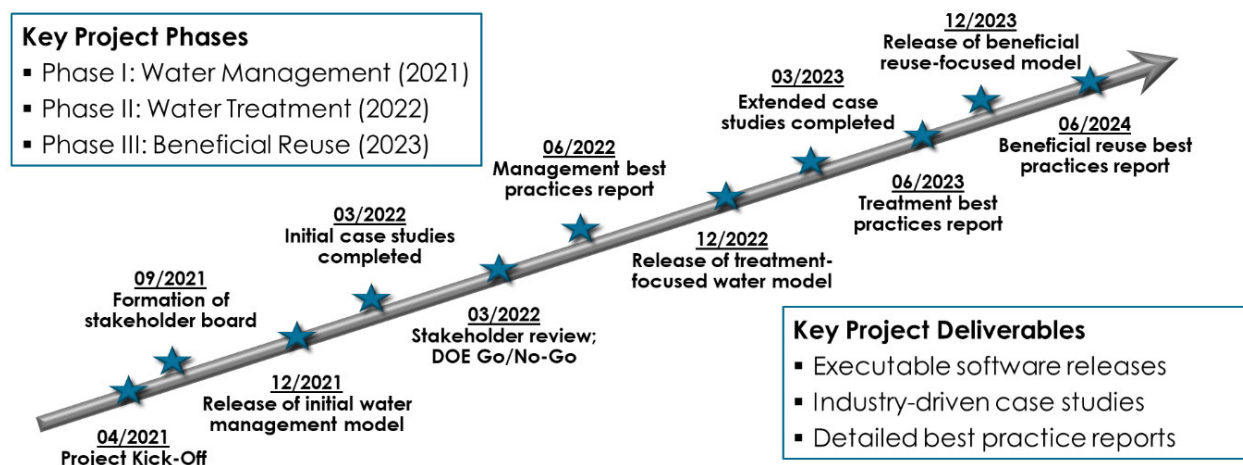- Detailed best practice reports

Fig. 1: Figure 4. Project PARETO Timeline

In terms of deliverables, PARETO itself will be released as free and open-source software every year of the initiative – with increasing capabilities and functionality becoming available over time. The project team is also committed to conducting case studies with industrial and other partners; and where possible findings from those collaborations will be shared with the produced water community as best practice reports. It should also be noted that the project will be continuously evaluated by a comprehensive stakeholder board that involves individuals representing upstream operators, midstream organizations, treatment technology providers, beneficial reuse entities, regulatory agencies and others – all of which will guide the project team and provide necessary input.

FOUR

CONTACT US

General, background and overview information is available at the PARETO main website. Framework development happens at our GitHub repo where you can report issues/bugs or make contributions. For further enquiries, send an email to: <pareto-support@pareto.org>

CONTENTS

5.1 Getting Started

5.1.1 Installation

To install the PARETO framework on Windows operating systems, follow the set of instructions below that are appropriate for your needs. If you need assistance please contact start a new discussion on our GitHub Discussion form or send an email to pareto-support@project-pareto.org.

Developer Role

The installation instructions vary slightly depending on the role you will have with Project Pareto. Below are the roles we've identified:

Users: Use the PARETO platform to develop models, but never contribute to development of the framework (i.e. never commit changes to the project-pareto repo). This includes people who only work with protected data.

Core-dev: Work primarily on PARETO platform development and never handle protected data.

Hybrid: Handle protected data, but also commit changes to the project-pareto repo (even occasionally) - needs approval from PhD. Markus Drouven

Developer Role	Installation Section
Users	*Users*
Core-dev	*Core-dev*
Hybrid	*Hybrid*

Install Miniconda (optional)

1. Download: https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe

2. Install anaconda from the downloaded file in (1).

3. Open the Anaconda Prompt (Start -> "Anaconda Prompt").

> **Warning:** If you are using Python for other complex projects, you may want to consider using environments of some sort to avoid conflicting dependencies. There are several good options including conda environments if you use Anaconda.

5.1.2 Users

Non-git tracked option

1. Install PARETO with pip by one of the following methods

a. To get the latest release:

```
pip install project-pareto
```

b. To get a specific release, for example 1.6.3:

```
pip install project-pareto==1.6.3
```

c. If you need unreleased cutting-edge development versions of PARETO, you can install PARETO directly from the GitHub repo either from the main PARETO repo or a developer's fork and branch (this installs from GitHub but does not create a local git clone/workspace):

```
pip install git+https://github.com/project-pareto/project-pareto.git
pip install git+https://github.com/ksbeattie/project-pareto@feature_1
```

5.1.3 Core-dev

1. Fork the repo on GitHub (your copy of the main repo)

2. Clone your fork locally, creating a workspace (github id is "myusername,"):

```
git clone https://github.com/myusername/project-pareto
git clone git@github.com:myusername/project-pareto.
```

3. In this new project-pareto directory, run the following command which installs PARETO in editable mode so that developers can make changes and push to their fork/branch:

```
pip install -e
```

5.1.4 Hybrid

User that can edit the base code

1. Create environment:

```
conda create -n pareto-env python=3.8pip --yes
conda activate pareto-env
```

2. Download zip files (project-pareto-main)

3. Unpack zip files (select directory)

4. Install pareto-project (non-git tracked repo):

```
pip install -r requirements-dev.txt
```

5.1.5 Building Documentation

We use Sphinx for writing and building our on-line documentation. This is a tool that translates a set of plain text *.rst* (reStructuredText) files into various output formats, such as HTML or PDF (via Latex).

After installing as a *Core-dev* or *Users* (as described above) you can build the documentation locally on your system by running the *make* command in the *docs* folder, as follows:

```
$ cd project-pareto/docs/
$ make html
```

Visit the Sphinx Style Guide for information on syntax rules, tips, and FAQ.

5.2 Model Library

5.2.1 Optimization Framework

The Produced Water Application for Beneficial Reuse, Environmental Impact and Treatment Optimization (PARETO) is specifically designed for produced water management and beneficial reuse. The major deliverable of this project will be an open-source, optimization-based, downloadable and executable produced water decision-support application, PARETO, that can be run by upstream operators, midstream companies, technology providers, water end users, research organizations and regulators.

PARETO is designed as an executable optimization-based decision-support application. In return for specifying select input data, users will be provided with specific, actionable recommendations as program outputs. The table below summarizes representative inputs and outputs.
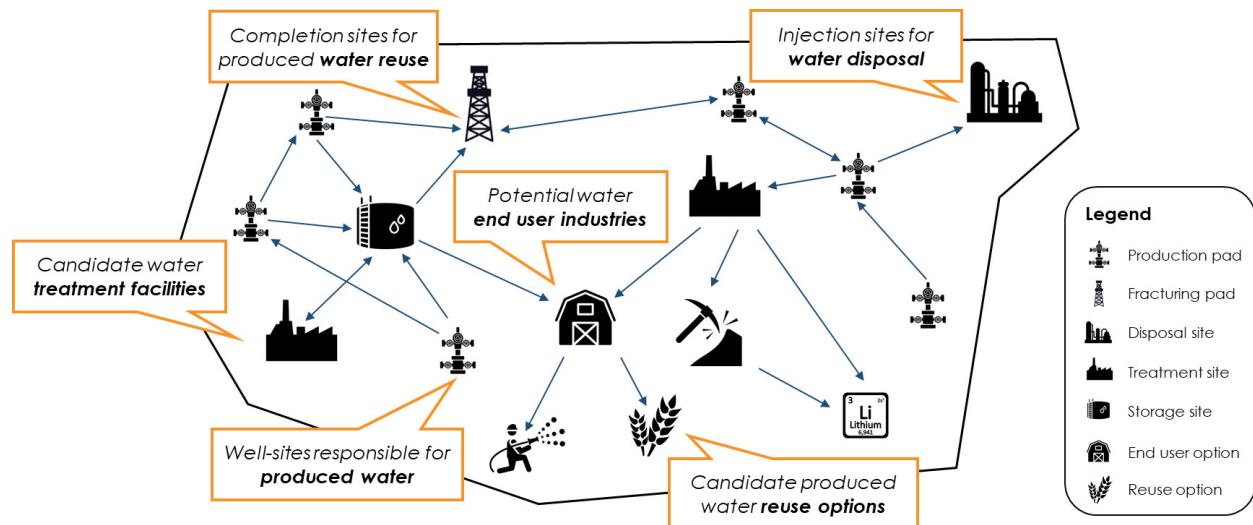

Fig. 1: Figure 1. Problem Description

Program Inputs

- **Produced water forecast**
    - Quantity
    - Quality
- **Transportation resources**

- – Water hauling trucks
- – Produced water pipelines

- **Reuse opportunities**
  - – Demand forecasts
  - – Quality constraints

- **Cost assumptions**
  - – cost of freshwater
  - – treatment, production, and storage cost

Fig. 2: Figure 2. Input Data

Program Outputs

- **Infrastructure buildout**
  - – Storage/treatment facilities
  - – Pipeline network

- **Source-to-sink matches**
  - – Volumetric flows
  - – Blending/treatment options

- **Planning and scheduling**
    - Water delivery coordination
    - Operational and strategic
- **Economic performance**
    - Operational costs
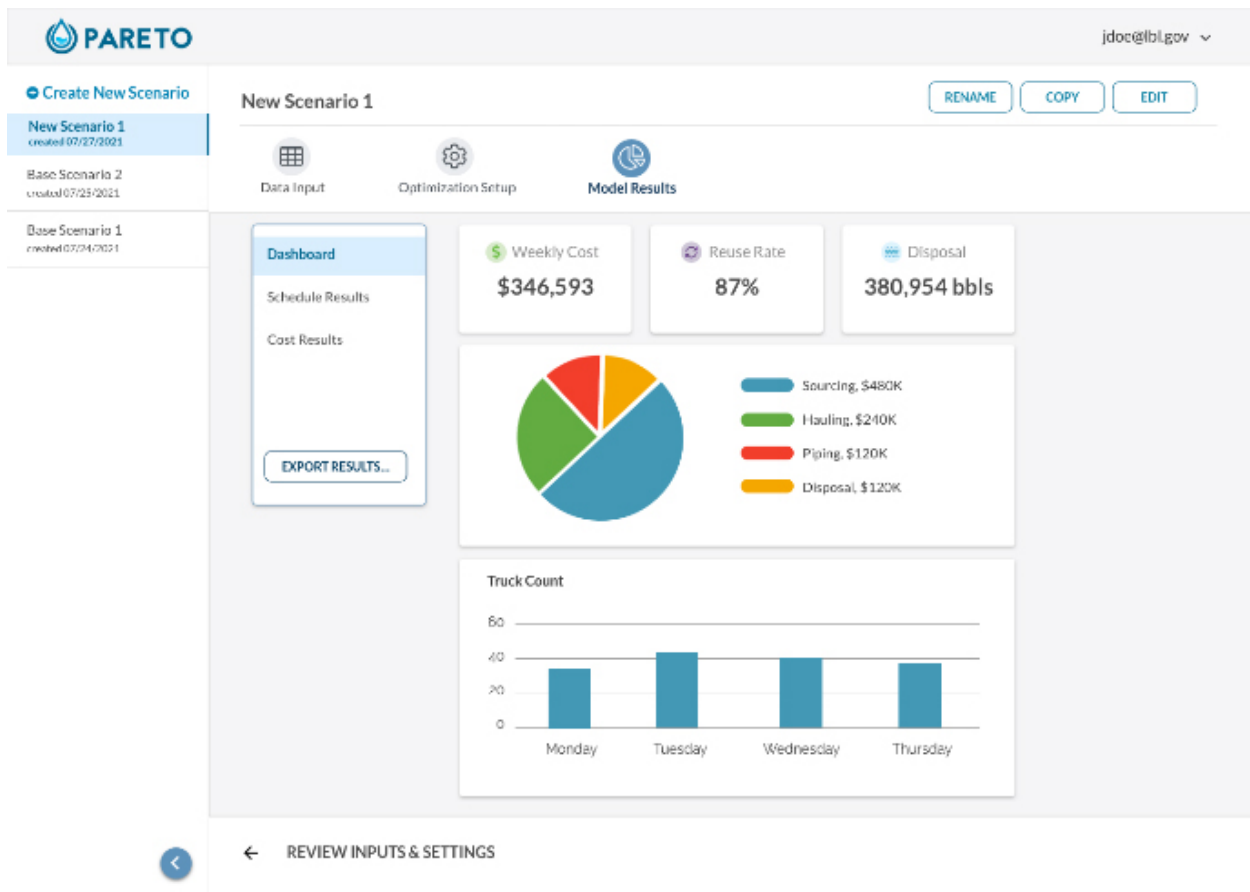   - Capital expenditures

Fig. 3: Figure 3. Output Data

It should be noted that PARETO users will be able to choose from a range of objectives for their optimization runs; these can range from minimizing costs to maximing the ruese of produced water (or combinations thereof).

5.2.2 Water Management Model Library

Operational Water Management

Overview

Given a fixed network of pads (completion and/or production), storage tanks, water forecasts (both consumption and production), and distribution options (trucks and/or pipelines), the operational water management model provides insight into the operational costs associated with water management. The operational model allows the user to

explore the tradeoff between minimizing costs (distribution, storage, treatment, disposal, etc.) and maximizing reuse water.

Operational Model Mathematical Notation

Sets

$t \in T$ Time periods (i.e. days)

$p \in P$ Well pads

$p \in PP$ Production pads (subset of well pads P)

$p \in CP$ Completions pads (subset of well pads P)

$f \in F$ Freshwater sources

$k \in K$ Disposal sites

$s \in S$ Storage sites

$r \in R$ Treatment sites

$o \in O$ Beneficial Reuse options

$n \in N$ Network nodes

$l \in L$ Locations (superset of well pads, disposal sites, nodes, ...)

$a \in A$ Production tanks

$(p, p) \in PCA$ Production-to-completions pipeline arcs

$(p, n) \in PNA$ Production-to-node pipeline arcs

$(p, p) \in PPA$ Production-to-production pipeline arcs

$(p, n) \in CNA$ Completions-to-node pipeline arcs

$(p, p) \in CCA$ Completions-to-completions pipeline arcs

$(n, n) \in NNA$ Node-to-node pipeline arcs

$(n, p) \in NCA$ Node-to-completions pipeline arcs

$(n, k) \in NKA$ Node-to-disposal pipeline arcs

$(n, s) \in NSA$ Node-to-storage pipeline arcs

$(n, r) \in NRA$ Node-to-treatment pipeline arcs

$(n, o) \in NOA$ Node-to-beneficial reuse pipeline arcs

$(f, p) \in FCA$ Freshwater-to-completions pipeline arcs

$(r, n) \in RNA$ Treatment-to-node pipeline arcs

$(r, p) \in RCA$ Treatment-to-completions pipeline arcs

$(r, k) \in RKA$ Treatment-to-disposal pipeline arcs

$(s, n) \in SNA$ Storage-to-node pipeline arcs

$(s, p) \in SCA$ Storage-to-completions pipeline arcs

$(s, k) \in SKA$ Storage-to-disposal pipeline arcs

$(s, r) \in SRA$ Storage-to-treatment pipeline arcs

$(s, o) \in SOA$ Storage-to-beneficial reuse pipeline arcs

$(p, p) \in PCT$ Production-to-completions trucking arcs

$(f, c) \in FCT$ Freshwater-to-completions trucking arcs

$(p, k) \in PKT$ Production-to-disposal trucking arcs

$(p, s) \in PST$ Production-to-storage trucking arcs

$(p, r) \in PRT$ Production-to-treatment trucking arcs

$(p, o) \in POT$ Production-to-beneficial reuse trucking arcs

$(p, k) \in CKT$ Completions-to-disposal trucking arcs

$(p, s) \in CST$ Completions-to-storage trucking arcs

$(p, r) \in CRT$ Completions-to-treatment trucking arcs

$(p, p) \in CCT$ Completions-to-completions trucking arcs (flowback reuse)

$(s, p) \in SCT$ Storage-to-completions trucking arcs

$(s, k) \in SKT$ Storage-to-disposal trucking arcs

$(r, k) \in RKT$ Treatment-to-disposal trucking arcs

$(p, a) \in PAL$ Pad-to-tank links

Continuous Variables

$F_{l,l,t}^{Piped}$ = Produced water piped from one location to another location

$F_{l,l,t}^{Trucked}$ = Produced water trucked from one location to another location

$F_{f,p,t}^{Sourced}$ = Fresh water sourced from source to completions

$F_{p,t}^{PadStorageIn}$ = Water put into completions pad storage

$F_{p,t}^{PadStorageOut}$ = Water removed from completions pad storage

$F_{r,t}^{TreatmentDestination}$ = Water delivered to treatment site

$F_{r,t}^{UnusedTreatedWater}$ = Water leftover from the treatment process

$F_{k,t}^{DisposalDestination}$ = Water injected at disposal site

$F_{o,t}^{BeneficialReuseDestination}$ = Water delivered to beneficial reuse site

If the production tanks are separate, water level and water drainage are tracked at each individual production tank:

$F_{p,a,t}^{DrainF}$ = Produced water drained from production tank

$L_{p,a,t}^{ProdTankL}$ = Water level in production tank at the end of time period t

Otherwise, if the production tanks are equalized, the water level and water drainage can be aggregated to a pad level:

$F_{p,t}^{DrainF}$ = Produced water drained from equalized production tanks

$L_{p,t}^{ProdTank}$ = Water level in equalized production tanks at the end of time period t

$B_{p,t}^{ProductionB}$ = Produced water for transport from pad

$L_{s,t}^{Storage}$ = Water level in storage site at the end of time period t

$L_{p,t}^{PadStorage}$ = Water level in completions pad storage at the end of time period t

$C_{l,l,t}^{Piped}$ = Cost of piping produced water from one location to another location

$C_{l,l,t}^{Trucked}$ = Cost of trucking produced water from one location to another location

$C_{f,p,t}^{Sourced}$ = Cost of sourcing fresh water from source to completions pad

$C_{k,t}^{Disposal}$ = Cost of injecting produced water at disposal site

$C_{r,t}^{Treatment}$ = Cost of treating produced water at treatment site

$C_{p,t}^{CompletionsReuse}$ = Cost of reusing produced water at completions site

$C_{s,t}^{Storage}$ = Cost of storing produced water at storage site (incl. treatment)

$R_{s,t}^{Storage}$ = Credit for retrieving stored produced water from storage site

