
PARETO

Release 0.10.dev0

The PARETO Project

Apr 26, 2024

CONTENTS

1	Project Goals	1
2	Collaborating institutions	3
3	Project Timeline	5
4	Contact us	7
5	Contents	9
5.1	Getting Started	9
5.2	Model Library	12
5.3	Case Studies	71
5.4	Tutorials and Examples	76
5.5	Utilities	76
5.6	Frequently Asked Questions	89
5.7	License Agreement	90
5.8	Copyright Notice	91
6	Indices and tables	93
	Index	95

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”.

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.

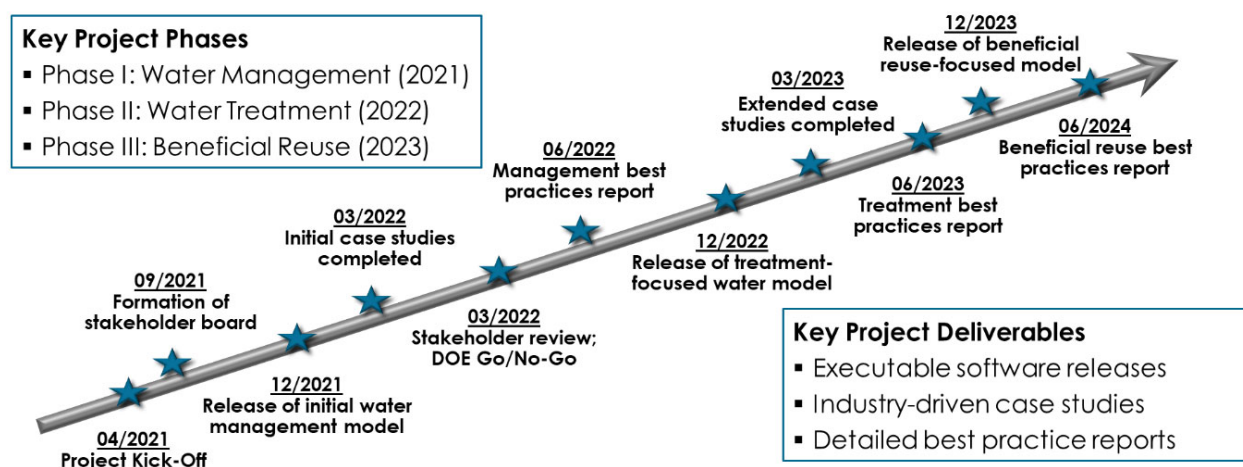


Fig. 1: Figure 4. Project PARETO Timeline

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.

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 the [support email list](#).

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 the [support email list](#).

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	<i>Users</i>
Core-dev	<i>Core-dev</i>
Hybrid	<i>Hybrid</i>

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. Create a Conda environment, named e.g. `pareto-env`:

```
conda create --yes --name pareto-env python=3.9
```

2. Activate the `pareto-env` Conda environment. This command must be run every time a new console/terminal window is opened:

```
conda activate pareto-env
```

3. 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"
```

4. After installing PARETO, install the open-source solvers provided by the IDAES project:

```
idaes get-extensions --verbose
```

5.1.3 Core-dev

Important: For more developer resources, see the [PARETO Wiki on GitHub](#).

1. Fork the repo on GitHub (your copy of the main repo)
2. Clone your fork locally, with only one of the following commands, creating a workspace (replacing `<githubid>` with your github user id):

```
git clone https://github.com/<githubid>/project-pareto  
git clone git@github.com:<githubid>/project-pareto
```

3. Create a dedicated Conda environment for development work:

```
conda create --name pareto-dev python=3.9 --yes  
conda activate pareto-dev
```

4. Activate the `pareto-dev` Conda environment. This command must be run every time a new console/terminal window is opened:

```
conda activate pareto-dev
```

5. Navigate into the new `project-pareto` directory, then run the following command to install PARETO in editable mode and the development-only dependencies:

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

6. After installing PARETO, install the open-source solvers provided by the IDAES project:

```
idaes get-extensions --verbose
```

7. (Recommended) install the pre-commit checks that will run automatically whenever `git commit` is used, preventing the commit from being created if any of the checks fail:

```
pre-commit install
```

Note: `pre-commit` can cause commits to fail for reasons unrelated to the pre-commit checks. For more information, check the [related GitHub issue\(s\)](#).

5.1.4 Hybrid

User that can edit the base code

Important: Unlike a local clone of the repository, ZIP archives of the repository are static snapshots that cannot be automatically updated, track changes, or publish (push) through Git, while still allowing to modify the PARETO codebase locally.

1. Create and activate environment:

```
conda create -n pareto-env python=3.9 pip --yes
conda activate pareto-env
```

2. Download a ZIP file containing a snapshot of the main branch of the repository by navigating to the following URL: <https://github.com/project-pareto/project-pareto/archive/refs/heads/main.zip>

Note: The URL can be modified to create a ZIP file for other repositories, branches or commits. e.g. for the fork belonging to the user `myuser` and the branch `mybranch`, the URL would be <https://github.com/myuser/project-pareto/archive/refs/heads/mybranch.zip>.

3. Unpack zip files (select directory)
4. Navigate to the directory where the ZIP files were extracted
5. Install `pareto-project` (non-git tracked repo):

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

6. After installing PARETO, install the open-source solvers provided by the IDAES project:

```
idaes get-extensions --verbose
```

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.

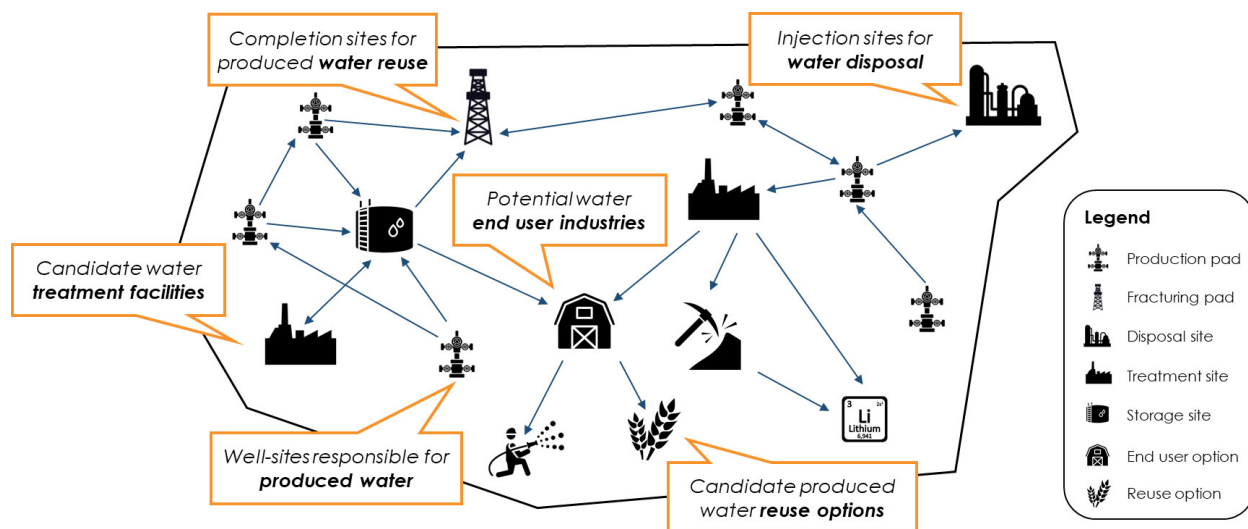


Fig. 1: Figure 1. Problem Description

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 externally sourced
- treatment, production, and storage cost

PARETO jdoe@bl.gov

Create New Scenario

- New Scenario 1** (draft edited 07/27/2021)
- Base Scenario 2 (created 07/21/2021)
- Base Scenario 1 (created 07/24/2021)

DRAFT New Scenario 1 RENAME COPY

Data Input Optimization Setup Model Results

INPUT DATA CATEGORIES

- Demand Forecast
- Supply Forecast**
- Cost Data

IMPORT DATA...

	Site A	Site B	Site C	Site D	Site E
06/2021	23k bbl	----	3k bbl	33k bbl	----
07/2021	20k bbl	----	3k bbl	----	----
08/2021	17k bbl	----	2k bbl	28k bbl	----
09/2021	15k bbl	54k bbl	2k bbl	25k bbl	14k bbl
10/2021	17k bbl	46k bbl	2k bbl	23k bbl	14k bbl

CONTINUE TO OPTIMIZATION SETUP

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

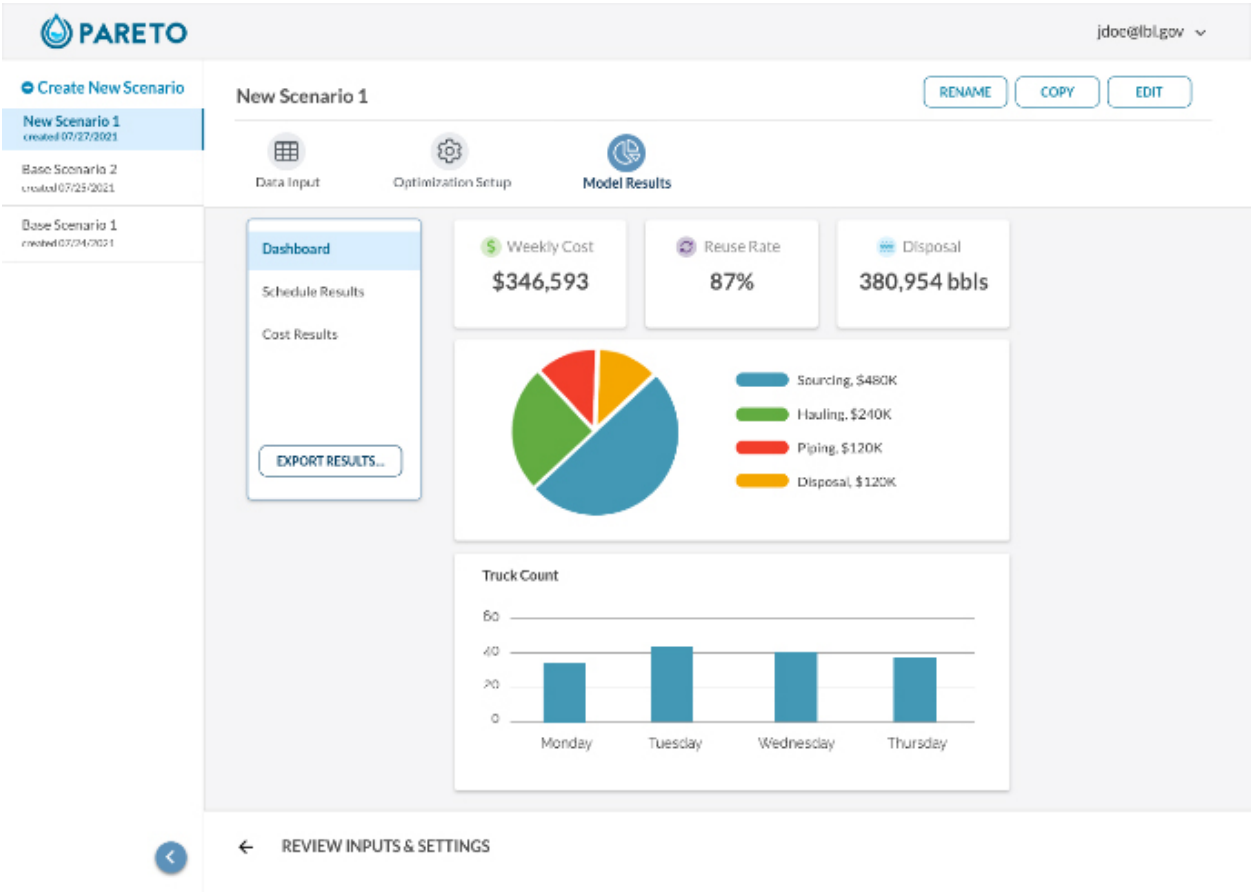


Fig. 3: Figure 3. Output Data

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 maximizing the reuse 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.

Sections
<i>Operational Model Mathematical Notation</i>
<i>Operational Model Mathematical Program Formulation</i>
<i>Operational Model Water Quality Extension</i>
<i>Operational Model Terminology</i>

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)
- $f \in F$ External water 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, ldots)
- $a \in A$ 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
 $(f, p) \in FCA$ Externally sourced water-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
 $(l, \tilde{l}) \in LLA$ All valid pipeline arcs
 $(p, p) \in PCT$ Production-to-completions trucking arcs
 $(f, c) \in FCT$ Externally sourced water-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
 $(l, \tilde{l}) \in LLT$ All valid trucking arcs
 $(p, a) \in PAL$ Pad-to-tank links

Continuous Variables

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

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

$F_{f,p,t}^{Sourced}$ = Externally sourced water 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:

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

$L_{p,a,t}^{ProdTank}$ = 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:

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

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

$B_{p,t}^{Production}$ = 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 external 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

$C_{p,t}^{PadStorage}$ = Cost of storing produced water at completions pad storage

$C^{TotalSourced}$ = Total cost of externally sourced water

$C^{TotalDisposal}$ = 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

$S_{l,i}^{PipelineCapacity}$ = 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

$y_{l,i,t}^{Flow}$ = 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

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

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

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

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

$\sigma_{l,i}^{Pipeline}$ = 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
- $\sigma_{f,t}^{ExternalWater}$ = Daily capacity of externally sourced water
- $\sigma_p^{Offloading,Pad}$ = 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
- $c_{r,w}^{Treatment}$ = 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
- $\lambda_{l,\tilde{l}}^{Pipeline}$ = 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 credit
- $\pi_{l,\tilde{l}}^{Pipeline}$ = Pipeline operational cost
- $\pi_l^{Trucking}$ = Trucking hourly cost (by source)
- $\pi_f^{Sourcing}$ = Externally sourced water 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
 $\psi^{TreatmentCapacity}$ = 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 externally sourced 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: $\forall p \in CP, t \in T$

$$\gamma_{p,t}^{Completions} = \sum_{l \in (L-F)|(l,p) \in LLA} F_{l,p,t}^{Piped} + \sum_{f \in F|(f,p) \in LLA} F_{f,p,t}^{Sourced} + \sum_{l \in L|(l,p) \in LLT} F_{l,p,t}^{Trucked} + F_{p,t}^{PadStorageOut} - F_{p,t}^{PadStorageIn} + S_{p,t}^{FracDemand}$$

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

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.

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

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

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

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

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

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

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

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.

$$\sum_{l \in L | (l,p) \in LLT} F_{l,p,t}^{Trucked} \leq \sigma_p^{Offloading,Pad}$$

Completions Pad Processing Capacity:

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

$$\begin{aligned} & \sum_{(n,p) \in NCA} F_{l,\tilde{l},t}^{Piped} + \sum_{(p,p) \in PCA} F_{l,\tilde{l},t}^{Piped} + \sum_{(s,p) \in SCA} F_{l,\tilde{l},t}^{Piped} \\ & + \sum_{(p,c) \in CCA} F_{l,\tilde{l},t}^{Piped} + \sum_{(r,p) \in RCA} F_{l,\tilde{l},t}^{Piped} + \sum_{(p,p) \in PCT} F_{l,\tilde{l},t}^{Trucked} \\ & + \sum_{(s,p) \in SCT} F_{l,\tilde{l},t}^{Trucked} + \sum_{(p,p) \in CCT} F_{l,\tilde{l},t}^{Trucked} \leq \sigma_p^{Processing,Pad} \end{aligned}$$

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.

$$\sum_{l \in L | (l,s) \in LLT} F_{l,s,t}^{Trucked} \leq \sigma_s^{Offloading,Storage}$$

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

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

$$\sum_{l \in L | (l,s) \in LLA} F_{l,s,t}^{Piped} + \sum_{l \in L | (l,s) \in LLT} F_{l,s,t}^{Trucked} \leq \sigma_s^{Processing,Storage}$$

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: $\forall (p,a) \in PAL, t \in T$

For $t = 1$:

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

For $t > 1$:

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

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

For $t = 1$:

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

For $t > 1$:

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

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: $\forall p \in P, t \in T$

$$\sum_{(p,a) \in PAL} F_{p,a,t}^{Drain} = B_{p,t}^{Production}$$

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

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

$$F_{p,t}^{Drain} = B_{p,t}^{Production}$$

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.

$$B_{p,t}^{Production} = \sum_{l \in L | (p,l) \in LLA} F_{p,l,t}^{Piped} + \sum_{l \in L | (p,l) \in LLT} F_{p,l,t}^{Trucked} + S_{p,t}^{Production}$$

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.

$$B_{p,t}^{Production} = \sum_{l \in L | (p,l) \in LLA} F_{p,l,t}^{Piped} + \sum_{l \in L | (p,l) \in LLT} F_{p,l,t}^{Trucked} + S_{p,t}^{Flowback}$$

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.

$$\sum_{l \in L | (l,n) \in LLA} F_{l,n,t}^{Piped} = \sum_{l \in L | (n,l) \in LLA} F_{n,l,t}^{Piped}$$