$C^{TotalSourced}$ = Total cost of sourcing freshwater

$C^{TotalDisposal}$ = Total cost of injecting produced water

$C^{TotalTreatment}$ = Total cost of treating produced water

$C^{TotalCompletionsReuse}$ = Total cost of reusing produced water

$C^{TotalPiping}$ = Total cost of piping produced water

$C^{TotalStorage}$ = Total cost of storing produced water

$C^{TotalPadStorage}$ = Total cost of storing produced water at completions pad

$C^{TotalTrucking}$ = Total cost of trucking produced water

C^{Slack} = Total cost of slack variables

$R^{TotalStorage}$ = Total credit for withdrawing produced water

$S_{p,t}^{FracDemand}$ = Slack variable to meet the completions water demand

$S_{p,t}^{Production}$ = Slack variable to process produced water production

$S_{p,t}^{Flowback}$ = Slack variable to process flowback water production

$S_{l,l}^{PipelineCapacity}$ = Slack variable to provide necessary pipeline capacity

$S_{s}^{StorageCapacity}$ = Slack variable to provide necessary storage capacity

$S_{k}^{DisposalCapacity}$ = Slack variable to provide necessary disposal capacity

$S_{r}^{TreatmentCapacity}$ = Slack variable to provide necessary treatment capacity

$S_{o}^{BeneficialReuseCapacity}$ = Slack variable to provide necessary beneficial reuse capacity

Binary Variables

$y_{l,l,t}^{Flow}$ = Directional flow between two locations

$z_{p,t}^{PadStorage}$ = Completions pad storage use

Parameters

$y_{p,t}^{Completions}$ = Completions demand at a completions site in a time period

If the production tanks are separate, water level and water drainage are tracked at each individual production tank:

$\beta_{p,a,t}^{Production}$ = Produced water supply forecast for a production pad

$\sigma_{p,a}^{ProdTank}$ = Production tank capacity

$\lambda_{p,a}^{ProdTank}$ = Initial water level in production tank

Otherwise, if the production tanks are equalized, the water level and water drainage can be aggregated to a pad level:

$\beta_{p,t}^{Production}$ = Produced water supply forecast for a production pad

$\sigma_{p}^{ProdTank}$ = Combined capacity of equalized production tanks

$\lambda_{p}^{ProdTank}$ = Initial water level in equalized production tanks

$\beta_{p,t}^{Flowback}$ = Flowback supply forecast for a completions pad

$\sigma_{l,l}^{Pipeline}$ = Daily pipeline capacity between two locations

$\sigma_{k}^{Disposal}$ = Daily disposal capacity at a disposal site

$\sigma_{s}^{Storage}$ = Storage capacity at a storage site

$\sigma_{p,t}^{PadStorage}$ = Storage capacity at completions site

$\sigma_{r}^{Treatment}$ = Daily treatment capacity at a treatment site

$\sigma_{o}^{BeneficialReuse}$ = Daily reuse capacity at a beneficial reuse site

$\sigma_{f,t}^{Freshwater}$ = Daily freshwater sourcing capacity at freshwater source

$\sigma_{p}^{Offloading,Pad}$ = Daily truck offloading sourcing capacity per pad

$\sigma_{s}^{Offloading,Storage}$ = Daily truck offloading sourcing capacity per storage site

$\sigma_{p}^{Processing,Pad}$ = Daily processing (e.g. clarification) capacity per pad

$\sigma_{s}^{Processing,Storage}$ = Daily processing (e.g. clarification) capacity at storage site

$\epsilon_{r,w}^{Treatment}$ = Treatment efficiency for water quality component at treatment site

δ^{Truck} = Truck Capacity

$\tau_{p,p}^{Trucking}$ = Drive time between two pads

$\tau_{p,k}^{Trucking}$ = Drive time from a pad to a disposal site

$\tau_{p,s}^{Trucking}$ = Drive time from a pad to a storage site

$\tau_{p,r}^{Trucking}$ = Drive time from a pad to a treatment site

$\tau_{p,o}^{Trucking}$ = Drive time from a pad to a beneficial reuse site

$\tau_{s,p}^{Trucking}$ = Drive time from a storage site to a completions site

$\tau_{s,k}^{Trucking}$ = Drive time from a storage site to a disposal site

$\tau_{r,k}^{Trucking}$ = Drive time from a treatment site to a disposal site

$\lambda_s^{Storage}$ = Initial storage level at storage site

$\lambda_p^{PadStorage}$ = Initial storage level at completions site

$\theta_p^{PadStorage}$ = Terminal storage level at completions site

$\lambda_{l,l}^{Pipeline}$ = Pipeline segment length

$\pi_k^{Disposal}$ = Disposal operational cost

$\pi_r^{Treatment}$ = Treatment operational cost (may include "clean brine")

$\pi_p^{CompletionReuse}$ = Completions reuse operational cost

$\pi_s^{Storage}$ = Storage deposit operational cost

$\pi_{p,t}^{PadStorage}$ = Completions pad operational cost

$\rho_s^{Storage}$ = Storage withdrawal operational cgreenit

$\pi_{l,l}^{Pipeline}$ = Pipeline operational cost

$\pi_l^{Trucking}$ = Trucking hourly cost (by source)

$\pi_f^{Sourcing}$ = Fresh sourcing cost (does not include transportation cost)

M^{Flow} = Big-M flow parameter

$\psi^{FracDemand}$ = Slack cost parameter

$\psi^{Production}$ = Slack cost parameter

$\psi^{Flowback}$ = Slack cost parameter

$\psi^{PipelineCapacity}$ = Slack cost parameter

$\psi^{StorageCapacity}$ = Slack cost parameter

$\psi^{DisposalCapacity}$ = Slack cost parameter

$\psi^{TreamentCapacity}$ = Slack cost parameter

$\psi^{BeneficialReuseCapacity}$ = Slack cost parameter

Operational Model Mathematical Program Formulation

The default objective function for this produced water operational model is to minimize costs, which includes operational costs associated with procurement of fresh water, the cost of disposal, trucking and piping produced water between well pads and treatment facilities, and the cost of storing, treating and reusing produced water. A credit for using treated water is also considered, and additional slack variables are included to facilitate the identification of potential issues with input data.

Objective

$$min = C^{TotalSourced} + C^{TotalDisposal} + C^{TotalTreatment} + C^{TotalCompletionsReuse} +$$

$$C^{TotalPiping} + C^{TotalStorage} + C^{TotalPadStorage} + C^{TotalTrucking} + C^{Slack} - R^{TotalStorage}$$

Completions Pad Demand Balance: $\forall p \in CP, t \in T$

$$\gamma_{p,t}^{Completions} = \sum_{(n,p) \in NCA} F_{l,l,t}^{Piped} + \sum_{(p,p) \in PCA} F_{l,l,t}^{Piped} + \sum_{(s,p) \in SCA} F_{l,l,t}^{Piped}$$

$$+ \sum_{(p,c) \in CCA} F_{l,l,t}^{Piped} + \sum_{(r,p) \in RCA} F_{l,l,t}^{Piped} + \sum_{(f,p) \in FCA} F_{l,l,t}^{Sourced}$$

$$+ \sum_{(p,p) \in PCT} F_{l,l,t}^{Trucked} + \sum_{(s,p) \in SCT} F_{l,l,t}^{Trucked} + \sum_{(p,p) \in CCT} F_{l,l,t}^{Trucked}$$

$$+ \sum_{(f,p) \in FCT} F_{l,l,t}^{Trucked} + F_{p,t}^{PadStorageOut} - F_{p,t}^{PadStorageIn} + S_{p,t}^{FracDemand}$$

Completions Pad Storage Balance: $\forall p \in CP, t \in T$

This constraint sets the storage level at the completions pad. For each completions pad and for each time period, completions pad storage is equal to storage in last time period plus water put in minus water removed. If it is the first time period, the pad storage is the initial pad storage.

$$L_{p,t}^{PadStorage} = \lambda_{p,t=1}^{PadStorage} + L_{p,t-1}^{PadStorage} + F_{p,t}^{StorageIn} - F_{p,t}^{StorageOut}$$

Completions Pad Storage Capacity: $\forall p \in CP, t \in T$

The storage at each completions pad must always be at or below its capacity in every time period.

$$L_{p,t}^{PadStorage} \le z_{p,t}^{PadStorage} \cdot \sigma_{p,t}^{PadStorage}$$

Terminal Completions Pad Storage Level: $\forall p \in CP, t \in T$

$$L_{p,t=T}^{PadStorage} \le \theta_p^{PadStorage}$$

The storage in the last period must be at or below its terminal storage level.

Freshwater Sourcing Capacity: $\forall f \in F, t \in T$

For each freshwater source and each time period, the outgoing water from the freshwater source is below the freshwater capacity.

$$\sum_{(f,p) \in FCA} F_{l,l,t}^{Sourced} + \sum_{(f,p) \in FCT} F_{l,l,t}^{Trucked} \le \sigma_{f,t}^{Freshwater}$$

Completions Pad Truck Offloading Capacity: $\forall p \in CP, t \in T$

For each completions pad and time period, the volume of water being trucked into the completions pad must be below the trucking offloading capacity.

$$\sum_{(p,p) \in PCT} F_{l,l,t}^{Trucked} + \sum_{(s,p) \in SCT} F_{l,l,t}^{Trucked} + \sum_{(f,p) \in FCT} F_{l,l,t}^{Trucked}$$

$$+ \sum_{(p,p) \in CCT} F_{l,l,t}^{Trucked} \le \sigma_p^{Offloading,Pad}$$

Completions Pad Processing Capacity:

For each completions pad and time period, the volume of water (excluding freshwater) coming in must be below the processing limit.

$$\sum_{(n,p) \in NCA} F_{l,l,t}^{Piped} + \sum_{(p,p) \in PCA} F_{l,l,t}^{Piped} + \sum_{(s,p) \in SCA} F_{l,l,t}^{Piped}$$

$$+ \sum_{(p,c) \in CCA} F_{l,l,t}^{Piped} + \sum_{(r,p) \in RCA} F_{l,l,t}^{Piped} + \sum_{(p,p) \in PCT} F_{l,l,t}^{Trucked}$$

$$+ \sum_{(s,p) \in SCT} F_{l,l,t}^{Trucked} + \sum_{(p,p) \in CCT} F_{l,l,t}^{Trucked} \le \sigma_p^{Processing,Pad}$$

Note: This constraint has not actually been implemented yet.

Storage Site Truck Offloading Capacity: $\forall p \in S, t \in T$

For each storage site and each time period, the volume of water being trucked into the storage site must be below the trucking offloading capacity for that storage site.

$$\sum\nolimits_{(p,s) \in PST} F_{l,l,t}^{Trucked} + \sum\nolimits_{(p,s) \in CST} F_{l,l,t}^{Trucked} \leq \sigma_s^{Offloading,Storage}$$

Storage Site Processing Capacity: $\forall s \in S, t \in T$

For each storage site and each time period, the volume of water being trucked into the storage site must be less than the processing capacity for that storage site.

$$\sum\nolimits_{(n,s) \in NSA} F_{l,l,t}^{Piped} + \sum\nolimits_{(p,s) \in PST} F_{l,l,t}^{Trucked} + \sum\nolimits_{(p,s) \in CST} F_{l,l,t}^{Trucked} \leq \sigma_s^{Processing,Storage}$$

Production Tank Balance:

If there are individual production tanks, the water level must be tracked at each tank. The water level at a given tank at the end of each period is equal to the water level at the previous period plus the flowback supply forecast at the pad minus the water that is drained. If it is the first period, it is equal to the initial water level.

For individual production tanks: $\forall (p,a) \in PAL, t \in T$

$$L_{p,a,t}^{ProdTank} = \lambda_{p,a,t=1}^{ProdTank} + L_{p,a,t-1}^{ProdTank} + \beta_{p,a,t}^{Production} - F_{p,a,t}^{Drain}$$

For equalized production tanks: $\forall p \in P, t \in T$

$$L_{p,t}^{ProdTank} = \lambda_{p,t=1}^{ProdTank} + L_{p,t-1}^{ProdTank} + \beta_{p,t}^{Production} - F_{p,t}^{Drain}$$

Production Tank Capacity:

The water level at the production tanks must always be below the production tank capacity.

For individual production tanks: $\forall (p,a) \in PAL, t \in T$

$$L_{p,a,t}^{ProdTank} \leq \sigma_{p,a}^{ProdTank}$$

For equalized production tanks: $\forall p \in P, t \in T$

$$L_{p,t}^{ProdTank} \leq \sigma_p^{ProdTank}$$

Terminal Production Tank Level Balance:

The water level at the production tanks in the final time period must be below the terminal production tank water level parameter.

For individual production tanks: $\forall (p,a) \in PAL, t \in T$

$$L_{p,a,t=T}^{ProdTank} \leq \lambda_{p,a,t=1}^{ProdTank}$$

For equalized production tanks: $\forall p \in P, t \in T$

$$L_{p,t=T}^{ProdTank} \leq \lambda_{p,t=1}^{ProdTank}$$

Tank-to-Pad Production Balance:

If there are individual production tanks, the water drained across all tanks at the completions pad must be equal to the produced water for transport at the pad.

For individual production tanks: $\forall p \in P, t \in T$

$$\sum_{(p,a)\in PAL} F^{Drain}_{p,a,t} = B^{Production}_{p,t}$$

Otherwise, if the production tanks are equalized, the production water drained is measured on an aggregated production pad level.

For equalized production tanks: $\forall p \in P, t \in T$

$$F^{Drain}_{p,t} = B^{Production}_{p,t}$$

Note: The constraint proposed above is not necessary but included to facilitate switching between (1) an equalized production tank version and (2) a non-equalized production tank version.

Production Pad Supply Balance: $\forall p \in PP, t \in T$

All produced water must be accounted for. For each production pad and for each time period, the volume of outgoing water must be equal to the produced water transported out of the production pad.

$$B^{Production}_{p,t} = \sum_{(p,n)\in PNA} F^{Piped}_{l,l,t} + \sum_{(p,p)\in PCA} F^{Piped}_{l,l,t} + \sum_{(p,p)\in PPA} F^{Piped}_{l,l,t}$$
$$+ \sum_{(p,p)\in PCT} F^{Trucked}_{l,l,t} + \sum_{(p,k)\in PKT} F^{Trucked}_{l,l,t} + \sum_{(p,s)\in PST} F^{Trucked}_{l,l,t}$$
$$+ \sum_{(p,r)\in PRT} F^{Trucked}_{l,l,t} + \sum_{(p,o)\in POT} F^{Trucked}_{l,l,t} + S^{Production}_{p,t}$$

Completions Pad Supply Balance (i.e. Flowback Balance): $\forall p \in CP, t \in T$

All flowback water must be accounted for. For each completions pad and for each time period, the volume of outgoing water must be equal to the forecasted flowback produced water for the completions pad.

$$\beta^{Flowback}_{P,t} = \sum_{(p,n)\in CNA} F^{Piped}_{l,l,t} + \sum_{(p,c)\in CCA} F^{Piped}_{l,l,t} + \sum_{(p,p)\in CCT} F^{Trucked}_{l,l,t} +$$
$$\sum_{(p,k)\in CKT} F^{Trucked}_{l,l,t} + \sum_{(p,s)\in CST} F^{Trucked}_{l,l,t} + \sum_{(p,r)\in CRT} F^{Trucked}_{l,l,t} + S^{Flowback}_{p,t}$$

Network Node Balance: $\forall n \in N, t \in T$

Flow balance constraint (i.e., inputs are equal to outputs). For each pipeline node and for each time period, the volume water into the node is equal to the volume of water out of the node.

$$\sum_{(p,n)\in PNA} F^{Piped}_{l,l,t} + \sum_{(p,n)\in CNA} F^{Piped}_{l,l,t} + \sum_{(n,n)\in NNA} F^{Piped}_{l,l,t} + \sum_{(s,n)\in SNA} F^{Piped}_{l,l,t}$$
$$= \sum_{(n,n)\in NNA} F^{Piped}_{l,l,t} + \sum_{(n,p)\in NCA} F^{Piped}_{l,l,t} + \sum_{(n,k)\in NKA} F^{Piped}_{l,l,t}$$
$$+ \sum_{(n,r)\in NRA} F^{Piped}_{l,l,t} + \sum_{(n,s)\in NSA} F^{Piped}_{l,l,t} + \sum_{(n,o)\in NOA} F^{Piped}_{l,l,t}$$

Bi-Directional Flow: $\forall (l,l) \in \{PCA,PNA,PPA,CNA,NNA,NCA,NKA,NSA,NRA,\dots,SOA\}, t \in T$

There can only be flow in one direction for a given pipeline arc in a given time period.

Flow is only allowed in a given direction if the binary indicator for that direction is "on".

$$y^{Flow}_{l,l,t} + y^{Flow}_{l,l,t} = 1$$

Note: Technically this constraint should only be enforced for truly reversible arcs (e.g. NCA and CNA); and even then it only needs to be defined per one reversible arc (e.g. NCA only and not NCA and CNA).

$$F_{l,l,t}^{Piped} \leq y_{l,l,t}^{Flow} \cdot M^{Flow}$$

Storage Site Balance: $\forall s \in S, t \in T$

For each storage site and for each time period, if it is the first time period, the storage level is the initial storage. Otherwise, the storage level is equal to the storage level in the previous time period plus water inputs minus water outputs.

$$L_{s,t}^{Storage} = \lambda_{s,t=1}^{Storage} + L_{s,t-1}^{Storage} + \sum_{(n,s)\in NSA} F_{l,l,t}^{Piped} + \sum_{(p,s)\in PST} F_{l,l,t}^{Trucked}$$

$$+ \sum_{(p,s)\in CST} F_{l,l,t}^{Trucked} - \sum_{(s,n)\in SNA} F_{l,l,t}^{Piped} - \sum_{(s,p)\in SCA} F_{l,l,t}^{Piped} - \sum_{(s,k)\in SKA} F_{l,l,t}^{Piped}$$

$$- \sum_{(s,r)\in SRA} F_{l,l,t}^{Piped} - \sum_{(s,o)\in SOA} F_{l,l,t}^{Piped} - \sum_{(s,p)\in SCT} F_{l,l,t}^{Trucked} - \sum_{(s,k)\in SKT} F_{l,l,t}^{Trucked}$$

Pipeline Capacity:

$\forall (l,l) \in \{PCA,PNA,PPA,CNA,NNA,NCA,NKA,NSA,NRA,\ldots,SOA\}, [t \in T]$

$$F_{l,l,[t]}^{Capacity} = \sigma_{l,l}^{Pipeline} + S_{l,l}^{PipelineCapacity}$$

$\forall (l,l) \in \{PCA,PNA,PPA,CNA,NNA,NCA,NKA,NSA,NRA,\ldots,SOA\}, t \in T$

$$F_{l,l,t}^{Piped} \leq F_{l,l,[t]}^{Capacity}$$

Storage Capacity:

The total stored water in a given time period must be less than the capacity. If the storage capacity limits the feasibility, the slack variable will be nonzero, and the storage capacity will be increased to allow a feasible solution.