Bi-Directional Flow: $\forall (l, \tilde{l}), t \in T$ such that $(l, \tilde{l}) \in LLA, (\tilde{l}, l) \in LLA, l \in L - F - O$, and $\tilde{l} \in L - F$

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”.

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

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

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 $t = 1$:

$$L_{s,t}^{Storage} = \lambda_{s,t=1}^{Storage} + \sum_{l \in L | (l,s) \in LLA} F_{l,s,t}^{Piped} + \sum_{l \in L | (l,s) \in LLT} F_{l,s,t}^{Trucked} - \sum_{l \in L | (s,l) \in LLA} F_{s,l,t}^{Piped} - \sum_{l \in L | (s,l) \in LLT} F_{s,l,t}^{Trucked}$$

For $t > 1$:

$$L_{s,t}^{Storage} = L_{s,t-1}^{Storage} + \sum_{l \in L | (l,s) \in LLA} F_{l,s,t}^{Piped} + \sum_{l \in L | (l,s) \in LLT} F_{l,s,t}^{Trucked} - \sum_{l \in L | (s,l) \in LLA} F_{s,l,t}^{Piped} - \sum_{l \in L | (s,l) \in LLT} F_{s,l,t}^{Trucked}$$

Pipeline Capacity:

$\forall (l, \tilde{l}) \in LLA, t \in T$

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

Pipeline Capacity Construction/Expansion: $\forall (l, \tilde{l}) \in LLA, [t \in T]$

$\forall (l, \tilde{l})$ if $(l, \tilde{l}) \in LLA$ and $(\tilde{l}, l) \in LLA$:

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

$\forall (l, \tilde{l})$ if $(l, \tilde{l}) \in LLA$ and $(\tilde{l}, l) \notin LLA$:

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

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.

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

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

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

$$\sum_{l \in L | (l,k) \in LLA} F_{l,k,t}^{Piped} + \sum_{l \in L | (l,k) \in LLT} F_{l,k,t}^{Trucked} \leq D_{k,[t]}^{Capacity}$$

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

$$F_{k,t}^{DisposalDestination} = \sum_{l \in L | (l,k) \in LLA} F_{l,k,t}^{Piped} + \sum_{l \in L | (l,k) \in LLT} F_{l,k,t}^{Trucked}$$

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.

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

$$\sum_{l \in L | (l,r) \in LLA} F_{l,r,t}^{Piped} + \sum_{l \in L | (l,r) \in LLT} F_{l,r,t}^{Trucked} \leq \sigma_r^{Treatment} + S_r^{TreatmentCapacity}$$

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

Treatment Destination Deliveries: The total water delivered to a treatment site is the sum of all piped and trucked flows into the site.

$$\sum_{l \in L | (l,r) \in LLA} F_{l,r,t}^{Piped} + \sum_{l \in L | (l,r) \in LLT} F_{l,r,t}^{Trucked} = F_{r,t}^{TreatmentDestination}$$

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.

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

$$\sum_{l \in L | (l,o) \in LLA} F_{l,o,t}^{Piped} + \sum_{l \in L | (l,o) \in LLT} F_{l,o,t}^{Trucked} \leq \sigma_o^{Reuse} + S_o^{ReuseCapacity}$$

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

$$\sum_{l \in L | (l,o) \in LLA} F_{l,o,t}^{Piped} + \sum_{l \in L | (l,o) \in LLT} F_{l,o,t}^{Trucked} = F_{o,t}^{BeneficialReuseDestination}$$

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

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

$$C_{f,p,t}^{Sourced} = (F_{f,p,t}^{Sourced} + F_{f,p,t}^{Trucked}) \cdot \pi_f^{Sourcing}$$

$$C_{TotalSourced} = \sum_{\forall t \in T} \sum_{f \in F, p \in CP} C_{f,p,t}^{Sourced}$$

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.

$$C_{k,t}^{Disposal} = (\sum_{(l,k) \in LLA} F_{l,k,t}^{Piped} + \sum_{(l,k) \in LLT} F_{l,k,t}^{Trucked}) \cdot \pi_k^{Disposal}$$

$$C_{TotalDisposal} = \sum_{\forall t \in T} \sum_{k \in K} C_{k,t}^{Disposal}$$

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.

$$C_{r,t}^{Treatment} = (\sum_{(l,r) \in LLA} F_{l,r,t}^{Piped} + \sum_{(l,r) \in LLT} F_{l,r,t}^{Trucked}) \cdot \pi_r^{Treatment}$$

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

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.

$$\varepsilon^{Treatment} \cdot (\sum_{l \in L | (l,r) \in LLA} F_{l,r,t}^{Piped} + \sum_{l \in L | (l,r) \in LLT} F_{l,r,t}^{Trucked}) = \sum_{l \in L | (l,r) \in LLA} F_{r,l,t}^{Piped} + F_{r,t}^{UnusedTreatedWater}$$

where $\varepsilon^{Treatment} \leq 1$

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

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

$$C_{p,t}^{CompletionsReuse} = (\sum_{(l,\tilde{l}) \in LLA, l \notin F} F_{l,\tilde{l},t}^{Piped} + \sum_{(l,\tilde{l}) \in LLT, l \notin F} F_{l,\tilde{l},t}^{Trucked}) \cdot \pi_p^{CompletionsReuse}$$

Note: Externally sourced water is excluded from completions reuse costs.

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

Piping Cost: $\forall l \in (L - O - K), \forall \tilde{l} \in (L - F), \forall (l, \tilde{l}) \in LLA, t \in T$

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

$\forall (l, \tilde{l})$ if $l \in F$:

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

Otherwise, $\forall(l, \tilde{l})$ if $l \notin F$:

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

Total Piping Cost:

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

Note: Note: Externally sourced water piping is tracked through Sourced Flow variable.

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.

$$C_{s,t}^{Storage} = \left(\sum_{l \in L | (l,s) \in LLA} F_{l,s,t}^{Piped} + \sum_{l \in L | (l,s) \in LLT} F_{l,s,t}^{Trucked} \right) \cdot \pi_s^{Storage}$$

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

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.

$$R_{s,t}^{Storage} = \left(\sum_{l \in L | (s,l) \in LLA} F_{s,l,t}^{Piped} + \sum_{l \in L | (s,l) \in LLT} F_{s,l,t}^{Trucked} \right) \cdot \rho_s^{Storage}$$

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

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

$$C_{p,t}^{PadStorage} = z_{p,t}^{PadStorage} \cdot \pi_{p,t}^{PadStorage}$$

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

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.

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

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

Note: The constraints above explicitly consider trucking externally sourced water via FCT arcs included in LLT.

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).

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

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.

$$\begin{aligned}
 & \sum_{(n,k) \in NKA} F_{l,l,t}^{Piped} \cdot Q_{n,w,t} + \sum_{(s,k) \in SKA} F_{l,l,t}^{Piped} \cdot Q_{s,w,t} + \sum_{(r,k) \in RKA} F_{l,l,t}^{Piped} \cdot Q_{r,w,t} \\
 & + \sum_{(s,k) \in SKT} F_{l,l,t}^{Trucked} \cdot Q_{s,w,t} + \sum_{(p,k) \in PKT} F_{l,l,t}^{Trucked} \cdot Q_{p,w,t} \\
 & + \sum_{(p,k) \in CKT} F_{l,l,t}^{Trucked} \cdot Q_{p,w,t} + \sum_{(r,k) \in RKT} F_{l,l,t}^{Trucked} \cdot Q_{r,w,t} \\
 & = F_{k,t}^{DisposalDestination} \cdot Q_{k,w,t}
 \end{aligned}$$

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.

$$\begin{aligned}
 & \lambda_{s,t=1}^{Storage} \cdot \xi_{s,w} + L_{s,t-1}^{Storage} \cdot Q_{s,w,t-1} + \sum_{(n,s) \in NSA} F_{l,\tilde{l},t}^{Piped} \cdot Q_{n,w,t} \\
 & + \sum_{(p,s) \in PST} F_{l,\tilde{l},t}^{Trucked} \cdot Q_{p,w,t} + \sum_{(p,s) \in CST} F_{l,\tilde{l},t}^{Trucked} \cdot Q_{p,w,t} \\
 & = Q_{s,w,t} \cdot (L_{s,t}^{Storage} + \sum_{(s,n) \in SNA} F_{l,\tilde{l},t}^{Piped} + \sum_{(s,p) \in SCA} F_{l,\tilde{l},t}^{Piped} + \sum_{(s,k) \in SKA} F_{l,\tilde{l},t}^{Piped} \\
 & + \sum_{(s,r) \in SRA} F_{l,\tilde{l},t}^{Piped} + \sum_{(s,o) \in SOA} F_{l,\tilde{l},t}^{Piped} + \sum_{(s,p) \in SCT} F_{l,\tilde{l},t}^{Trucked} + \sum_{(s,k) \in SKT} F_{l,\tilde{l},t}^{Trucked})
 \end{aligned}$$

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

$$\begin{aligned}
 & \epsilon_{r,w}^{Treatment} \cdot (\sum_{(n,r) \in NRA} F_{l,\tilde{l},t}^{Piped} \cdot Q_{n,w,t} + \sum_{(s,r) \in SRA} F_{l,\tilde{l},t}^{Piped} \cdot Q_{s,w,t} \\
 & + \sum_{(p,r) \in PRT} F_{l,\tilde{l},t}^{Trucked} \cdot Q_{p,w,t} + \sum_{(p,r) \in CRT} F_{l,\tilde{l},t}^{Trucked} \cdot Q_{p,w,t}) \\
 & = Q_{r,w,t} \cdot (\sum_{(r,p) \in RCA} F_{l,\tilde{l},t}^{Piped} + F_{r,t}^{UnusedTreatedWater})
 \end{aligned}$$

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.

$$\begin{aligned}
 & \sum_{(p,n) \in PNA} F_{l,\tilde{l},t}^{Piped} \cdot Q_{p,w,t} + \sum_{(p,n) \in CNA} F_{l,\tilde{l},t}^{Piped} \cdot Q_{p,w,t} \\
 & + \sum_{(\tilde{n},n) \in NNA} F_{l,\tilde{l},t}^{Piped} \cdot Q_{n,w,t} + \sum_{(s,n) \in SNA} F_{l,\tilde{l},t}^{Piped} \cdot Q_{s,w,t} \\
 & = Q_{n,w,t} \cdot (\sum_{(n,\tilde{n}) \in NNA} F_{l,\tilde{l},t}^{Piped} + \sum_{(n,p) \in NCA} F_{l,\tilde{l},t}^{Piped} \\
 & + \sum_{(n,k) \in NKA} F_{l,\tilde{l},t}^{Piped} + \sum_{(n,r) \in NRA} F_{l,\tilde{l},t}^{Piped} \\
 & + \sum_{(n,s) \in NSA} F_{l,\tilde{l},t}^{Piped} + \sum_{(n,o) \in NOA} F_{l,\tilde{l},t}^{Piped})
 \end{aligned}$$

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.

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

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 externally sourced water.

Completions Reuse Water: Water that meets demand at a completions site. This does not include externally sourced water 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 externally sourced water consumption.

Section
<i>Terminology</i>
<i>Strategic Model Mathematical Notation</i>
<i>Strategic Model Mathematical Program Formulation</i>
<i>Strategic Model Water Quality Extension</i>
<i>Strategic Model Discrete Water Quality Extension</i>
<i>References</i>

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 externally sourced water.

Completions reuse water: Water that meets demand at a completions site. This does not include externally sourced water 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)
 $f \in F$ External water 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

Note: The sets for capacity sizes $d \in D, c \in C, j \in J, i \in I$ should also include the 0th case (e.g., 0 bbl) that indicates the choice to not expand capacity. Alternatively, if it is desired to only consider sizes to build, the 0th case can be left out.

$wt \in WT$ Treatment technologies
 $(p, p) \in PCA$ 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
 $(f, p) \in FCA$ Externally sourced water-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
 $(l, \tilde{l}) \in LLA$ All valid 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
 $(f, p) \in FCT$ Externally sourced water-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
 $(l, \tilde{l}) \in LLT$ All valid 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

$F_{f,p,t}^{Sourced}$ = Externally sourced water to completions

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

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

$F_{s,t}^{StorageEvaporationStream}$ = Water at storage lost to evaporation

$F_{r,t}^{TreatmentFeed}$ = Flow of feed to a treatment site

$F_{r,t}^{ResidualWater}$ = Flow of residual water out of a treatment site

$F_{r,t}^{TreatedWater}$ = Flow of treated water out of a treatment site

$F_{p,t}^{CompletionsReuseDestination}$ = 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 option
- $F_{p,t}^{CompletionsDestination}$ = All water delivered to completions pad
- $L_{s,t}^{Storage}$ = 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 externally sourced water
- $F^{TotalDisposed}$ = Total volume of produced water disposed
- $F^{TotalCompletionsReuse}$ = Total volume of produced water reused at completions
- $F^{TotalBeneficialReuse}$ = Total volume of water beneficially reused
- $C_{l,l,t}^{Piped}$ = 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 external 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)
- $C_{o,t}^{BeneficialReuse}$ = Processing cost of sending water to beneficial reuse
- $R_{s,t}^{Storage}$ = Credit for retrieving stored produced water from storage site
- $R_{o,t}^{BeneficialReuse}$ = Credit for sending water to beneficial reuse
- $C^{TotalSourced}$ = Total cost of externally sourced water
- $C^{TotalDisposal}$ = 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
- $C^{TotalBeneficialReuse}$ = Total processing cost for sending water to beneficial reuse
- C^{Slack} = Total cost of slack variables
- $R^{TotalStorage}$ = Total credit for withdrawing produced water
- $R^{TotalBeneficialReuse}$ = Total credit for sending water to beneficial reuse
- $D_{k,[t]}^{Capacity}$ = 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
- $S_r^{TreatmentCapacity}$ = Slack variable to provide necessary treatment capacity
- $S_o^{BeneficialReuseCapacity}$ = 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
- $y_{r,wt,j,[t]}^{Treatment}$ = New or additional treatment capacity installed at treatment site with specific treatment capacity and treatment technology
- $y_{k,i,[t]}^{Disposal}$ = New or additional disposal facility installed at disposal site with specific injection capacity
- $y_{l,\tilde{l},t}^{Flow}$ = Directional flow between two locations
- $y_{o,t}^{BeneficialReuse}$ = Beneficial reuse option selection

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 disposal site
- $\sigma_s^{Storage}$ = Initial storage capacity at storage site

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

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

$\sigma_{o,t}^{BeneficialReuseMinimum}$ = Minimum flow that must be sent to beneficial reuse option

$\sigma_{o,t}^{BeneficialReuse}$ = Capacity of beneficial reuse option

$\sigma_{f,t}^{ExternalWater}$ = Capacity of externally sourced water

$\sigma_p^{Offloading,Pad}$ = 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

$\epsilon_{r,wt}^{Treatment}$ = Treatment efficiency for technology wt at treatment site

$\epsilon_{r,wt,qc}^{TreatmentRemoval}$ = Removal efficiency for technology wt and quality component qc at treatment site

$\epsilon_{k,t}^{DisposalOperatingCapacity}$ = Operating capacity of disposal site [%]

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

$\chi_p^{OutsideCompletionsPad}$ = Binary parameter designating the completion pads that are outside the system

$\chi_{wt}^{DesalinationTechnology}$ = Binary parameter designating which treatment technologies are for desalination (1) and which are not (0)

$\chi_r^{DesalinationSites}$ = Binary parameter designating which treatment sites are for desalination (1) and which are not (0)

$\chi_k^{DisposalExpansionAllowed}$ = Binary parameter indicating if expansion is allowed at site k

$\omega^{EvaporationRate}$ = Evaporation rate per week

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

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

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