$\forall s \in S, [t \in T]$

$$X_{s,[t]}^{Capacity} = \sigma_s^{Storage} + S_s^{StorageCapacity}$$

$\forall s \in S, t \in T$

$$L_{s,t}^{Storage} \leq X_{s,[t]}^{Capacity}$$

Disposal Capacity:

The total disposed water in a given time period must be less than the capacity. If the disposal capacity limits the feasibility, the slack variable will be nonzero, and the disposal capacity will be increased to allow a feasible solution.

$\forall k \in K, [t \in T]$

$$D_{k,[t]}^{Capacity} = \sigma_k^{Disposal} + S_k^{DisposalCapacity}$$

$\forall k \in K, t \in T$

$$\sum_{(n,k)\in NKA} F_{l,l,t}^{Piped} + \sum_{(s,k)\in SKA} F_{l,l,t}^{Piped} + \sum_{(s,k)\in SKT} F_{l,l,t}^{Trucked} + \sum_{(p,k)\in PKT} F_{l,l,t}^{Trucked}$$
$$+ \sum_{(p,k)\in CKT} F_{l,l,t}^{Trucked} + \sum_{(r,k)\in RKT} F_{l,l,t}^{Trucked} \leq D_{k,[t]}^{Capacity}$$

$\forall k \in K, t \in T$

$$\sum_{(n,k)\in NKA} F_{l,l,t}^{Piped} + \sum_{(s,k)\in SKA} F_{l,l,t}^{Piped} + \sum_{(s,k)\in SKT} F_{l,l,t}^{Trucked} + \sum_{(p,k)\in PKT} F_{l,l,t}^{Trucked}$$
$$+ \sum_{(p,k)\in CKT} F_{l,l,t}^{Trucked} + \sum_{(r,k)\in RKT} F_{l,l,t}^{Trucked} = F_{k,t}^{DisposalDestination}$$

Treatment Capacity:

The total treated water in a given time period must be less than the capacity. If the treatment capacity limits the feasibility, the slack variable will be nonzero, and the treatment capacity will be increased to allow a feasible solution.

$\forall r \in R, t \in T$

$$\sum_{(n,r)\in NRA} F_{l,l,t}^{Piped} + \sum_{(s,r)\in SRA} F_{l,l,t}^{Piped} + \sum_{(p,r)\in PRT} F_{l,l,t}^{Trucked}$$
$$+ \sum_{(p,r)\in CRT} F_{l,l,t}^{Trucked} \leq \sigma_r^{Treatment} + S_r^{TreatmentCapacity}$$

$\forall r \in R, t \in T$

$$\sum_{(n,r)\in NRA} F_{l,l,t}^{Piped} + \sum_{(s,r)\in SRA} F_{l,l,t}^{Piped} + \sum_{(p,r)\in PRT} F_{l,l,t}^{Trucked}$$
$$+ \sum_{(p,r)\in CRT} F_{l,l,t}^{Trucked} = F_{r,t}^{TreatmentDestination}$$

Beneficial Reuse Capacity:

The total water for beneficial reuse in a given time period must be less than the capacity. If the beneficial reuse capacity limits the feasibility, the slack variable will be nonzero, and the beneficial reuse capacity will be increased to allow a feasible solution.

$\forall o \in O, t \in T$

$$\sum_{(n,o)\in NOA} F_{l,l,t}^{Piped} + \sum_{(s,o)\in SOA} F_{l,l,t}^{Piped} + \sum_{(p,o)\in POT} F_{l,l,t}^{Trucked} \leq \sigma_o^{Reuse} + S_o^{ReuseCapacity}$$

$\forall o \in O, t \in T$

$$\sum_{(n,o)\in NOA} F_{l,l,t}^{Piped} + \sum_{(s,o)\in SOA} F_{l,l,t}^{Piped} + \sum_{(p,o)\in POT} F_{l,l,t}^{Trucked} = F_{o,t}^{BeneficialReuseDestination}$$

Fresh Sourcing Cost: $\forall f \in F, p \in CP, t \in T$

For each freshwater source, for each completions pad, and for each time period, the freshwater sourcing cost is equal to all output from the freshwater source times the freshwater sourcing cost.

$$C_{f,p,t}^{Sourced} = (F_{f,p,t}^{Sourced} + F_{f,p,t}^{Trucked}) \cdot \pi_f^{Sourcing}$$
$$C^{TotalSourced} = \sum_{\forall t\in T} \sum_{(f,p)\in FCA} C_{f,p,t}^{Sourced}$$

Disposal Cost: $\forall k \in K, t \in T$

For each disposal site, for each time period, the disposal cost is equal to all water moved into the disposal site multiplied by the operational disposal cost. Total disposal cost is the sum of disposal costs over all time periods and all disposal sites.

$$C_{k,t}^{Disposal} = (\sum_{(l,l)\in NKA,RKA,SKA} F_{l,l,t}^{Piped} + \sum_{(l,l)\in PKT,CKT,SKT,RKT} F_{l,l,t}^{Trucked}) \cdot \pi_k^{Disposal}$$
$$C^{TotalDisposal} = \sum_{\forall t\in T} \sum_{k\in K} C_{k,t}^{Disposal}$$

Treatment Cost: $\forall r \in R, t \in T$

For each treatment site, for each time period, the treatment cost is equal to all water moved to the treatment site multiplied by the operational treatment cost. The total treatments cost is the sum of treatment costs over all time periods and all treatment sites.

$$C_{r,t}^{Treatment} = (\sum_{(l,l)\in NRA,SRA} F_{l,l,t}^{Piped} + \sum_{(l,l)\in PRT,CRT} F_{l,l,t}^{Trucked}) \cdot \pi_r^{Treatment}$$
$$C^{TotalTreatment} = \sum_{\forall t\in T} \sum_{r\in R} C_{r,t}^{Treatment}$$

Treatment Balance: $\forall r \in R, t \in T$

Water input into treatment facility is treated with a level of efficiency, meaning only a given percentage of the water input is outputted to be reused at the completions pads.

$$\epsilon^{Treatment} \cdot \left(\sum_{(n,r)\in NRA} F^{Piped}_{l,l,t} + \sum_{(s,r)\in SRA} F^{Piped}_{l,l,t} + \sum_{(p,r)\in PRT} F^{Trucked}_{l,l,t}\right.$$
$$\left.+ \sum_{(p,r)\in CRT} F^{Trucked}_{l,l,t}\right) = \sum_{(r,p)\in RCA} F^{Piped}_{l,l,t} + F^{UnusedTreatedWater}_{r,t}$$

where $\epsilon^{Treatment} < 1$

Completions Reuse Cost: $\forall p \in P, t \in T$

Completions reuse water is all water that meets completions pad demand, excluding freshwater. Completions reuse cost is the volume of completions reused water multiplied by the cost for reuse.

$$C^{CompletionsReuse}_{p,t} = \left(\sum_{(n,p)\in NCA} F^{Piped}_{l,l,t} + \sum_{(p,p)\in PCA} F^{Piped}_{l,l,t} + \sum_{(r,p)\in RCA} F^{Piped}_{l,l,t}\right.$$
$$+ \sum_{(s,p)\in SCA} F^{Piped}_{l,l,t} + \sum_{(p,c)\in CCA} F^{Piped}_{l,l,t} + \sum_{(p,p)\in CCT} F^{Trucked}_{l,l,t}$$
$$\left.+ \sum_{(p,p)\in PCT} F^{Trucked}_{l,l,t} + \sum_{(s,p)\in SCT} F^{Trucked}_{l,l,t}\right) \cdot \pi^{CompletionsReuse}_{p}$$

Note: Freshwater sourcing is excluded from completions reuse costs.

$$C^{TotalCompletionsReuse} = \sum_{\forall t \in T} \sum_{p\in CP} C^{CompletionsReuse}_{p,t}$$

Piping Cost: $\forall (l,l) \in \{PPA,...,CCA\}, t \in T$

Piping cost is the total volume of piped water multiplied by the cost for piping.

$$C^{Piped}_{l,l,t} = (F^{Piped}_{l,l,t} + F^{Sourced}_{l,l,t}) \cdot \pi^{Pipeline}_{l,l}$$
$$C^{TotalPiping} = \sum_{(} t \in T \sum_{\forall (l,l)\in PPA,...} C^{Piped}_{l,l,t}$$

Note: The constraints above explicitly consider freshwater piping via FCA arcs.

Storage Deposit Cost: $\forall s \in S, t \in T$

Cost of depositing into storage is equal to the total volume of water moved into storage multiplied by the storage operation cost rate.

$$C^{Storage}_{s,t} = \left(\sum_{(l,l)\in NSA} F^{Piped}_{l,l,t} + \sum_{(l,l)\in CST} F^{Trucked}_{l,l,t} + \sum_{(l,s)\in PST} F^{Trucked}_{l,s,t}\right) \cdot \pi^{Storage}_{s}$$
$$C^{TotalStorage} = \sum_{\forall t \in T} \sum_{\forall s \in S} C^{Storage}_{s,t}$$

Storage Withdrawal Credit: $\forall s \in S, t \in T$

Credits from withdrawing from storage is equal to the total volume of water moved out from storage multiplied by the storage operation credit rate.

$$R^{Storage}_{s,t} = \left(\sum_{(l,l)\in SNA,SCA,SKA,SRA,SOA} F^{Piped}_{l,l,t} + \sum_{(l,l)\in SCT,SKT} F^{Trucked}_{l,l,t}\right) \cdot \rho^{Storage}_{s}$$
$$R^{TotalStorage} = \sum_{\forall t \in T} \sum_{\forall s \in S} R^{Storage}_{s,t}$$

Pad Storage Cost: $\forall l \in L, \hat{l} \in L, t \in T$

$$C^{TotalPadStorage} = \sum_{\forall t \in T} \sum_{\forall p \in CP} z_{p,t}^{PadStorage} \cdot \pi_{p,t}^{PadStorage}$$

Trucking Cost (Simplified)

Trucking cost between two locations for time period is equal to the trucking volume between locations in time t divided by the truck capacity [this gets # of truckloads] multiplied by the lead time between two locations and hourly trucking cost.

$$C_{l,l,t}^{Trucked} = F_{l,l,t}^{Trucked} \cdot 1/\delta^{Truck} \cdot \tau_{p,p}^{Trucking} \cdot \pi_l^{Trucking}$$

$$C^{TotalTrucking} = \sum_{\forall t \in T} \sum_{\forall (l,l) \in PPA,\ldots,CCT} C_{l,l,t}^{Trucked}$$

Note: The constraints above explicitly consider freshwater trucking via FCT arcs.

Slack Costs:

Weighted sum of the slack variables. In the case that the model is infeasible, these slack variables are used to determine where the infeasibility occurs (e.g. pipeline capacity is not sufficient).

$$C^{Slack} = \sum_{p \in CP} \sum_{t \in T} S_{p,t}^{FracDemand} \cdot \psi^{FracDemand} + \sum_{p \in PP} \sum_{t \in T} S_{p,t}^{Production} \cdot \psi^{Production}$$

$$+ \sum_{p \in CP} \sum_{t \in T} S_{p,t}^{Flowback} \cdot \psi^{Flowback} + \sum_{(l,l) \in \ldots} S_{l,l}^{PipelineCapacity} \cdot \psi^{PipeCapacity}$$

$$+ \sum_{s \in S} S_s^{StorageCapacity} \cdot \psi^{StorageCapacity} + \sum_{k \in K} S_k^{DisposalCapacity} \cdot \psi^{DisposalCapacity}$$

$$+ \sum_{r \in R} S_r^{TreatmentCapacity} \cdot \psi^{TreatmentCapacity} + \sum_{o \in O} S_o^{BeneficialReuseCapacity} \cdot \psi^{BeneficialReuseCapacity}$$

Operational Model Water Quality Extension

An extension to this operational optimization model measures the water quality across all locations over time. As of now, water quality is not a decision variable. It is calculated after optimization of the operational model. The process for calculating water quality is as follows: the operational model is first solved to optimality, water quality variables and constraints are added, flow rates and storage levels are fixed to the solved values at optimality, and the water quality is calculated.

Note: Fixed variables are denoted in purple in the documentation.

Assumptions:

- Water quality at a production pad or completions pad remains the same across all time periods
- When blending flows of different water quality, they blend linearly
- Treatment does not affect water quality

Water Quality Sets

$w \in W$ Water Quality Components (e.g., TDS)

Water Quality Parameters

$v_{l,w,[t]}$ = Water quality at well pad

$\xi_{l,w}$ = Initial water quality at storage

Water Quality Variables

$Q_{l,w,t}$ = Water quality at location

Disposal Site Water Quality $\forall k \in K, w \in W, t \in T$

The water quality of disposed water is dependent on the flow rates into the disposal site and the quality of each of these flows.

$$\sum_{(n,k)\in NKA} F_{l,l,t}^{Piped} \cdot Q_{n,w,t} + \sum_{(s,k)\in SKA} F_{l,l,t}^{Piped} \cdot Q_{s,w,t} + \sum_{(r,k)\in RKA} F_{l,l,t}^{Piped} \cdot Q_{r,w,t}$$

$$+ \sum_{(s,k)\in SKT} F_{l,l,t}^{Trucked} \cdot Q_{s,w,t} + \sum_{(p,k)\in PKT} F_{l,l,t}^{Trucked} \cdot Q_{p,w,t}$$

$$+ \sum_{(p,k)\in CKT} F_{l,l,t}^{Trucked} \cdot Q_{p,w,t} + \sum_{(r,k)\in RKT} F_{l,l,t}^{Trucked} \cdot Q_{r,w,t}$$

$$= F_{k,t}^{DisposalDestination} \cdot Q_{k,w,t}$$

Storage Site Water Quality $\forall s \in S, w \in W, t \in T$

The water quality at storage sites is dependent on the flow rates into the storage site, the volume of water in storage in the previous time period, and the quality of each of these flows. Even mixing is assumed, so all outgoing flows have the same water quality. If it is the first time period, the initial storage level and initial water quality replaces the water stored and water quality in the previous time period respectively.

$$\lambda_{s,t=1}^{Storage} \cdot \xi_{s,w} + L_{s,t-1}^{Storage} \cdot Q_{s,w,t-1} + \sum_{(n,s)\in NSA} F_{l,l,t}^{Piped} \cdot Q_{n,w,t}$$

$$+ \sum_{(p,s)\in PST} F_{l,l,t}^{Trucked} \cdot Q_{p,w,t} + \sum_{(p,s)\in CST} F_{l,l,t}^{Trucked} \cdot Q_{p,w,t}$$

$$= Q_{s,w,t} \cdot (L_{s,t}^{Storage} + \sum_{(s,n)\in SNA} F_{l,l,t}^{Piped} + \sum_{(s,p)\in SCA} F_{l,l,t}^{Piped} + \sum_{(s,k)\in SKA} F_{l,l,t}^{Piped}$$

$$+ \sum_{(s,r)\in SRA} F_{l,l,t}^{Piped} + \sum_{(s,o)\in SOA} F_{l,l,t}^{Piped} + \sum_{(s,p)\in SCT} F_{l,l,t}^{Trucked} + \sum_{(s,k)\in SKT} F_{l,l,t}^{Trucked})$$

Treatment Site Water Quality $\forall r \in R, w \in W, t \in T$

The water quality at treatment sites is dependent on the flow rates into the treatment site, the efficiency of treatment, and the water quality of the flows. Even mixing is assumed, so all outgoing flows have the same water quality. The treatment process does not affect water quality

$$\epsilon_{r,w}^{Treatment} \cdot (\sum_{(n,r)\in NRA} F_{l,l,t}^{Piped} \cdot Q_{n,w,t} + \sum_{(s,r)\in SRA} F_{l,l,t}^{Piped} \cdot Q_{s,w,t}$$

$$+ \sum_{(p,r)\in PRT} F_{l,l,t}^{Trucked} \cdot Q_{p,w,t} + \sum_{(p,r)\in CRT} F_{l,l,t}^{Trucked} \cdot Q_{p,w,t})$$

$$= Q_{r,w,t} \cdot (\sum_{(r,p)\in RCA} F_{l,l,t}^{Piped} + F_{r,t}^{UnusedTreatedWater})$$

where $\epsilon_{r,w}^{Treatment} < 1$

Network Node Water Quality $\forall n \in N, w \in W, t \in T$

The water quality at nodes is dependent on the flow rates into the node and the water quality of the flows. Even mixing is assumed, so all outgoing flows have the same water quality.

$$\sum_{(p,n)\in PNA} F_{l,l,t}^{Piped} \cdot Q_{p,w,t} + \sum_{(p,n)\in CNA} F_{l,l,t}^{Piped} \cdot Q_{p,w,t}$$

$$+ \sum_{(n,n)\in NNA} F_{l,l,t}^{Piped} \cdot Q_{n,w,t} + \sum_{(s,n)\in SNA} F_{l,l,t}^{Piped} \cdot Q_{s,w,t}$$

$$= Q_{n,w,t} \cdot (\sum_{(n,n)\in NNA} F_{l,l,t}^{Piped} + \sum_{(n,p)\in NCA} F_{l,l,t}^{Piped}$$

$$+ \sum_{(n,k)\in NKA} F_{l,l,t}^{Piped} + \sum_{(n,r)\in NRA} F_{l,l,t}^{Piped}$$

$$+ \sum_{(n,s)\in NSA} F_{l,l,t}^{Piped} + \sum_{(n,o)\in NOA} F_{l,l,t}^{Piped})$$

Beneficial Reuse Water Quality $\forall o \in O, w \in W, t \in T$

The water quality at beneficial reuse sites is dependent on the flow rates into the site and the water quality of the flows.

$$\sum_{(n,o)\in NOA} F_{l,l,t}^{Piped} \cdot Q_{n,w,t} + \sum_{(s,o)\in SOA} F_{l,l,t}^{Piped} \cdot Q_{s,w,t} + \sum_{(p,o)\in POT} F_{l,l,t}^{Trucked} \cdot Q_{p,w,t}$$
$$= Q_{o,w,t} \cdot F_{o,t}^{BeneficialReuseDestination}$$

Operational Model Terminology

Beneficial Reuse Options: This term refers to the reuse of water at mining facilities, farms, etc.

Completions Demand: Demand set by completions pads. This demand can be met by produced water, treated water, or freshwater.

Completions Reuse Water: Water that meets demand at a completions site. This does not include freshwater or water for beneficial reuse.

Network Nodes: These are branch points for pipelines only.