δ^{Truck} = Truck capacity

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

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

$\tau_{r,wt,j}^{Treatment}$ = Treatment construction or expansion lead time

$\tau_{l,l}^{Trucking}$ = 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

$\kappa_{r,wt,j}^{Treatment}$ = 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]

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

$\tau^{Pipeline}$ = Pipeline construction or expansion lead time [time/distance]

Otherwise, if the pipeline cost configuration is capacity based:

$\kappa_{l,i,d}^{Pipeline}$ = 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]

$\tau_{l,i,d}^{Pipeline}$ = Pipeline construction or expansion lead time [time]

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

$\pi_{r,wt}^{Treatment}$ = Treatment operational cost

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

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

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

$\pi_o^{BeneficialReuse}$ = Processing cost for sending water to beneficial reuse

$\rho_o^{BeneficialReuse}$ = Credit for sending water to beneficial reuse

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

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

$\pi_f^{Sourcing}$ = Externally sourced water cost

M^{Flow} = Big-M flow parameter

$M^{Concentration}$ = Big-M concentration parameter

$M^{FlowConcentration}$ = Big-M flow concentration 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

$\psi_{TreatmentCapacity}$ = 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 externally sourced 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.

$$\begin{aligned}
 \min \quad & C^{TotalSourced} + C^{TotalDisposal} + C^{TotalTreatment} \\
 & + C^{TotalCompletionsReuse} + C^{TotalPiping} + C^{TotalStorage} \\
 & + C^{TotalTrucking} + C^{TotalBeneficialReuse} \\
 & + \alpha^{AnnualizationRate} \cdot (C^{DisposalCapEx} + C^{StorageCapEx} \\
 & + C^{TreatmentCapEx} + C^{PipelineCapEx}) + C^{Slack} \\
 & - R^{TotalStorage} - R^{TotalBeneficialReuse} \\
 \max \quad & F^{TotalCompletionsReuse} / \beta^{TotalProd}
 \end{aligned}$$

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: $\forall p \in CP, t \in T$

If the completions pad lies outside the system, the demand is optional. Otherwise, if the completions pad is within the system, completions demand must be met. Demand can be met by trucked or piped water moved into the pad in addition to water in completions pad storage.

If $\chi_p^{OutsideCompletionsPad} = 1$:

$$\begin{aligned}
 Y_{p,t}^{Completions} \geq & \sum_{l \in (L-F) | (l,p) \in LLA} F_{l,p,t}^{Piped} + \sum_{f \in F | (f,p) \in LLA} F_{f,p,t}^{Sourced} + \sum_{l \in L | (l,p) \in LLT} F_{l,p,t}^{Trucked} \\
 & + F_{p,t}^{PadStorageOut} - F_{p,t}^{PadStorageIn} + S_{p,t}^{FracDemand}
 \end{aligned}$$

Else if $\chi_p^{OutsideCompletionsPad} = 0$:

$$\begin{aligned}
 Y_{p,t}^{Completions} = & \sum_{l \in (L-F) | (l,p) \in LLA} F_{l,p,t}^{Piped} + \sum_{f \in F | (f,p) \in LLA} F_{f,p,t}^{Sourced} + \sum_{l \in L | (l,p) \in LLT} F_{l,p,t}^{Trucked} \\
 & + F_{p,t}^{PadStorageOut} - F_{p,t}^{PadStorageIn} + S_{p,t}^{FracDemand}
 \end{aligned}$$

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.

Externally Sourced Water Capacity: $\forall f \in F, t \in T$

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

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

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.

$$\sum_{l \in L | (l,p) \in LLT} F_{l,p,t}^{Trucked} \leq \sigma_p^{Offloading,Pad}$$

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

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

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

Note: The above constraint has not 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.

$$\sum_{l \in L | (l,s) \in LLT} F_{l,s,t}^{Trucked} \leq \sigma_s^{Offloading,Storage}$$

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.

$$\sum_{l \in L | (l,s) \in LLA} F_{l,s,t}^{Piped} + \sum_{l \in L | (l,s) \in LLT} F_{l,s,t}^{Trucked} \leq \sigma_s^{Processing,Storage}$$

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.

$$\beta_{p,t}^{Production} = \sum_{l \in L | (p,l) \in LLA} F_{p,l,t}^{Piped} + \sum_{l \in L | (p,l) \in LLT} F_{p,l,t}^{Trucked} + S_{p,t}^{Production}$$

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.

$$\beta_{p,t}^{Flowback} = \sum_{l \in L | (p,l) \in LLA} F_{p,l,t}^{Piped} + \sum_{l \in L | (p,l) \in LLT} F_{p,l,t}^{Trucked} + S_{p,t}^{Flowback}$$

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.

$$\sum_{l \in L | (l,n) \in LLA} F_{l,n,t}^{Piped} = \sum_{l \in L | (n,l) \in LLA} F_{n,l,t}^{Piped}$$

Bi-Directional Flow: $\forall l \in (L - F - O), \tilde{l} \in (L - F), (l, \tilde{l}) \in LLA, t \in T$

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”.

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

Note: Technically the above 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 determined by the initial storage and storage inputs and outputs. Otherwise, the storage level is determined by the storage level in the

previous time period and storage inputs and outputs. Water outputs include other system nodes (i.e., pipeline nodes and completions pads) and an evaporation stream.

For $t = 1$:

$$L_{s,t}^{Storage} = \lambda_{s,t=1}^{Storage} + \sum_{l \in L | (l,s) \in LLA} F_{l,s,t}^{Piped} + \sum_{l \in L | (l,s) \in LLT} F_{l,s,t}^{Trucked} - \sum_{l \in L | (s,l) \in LLA} F_{s,l,t}^{Piped} - \sum_{l \in L | (s,l) \in LLT} F_{s,l,t}^{Trucked} - F_{s,t}^{StorageEvaporationStream}$$

For $t > 1$:

$$L_{s,t}^{Storage} = L_{s,t-1}^{Storage} + \sum_{l \in L | (l,s) \in LLA} F_{l,s,t}^{Piped} + \sum_{l \in L | (l,s) \in LLT} F_{l,s,t}^{Trucked} - \sum_{l \in L | (s,l) \in LLA} F_{s,l,t}^{Piped} - \sum_{l \in L | (s,l) \in LLT} F_{s,l,t}^{Trucked} - F_{s,t}^{StorageEvaporationStream}$$

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.

$$\sum_{l \in L | (l,n) \in LLA} F_{l,n,t}^{Piped} \leq \sigma_n^{Node}$$

Pipeline Capacity Construction/Expansion:

Sets the flow capacity in a given pipeline during a given time period. The set D should also include the 0th case (e.g. 0 bbl/day) that indicates the choice to not expand capacity. Different constraints apply based on whether a pipeline is allowed to reverse flows at any time. Thus, the following constraint applies to all pipelines that allow reversible flows:

$$\forall (l, \tilde{l}) \in LLA, (\tilde{l}, l) \in LLA, [t \in T]$$

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

The following constraint applies to all pipelines that do not allow reversible flows:

$$\forall (l, \tilde{l}) \in LLA, [t \in T]$$

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

Note: While populating the input data into the spreadsheet for initial pipeline capacities, users must use the following guidelines.

1. For uni-directional pipelines, the initial pipeline capacity must be populated only in the direction of flow else, it will be ignored by the model.
2. For bi-directional pipelines, the initial pipeline capacity should be populated for only one of the allowable flow directions, not both. The pipeline capacities are aggregated for both directions, so the choice of direction for the capacity is irrelevant.