Note: Well pads are not a subset of network nodes.

---

[t]: This notation indicates that timing of capacity expansion has not yet been implemented.

Terminal Storage Level: These are goal storage levels for the final time period. Without this, the storage levels would likely be depleted in the last time period.

Strategic Water Management

Overview

Given a set of existing network components (completion pads, storage pads, production pads, and distribution options like trucks and/or pipelines) and capacity expansion options, the strategic water management model provides an insight into financial opportunities and mid-long term investment decisions to reduce operational costs or maximize reuse or reduce fresh water consumption.

Section
Terminology
Strategic Model Mathematical Notation
Strategic Model Mathematical Program Formulation
Strategic Model Water Quality Extension
Strategic Model Discrete Water Quality Extension
References

Terminology

Beneficial reuse options: This term refers to the reuse of water at mining facilities, farms, etc.

Completions demand: Demand set by completions pads. This demand can be met by produced water, treated water, or freshwater.

Completions reuse water: Water that meets demand at a completions site. This does not include freshwater or water for beneficial reuse.

Network Nodes: These are branch points for pipelines only.

Note: Well pads are not a subset of network nodes.

$[t]$ **or** $[t \in T]$ **:** This notation indicates that timing of capacity expansion has not yet been implemented.

Terminal storage level: These are goal storage levels for the final time period. Without this, the storage levels would likely be depleted in the last time period.

Water boosting: Moving large volumes of water requires water pumps. Water boosting refers to the infrastructure required to maintain water pressure.

Strategic Model Mathematical Notation

Sets

$t \in T$ Time periods (i.e. days)

$p \in P$ Well pads

$p \in PP$ Production pads (subset of well pads P)

$p \in CP$ Completions pads (subset of well pads P)

$f \in F$ Freshwater sources

$k \in K$ Disposal sites

$s \in S$ Storage sites

$r \in R$ Treatment sites

$o \in O$ Beneficial reuse options

$n \in N$ Network nodes

$l \in L$ Locations (superset of well pads, disposal sites, nodes, ...)

$d \in D$ Pipeline diameters

$c \in C$ Storage capacities

$j \in J$ Treatment capacities

$i \in I$ Injection (i.e. disposal) capacities

$(p, p) \in PCA$ Production-to-completions pipeline arcs

$(p, n) \in PNA$ Production-to-node pipeline arcs

$(p, p) \in PPA$ Production-to-production pipeline arcs

$(p, n) \in CNA$ Completions-to-node pipeline arcs

$(p, p) \in CCA$ Completions-to-completions pipeline arcs

$(n, n) \in NNA$ Node-to-node pipeline arcs

$(n, p) \in NCA$ Node-to-completions pipeline arcs

$(n, k) \in NKA$ Node-to-disposal pipeline arcs

$(n, s) \in NSA$ Node-to-storage pipeline arcs

$(n, r) \in NRA$ Node-to-treatment pipeline arcs

$(n, o) \in NOA$ Node-to-beneficial reuse pipeline arcs

$(f, p) \in FCA$ Freshwater-to-completions pipeline arcs

$(r, n) \in RNA$ Treatment-to-node pipeline arcs

$(r, p) \in RCA$ Treatment-to-completions pipeline arcs

$(r, k) \in RKA$ Treatment-to-disposal pipeline arcs

$(r, s) \in RSA$ Treatment-to-storage pipeline arcs

$(s, n) \in SNA$ Storage-to-node pipeline arcs

$(s, p) \in SCA$ Storage-to-completions pipeline arcs

$(s, k) \in SKA$ Storage-to-disposal pipeline arcs

$(s, r) \in SRA$ Storage-to-treatment pipeline arcs

$(s, o) \in SOA$ Storage-to-beneficial reuse pipeline arcs

$(p, p) \in PCT$ Production-to-completions trucking arcs

$(p, k) \in PKT$ Production-to-disposal trucking arcs

$(p, s) \in PST$ Production-to-storage trucking arcs

$(p, r) \in PRT$ Production-to-treatment trucking arcs

$(p, o) \in POT$ Production-to-beneficial reuse trucking arcs

$(f, p) \in FCT$ Freshwater-to-completions trucking arcs

$(p, k) \in CKT$ Completions-to-disposal trucking arcs

$(p, s) \in CST$ Completions-to-storage trucking arcs

$(p, r) \in CRT$ Completions-to-treatment trucking arcs

$(p, p) \in CCT$ Completions-to-completions trucking arcs (flowback reuse)

$(s, p) \in SCT$ Storage-to-completions trucking arcs

$(s, k) \in SKT$ Storage-to-disposal trucking arcs

$(r, k) \in RKT$ Treatment-to-disposal trucking arcs

Continuous Variables

$F_{l,\tilde{l},t}^{Piped}$ = Produced water piped from one location to another location

$F_{l,\tilde{l},t}^{Trucked}$ = Water trucked from one location to another location

$F_{f,p,t}^{Sourced}$ = Fresh water sourced from source to completions

$F_{p,t}^{PadStorageIn}$ = Water put into completions pad storage

$F_{p,t}^{PadStorageOut}$ = Water removed from completions pad storage

$F_{r,t}^{TreatmentDestination}$ = Water delivered to treatment site

$F_{r,t}^{UnusedTreatedWater}$ = Treated water that is not used

$F_{p,t}^{CompletionsReuseDestination}$ = Water delivered to completions pad for reuse

$F_{k,t}^{DisposalDestination}$ = Water injected at disposal site

$F_{o,t}^{BeneficialReuseDestination}$ = Water delivered to beneficial reuse

$F_{p,t}^{CompletionsDestination}$ = All water delivered to completions pad

$F_{p,t}^{BeneficialReuseDestination}$ = Water delivered to beneficial reuse site

$L_{s,t}^{Storage}$ = Water level at storage site at the end of time period t

$L_{p,t}^{PadStorage}$ = Water level in completions pad storage at the end of time period t

$F^{TotalTrucked}$ = Total volume of water trucked

$F^{TotalSourced}$ = Total volume of freshwater sourced

$F^{TotalDisposed}$ = Total volume of produced water disposed

$F^{TotalCompletionsReuse}$ = Total volume of produced water reused

$C_{l,\tilde{l},t}^{Piped}$ = Cost of piping produced water from one location to another location

$C_{l,\tilde{l},t}^{Trucked}$ = Cost of trucking produced water from one location to another location

$C_{f,p,t}^{Sourced}$ = Cost of sourcing fresh water from source to completions pad

$C_{k,t}^{Disposal}$ = Cost of injecting produced water at disposal site

$C_{r,t}^{Treatment}$ = Cost of treating produced water at treatment site

$C_{p,t}^{CompletionsReuse}$ = Cost of reusing produced water at completions site

$C_{s,t}^{Storage}$ = Cost of storing produced water at storage site (incl. treatment)

$R_{s,t}^{Storage}$ = Credit for retrieving stored produced water from storage site

$C^{TotalSourced}$ = Total cost of sourcing freshwater

$C^{TotalDisposal}$ = Total cost of injecting produced water

$C^{TotalTreatment}$ = Total cost of treating produced water

$C^{TotalCompletionsReuse}$ = Total cost of reusing produced water

$C^{TotalPiping}$ = Total cost of piping produced water

$C^{TotalStorage}$ = Total cost of storing produced water

$C^{TotalTrucking}$ = Total cost of trucking produced water

C^{Slack} = Total cost of slack variables

$R^{TotalStorage}$ = Total credit for withdrawing produced water

$D_{k,[t]}^{Capacity}$ = Disposal capacity in a given time period at disposal site

$X_{s,[t]}^{Capacity}$ = Storage capacity in a given time period at storage site

$T_{r,[t]}^{Capacity}$ = Treatment capacity in a given time period at treatment site

$F_{l,\tilde{l},[t]}^{Capacity}$ = Flow capacity in a given time period between two locations

$C_{[t]}^{DisposalCapEx}$ = Capital cost of constructing or expanding disposal capacity

$C_{[t]}^{PipelineCapEx}$ = Capital cost of constructing or expanding piping capacity

$C_{[t]}^{StorageCapEx}$ = Capital cost of constructing or expanding storage capacity

$C_{[t]}^{TreatmentCapEx}$ = Capital cost of constructing or expanding treatment capacity

$S_{p,t}^{FracDemand}$ = Slack variable to meet the completions water demand

$S_{p,t}^{Production}$ = Slack variable to process produced water production

$S_{p,t}^{Flowback}$ = Slack variable to process flowback water production

$S_{l,\tilde{l}}^{PipelineCapacity}$ = Slack variable to provide necessary pipeline capacity

$S_{s}^{StorageCapacity}$ = Slack variable to provide necessary storage capacity

$S_{k}^{DisposalCapacity}$ = Slack variable to provide necessary disposal capacity

$S_{r}^{TreamentCapacity}$ = Slack variable to provide necessary treatment capacity

$S_{o}^{BeneficialResueCapacity}$ = Slack variable to provide necessary beneficial reuse capacity

Binary Variables

$y_{l,\tilde{l},d,[t]}^{Pipeline}$ = New pipeline installed between one location and another location with specific diameter

$y_{s,c,[t]}^{Storage}$ = New or additional storage facility installed at storage site with specific storage capacity

$y_{r,j,[t]}^{Treatment}$ = New or additional treatment facility installed at treatment site with specific treatment capacity

$y_{k,i,[t]}^{Disposal}$ = New or additional disposal facility installed at disposal site with specific injection capacity

$y_{l,\tilde{l},t}^{Flow}$ = Directional flow between two locations

Parameters

$\gamma_{p,t}^{Completions}$ = Completions demand at a completions site in a time period

$\gamma^{TotalDemand}$ = Total water demand over the planning horizon

$\beta_{p,t}^{Production}$ = Produced water supply forecast for a production pad

$\beta_{p,t}^{Flowback}$ = Flowback supply forecast for a completions pad

$\beta^{TotalProd}$ = Total water production (production & flowback) over the planning horizon

$\sigma_{l,\tilde{l}}^{Pipeline}$ = Initial pipeline capacity between two locations

$\sigma_{k}^{Disposal}$ = Initial disposal capacity at a disposal site

$\sigma_{s}^{Storage}$ = Initial storage capacity at a storage site

---

$\sigma_{p,t}^{PadStorage}$ = Storage capacity at completions site

$\sigma_{r}^{Treatment}$ = Initial treatment capacity at a treatment site

$\sigma_{o}^{BeneficialReuse}$ = Initial reuse capacity at a reuse site

$\sigma_{f,t}^{Freshwater}$ = Freshwater sourcing capacity at freshwater source

$\sigma_{p}^{Offloading,Pad}$ = Truck offloading sourcing capacity per pad

$\sigma_{s}^{Offloading,Storage}$ = Truck offloading sourcing capacity per storage site

$\sigma_{p}^{Processing,Pad}$ = Processing (e.g. clarification) capacity per pad

$\sigma_{s}^{Processing,Storage}$ = Processing (e.g. clarification) capacity at storage site

σ_{n}^{Node} = Capacity per network node

$W_{r}^{TreatmentComponent}$ = Water quality component treated for at treatment site

$\epsilon_{r,w}^{Treatment}$ = Treatment efficiency at treatment site for quality component w

$\alpha^{AnnualizationRate}$ = Annualization Rate [%]

$\delta_{i}^{Disposal}$ = Increments for installation/expansion of disposal capacity

$\delta_{c}^{Storage}$ = Increments for installation/expansion of storage capacity

$\delta_{j}^{Treatment}$ = Increments for installation/expansion of treatment capacity

δ^{Truck} = Truck capacity

$\tau_{k}^{Disposal}$ = Disposal construction or expansion lead time

$\tau_{s}^{Storage}$ = Storage construction or expansion lead time

$\tau_{l,\tilde{l}}^{Pipeline}$ = Pipeline construction or expansion lead time

$\tau_{l,\tilde{l}}^{Trucking}$ = Drive time between two locations

$\lambda_{s}^{Storage}$ = Initial storage level at storage site

$\lambda_{p}^{PadStorage}$ = Initial storage level at completions site

$\theta_{s}^{Storage}$ = Terminal storage level at storage site

$\theta_{p}^{PadStorage}$ = Terminal storage level at completions site

$\kappa_{k,i}^{Disposal}$ = Disposal construction or expansion capital cost for selected capacity increment

$\kappa_{s,c}^{Storage}$ = Storage construction or expansion capital cost for selected capacity increment

$\kappa_{r,j}^{Treatment}$ = Treatment construction or expansion capital cost for selected capacity increment

The cost parameter for expanding or constructing new pipeline capacity is structured differently depending on model configuration settings. If the pipeline cost configuration is distance based:

$\kappa^{Pipeline}$ = Pipeline construction or expansion capital cost [currency/(diameter-distance)]

$\mu_{d}^{Pipeline}$ = Pipeline diameter installation or expansion increments [diameter]

$\lambda_{l,\tilde{l}}^{Pipeline}$ = Pipeline segment length [distance]

Otherwise, if the pipeline cost configuration is capacity based:

$\kappa_{l,\tilde{l},d}^{Pipeline}$ = Pipeline construction or expansion capital cost for selected diameter capacity [currency/(volume/time)]

$\delta_d^{Pipeline}$ = Increments for installation/expansion of pipeline capacity [volume/time]

$\pi_k^{Disposal}$ = Disposal operational cost

$\pi_r^{Treatment}$ = Treatment operational cost (may include "clean brine")

$\pi_p^{CompletionReuse}$ = Completions reuse operational cost

$\pi_s^{Storage}$ = Storage deposit operational cost

$\rho_s^{Storage}$ = Storage withdrawal operational credit

$\pi_{l,\tilde{l}}^{Pipeline}$ = Pipeline operational cost

$\pi_l^{Trucking}$ = Trucking hourly cost (by source)

$\pi_f^{Sourcing}$ = Fresh sourcing cost

M^{Flow} = Big-M flow parameter

$\psi^{FracDemand}$ = Slack cost parameter

$\psi^{Production}$ = Slack cost parameter

$\psi^{Flowback}$ = Slack cost parameter

$\psi^{PipelineCapacity}$ = Slack cost parameter

$\psi^{StorageCapacity}$ = Slack cost parameter

$\psi^{DisposalCapacity}$ = Slack cost parameter

$\psi^{TreamentCapacity}$ = Slack cost parameter

$\psi^{BeneficialReuseCapacity}$ = Slack cost parameter

Strategic Model Mathematical Program Formulation

Objectives

Two objective functions can be considered for the optimization of a produced water system: first, the minimization of costs, which includes operational costs associated with procurement of fresh water, the cost of disposal, trucking and piping produced water between well pads and treatment facilities, and the cost of storing, treating and reusing produced water. Capital costs are also considered due to infrastructure build out such as the installation of pipelines, treatment, and storage facilities. A credit for (re)using treated water is also considered, and additional slack variables are included to facilitate the identification of potential issues with input data. The second objective is the maximization of water reused which is defined as the ratio between the treated produced water that is used in completions

operations and the total produced water coming to surface.

$$\min\ C^{TotalSourced} + C^{TotalDisposal} + C^{TotalTreatment}$$
$$+ C^{TotalCompletionsReuse} + C^{TotalPiping} + C^{TotalStorage}$$
$$+ C^{TotalTrucking} + \alpha^{AnnualizationRate} \cdot (C^{DisposalCapEx}$$
$$+ C^{StorageCapEx} + C^{TreatmentCapEx} + C^{PipelineCapEx})$$
$$+ C^{Slack} - R^{TotalStorage}$$

$$\max\ F^{TotalCompletionsReuse} / \beta^{TotalProd}$$

Annualization Rate Calculation:

The annualization rate is calculated using the formula described at this website: https://www.investopedia.com/terms/e/eac.asp. The annualization rate takes the discount rate (rate) and the number of years the CAPEX investment is expected to be used (life) as input.

$$\alpha^{AnnualizationRate} = \frac{rate}{(1 - (1 + rate)^{-life})}$$

Completions Pad Demand Balance: $\forall p \in CP, t \in T$

Completions pad demand can be met by trucked or piped water moved into the pad in addition to water in completions pad storage. For each completions pad and for each time period, completions demand at the given pad is equal to the sum of all piped and trucked water moved into the completions pad plus water removed from the pad storage minus water put into the pad storage plus a slack.

$$\gamma_{p,t}^{Completions} = \sum_{n \in N|(n,p) \in NCA} F_{n,p,t}^{Piped} + \sum_{\tilde{p} \in PP|(\tilde{p},p) \in PCA} F_{\tilde{p},p,t}^{Piped} + \sum_{s \in S|(s,p) \in SCA} F_{s,p,t}^{Piped}$$
$$+ \sum_{\tilde{p} \in CP|(\tilde{p},p) \in CCA} F_{\tilde{p},p,t}^{Piped} + \sum_{r \in R|(r,p) \in RCA} F_{r,p,t}^{Piped} + \sum_{f \in F|(f,p) \in FCA} F_{f,p,t}^{Sourced}$$
$$+ \sum_{\tilde{p} \in PP|(\tilde{p},p) \in PCT} F_{\tilde{p},p,t}^{Trucked} + \sum_{\tilde{p} \in CP|(\tilde{p},p) \in CCT} F_{\tilde{p},p,t}^{Trucked} + \sum_{s \in S|(s,p) \in SCT} F_{s,p,t}^{Trucked}$$
$$+ \sum_{f \in F|(f,p) \in FCT} F_{f,p,t}^{Trucked} + F_{p,t}^{PadStorageOut} - F_{p,t}^{PadStorageIn} + S_{p,t}^{FracDemand}$$

Completions Pad Storage Balance: $\forall p \in CP, t \in T$

Sets the storage level at the completions pad. For each completions pad and for each time period, completions pad storage is equal to storage in last time period plus water put in minus water removed. If it is the first time period, the pad storage is the initial pad storage.

For $t = 1$:

$$L_{p,t}^{PadStorage} = \lambda_{p,t=1}^{PadStorage} + F_{p,t}^{PadStorageIn} - F_{p,t}^{PadStorageOut}$$

For $t > 1$:

$$L_{p,t}^{PadStorage} = L_{p,t-1}^{PadStorage} + F_{p,t}^{PadStorageIn} - F_{p,t}^{PadStorageOut}$$

Completions Pad Storage Capacity: $\forall p \in CP, t \in T$

The storage at each completions pad must always be at or below its capacity in every time period.

$$L_{p,t}^{PadStorage} \leq \sigma_p^{PadStorage}$$

Terminal Completions Pad Storage Level: $\forall p \in CP$

The storage in the last period must be at or below its terminal storage level.

$$L_{p,t=T}^{PadStorage} \leq \theta_p^{PadStorage}$$

The storage in the last period must be at or below its terminal storage level.