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:

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

$$\forall (l, \tilde{l}) \in LLA, t \in T$$

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

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

The following 2 constraints account 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 set C should also include the 0th case (0 bbl) that indicates the choice to not expand capacity.

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

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

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

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

The following 2 constraints account 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. The set I should also include the 0th case (e.g. 0 bbl/day) that indicates the choice to not expand capacity.

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

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

$$\sum_{l \in L | (l,k) \in LLA} F_{l,k,t}^{Piped} + \sum_{l \in L | (l,k) \in LLT} F_{l,k,t}^{Trucked} \leq D_{k,[t]}^{Capacity}$$

Treatment Capacity Construction/Expansion: $\forall r \in R$

Similarly to disposal and storage capacity construction/expansion constraints, the current treatment capacity can be expanded as required or new facilities may be installed. The set J should also include the 0th case (e.g. 0 bbl/day) that indicates the choice to not expand capacity.

$$\sum_{wt \in WT, j \in J} (\sigma_{r,wt}^{Treatment} \cdot y_{r,wt,j}^{Treatment} + \delta_{wt,j}^{Treatment} \cdot y_{r,wt,j}^{Treatment}) = T_r^{Capacity}$$

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

$$\sum_{l \in L | (l,r) \in LLA} F_{l,r,t}^{Piped} + \sum_{l \in L | (l,r) \in LLT} F_{l,r,t}^{Trucked} \leq T_{r,[t]}^{Capacity}$$

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

At a treatment facility, the inlet raw produced water is combined into a single input treatment feed.

$$\sum_{l \in L | (l,r) \in LLA} F_{l,r,t}^{Piped} + \sum_{l \in L | (l,r) \in LLT} F_{l,r,t}^{Trucked} = F_{r,t}^{TreatmentFeed}$$

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

At a treatment facility, the input treatment feed is treated and separated into treated water and residual water.

$$F_{r,t}^{TreatmentFeed} = F_{r,t}^{ResidualWater} + F_{r,t}^{TreatedWater}$$

Residual Water: $\forall r \in R, wt \in WT, t \in T$

The efficiency of a treatment technology determines the amount of residual water produced.

Residual Water LHS

$$F_{r,t}^{TreatmentFeed} \cdot (1 - \epsilon_{r,wt}^{Treatment}) - M^{Flow} \cdot (1 - \sum_{j \in J} y_{r,wt,j}^{Treatment}) \leq F_{r,t}^{ResidualWater}$$

Residual Water RHS

$$F_{r,t}^{TreatmentFeed} \cdot (1 - \epsilon_{r,wt}^{Treatment}) + M^{Flow} \cdot (1 - \sum_{j \in J} y_{r,wt,j}^{Treatment}) \geq F_{r,t}^{ResidualWater}$$

Treated and Residual Water Balances:

For all piping or trucking arcs (r, l) immediately downstream of a treatment site r , the user must specify whether the arc carries treated water or residual water away from the treatment site. Moreover,

Treated Water Balance: $\forall r \in R$ | there exists at least one arc (r, l) carrying treated water away from r , $t \in T$

$$F_{r,t}^{TreatedWater} = \sum_{l \in L | (r,l) \in LLA \text{ and } (r,l) \text{ carries treated water}} F_{r,l,t}^{Piped} + \sum_{l \in L | (r,l) \in LLT \text{ and } (r,l) \text{ carries treated water}} F_{r,l,t}^{Trucked}$$

Residual Water Balance: $\forall r \in R$ | there exists at least one arc (r, l) carrying residual water away from r , $t \in T$

$$F_{r,t}^{ResidualWater} = \sum_{l \in L | (r,l) \in LLA \text{ and } (r,l) \text{ carries residual water}} F_{r,l,t}^{Piped} + \sum_{l \in L | (r,l) \in LLT \text{ and } (r,l) \text{ carries residual water}} F_{r,l,t}^{Trucked}$$

Note: The user is not required to specify any arcs carrying away treated or residual water immediately downstream of a treatment site. In reality, water that enters a treatment site must eventually leave and go somewhere, but for the sake of modeling flexibility, it is not required to include such arcs. If the user chooses to omit downstream treated and/or residual water arcs for a treatment site, then the treatment site acts as a sink within the greater network model for the water which is not propagated downstream.

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

If a beneficial reuse option is selected ($y_{o,t}^{BeneficialReuse} = 1$), the flow to it must meet the minimum required value.

If $\sigma_{o,t}^{BeneficialReuseMinimum} > 0$:

$$F_{o,t}^{BeneficialReuseDestination} \geq \sigma_{o,t}^{BeneficialReuseMinimum} \cdot y_{o,t}^{BeneficialReuse}$$

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

If a beneficial reuse option is not selected ($y_{o,t}^{BeneficialReuse} = 0$), the flow to it must be zero. Furthermore, the specified capacities of beneficial reuse options must be respected.

It is optional to specify capacities ($\sigma_{o,t}^{BeneficialReuse}$) for reuse options. If a capacity is provided for reuse option o :

$$F_{o,t}^{BeneficialReuseDestination} \leq \sigma_{o,t}^{BeneficialReuse} \cdot y_{o,t}^{BeneficialReuse} + S_o^{BeneficialReuseCapacity}$$

Otherwise:

$$F_{o,t}^{BeneficialReuseDestination} \leq M^{Flow} \cdot y_{o,t}^{BeneficialReuse} + S_o^{BeneficialReuseCapacity}$$

Total Beneficial Reuse Volume:

$$F^{TotalBeneficialReuse} = \sum_{t \in T} \sum_{o \in O} F_{o,t}^{BeneficialReuseDestination}$$

Externally Sourced Water Cost: $\forall f \in F, p \in CP, t \in T$

For each external water source, for each completions pad, and for each time period, the sourcing cost is equal to all output from the source times the 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 Externally Sourced Volume: $\forall f \in F, p \in CP, t \in T$

The total externally sourced volume is the sum of externally sourced water movements by truck and pipeline over all time periods, completions pads, and external water 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.

$$C_{k,t}^{Disposal} = \left(\sum_{l \in L | (l,k) \in LLA} F_{l,k,t}^{Piped} + \sum_{l \in L | (l,k) \in LLT} F_{l,k,t}^{Trucked} \right) \cdot \pi_k^{Disposal}$$

$$C^{TotalDisposal} = \sum_{t \in T} \sum_{k \in K} C_{k,t}^{Disposal}$$

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.

$$F^{TotalDisposed} = \sum_{t \in T} \sum_{k \in K} F_{k,t}^{DisposalDestination}$$

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

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.

$$C_{r,t}^{Treatment} \geq \left(\sum_{l \in L | (l,r) \in LLA} F_{l,r,t}^{Piped} + \sum_{l \in L | (l,r) \in LLT} F_{l,r,t}^{Trucked} - M^{Flow} \cdot \left(1 - \sum_{j \in J} y_{r,wt,j}^{Treatment}\right) \right) \cdot \pi_{r,wt}^{Treatment}$$

$$C_{r,t}^{Treatment} \leq \left(\sum_{l \in L | (l,r) \in LLA} F_{l,r,t}^{Piped} + \sum_{l \in L | (l,r) \in LLT} F_{l,r,t}^{Trucked} + M^{Flow} \cdot \left(1 - \sum_{j \in J} y_{r,wt,j}^{Treatment}\right) \right) \cdot \pi_{r,wt}^{Treatment}$$

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

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