Freshwater Sourcing Capacity: $\forall f \in F, t \in T$

For each freshwater source and each time period, the outgoing water from the freshwater source is below the freshwater capacity.

$$\sum_{p \in P|(f,p) \in FCA} F_{f,p,t}^{Sourced} + \sum_{p \in P|(f,p) \in FCT} F_{f,p,t}^{Trucked} \leq \sigma_{f,t}^{Freshwater}$$

Completions Pad Truck Offloading Capacity: $\forall p \in CP, t \in T$

For each completions pad and time period, the volume of water being trucked into the completions pad must be below the trucking offloading capacity.

$$\sum_{\tilde{p} \in P|(\tilde{p},p) \in PCT} F_{\tilde{p},p,t}^{Trucked} + \sum_{s \in S|(s,p) \in SCT} F_{s,p,t}^{Trucked} + \sum_{f \in F|(f,p) \in FCT} F_{f,p,t}^{Trucked}$$
$$+ \sum_{\tilde{p} \in P|(\tilde{p},p) \in CCT} F_{\tilde{p},p,t}^{Trucked} \leq \sigma_p^{Offloading,Pad}$$

Completions Pad Processing Capacity: $\forall p \in CP, t \in T$

For each completions pad and time period, the volume of water (excluding freshwater) coming in must be below the processing limit.

$$\sum_{n \in N|(n,p) \in NCA} F_{n,p,t}^{Piped} + \sum_{\tilde{p} \in P|(\tilde{p},p) \in PCA} F_{\tilde{p},p,t}^{Piped} + \sum_{s \in S|(s,p) \in SCA} F_{s,p,t}^{Piped}$$
$$+ \sum_{\tilde{p} \in P|(\tilde{p},p) \in CCA} F_{\tilde{p},p,t}^{Piped} + \sum_{r \in R|(r,p) \in RCA} F_{r,p,t}^{Piped} + \sum_{\tilde{p} \in P|(\tilde{p},p) \in PCT} F_{\tilde{p},p,t}^{Trucked}$$
$$+ \sum_{s \in S|(s,p) \in SCT} F_{s,p,t}^{Trucked} + \sum_{\tilde{p} \in P|(\tilde{p},p) \in CCT} F_{\tilde{p},p,t}^{Trucked} \leq \sigma_p^{Processing,Pad}$$

Note: This constraint has not actually been implemented yet.

Storage Site Truck Offloading Capacity: $\forall s \in S, t \in T$

For each storage site and each time period, the volume of water being trucked into the storage site must be below the trucking offloading capacity for that storage site.

$$\sum_{p \in P|(p,s) \in PST} F_{p,s,t}^{Trucked} + \sum_{p \in P|(p,s) \in CST} F_{p,s,t}^{Trucked} \leq \sigma_s^{Offloading,Storage}$$

Storage Site Processing Capacity: $\forall s \in S, t \in T$

For each storage site and each time period, the volume of water being piped and trucked into the storage site must be less than the processing capacity for that storage site.

$$\sum_{n \in N|(n,s) \in NSA} F_{n,s,t}^{Piped} + \sum_{r \in R|(r,s) \in RSA} F_{r,s,t}^{Piped} + \sum_{p \in P|(p,s) \in PST} F_{p,s,t}^{Trucked}$$
$$+ \sum_{p \in P|(p,s) \in CST} F_{p,s,t}^{Trucked} \leq \sigma_s^{Processing,Storage}$$

Production Pad Supply Balance: $\forall p \in PP, t \in T$

All produced water must be accounted for. For each production pad and for each time period, the volume of outgoing water must be equal to the forecasted produced water for the production pad.

$$\beta_{p,t}^{Production} = \sum_{n \in N|(p,n) \in PNA} F_{p,n,t}^{Piped} + \sum_{\tilde{p} \in P|(p,\tilde{p}) \in PCA} F_{p,\tilde{p},t}^{Piped} + \sum_{\tilde{p} \in P|(p,\tilde{p}) \in PPA} F_{p,\tilde{p},t}^{Piped}$$

$$+ \sum_{\tilde{p} \in P|(p,\tilde{p}) \in PCT} F_{p,\tilde{p},t}^{Trucked} + \sum_{k \in K|(p,k) \in PKT} F_{p,k,t}^{Trucked} + \sum_{s \in S|(p,s) \in PST} F_{p,s,t}^{Trucked}$$

$$+ \sum_{r \in R|(p,r) \in PRT} F_{p,r,t}^{Trucked} + \sum_{o \in O|(p,o) \in POT} F_{p,o,t}^{Trucked} + S_{p,t}^{Production}$$

Completions Pad Supply Balance (i.e. Flowback Balance): $\forall p \in CP, t \in T$

All flowback water must be accounted for. For each completions pad and for each time period, the volume of outgoing water must be equal to the forecasted flowback produced water for the completions pad.

$$\beta_{P,t}^{Flowback} = \sum_{n \in N|(p,n) \in CNA} F_{p,n,t}^{Piped} + \sum_{c \in C|(p,c) \in CCA} F_{p,c,t}^{Piped} + \sum_{\tilde{p} \in P|(p,\tilde{p}) \in CCT} F_{p,\tilde{p},t}^{Trucked}$$

$$+ \sum_{k \in K|(p,k) \in CKT} F_{p,k,t}^{Trucked} + \sum_{s \in S|(p,s) \in CST} F_{p,s,t}^{Trucked} + \sum_{r \in R|(p,r) \in CRT} F_{p,r,t}^{Trucked} + S_{p,t}^{Flowback}$$

Network Node Balance: $\forall n \in N, t \in T$

Flow balance constraint (i.e., inputs are equal to outputs). For each pipeline node and for each time period, the volume water into the node is equal to the volume of water out of the node.

$$\sum_{p \in P|(p,n) \in PNA} F_{p,n,t}^{Piped} + \sum_{p \in P|(p,n) \in CNA} F_{p,n,t}^{Piped} + \sum_{\tilde{n} \in N|(\tilde{n},n) \in NNA} F_{\tilde{n},n,t}^{Piped}$$

$$+ \sum_{s \in S|(s,n) \in SNA} F_{s,n,t}^{Piped} + \sum_{r \in R|(r,n) \in RNA} F_{r,n,t}^{Piped}$$

$$= \sum_{\tilde{n} \in N|(n,\tilde{n}) \in NNA} F_{n,\tilde{n},t}^{Piped} + \sum_{p \in P|(n,p) \in NCA} F_{n,p,t}^{Piped} + \sum_{k \in K|(n,k) \in NKA} F_{n,k,t}^{Piped}$$

$$+ \sum_{r \in R|(n,r) \in NRA} F_{n,r,t}^{Piped} + \sum_{s \in S|(n,s) \in NSA} F_{n,s,t}^{Piped} + \sum_{o \in O|(n,o) \in NOA} F_{n,o,t}^{Piped}$$

Bi-Directional Flow: $\forall (l,\tilde{l}) \in \{PCA, PNA, PPA, CNA, NNA, NCA, NKA, NSA, NRA, \dots, SOA\}, t \in T$

There can only be flow in one direction for a given pipeline arc in a given time period. Flow is only allowed in a given direction if the binary indicator for that direction is "on".

$$y_{l,\tilde{l},t}^{Flow} + y_{l,l,t}^{Flow} = 1$$

Note: Technically this constraint should only be enforced for truly reversible arcs (e.g. NCA and CNA); and even then it only needs to be defined per one reversible arc (e.g. NCA only and not NCA and CNA).

$$F_{l,\tilde{l},t}^{Piped} \leq y_{l,\tilde{l},t}^{Flow} \cdot M^{Flow}$$

Storage Site Balance: $\forall s \in S, t \in T$

For each storage site and for each time period, if it is the first time period, the storage level is the initial storage. Otherwise, the storage level is equal to the storage level in the previous time period plus water inputs minus water outputs.

For $t = 1$:

$$L_{s,t}^{Storage} = \lambda_{s,t=1}^{Storage} + \sum_{n \in N|(n,s) \in NSA} F_{n,s,t}^{Piped} + \sum_{r \in R|(r,s) \in RSA} F_{r,s,t}^{Piped} + \sum_{p \in P|(p,s) \in PST} F_{p,s,t}^{Trucked}$$

$$+ \sum_{p \in P|(p,s) \in CST} F_{p,s,t}^{Trucked} - \sum_{n \in N|(s,n) \in SNA} F_{s,n,t}^{Piped} - \sum_{p \in P|(s,p) \in SCA} F_{s,p,t}^{Piped} - \sum_{k \in K|(s,k) \in SKA} F_{s,k,t}^{Piped}$$

$$- \sum_{r \in R|(s,r) \in SRA} F_{s,r,t}^{Piped} - \sum_{o \in O|(s,o) \in SOA} F_{s,o,t}^{Piped} - \sum_{p \in P|(s,p) \in SCT} F_{s,p,t}^{Trucked} - \sum_{k \in K|(s,k) \in SKT} F_{s,k,t}^{Trucked}$$

For $t > 1$:

$$L_{s,t}^{Storage} + L_{s,t-1}^{Storage} + \sum_{n \in N|(n,s) \in NSA} F_{n,s,t}^{Piped} + \sum_{r \in R|(r,s) \in RSA} F_{r,s,t}^{Piped} + \sum_{p \in P|(p,s) \in PST} F_{p,s,t}^{Trucked}$$

$$+ \sum_{p \in P|(p,s) \in CST} F_{p,s,t}^{Trucked} - \sum_{n \in N|(s,n) \in SNA} F_{s,n,t}^{Piped} - \sum_{p \in P|(s,p) \in SCA} F_{s,p,t}^{Piped} - \sum_{k \in K|(s,k) \in SKA} F_{s,k,t}^{Piped}$$

$$- \sum_{r \in R|(s,r) \in SRA} F_{s,r,t}^{Piped} - \sum_{o \in O|(s,o) \in SOA} F_{s,o,t}^{Piped} - \sum_{p \in P|(s,p) \in SCT} F_{s,p,t}^{Trucked} - \sum_{k \in K|(s,k) \in SKT} F_{s,k,t}^{Trucked}$$

Terminal Storage Level: $\forall s \in S, t \in T$

For each storage site, the storage in the last time period must be less than or equal to the predicted/set terminal storage level.

$$L_{s,t=T}^{Storage} \leq \theta_s^{Storage}$$

Network Node Capacity: $\forall n \in N, t \in T$

Flow capacity constraint. For each pipeline node and for each time period, the volume should not exceed the node capacity.

$$\sum_{p \in P|(p,n) \in PNA} F_{p,n,t}^{Piped} + \sum_{p \in P|(p,n) \in CNA} F_{p,n,t}^{Piped} + \sum_{\tilde{n} \in N|(\tilde{n},n) \in NNA} F_{\tilde{n},n,t}^{Piped}$$

$$+ \sum_{s \in S|(s,n) \in SNA} F_{s,n,t}^{Piped} + \sum_{r \in R|(r,n) \in RNA} F_{r,n,t}^{Piped} \leq \sigma_n^{Node}$$

Pipeline Capacity Construction/Expansion: $\forall (l,\tilde{l}) \in \{PCA, PNA, PPA, NKA, CNA, NCA, NSA, NOA, FCA, RCA, SKA, SOA, RSA,$

Sets the flow capacity in a given pipeline during a given time period. Different constraints apply depending on if the pipeline is reversible or not.

$$F_{l,\tilde{l},[t]}^{Capacity} = \sigma_{l,\tilde{l}}^{Pipeline} + \sum_{d \in D} \delta_d^{Pipeline} \cdot y_{l,\tilde{l},d}^{Pipeline} + S_{l,\tilde{l}}^{PipelineCapacity}$$

$\forall (l,\tilde{l}) \in \{PPA, CNA, NNA, NCA, NSA, NRA, RNA, RKA, SNA, SCA\}, [t \in T]$

$$F_{l,\tilde{l},[t]}^{Capacity} = \sigma_{l,\tilde{l}}^{Pipeline} + \sum_{d \in D} \delta_d^{Pipeline} \cdot (y_{l,\tilde{l},d}^{Pipeline} + y_{\tilde{l},l,d}^{Pipeline}) + S_{l,\tilde{l}}^{PipelineCapacity}$$

Note: $\delta_d^{Pipeline}$ can be input by user or calculated. If the user chooses to calculate pipeline capacity, the parameter will be calculated by the equation below where $\kappa_{l,\tilde{l}}$ is Hazen-Williams constant and ω is Hazen-Williams exponent as per Cafaro & Grossmann (2021) and d represents the pipeline diameter as per the set $d \in D$.

See equation:

$$\delta_d^{Pipeline} = \kappa_{l,\tilde{l}} \cdot d^{\omega}$$

$\forall (l, \tilde{l}) \in \{PCA, PNA, PPA, CNA, RCANNA, NCA, NKA, NSA, NRA, \dots, SOA\}, t \in T$

$$F_{l,\tilde{l},t}^{Piped} \leq F_{l,\tilde{l},[t]}^{Capacity}$$

Storage Capacity Construction/Expansion: $\forall s \in S, [t \in T]$

This constraint accounts for the expansion of available storage capacity or installation of storage facilities. If expansion/construction is selected, expand the capacity by the set expansion amount. The water level at the storage site must be less than this capacity. As of now, the model considers that a storage facility is expanded or built at the beginning of the planning horizon. The C_0 notation indicates that we also include the 0th case, meaning that there is no selection in the set C where no capacity is added.

$$X_{s,[t]}^{Capacity} = \sigma_s^{Storage} + \sum_{c \in C_0} \delta_c^{Storage} \cdot y_{s,c}^{Storage} + S_s^{StorageCapacity}$$

$\forall s \in S, t \in T$

$$L_{s,t}^{Storage} \leq X_{s,[t]}^{Capacity}$$

Disposal Capacity Construction/Expansion: $\forall k \in K, [t \in T]$

This constraint accounts for the expansion of available disposal sites or installation of new disposal sites. If expansion/construction is selected, expand the capacity by the set expansion amount. The total disposed water in a given time period must be less than this new capacity.

$$D_{k,[t]}^{Capacity} = \sigma_k^{Disposal} + \sum_{i \in I_0} \delta_i^{Disposal} \cdot y_{k,i}^{Disposal} + S_k^{DisposalCapacity}$$

$\forall k \in K, t \in T$

$$\sum_{n \in N|(n,k) \in NKA} F_{n,k,t}^{Piped} + \sum_{s \in S|(s,k) \in SKA} F_{s,k,t}^{Piped} + \sum_{r \in R|(r,k) \in RKA} F_{r,k,t}^{Trucked} + \sum_{s \in S|(s,k) \in SKT} F_{s,k,t}^{Trucked}$$
$$+ \sum_{p \in P|(p,k) \in PKT} F_{p,k,t}^{Trucked} + \sum_{p \in P|(p,k) \in CKT} F_{p,k,t}^{Trucked} + \sum_{r \in R|(r,k) \in RKT} F_{r,k,t}^{Trucked} \leq D_{k,[t]}^{Capacity}$$

Treatment Capacity Construction/Expansion: $\forall r \in R, [t \in T]$

Similarly to disposal and storage capacity construction/expansion constraints, the current treatment capacity can be expanded as required or new facilities may be installed.

$$T_{r,[t]}^{Capacity} = \sigma_r^{Treatment} + \sum_{j \in J_0} \delta_j^{Treatment} \cdot y_{r,j}^{Treatment} + S_r^{TreatmentCapacity}$$

$\forall r \in R, t \in T$

$$\sum_{n \in N|(n,r) \in NRA} F_{n,r,t}^{Piped} + \sum_{s \in S|(s,r) \in SRA} F_{s,r,t}^{Piped} + \sum_{p \in P|(p,r) \in PRT} F_{p,r,t}^{Trucked}$$
$$+ \sum_{p \in P|(p,r) \in CRT} F_{p,r,t}^{Trucked} \leq T_{r,[t]}^{Capacity}$$

Treatment Balance: $\forall r \in R, t \in T$

Water input into treatment facility is treated with a level of efficiency, meaning only a given percentage of the water input is outputted to be reused at the completions pads.

$$\epsilon_{r,W_r^{TreatmentComponent}}^{Treatment} \cdot \left(\sum_{n \in N | (n,r) \in NRA} F_{n,r,t}^{Piped} + \sum_{s \in S | (s,r) \in SRA} F_{s,r,t}^{Piped} + \sum_{p \in P | (p,r) \in PRT} F_{p,r,t}^{Trucked} \right.$$
$$\left. + \sum_{p \in P | (p,r) \in CRT} F_{p,r,t}^{Trucked} \right) = \sum_{p \in P | (r,p) \in RCA} F_{r,p,t}^{Piped} + F_{r,t}^{UnusedTreatedWater}$$

where $\epsilon_{r,w}^{Treatment} \leq 1$.

Beneficial Reuse Capacity: $\forall o \in O, t \in T$

For each beneficial reuse site and for each time period, water sent to a site must be less than or equal to the capacity.

$$\sum_{n \in N | (n,o) \in NOA} F_{n,o,t}^{Piped} + \sum_{s \in S | (s,o) \in SOA} F_{s,o,t}^{Piped} + \sum_{p \in P | (p,o) \in POT} F_{p,o,t}^{Trucked}$$
$$\leq \sigma_o^{BeneficialReuse} + S_o^{BeneficialReuseCapacity}$$

Fresh Sourcing Cost: $\forall f \in F, p \in CP, t \in T$

For each freshwater source, for each completions pad, and for each time period, the freshwater sourcing cost is equal to all output from the freshwater source times the freshwater sourcing cost.

$$C_{f,p,t}^{Sourced} = (F_{f,p,t}^{Sourced} + F_{f,p,t}^{Trucked}) \cdot \pi_f^{Sourcing}$$
$$C^{TotalSourced} = \sum_{t \in T} \sum_{(f,p) \in FCA} C_{f,p,t}^{Sourced}$$

Total Fresh Sourced Volume: $\forall f \in F, p \in CP, t \in T$

The total fresh sourced volume is the sum of freshwater movements by truck and pipeline over all time periods, completions pads, and freshwater sources.

$$F^{TotalSourced} = \sum_{t \in T} \sum_{f \in F} \sum_{p \in CP} (F_{f,p,t}^{Sourced} + F_{f,p,t}^{Trucked})$$

Disposal Cost: $\forall k \in K, t \in T$

For each disposal site, for each time period, the disposal cost is equal to all water moved into the disposal site multiplied by the operational disposal cost. Total disposal cost is the sum of disposal costs over all time periods and all disposal sites.

$$C_{k,t}^{Disposal} = \left(\sum_{l \in L | (l,k) \in \{NKA,RKA,SKA\}} F_{l,k,t}^{Piped} + \sum_{l \in L | (l,k) \in \{PKT,CKT,SKT,RKT\}} F_{l,k,t}^{Trucked} \right) \cdot \pi_k^{Disposal}$$
$$C^{TotalDisposal} = \sum_{t \in T} \sum_{k \in K} C_{k,t}^{Disposal}$$

Total Disposed Volume:

Total disposed volume over all time is the sum of all piped and trucked water to disposal summed over all time periods.

$$F^{TotalDisposed} = \sum_{t \in T} \left(\sum_{(l,k) \in \{NKA,RKA,SKA\}} F_{l,k,t}^{Piped} + \sum_{(l,k) \in \{PKT,CKT,SKT,RKT\}} F_{l,k,t}^{Trucked} \right)$$

Treatment Cost: $\forall r \in R, t \in T$

For each treatment site, for each time period, the treatment cost is equal to all water moved to the treatment site multiplied by the operational treatment cost. The total treatments cost is the sum of treatment costs over all time periods and all treatment sites.

$$C_{r,t}^{Treatment} = (\sum_{l \in L|(l,r) \in \{NRA,SRA\}} F_{l,r,t}^{Piped} + \sum_{l \in L|(l,r) \in \{PRT,CRT\}} F_{l,r,t}^{Trucked}) \cdot \pi_r^{Treatment}$$

$$C^{TotalTreatment} = \sum_{t \in T} \sum_{r \in R} C_{r,t}^{Treatment}$$

Completions Reuse Cost: $\forall p \in P, t \in T$

Completions reuse water is all water that meets completions pad demand, excluding freshwater. Completions reuse cost is the volume of completions reused water multiplied by the cost for reuse.

$$C_{p,t}^{CompletionsReuse} = (\sum_{n \in N|(n,p) \in NCA} F_{n,p,t}^{Piped} + \sum_{\tilde{p} \in P|(\tilde{p},p) \in PCA} F_{\tilde{p},p,t}^{Piped} + \sum_{r \in R|(r,p) \in RCA} F_{r,p,t}^{Piped}$$

$$+ \sum_{s \in S|(s,p) \in SCA} F_{s,p,t}^{Piped} + \sum_{\tilde{p} \in P|(\tilde{p},p) \in CCA} F_{\tilde{p},p,t}^{Piped} + \sum_{\tilde{p} \in P|(\tilde{p},p) \in CCT} F_{\tilde{p},p,t}^{Trucked}$$

$$+ \sum_{\tilde{p} \in P|(\tilde{p},p) \in PCT} F_{\tilde{p},p,t}^{Trucked} + \sum_{s \in S|(s,p) \in SCT} F_{s,p,t}^{Trucked}) \cdot \pi_p^{CompletionsReuse}$$

Note: Freshwater sourcing is excluded from completions reuse costs.

$$C^{TotalReuse} = \sum_{t \in T} \sum_{p \in CP} C_{p,t}^{Reuse}$$

Total Completions Reuse Volume:

The total reuse volume is the total volume of produced water reused, or the total water meeting completions pad demand over all time periods, excluding freshwater.

$$F^{TotalCompletionsReused} = \sum_{t \in T} (\sum_{(n,p) \in NCA} F_{n,p,t}^{Piped} + \sum_{(p,\tilde{p}) \in PCA} F_{p,\tilde{p},t}^{Piped}$$

$$+ \sum_{(s,p) \in SCA} F_{s,p,t}^{Piped} + \sum_{(r,p) \in RCA} F_{r,p,t}^{Piped} + \sum_{(p,\tilde{p}) \in PCT} F_{p,\tilde{p},t}^{Trucked}$$

$$+ \sum_{(p,\tilde{p}) \in CCA} F_{p,\tilde{p},t}^{Piped} + \sum_{(p,\tilde{p}) \in CCT} F_{p,\tilde{p},t}^{Trucked} + \sum_{(s,p) \in SCT} F_{s,p,t}^{Trucked})$$

Piping Cost: $\forall (l,\tilde{l}) \in \{PPA, \dots, CCA\}, t \in T$

Piping cost is the total volume of piped water multiplied by the cost for piping.

$$C_{l,\tilde{l},t}^{Piped} = (F_{l,\tilde{l},t}^{Piped} + F_{l,\tilde{l},t}^{Sourced}) \cdot \pi_{l,\tilde{l}}^{Pipeline}$$

$$C^{TotalPiping} = \sum_{t \in T} \sum_{(l,\tilde{l}) \in \{PPA,\dots\}} C_{l,\tilde{l},t}^{Piped}$$

Note: The constraints above explicitly consider freshwater piping via *FCA* arcs.

Storage Deposit Cost: $\forall s \in S, t \in T$

Cost of depositing into storage is equal to the total volume of water moved into storage multiplied by the storage operation cost rate.

$$C_{s,t}^{Storage} = (\sum_{n \in N | (n,s) \in NSA} F_{n,s,t}^{Piped} + \sum_{r \in R | (r,s) \in RSA} F_{r,s,t}^{Piped}$$

$$+ \sum_{p \in P | (p,s) \in CST} F_{p,s,t}^{Trucked} + \sum_{p \in P | (p,s) \in PST} F_{p,s,t}^{Trucked}) \cdot \pi_s^{Storage}$$

$$C^{TotalStorage} = \sum_{t \in T} \sum_{s \in S} C_{s,t}^{Storage}$$

Storage Withdrawal Credit: $\forall s \in S, t \in T$

Credits from withdrawing from storage is equal to the total volume of water moved out from storage multiplied by the storage operation credit rate.

$$R_{s,t}^{Storage} = (\sum_{l \in L | (s,l) \in \{SNA,SCA,SKA,SRA,SOA\}} F_{s,l,t}^{Piped} + \sum_{l \in L | (s,l) \in \{SCT,SKT\}} F_{s,l,t}^{Trucked}) \cdot \rho_s^{Storage}$$

$$R^{TotalStorage} = \sum_{t \in T} \sum_{s \in S} R_{s,t}^{Storage}$$

Trucking Cost (Simplified)

Trucking cost between two locations for time period is equal to the trucking volume between locations in time t divided by the truck capacity [this gets # of truckloads] multiplied by the lead time between two locations and hourly trucking cost.

$$C_{l,\tilde{l},t}^{Trucked} = F_{l,\tilde{l},t}^{Trucked} \cdot 1/\delta^{Truck} \cdot \tau_{l,\tilde{l}}^{Trucking} \cdot \pi_l^{Trucking}$$

$$C^{TotalTrucking} = \sum_{t \in T} \sum_{(l,\tilde{l}) \in \{PCT,...,RKT\}} C_{l,\tilde{l},t}^{Trucked}$$

Note: The constraints above explicitly consider freshwater trucking via *FCT* arcs.

Total Trucking Volume: $\forall t \in T$

The total trucking volume is estimated as the summation of trucking movements over all time periods and locations.

$$F^{TotalTrucking} = \sum_{t \in T} \sum_{(l,\tilde{l}) \in \{PCT,...,RKT\}} F_{l,\tilde{l},t}^{Trucked}$$

Disposal Construction or Capacity Expansion Cost: $\forall t \in T$

Cost related to expanding or constructing new disposal capacity. Takes into consideration capacity increment, cost for selected capacity increment, and if the construction/expansion is selected to occur.

$$C_{[t]}^{DisposalCapEx} = \sum_{i \in I_0} \sum_{k \in K} \kappa_{k,i}^{Disposal} \cdot \delta_i^{Disposal} \cdot y_{k,i}^{Disposal}$$

Storage Construction or Capacity Expansion Cost: $\forall t \in T$

Cost related to expanding or constructing new storage capacity. Takes into consideration capacity increment, cost for selected capacity increment, and if the construction/expansion is selected to occur.

$$C_{[t]}^{StorageCapEx} = \sum_{s \in S} \sum_{c \in C_0} \kappa_{s,c}^{Storage} \cdot \delta_c^{Storage} \cdot y_{s,c}^{Storage}$$

Treatment Construction or Capacity Expansion Cost: $\forall t \in T$

Cost related to expanding or constructing new treatment capacity. Takes into consideration capacity increment, cost for selected capacity increment, and if the construction/expansion is selected to occur.

$$C_{[t]}^{TreatmentCapEx} = \sum_{r \in R} \sum_{j \in J_0} \kappa_{r,j}^{Treatment} \cdot \delta_j^{Treatment} \cdot y_{r,j}^{Treatment}$$

Pipeline Construction or Capacity Expansion Cost: $\forall t \in T$

Cost related to expanding or constructing new pipeline capacity is calculated differently depending on model configuration settings.

If the pipeline cost configuration is **capacity based**, pipeline expansion cost is calculated using capacity increments, cost for selected capacity increment, and if the construction/expansion is selected to occur.

$$C_{[t]}^{PipelineCapEx} = \sum_{l \in L} \sum_{\tilde{l} \in L} \sum_{d \in D_0} \kappa_{l,\tilde{l},d}^{Pipeline} \cdot \delta_d^{Pipeline} \cdot y_{l,\tilde{l},d}^{Pipeline}$$

If the pipeline cost configuration is **distance based**, pipeline expansion cost is calculated using pipeline distances, pipeline diameters, cost per inch mile, and if the construction/expansion is selected to occur.

$$C_{[t]}^{PipelineCapEx} = \sum_{l \in L} \sum_{\tilde{l} \in L} \sum_{d \in D_0} \kappa^{Pipeline} \cdot \mu_d^{Pipeline} \cdot \lambda_{l,\tilde{l}}^{Pipeline} \cdot y_{l,\tilde{l},d}^{Pipeline}$$

Slack Costs:

Weighted sum of the slack variables. In the case that the model is infeasible, these slack variables are used to determine where the infeasibility occurs (e.g. pipeline capacity is not sufficient).

$$
\begin{aligned}
C^{Slack} = & \sum_{p \in CP} \sum_{t \in T} S_{p,t}^{FracDemand} \cdot \psi^{FracDemand} + \sum_{p \in PP} \sum_{t \in T} S_{p,t}^{Production} \cdot \psi^{Production} \\
& + \sum_{p \in CP} \sum_{t \in T} S_{p,t}^{Flowback} \cdot \psi^{Flowback} + \sum_{(l,\tilde{l}) \in \dots} S_{l,\tilde{l}}^{PipelineCapacity} \cdot \psi^{PipeCapacity} \\
& + \sum_{s \in S} S_s^{StorageCapacity} \cdot \psi^{StorageCapacity} + \sum_{k \in K} S_k^{DisposalCapacity} \cdot \psi^{DisposalCapacity} \\
& + \sum_{r \in R} S_r^{TreatmentCapacity} \cdot \psi^{TreatmentCapacity} + \sum_{o \in O} S_o^{BeneficialReuseCapacity} \cdot \psi^{BeneficialReuseCapacity}
\end{aligned}
$$

Logic Constraints:

New pipeline or facility capacity constraints: e.g., only one injection capacity can be used for a given site

$\forall k \in K$

$$\sum_{i \in I_0} y_{k,i,[t]}^{Disposal} = 1$$

$\forall s \in S$

$$\sum_{c \in C_0} y_{s,c,[t]}^{Storage} = 1$$

$\forall r \in R$

$$\sum_{j \in J_0} y_{r,j,[t]}^{Treatment} = 1$$

$$\sum_{d \in D_0} y^{Pipeline}_{l,\tilde{l},d,[t]} = 1$$

$$\forall l \in L, \tilde{l} \in L$$

Deliveries Destination Constraints:

Completions reuse deliveries at a completions pad in time period t is equal to all piped and trucked water moved into the completions pad, excluding freshwater. $\forall p \in CP, t \in T$

$$F^{CompletionsReuseDestination}_{p,t} = \sum_{l \in L|(l,p) \in \{PCA,CCA,NCA,RCA,SCA,PCT,CCT,SCT\}} F^{Piped}_{l,p,t} + F^{Trucked}_{l,p,t}$$

Disposal deliveries for disposal site k at time t is equal to all piped and trucked water moved to the disposal site k. $\forall k \in K, t \in T$

$$F^{DisposalDestination}_{k,t} = \sum_{l \in L|(l,k) \in \{NKA,RKA,SKA,PKT,CKT,SKT,RKT\}} F^{Piped}_{l,k,t} + F^{Trucked}_{l,k,t}$$

Beneficial reuse deliveries for beneficial reuse site o at time t is equal to all piped and trucked water moved to the beneficial reuse site o. $\forall o \in O, t \in T$

$$F^{BeneficialReuseDestination}_{o,t} = \sum_{l \in L|(l,o) \in \{NOA,SOA,POT\}} F^{Piped}_{l,o,t} + F^{Trucked}_{l,o,t}$$

Completions deliveries destination for completions pad p at time t is equal to all piped and trucked water moved to the completions pad. $\forall p \in CP, t \in T$

$$\begin{aligned}
F^{CompletionsDestination}_{p,t} = &\sum_{n \in N|(n,p) \in NCA} F^{Piped}_{n,p,t} + \sum_{\tilde{p} \in P|(\tilde{p},p) \in PCA} F^{Piped}_{\tilde{p},p,t} + \sum_{s \in S|(s,p) \in SCA} F^{Piped}_{s,p,t} \\
&+ \sum_{\tilde{p} \in P|(\tilde{p},p) \in CCA} F^{Piped}_{\tilde{p},p,t} + \sum_{r \in R|(r,p) \in RCA} F^{Piped}_{r,p,t} + \sum_{f \in F|(f,p) \in FCA} F^{Sourced}_{f,p,t} \\
&+ \sum_{\tilde{p} \in P|(\tilde{p},p) \in PCT} F^{Trucked}_{\tilde{p},p,t} + \sum_{s \in S|(s,p) \in SCT} F^{Trucked}_{s,p,t} + \sum_{\tilde{p} \in P|(\tilde{p},p) \in CCT} F^{Trucked}_{\tilde{p},p,t} \\
&+ \sum_{f \in F|(f,p) \in FCT} F^{Trucked}_{f,p,t} + F^{PadStorageOut}_{p,t} - F^{PadStorageIn}_{p,t}
\end{aligned}$$

Strategic Model Water Quality Extension

An extension to this strategic optimization model measures the water quality across all locations over time. As of now, water quality is not a decision variable. It is calculated after optimization of the strategic model. The process for calculating water quality is as follows: the strategic model is first solved to optimality, water quality variables and constraints are added, flow rates and storage levels are fixed to the solved values at optimality, and the water quality is calculated.

Note: Fixed variables are colored purple in the documentation.

Assumptions:

- Water quality of produced water from production pads and completions pads remains the same across all time periods
- When blending flows of different water quality, they blend linearly

---

- Treatment does not affect water quality

Water Quality Sets

$w \in W$ Water Quality Components (e.g., TDS)

$p^{IntermediateNode} \in CP$ Intermediate Completions Pad Nodes

$p^{PadStorage} \in CP$ Pad Storage

Water Quality Parameters

$v_{p,w,[t]}$ = Water quality at well pad

$\xi_{s,w}^{StorageSite}$ = Initial water quality at storage

$\xi_{p,w}^{PadStorage}$ = Initial water quality at pad storage

Water Quality Variables

$Q_{l,w,t}$ = Water quality at location

Disposal Site Water Quality $\forall k \in K, w \in W, t \in T$

The water quality of disposed water is dependent on the flow rates into the disposal site and the quality of each of these flows.

$$
\sum_{n \in N | (n,k) \in NKA} F_{n,k,t}^{Piped} \cdot Q_{n,w,t} + \sum_{s \in S | (s,k) \in SKA} F_{s,k,t}^{Piped} \cdot Q_{s,w,t} + \sum_{r \in R | (r,k) \in RKA} F_{r,k,t}^{Piped} \cdot Q_{r,w,t}
$$
$$
+ \sum_{s \in S | (s,k) \in SKT} F_{s,k,t}^{Trucked} \cdot Q_{s,w,t} + \sum_{p \in P | (p,k) \in PKT} F_{p,k,t}^{Trucked} \cdot v_{p,w,[t]}
$$
$$
+ \sum_{p \in P | (p,k) \in CKT} F_{p,k,t}^{Trucked} \cdot v_{p,w,[t]} + \sum_{r \in R | (r,k) \in RKT} F_{r,k,t}^{Trucked} \cdot Q_{r,w,t}
$$
$$
= F_{k,t}^{DisposalDestination} \cdot Q_{k,w,t}
$$

Storage Site Water Quality $\forall s \in S, w \in W, t \in T$

The water quality at storage sites is dependent on the flow rates into the storage site, the volume of water in storage in the previous time period, and the quality of each of these flows. Even mixing is assumed, so all outgoing flows have the same water quality. If it is the first time period, the initial storage level and initial water quality, respectively, replace the water stored and water quality in the previous time period.