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

$$C_{p,t}^{CompletionsReuse} = \left(\sum_{l \in (L-F) \mid (l,p) \in LLA} F_{l,p,t}^{Piped} + \sum_{l \in (L-F) \mid (l,p) \in LLT} F_{l,p,t}^{Trucked} \right) \cdot \pi_p^{CompletionsReuse}$$

Note: Externally sourced water is excluded from completions reuse costs.

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

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 externally sourced water.

$$F^{TotalCompletionsReused} = \sum_{t \in T} \left(\sum_{l \in (L-F) \mid (l,p) \in LLA} F_{l,p,t}^{Piped} + \sum_{l \in (L-F) \mid (l,p) \in LLT} F_{l,p,t}^{Trucked} \right)$$

Piping Cost: $\forall l \in (L-O-K), \forall \tilde{l} \in (L-F), \forall (l, \tilde{l}) \in LLA, t \in T$

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

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

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

Note: The constraints above explicitly consider piping externally sourced water 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.

$$C_{s,t}^{Storage} = \left(\sum_{l \in L \mid (l,s) \in LLA} F_{l,s,t}^{Piped} + \sum_{l \in L \mid (l,s) \in LLT} F_{l,s,t}^{Trucked} \right) \cdot \pi_s^{Storage}$$

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

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

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

$$R_{s,t}^{Storage} = \left(\sum_{l \in L \mid (s,l) \in LLA} F_{s,l,t}^{Piped} + \sum_{l \in L \mid (s,l) \in LLT} F_{s,l,t}^{Trucked} \right) \cdot \rho_s^{Storage}$$

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

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

Processing cost for sending water to beneficial reuse is equal to the volume of water sent to beneficial reuse multiplied by the beneficial reuse cost rate.

$$C_{o,t}^{BeneficialReuse} = \left(\sum_{l \in L | (l,o) \in LLA} F_{l,o,t}^{Piped} + \sum_{l \in L | (l,o) \in LLT} F_{l,o,t}^{Trucked} \right) \cdot \pi_o^{BeneficialReuse}$$

$$C^{TotalBeneficialReuse} = \sum_{t \in T} \sum_{o \in O} C_{o,t}^{BeneficialReuse}$$

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

Credit for sending water to beneficial reuse is equal to the volume of water sent to beneficial reuse multiplied by the beneficial reuse credit rate.

$$R_{o,t}^{BeneficialReuse} = \left(\sum_{l \in L | (l,o) \in LLA} F_{l,o,t}^{Piped} + \sum_{l \in L | (l,o) \in LLT} F_{l,o,t}^{Trucked} \right) \cdot \rho_o^{BeneficialReuse}$$

$$R^{TotalBeneficialReuse} = \sum_{t \in T} \sum_{o \in O} R_{o,t}^{BeneficialReuse}$$

Trucking Cost (Simplified) $\forall (l, \tilde{l}) \in LLT, [t \in T]$

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.

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

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

Note: The constraints above explicitly consider trucking externally sourced water 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.

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

Disposal Construction or Capacity Expansion Cost:

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.

$$C^{DisposalCapEx} = \sum_{i \in I} \sum_{k \in K} \kappa_{k,i}^{Disposal} \cdot \delta_{k,i}^{Disposal} \cdot y_{k,i}^{Disposal}$$

Storage Construction or Capacity Expansion Cost:

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.

$$C^{StorageCapEx} = \sum_{s \in S} \sum_{c \in C} \kappa_{s,c}^{Storage} \cdot \delta_c^{Storage} \cdot y_{s,c}^{Storage}$$

Treatment Construction or Capacity Expansion Cost:

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.

$$C^{TreatmentCapEx} = \sum_{r \in R} \sum_{j \in J} \sum_{wt \in WT} \kappa_{r,wt,j}^{Treatment} \cdot \delta_{wt,j}^{Treatment} \cdot y_{r,wt,j}^{Treatment}$$

Pipeline Construction or Capacity Expansion Cost:

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.

$$C^{PipelineCapEx} = \sum_{l \in L} \sum_{\tilde{l} \in L} \sum_{d \in D} \kappa_{l,\tilde{l},d}^{Pipeline} \cdot \delta_d^{Pipeline} \cdot y_{l,\tilde{l},d}^{Pipeline}$$

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.

$$C^{PipelineCapEx} = \sum_{l \in L} \sum_{\tilde{l} \in L} \sum_{d \in D} \kappa_{l,\tilde{l},d}^{Pipeline} \cdot \mu_d^{Pipeline} \cdot \lambda_{l,\tilde{l}}^{Pipeline} \cdot y_{l,\tilde{l},d}^{Pipeline}$$

Seismic Response Area - Disposal Operating Capacity Reduction: $\forall k \in K, t \in T$

Seismic Response Areas (SRAs) can reduce the operating capacity at disposal wells. The operating capacity is set by the full built capacity and the max percentage of capacity the disposal site is allowed to use.

$$F_{k,t}^{DisposalDestination} \leq \epsilon_{k,t}^{DisposalOperatingCapacity} \cdot D_k^{Capacity}$$

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).

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

Logic Constraints:

New pipeline or facility capacity constraints: e.g., only one injection capacity can be used for a given site. The sets for capacity sizes should also include the 0th case (e.g., 0 bbl) that indicates the choice to not expand capacity. Alternatively, if it is desired to only consider sizes to build, the 0th case can be left out.

$\forall k \in K$

$$\sum_{i \in I} y_{k,i,t}^{Disposal} = 1$$

$\forall s \in S$

$$\sum_{c \in C} y_{s,c,t}^{Storage} = 1$$

$$\forall (l, \tilde{l}) \in LLA$$

$$\sum_{d \in D} y_{l, \tilde{l}, d, [t]}^{Pipeline} = 1$$

$$\forall r \in R$$

$$\sum_{j \in J, wt \in WT} y_{r, wt, j}^{Treatment} = 1$$

Logic Constraints for Desalination:

Desalination technology is assigned to a pre-determined site.

$$\forall r \in R$$

if $\chi_r^{DesalinationSite}$

$$\sum_{j \in J, wt \in WT | \chi_b^{DesalinationTechnology}} y_{r, wt, j}^{Treatment} = 1$$

Clean brine technology is assigned to a non-desalination site.

$$\forall r \in R$$

if NOT $\chi_r^{DesalinationSite}$

$$\sum_{j \in J, wt \notin WT | \chi_b^{DesalinationTechnology}} y_{r, wt, j}^{Treatment} = 1$$

Evaporation Flow Constraint Evaporation flow for a given time period and storage site is 0 if it is the first time period. Otherwise, evaporation is a constant flow set by the parameter $\omega^{EvaporationRate}$.

For $t = 1$:

$$F_{s, t}^{StorageEvaporationStream} = 0$$

For $t > 1$:

$$F_{s, t}^{StorageEvaporationStream} = \omega^{EvaporationRate} \cdot \sum_{j \in J, r \in R | (r, s) \in RSA} y_{r, 'CB-EV', j}^{Treatment}$$

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 externally sourced water. $\forall p \in CP, t \in T$

$$F_{p, t}^{CompletionsReuseDestination} = \sum_{l \in L | (l, p) \in LLA, l \notin F} F_{l, p, t}^{Piped} + \sum_{l \in L | (l, p) \in LLT, l \notin F} F_{l, p, t}^{Trucked}$$

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

$$F_{k, t}^{DisposalDestination} = \sum_{l \in L | (l, k) \in LLA} F_{l, k, t}^{Piped} + \sum_{l \in L | (l, k) \in LLT} F_{l, k, t}^{Trucked}$$

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

$$F_{o, t}^{BeneficialReuseDestination} = \sum_{l \in L | (l, o) \in LLA} F_{l, o, t}^{Piped} + \sum_{l \in L | (l, o) \in LLT} F_{l, o, t}^{Trucked}$$

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

$$F_{p,t}^{CompletionsDestination} = \sum_{l \in (L-F) | (l,p) \in LLA} F_{l,p,t}^{Piped} + \sum_{f \in F | (f,p) \in FCA} F_{f,p,t}^{Sourced} \\ + \sum_{l \in (L-F) | (l,p) \in LLT} F_{l,p,t}^{Trucked} + F_{p,t}^{PadStorageOut} - F_{p,t}^{PadStorageIn}$$