For $t = 1$:

$$
\lambda_{s,t=1}^{Storage} \cdot \xi_{s,w}^{StorageSite} + \sum_{n \in N | (n,s) \in NSA} F_{n,s,t}^{Piped} \cdot Q_{n,w,t}
$$
$$
+ \sum_{r \in R | (r,s) \in RSA} F_{r,s,t}^{Piped} \cdot Q_{r,w,t} + \sum_{p \in P | (p,s) \in PST} F_{p,s,t}^{Trucked} \cdot v_{p,w,[t]} + \sum_{p \in P | (p,s) \in CST} F_{p,s,t}^{Trucked} \cdot v_{p,w,[t]}
$$
$$
= Q_{s,w,t} \cdot (L_{s,t}^{Storage} + \sum_{n \in N | (s,n) \in SNA} F_{s,n,t}^{Piped} + \sum_{p \in P | (s,p) \in SCA} F_{s,p,t}^{Piped} + \sum_{k \in K | (s,k) \in SKA} F_{s,k,t}^{Piped}
$$
$$
+ \sum_{r \in R | (s,r) \in SRA} F_{s,r,t}^{Piped} + \sum_{o \in O | (s,o) \in SOA} F_{s,o,t}^{Piped} + \sum_{p \in P | (s,p) \in SCT} F_{s,p,t}^{Trucked} + \sum_{k \in K | (s,k) \in SKT} F_{s,k,t}^{Trucked})
$$

For $t > 1$:

$$L_{s,t-1}^{Storage} \cdot Q_{s,w,t-1} + \sum_{n \in N|(n,s) \in NSA} F_{n,s,t}^{Piped} \cdot Q_{n,w,t}$$

$$+ \sum_{r \in R|(r,s) \in RSA} F_{r,s,t}^{Piped} \cdot Q_{r,w,t} + \sum_{p \in P|(p,s) \in PST} F_{p,s,t}^{Trucked} \cdot v_{p,w,[t]} + \sum_{p \in P|(p,s) \in CST} F_{p,s,t}^{Trucked} \cdot v_{p,w,[t]}$$

$$= Q_{s,w,t} \cdot (L_{s,t}^{Storage} + \sum_{n \in N|(s,n) \in SNA} F_{s,n,t}^{Piped} + \sum_{p \in P|(s,p) \in SCA} F_{s,p,t}^{Piped} + \sum_{k \in K|(s,k) \in SKA} F_{s,k,t}^{Piped}$$

$$+ \sum_{r \in R|(s,r) \in SRA} F_{s,r,t}^{Piped} + \sum_{o \in O|(s,o) \in SOA} F_{s,o,t}^{Piped} + \sum_{p \in P|(s,p) \in SCT} F_{s,p,t}^{Trucked} + \sum_{k \in K|(s,k) \in SKT} F_{s,k,t}^{Trucked})$$

Treatment Site Water Quality $\forall r \in R, w \in W, t \in T$

The water quality at treatment sites is dependent on the flow rates into the treatment site, the efficiency of treatment, and the water quality of the flows. Even mixing is assumed, so all outgoing flows have the same water quality. The treatment process does not affect water quality.

$$\epsilon_{r,W_r^{TreatmentComponent}}^{Treatment} \cdot (\sum_{n \in N|(n,r) \in NRA} F_{n,r,t}^{Piped} \cdot Q_{n,w,t} + \sum_{s \in S|(s,r) \in SRA} F_{s,r,t}^{Piped} \cdot Q_{s,w,t}$$

$$+ \sum_{p \in P|(p,r) \in PRT} F_{p,r,t}^{Trucked} \cdot v_{p,w,[t]} + \sum_{p \in P|(p,r) \in CRT} F_{p,r,t}^{Trucked} \cdot v_{p,w,[t]})$$

$$= Q_{r,w,t} \cdot (\sum_{p \in P|(r,p) \in RCA} F_{r,p,t}^{Piped} + F_{r,t}^{UnusedTreatedWater})$$

where $\epsilon_{r,w}^{Treatment} \leq 1$.

Network Node Water Quality $\forall n \in N, w \in W, t \in T$

The water quality at nodes is dependent on the flow rates into the node and the water quality of the flows. Even mixing is assumed, so all outgoing flows have the same water quality.

$$\sum_{p \in P|(p,n) \in PNA} F_{p,n,t}^{Piped} \cdot v_{p,w,[t]} + \sum_{p \in P|(p,n) \in CNA} F_{p,n,t}^{Piped} \cdot v_{p,w,[t]}$$

$$+ \sum_{\tilde{n} \in N|(\tilde{n},n) \in NNA} F_{\tilde{n},n,t}^{Piped} \cdot Q_{\tilde{n},w,t} + \sum_{s \in S|(s,n) \in SNA} F_{s,n,t}^{Piped} \cdot Q_{s,w,t}$$

$$+ \sum_{r \in R|(r,n) \in RNA} F_{r,n,t}^{Piped} \cdot Q_{r,w,t}$$

$$= Q_{n,w,t} \cdot (\sum_{\tilde{n} \in N|(n,\tilde{n}) \in NNA} F_{n,\tilde{n},t}^{Piped} + \sum_{p \in P|(n,p) \in NCA} F_{n,p,t}^{Piped}$$

$$+ \sum_{k \in K|(n,k) \in NKA} F_{n,k,t}^{Piped} + \sum_{r \in R|(n,r) \in NRA} F_{n,r,t}^{Piped}$$

$$+ \sum_{s \in S|(n,s) \in NSA} F_{n,s,t}^{Piped} + \sum_{o \in O|(n,o) \in NOA} F_{n,o,t}^{Piped})$$

Completions Pad Intermediate Node Water Quality $\forall p \in P, w \in W, t \in T$

Water Quality at Completions Pads

Water that is piped and trucked to a completions pad is mixed and split into two output streams: Stream (1) goes to the completions pad and stream (2) is input to the completions storage. This mixing happens at an intermediate

node. Finally, water that meets completions demand comes from two inputs: The first input is output stream (1) from the intermediate step. The second is outgoing flow from the storage tank.

The water quality at the completions pad intermediate node is dependent on the flow rates of water from outside of the pad to the pad. Even mixing is assumed, so the water to storage and water to completions input have the same water quality.

$$
\sum_{n \in N | (n,p) \in NCA} F_{n,p,t}^{Piped} \cdot Q_{n,w,t} + \sum_{\tilde{p} \in P | (\tilde{p},p) \in PCA} F_{\tilde{p},p,t}^{Piped} \cdot v_{\tilde{p},w,[t]} + \sum_{s \in S | (s,p) \in SCA} F_{s,p,t}^{Piped} \cdot Q_{s,w,t}
$$
$$
+ \sum_{\tilde{p} \in P | (\tilde{p},p) \in CCA} F_{\tilde{p},p,t}^{Piped} \cdot v_{\tilde{p},w,[t]} + \sum_{r \in R | (r,p) \in RCA} F_{r,p,t}^{Piped} \cdot Q_{r,w,t} + \sum_{f \in F | (f,p) \in FCA} F_{f,p,t}^{Sourced} \cdot Q_{f,w,t}
$$
$$
+ \sum_{\tilde{p} \in P | (\tilde{p},p) \in PCT} F_{\tilde{p},p,t}^{Trucked} \cdot v_{\tilde{p},w,[t]} + \sum_{s \in S | (s,p) \in SCT} F_{s,p,t}^{Trucked} \cdot Q_{s,w,t} + \sum_{\tilde{p} \in P | (\tilde{p},p) \in CCT} F_{\tilde{p},p,t}^{Trucked} \cdot v_{\tilde{p},w,[t]}
$$
$$
+ \sum_{f \in F | (f,p) \in FCT} F_{f,p,t}^{Trucked} \cdot Q_{f,w,t} = Q_{p^{IntermediateNode},w,t} \cdot (F_{p,t}^{PadStorageIn} + F_{p,t}^{CompletionsDestination})
$$

Completions Pad Input Node Water Quality $\forall p \in P, w \in W, t \in T$

The water quality at the completions pad input is dependent on the flow rates of water from pad storage and water from the intermediate node. Even mixing is assumed, so all water into the pad is of the same water quality.

$$
F_{p,t}^{PadStorageOut} \cdot Q_{p^{PadStorage},w,t} + F_{p,t}^{CompletionsDestination} \cdot Q_{p^{IntermediateNode},w,t} = Q_{p,w,t} \cdot \gamma_{p,t}^{Completions}
$$

Completions Pad Storage Node Water Quality $\forall p \in P, w \in W, t \in T$

The water quality at pad storage sites is dependent on the flow rates into the pad storage site, the volume of water in pad storage in the previous time period, and the quality of each of these flows. Even mixing is assumed, so the outgoing flow to completions pad and water in storage at the end of the period have the same water quality. If it is the first time period, the initial storage level and initial water quality, respectively, replace the water stored and water quality in the previous time period.

For $t = 1$:

$$
\lambda_{s,t=1}^{PadStorage} \cdot \xi_{p,w}^{PadStorage} + F_{p,t}^{PadStorageIn} \cdot Q_{p^{IntermediateNode},w}
$$
$$
= Q_{p^{PadStorage},w,t} \cdot (L_{s,t}^{PadStorage} + F_{p,t}^{PadStorageOut})
$$

For $t > 1$:

$$
L_{s,t-1}^{PadStorage} \cdot Q_{p^{PadStorage},w,t-1} + F_{p,t}^{PadStorageIn} \cdot Q_{p^{IntermediateNode},w}
$$
$$
= Q_{p^{PadStorage},w,t} \cdot (L_{s,t}^{PadStorage} + F_{p,t}^{PadStorageOut})
$$

Beneficial Reuse Water Quality $\forall o \in O, w \in W, t \in T$

The water quality at beneficial reuse sites is dependent on the flow rates into the site and the water quality of the flows.

$$
\sum_{n \in N | (n,o) \in NOA} F_{n,o,t}^{Piped} \cdot Q_{n,w,t} + \sum_{s \in S | (s,o) \in SOA} F_{s,o,t}^{Piped} \cdot Q_{s,w,t} + \sum_{p \in P | (p,o) \in POT} F_{p,o,t}^{Trucked} \cdot v_{p,w,[t]}
$$
$$
= Q_{o,w,t} \cdot F_{o,t}^{BeneficialReuseDestination}
$$

Strategic Model Discrete Water Quality Extension

In the previous chapter a model for tracking the water quality was shown. Without fixing the flows this model is non-linear. By discretizing the number of water qualities for all locations over time we can make the model linear again.

The discretization works as follows.

Take for example this term from the Disposal Site Water Quality:

$$F_{k,t}^{DisposalDestination} \cdot Q_{k,w,t}$$

Both terms are continuous, so this is non-linear.

First we introduce a set, parameter, variables and constraints

Discrete Water Quality Sets

$q \in Q$ Discrete Water Qualities

Discrete Water Quality Parameters

$Q_{w,q}^{DiscreteQuality}$ = Values for discrete Water Qualities

Discrete Water Quality Variables

$Z_{l,t,w,q}$ = Binary decision variable for which discrete quality chosen

$F_{k,t,w,q}^{DiscreteDisposalDestination}$ = Water injected at disposal site for each discrete quality

Only One Discrete Quality Per Location $\forall$l in L, t in T, w in W

For each location in time only one discrete water quality can be chosen for a water quality component.

$$\sum_{(q)\in Q} Z_{l,t,w,q} = 1$$

Discrete Max Disposal Destination $\forall$l in L, t in T, w in W, q in Q

For each location in time only for one discrete quality there can be water injected at the disposal site and at most the capacity for that disposal site. For all the others it is equal to zero.

$$F_{k,t,w,q}^{DiscreteDisposalDestination} \leq D_{k,[t]}^{Capacity} \cdot Z_{l,t,w,q}$$

Sum Flow Discrete Disposal Destinations is Flow Disposal Destination $\forall$l in L, t in T, w in W

For each location in time the sum of the flows for all the discrete qualities is equal to the actual flow going to the disposal site.

$$\sum_{(q)\in Q} F_{k,t,w,q}^{DiscreteDisposalDestination} = F_{k,t}^{DisposalDestination}$$

We can now rewrite the non linear equation showed before to:

$$\sum_{(q)\in Q} F_{k,t,w,q}^{DiscreteDisposalDestination} \cdot Q_{w,q}^{DiscreteQuality}$$

Rewriting the whole constraints goes as follows:

Disposal Site Water Quality $\forall$k in K, w in W, t in T

The water quality of disposed water is dependent on the flow rates into the disposal site and the quality of each of these flows.

$$
\sum_{n\in N|(n,k)\in NKA} F_{n,k,t}^{Piped}\cdot Q_{n,w,t} + \sum_{s\in S|(s,k)\in SKA} F_{s,k,t}^{Piped}\cdot Q_{s,w,t} + \sum_{r\in R|(r,k)\in RKA} F_{r,k,t}^{Piped}\cdot Q_{r,w,t}
$$
$$
+ \sum_{s\in S|(s,k)\in SKT} F_{s,k,t}^{Trucked}\cdot Q_{s,w,t} + \sum_{p\in P|(p,k)\in PKT} F_{p,k,t}^{Trucked}\cdot v_{p,w,[t]}
$$
$$
+ \sum_{p\in P|(p,k)\in CKT} F_{p,k,t}^{Trucked}\cdot v_{p,w,[t]} + \sum_{r\in R|(r,k)\in RKT} F_{r,k,t}^{Trucked}\cdot Q_{r,w,t}
$$
$$
= F_{k,t}^{DisposalDestination}\cdot Q_{k,w,t}
$$

Can be rewritten as

Discrete Disposal Site Water Quality ∀k in K, w in W, t in T

The water quality of disposed water is dependent on the flow rates into the disposal site and the quality of each of these flows.

$$
\sum_{n\in N|(n,k)\in NKA}\sum_{(q)\in Q} F_{n,k,t,q}^{DiscretePiped}\cdot Q_{w,q}^{DiscreteQuality}
$$
$$
+ \sum_{s\in S|(s,k)\in SKA}\sum_{(q)\in Q} F_{s,k,t,q}^{DiscretePiped}\cdot Q_{w,q}^{DiscreteQuality}
$$
$$
+ \sum_{r\in R|(r,k)\in RKA}\sum_{(q)\in Q} F_{r,k,t,q}^{DiscretePiped}\cdot Q_{w,q}^{DiscreteQuality}
$$
$$
+ \sum_{s\in S|(s,k)\in SKT}\sum_{(q)\in Q} F_{s,k,t,q}^{DiscreteTrucked}\cdot Q_{w,q}^{DiscreteQuality}
$$
$$
+ \sum_{p\in P|(p,k)\in PKT} F_{p,k,t}^{Trucked}\cdot v_{p,w,[t]}
$$
$$
+ \sum_{p\in P|(p,k)\in CKT} F_{p,k,t}^{Trucked}\cdot v_{p,w,[t]}
$$
$$
+ \sum_{r\in R|(r,k)\in RKT}\sum_{(q)\in Q} F_{r,k,t,q}^{DiscreteTrucked}\cdot Q_{w,q}^{DiscreteQuality}
$$
$$
\leq \sum_{(q)\in Q} F_{k,t,w,q}^{DiscreteDisposalDestination}\cdot Q_{w,q}^{DiscreteQuality}
$$

The constraints for the DiscretePiped and DiscreteTrucked are similar to the DiscreteDisposalDestination.

Note: The = sign in the original constraint is changed to ≤ sign in the discretized version.

References

Cafaro, D. C., & Grossmann, I. (2021). Optimal design of water pipeline networks for the development of shale gas resources. AIChE Journal, 67(1), e17058.

5.3 Tutorials

PARETO tutorials are currently under development and will be made publicly available as soon as possible.

Since PARETO project is an open source project and we are in collaboration with WaterTAP and IDAES projects, the IDAES tutorials and WaterTAP tutorials are good learning materials.

In particular, new users should be familiar with Python and Pyomo. Because PARETO and IDAES share this common base, we refer users unfamiliar with Python or Pyomo to the IDAES tutorials Short Introduction to Python and Pyomo or the Longer Introduction to Python and Pyomo.

PARETO will leverage WaterTAP models for desalination technologies, therefore, WaterTAP tutorials can be found here: https://watertap.readthedocs.io/en/latest/tutorials/index.html

The IDAES documentation and tutorials can be found here: https://idaes-pse.readthedocs.io/en/stable/

5.4 Utilities

PARETO project provides a set of user-friendly utility methods to display and analyze results. These methods include debugging tools, plotting utilities, and Python-Excel interfaces.

5.4.1 Get Data Functions

The following functions are used to conveniently acquire data.

Function	Section
get_data	*Get Data*
set_consistency_check	*Set Consistency Check*
od_matrix	*Origin/Destination Matrix*

Get Data

Method Description:

This method uses Pandas methods to read data for sets and parameters from an Excel spreadsheet. Sets are assumed to not have neither a header nor an index column. In addition, the data should be placed in column A, row 2, for example:

Parameters can be in either table or column format. Table format requires a header (usually time periods) and index columns whose elements should be contained in a set. Each index column should be labeled with a header starting in cell A2. Spreadsheet names for sets should be used as headers; however, generic keywords "NODES" and "INDEX" can also be used. Column format requires that each set be placed in one column, starting from cell A3. Spreadsheet names for sets should be used as headers in row 2 for each column "NODES" and "INDEX" can also be used. Data should be provided in the last column, and the keyword "VALUE" should be used as header.

This method outputs a dictionary that contains a list for each set and a dictionary that contains parameters in the following format: {'param1': {(set1, set2): value}, 'param1': {(set1, set2): value}}

How to Use:

Fig. 4: Figure 1. get_data Set Setup Format

Fig. 5: Figure 2. get_data Parameter Setup Format

```
set_list = ['ProductionPads','CompletionsPads'], parameter_list = ['DriveTimes',
↪'CompletionsDemand]
fpath = 'path\\to\\excel\\file.xlsx'
[df_sets, df_parameters] = get_data(fpath, set_list, parameter_list)
```

Set Consistency Check

Method Description:

This method checks if the elements included in a table or parameter have been defined as part of the Sets that index such parameter. set_consistency_check() raises a TypeError exception If there are entries in the Parameter that are not contained in the Sets, and prints out a list with all the entries that require revision.

How to Use:

The method requires one specified parameter (e.g. ProductionRates) AND one OR several sets over which the afore-mentioned parameter is declared (e.g.ProductionPads, ProductionTanks, TimePeriods). In general, the method can be run as follows: set_consistency_check(Parameter, set_1, set_2, etc)

Origin/Destination Matrix

Method Description:

This method allows the user to request drive distances and drive times using Bing maps API and Open Street Maps API. The method accept the following input arguments: - origin:

> REQUIRED. Data containing information regarding location name, and coordinates latitude and longitude. Two formats are acceptable:
>
> - {(origin1,"latitude"): value1, (origin1,"longitude"): value2} or
>
> - {origin1:{"latitude":value1, "longitude":value2}}
>
> The first format allows the user to include a tab with the corresponding data in a table format as part of the workbook casestudy.

- **destination:**
  OPTIONAL. If no data for destination is provided, it is assumed that the origins are also destinations.

- **api:**
  OPTIONAL. Specify the type of API service, two options are supported:

  - Bing maps: https://docs.microsoft.com/en-us/bingmaps/rest-services/

  - Open Street Maps: https://www.openstreetmap.org/

  If no API is selected, Open Street Maps is used by default

- **api_key:**
  An API key should be provided in order to use Bing maps. The key can be obtained at: https://www.microsoft.com/en-us/maps/create-a-bing-maps-key

- **output:**
  OPTIONAL. Define the parameters that the method will output. The user can select:

  - 'time': A list containing the drive times between the locations is returned

  - 'distance': A list containing the drive distances between the locations is returned

  - 'time_distance': Two lists containing the drive times and drive distances between the locations is returned

  If not output is specified, 'time_distance' is the default

- **fpath:**
  OPTIONAL. od_matrix() will ALWAYS output an Excel workbook with two tabs, one that contains drive times, and another that contains drive distances. If not path is specified, the excel file is saved with the name 'od_output.xlsx' in the current directory.

- **create_report:**
  OPTIONAL. if True an Excel report with drive distances and drive times is created

5.4.2 Results Functions

The following functions are used to conveniently display and analyze data.

Function	Section
generate_report	*Generate Report*
generate_sankey	*Generate Sankey*
plot_sankey	*Plot Sankey*
plot_bars	*Plot Bars*
plot_scatter	*Plot Scatter*

Generate Report

Method Description:

This method identifies the type of model: [strategic, operational], creates a printing list based on is_print, and creates a dictionary that contains headers for all the variables that will be included in an Excel report. The dictionaries are used to create separate excel sheets which categorize the data by variable name or type. This same data is put into excel sheets named after each variable as well as an overview sheet which contains totals and Key Performance Indicators (KPI) information.

> **Warning:** If an indexed variable is added or removed from a model, the printing lists and headers should be updated accordingly.

The output of this method prints out each variable's information in the terminal as specified by the user, as shown below.

Fig. 6: Figure 1. Example of Terminal Output

How to Use:

This method requires two parameters:

1.) The model that is being returned from the create_model() method after a solution is found

2.) An array of an "enum" class value specifying which variables to print which are chosen by the user. These values are:

> *"PrintValues.Essential"* – Specifies that the overview information will be printed

Fig. 7: Figure 2. Example of Excel Output

"PrintValues.Nominal" – Specifies that all information from PrintValues.Essential + Trucked, Piped, and Sourced water information will be printed

"PrintValues.Detailed" – Specifies that all information including slack information will be printed

Example of how this method is used:

```
strategic_model = create_model(df_sets, df_parameters)
results = opt.solve(strategic_model, tee=True)
[model, results] = generate_report(strategic_model, is_print=[PrintValues.Detailed],␣
 ↪fname="..\\..\\PARETO_report.xlsx")
```

Generate Sankey

Method Description:

Sankey diagrams are a graphic tool used to easily visualize supply-sink flows across a given infrastructure (source/destination). The relative width of each "flow" is proportional to the amount of water that is being transported between locations. Such diagrams are commonly used to visualize the complex nature of money, energy or material flows.

This method receives the final lists for source, destination, value, and labels to be used in generating the Sankey diagram. It also receives arguments that determine font size and plot titles. The user can save the Sankey diagram in the following formats: jpg, jpeg, pd, png, svg, and html. Html format is set by default.

How to Use:

```
# Creating links and nodes based on the passed in lists to be used as the data for␣
 ↪generating the Sankey diagram
 link = dict(source=source, target=destination, value=value)
 node = dict(label=label, pad=30, thickness=15, line=dict(color="black", width=0.5))
 data = go.Sankey(link=link, node=node)

 # Assigning sankey diagram to fig variable
 fig = go.Figure(data)
 fig.write_html("first_figure.html", auto_open=True)
```

Plot Sankey

Method Description:

This method receives data in the form of 3 seperate lists (origin, destination, value lists), generate_report dictionary output format, or get_data dictionary output format. It then places this data into 4 lists of unique elements so that proper indexes can be assigned for each list so that the elements will correspond with each other based off of the indexes. These lists are then passed into the outlet_flow method which gives an output which is passed into the method to generate the sankey diagram.

How to Use:

This method requires two parameters:

1.) An input data dictionary that includes the time periods requested as well as said data. The data is passed in as 'pareto_var' and can be in get_data() format, which requires labels, generate_report() format, or 3 separate lists:

"pareto_var" – This parameter can be variable data returned from the get_data() or generate_report() format
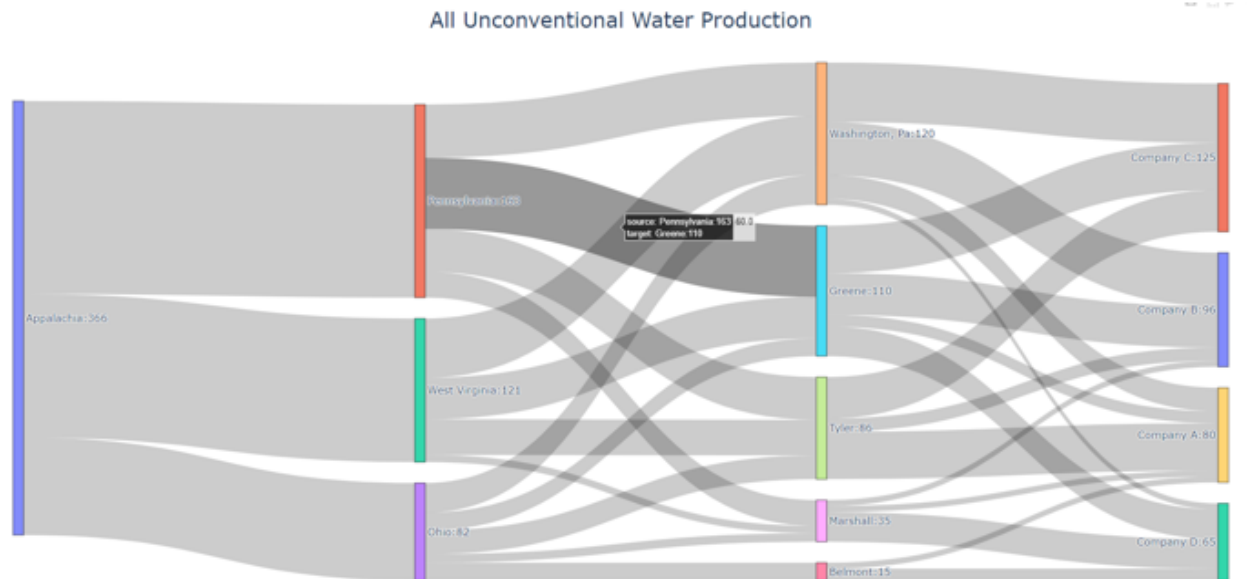
Fig. 8: Figure 3. Example of Sankey Diagram Showing Water Production Flows

"time_period" – This is used to specify which time periods from the data that the user wants shown in the diagram. If the user passes no time periods in, then all time periods are used in the data.

"labels" – This is only required if the data being passed in is in get_data() format. The labels are used to distinguish between the columns.

2.) A dictionary of arguments that include formatting options like font size, title of the plot and output file:

output_file – This parameter is used for creating the file that contains the Sankey Diagram created by this method

Example of how this method is used:

```
args = {"font_size": 15,
        "plot_title": "Completion Storage",
        "output_file": "..\\first_sankey.png"}

input_data = {"pareto_var": df_parameters["v_f_Trucked"]
              "labels": [("Origin", "Destination", "Time", "Trucked Water")]}

plot_sankey(input_data, args)
```

Plot Bars

Method Description:

This method generates a bar chart based on the variable data that the user passes in. It automatically creates either an animated bar chart (if the variable is indexed by time) or a static bar chart.

How to Use

This method requires two parameters:

1.) A dictionary including the data and labels that are being used, either in get_data() output format or generate_report() output format. (Labels only required for get_data() format).
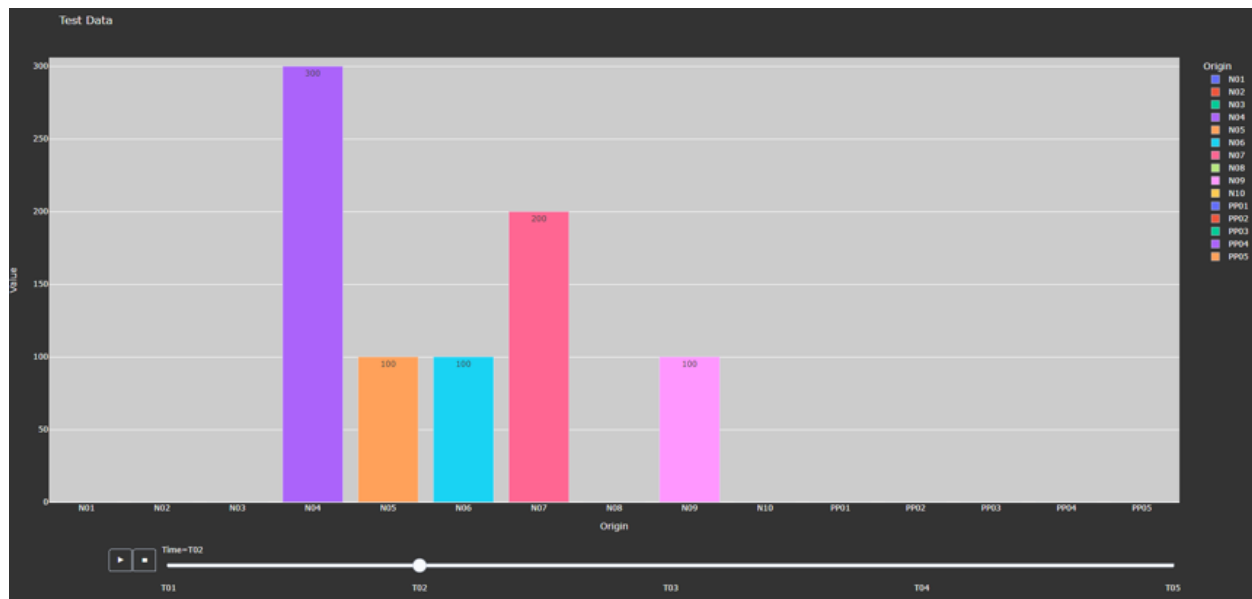
Fig. 9: Figure 4. Animated Bar Chart

"pareto_var"– This parameter contains the data that the user wants to use

"labels"– This is a tuple that contains the labels for each column of the data provided.

2.) A dictionary of arguments that include the title of the plot, a group by parameter, and an output file. Here is an example of the arguments:

"group_by" - This specifies what field will be used as the x axis in the plot

"output_file" - This parameter is used for creating the file that contains the Bar Chart created by this method.

"y_axis" - This specifies if the user wants to take the logarithm of the y axis. If not provided, then the y axis remains the default(linear).

Example of how this method is used:

```
args = {"plot_title": "Completion Storage",
        "y_axis": "log",
        "group_by": "Destination",
        "output_file": "first_bar.html"}

input_data = {"pareto_var": df_parameters["v_f_PadStorage"]
              "labels": [("Completion Pad", "Time", "Storage Levels")]}

plot_bars(input_data, args)
```

Plot Scatter

Method Description:

This method creates the scatter plot that is generated from the variable data that the user passes in. It creates either an animated scatter plot(if the variable is indexed by time) or a static scatter plot.
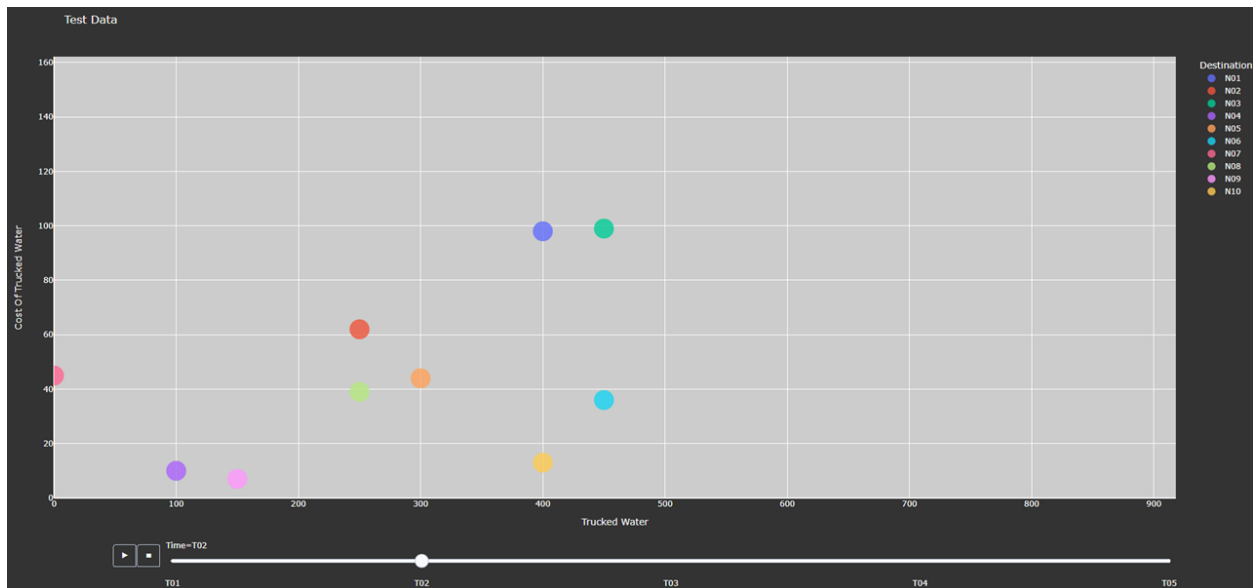

Fig. 10: Figure 5. Animated Scatter Chart. Notice the time period slider at the bottom.

How to Use

This method requires two parameters:

1.) An input data dictionary that include the variables for x and y axis, a size parameter, and labels parameters that provides a tuple of labels (only required for get_data() format) for x, y, and size variables.

"pareto_var"– This parameter contains the data that the user wants to use.

"labels"– This is a tuple that contains the labels for each column of the data provided.

"size"- This specifies what will be used for the size of each individual marker on the plot. If the size parameter is not provided, a default size is given to all the markers. There are 3 options for the size parameter:

- "x/y" - This specifies that size will be calculated as a ratio of the x variable data over the y variable data
- "y/x" - This specifies that size will be calculated as a ratio of the y variable data over the x variable data
- A Pareto variable that contains data for the size of the bubbles. The data must match the column used for grouping the data in the option "group_by".

2.) A dictionary of arguments that include the title of the plot, a group by parameter, and an output file. Here is an example of the arguments:

"group_by" - This specifies what field will be used as the x axis in the plot. The column name should be used to indicate how to group the data. If "group_by" is not specified, then first column is used.

"output_file" - This parameter is used to name a file that the figure will be output to. It can be a file path such as "..\first_figure.html" or just the file name itself "first_figure.html". There will always need to be

```
"size": df_parameters["v_F_Trucked"],
"size": 'x/y',
"size": 'y/x',
```

Fig. 11: Figure 6. Options for specifying the bubbles size.

a specified extension to the file. The accepted file extensions are as follows: .html, .png, .jpg, .jpeg, .svg, .pdf

"print_data" - The PARETO methods allow the user to specify if they want the plotted data to be printed in the console (default is False):

- True: The dataframe used for creating the figure is printed in the console

```
     Completion Pad  Time  Storage Levels
0                 A    T1          9.0e+03
252               B    T1          1.0e-10
367               C    T1          1.0e-10
392               D    T1          1.0e-10
1                 A    T2          9.0e+03
253               B    T2          1.0e-10
368               C    T2          1.0e-10
393               D    T2          1.0e-10
131               A    T3          1.0e-10
254               B    T3          1.0e-10
10                C    T3          2.0e+02
394               D    T3          1.0e-10
132               A    T4          1.0e-10
255               B    T4          1.0e-10
11                C    T4          3.0e+02
395               D    T4          1.0e-10
```

Fig. 12: Figure 7. Setting print_data to True will print out a dataframe for easy inspection.

"group_by_category" - This specifies how the color of the nodes will be assigned for easy visualization. There are 3 options:

- True: This will cause the color of the chart markers to be grouped based on the names of the nodes. For example: PP, CP, N, R, S, K, etc will be assigned a unique color.

- False: The data won't be categorized by color, therefore one color will be used for the chart markers.

- A Pareto variable containing a custom categorization. The method will recognize the variable automatically and the values in this variable will be used for assigning colors to the categories that are provided. An excel sheet should be created with all Node names, removing all duplicates, and assigning a numerical value to each specific node with the category the user would like it to be associated with. This approach is best for the situations where nodes of different types are to be

categorized together.

Fig. 13: Figure 8. Data used for custom categories.

Example of how this method can be used:

```
args = {"plot_title": "Trucked Water",
         "y_axis": "log",
         "group_by": "Origin",
         "output_file": "first_bar.html",
         "print_data": True,
         "group_by_category": df_parameters["plot_scatter_categories"]}

input_data = {"pareto_var_x": df_parameters["plot_scatter_vFPiped"],
              "pareto_var_y": df_parameters["plot_scatter_vCPiped"],
              "size": df_parameters["plot_scatter_vSize"], # 'x/y', 'y/x'
              "labels_x": [("Origin", "Destination", "Time", "Trucked Water")],
              "labels_y": [("Origin", "Destination", "Time", "Cost of Trucked Water
 ↪")],
              "labels_size": [("Origin", "Destination", "Time", "Size")],
              }

plot_scatter(input_data, args)
```

5.5 Frequently Asked Questions

5.5.1 How to ...

... Run examples?

PARETO project provides examples to run the operational produced water management model and the strategic produced water management model (see pareto/case_studies/). To run the examples, go to:

- pareto/operatinal_water_management/run_operational_model.py

- pareto/strategic_water_management/run_strategic_model.py

... Get more help?

Use the website to register for the PARETO support mailing list. Then you can send questions to pareto-support@pareto.org. For more specific technical questions, we recommend our new PARETO discussions board on Github.

5.5.2 Troubleshooting

Missing win32api DLL

For Python 3.8 and maybe others, you can get an error when running Jupyter on Windows 10 about missing the win32api DLL. There is a relatively easy fix:

```
pip uninstall pywin32
pip install pywin32==225
```

5.6 License Agreement

PARETO Copyright (c) 2021, by the software owners: The Regents of the University of California, through Lawrence Berkeley National Laboratory, et al. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the Produced Water Application for Beneficial Reuse Environmental Impact and Treatment Optimization (PARETO), University of California, Lawrence Berkeley National Laboratory, U.S. Dept. of Energy, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE S PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES: LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You are under no obligation whatsoever to provide any bug fixes, patches, or upgrades to the features, functionality or performance of the source code ("Enhancements") to anyone; however, if you choose to make your Enhancements available either publicly, or directly to Lawrence Berkeley National Laboratory, without imposing a separate written license agreement for such Enhancements, then you hereby grant Lawrence Berkeley National Laboratory the following license: a non-exclusive, royalty-free perpetual license to install, use, modify, prepare derivative works, incorporate into other computer software, distribute, and sublicense such enhancements or derivative works thereof, in binary and source code form

5.7 Copyright Notice

PARETO was produced under the DOE Produced Water Application for Beneficial Reuse Environmental Impact and Treatment Optimization (PARETO), and is copyright (c) 2021 by the software owners: The Regents of the University of California, through Lawrence Berkeley National Laboratory, et al. All rights reserved.

NOTICE. This Software was developed under funding from the U.S. Department of Energy and the U.S. Government consequently retains certain rights. As such, the U.S. Government has been granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable, worldwide license in the Software to reproduce, distribute copies to the public, prepare derivative works, and perform publicly and display publicly, and to permit other to do so.

INDICES AND TABLES

- genindex

H

Home
    pareto, 1

P

pareto
    Home, 1