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

Water Quality Sets

$qc \in QC$ Water Quality Components (e.g., TDS)

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

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

$r^{TreatedWaterNodes} \in R$ Treated Water Nodes

$r^{ResidualWaterNodes} \in R$ Residual Water Nodes

Water Quality Parameters

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

$v_{f,qc,[t]}$ = Water quality of externally sourced water

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

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

Water Quality Variables

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

Disposal Site Water Quality $\forall k \in K, qc \in QC, t \in T$

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

these flows.

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

Storage Site Water Quality $\forall s \in S, qc \in QC, t \in T$

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 $t = 1$:

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

For $t > 1$:

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

Treatment Feed Water Quality $\forall r \in R, qc \in QC, t \in T$

The water quality at treatment sites is dependent on the flow rates and qualities of the feed streams into the treatment site. Even mixing is assumed in calculating the quality of the combined feed stream.

$$\begin{aligned}
 & \sum_{n \in N | (n,r) \in NRA} F_{n,r,t}^{Piped} \cdot Q_{n,qc,t} + \sum_{s \in S | (s,r) \in SRA} F_{s,r,t}^{Piped} \cdot Q_{s,qc,t} \\
 & + \sum_{p \in P | (p,r) \in PRT} F_{p,r,t}^{Trucked} \cdot v_{p,qc,[t]} + \sum_{p \in P | (p,r) \in CRT} F_{p,r,t}^{Trucked} \cdot v_{p,qc,[t]} \\
 & = Q_{r,qc,t} \cdot F_{r,t}^{TreatmentFeed}
 \end{aligned}$$

Treated Water Quality $\forall r \in R, qc \in QC, t \in T$

All treated water from a single treatment site and single time period will have the same water quality. The following constraints allow us to easily track the water quality at treated water end points like desalinated water.

Treated Water Quality General Constraint

$$Q_{r,qc,t} \cdot F_{r,t}^{TreatmentFeed} = Q_{r,TreatedWaterNodes,qc,t} \cdot F_{r,t}^{TreatedWater} + Q_{r,ResidualWaterNodes,qc,t} \cdot F_{r,t}^{ResidualWater}$$

Treated Water Quality Concentration-Based LHS Constraint

$$Q_{r,qc,t} \cdot (1 - \epsilon_{r,wt}^{TreatmentRemoval}) + M^{Concentration} \cdot (1 - \sum_{j \in J} y_{r,wt,j}^{Treatment}) \geq Q_{r,TreatedWaterNodes,qc,t}$$

Treated Water Quality Concentration-Based RHS Constraint

$$Q_{r,qc,t} \cdot (1 - \epsilon_{r,wt}^{TreatmentRemoval}) - M^{Concentration} \cdot (1 - \sum_{j \in J} y_{r,wt,j}^{Treatment}) \leq Q_{r,TreatedWaterNodes,qc,t}$$

Treated Water Quality Load-Based LHS Constraint

$$Q_{r,qc,t} \cdot F_{r,t}^{TreatmentFeed} \cdot (1 - \epsilon_{r,wt}^{TreatmentRemoval}) + M^{FlowConcentration} \cdot (1 - \sum_{j \in J} y_{r,wt,j}^{Treatment}) \geq Q_{r,TreatedWaterNodes,qc,t} \cdot F_{r,t}^{TreatedWater}$$

Treated Water Quality Load-Based RHS Constraint

$$Q_{r,qc,t} \cdot F_{r,t}^{TreatmentFeed} \cdot (1 - \epsilon_{r,wt}^{TreatmentRemoval}) - M^{FlowConcentration} \cdot (1 - \sum_{j \in J} y_{r,wt,j}^{Treatment}) \leq Q_{r,TreatedWaterNodes,qc,t} \cdot F_{r,t}^{TreatedWater}$$

Network Node Water Quality $\forall n \in N, qc \in QC, t \in T$

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.

$$\begin{aligned} & \sum_{p \in P | (p,n) \in PNA} F_{p,n,t}^{Piped} \cdot v_{p,qc,[t]} + \sum_{p \in P | (p,n) \in CNA} F_{p,n,t}^{Piped} \cdot v_{p,qc,[t]} \\ & + \sum_{\tilde{n} \in N | (\tilde{n},n) \in NNA} F_{\tilde{n},n,t}^{Piped} \cdot Q_{\tilde{n},qc,t} + \sum_{s \in S | (s,n) \in SNA} F_{s,n,t}^{Piped} \cdot Q_{s,qc,t} \\ & + \sum_{r \in R | (r,n) \in RNA} F_{r,n,t}^{Piped} \cdot Q_{r,qc,t} \\ & = Q_{n,qc,t} \cdot \left(\sum_{\tilde{n} \in N | (\tilde{n},n) \in NNA} F_{\tilde{n},n,t}^{Piped} + \sum_{p \in P | (n,p) \in NCA} F_{n,p,t}^{Piped} \right. \\ & \quad + \sum_{k \in K | (n,k) \in NKA} F_{n,k,t}^{Piped} + \sum_{r \in R | (n,r) \in NRA} F_{n,r,t}^{Piped} \\ & \quad \left. + \sum_{s \in S | (n,s) \in NSA} F_{n,s,t}^{Piped} + \sum_{o \in O | (n,o) \in NOA} F_{n,o,t}^{Piped} \right) \end{aligned}$$

Completions Pad Intermediate Node Water Quality $\forall p \in P, qc \in QC, t \in T$

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.

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

Completions Pad Input Node Water Quality $\forall p \in P, qc \in QC, t \in T$

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.

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

Completions Pad Storage Node Water Quality $\forall p \in P, qc \in QC, t \in T$

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 $t = 1$:

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

For $t > 1$:

$$\begin{aligned}
 & L_{s,t-1}^{PadStorage} \cdot Q_{p,PadStorage,qc,t-1} + F_{p,t}^{PadStorageIn} \cdot Q_{p,IntermediateNode,qc} \\
 & = Q_{p,PadStorage,qc,t} \cdot (L_{s,t}^{PadStorage} + F_{p,t}^{PadStorageOut})
 \end{aligned}$$

Beneficial Reuse Water Quality $\forall o \in O, qc \in QC, t \in T$

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

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

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:

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

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

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

Discrete Water Quality Variables

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

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

Only One Discrete Quality Per Location $\forall l$ in L , t in T , qc in QC

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

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

Discrete Max Disposal Destination $\forall l$ in L , t in T , qc in QC , 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.

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

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

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.

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

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

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

Rewriting the whole constraints goes as follows:

Disposal Site Water Quality $\forall k$ in K , qc in QC , 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.

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

Can be rewritten as

Discrete Disposal Site Water Quality $\forall k$ in K, qc in QC, 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.

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

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

Note: The = sign in the original constraint is changed to \leq 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.

Water Treatment

Note: The treatment model discussed in this section primarily applies to the strategic model within PARETO. The content presented here is therefore focused on its relevance to the strategic model. Although some concepts might be applicable to PARETO as a whole, it is essential to note that component removal efficiency has not been implemented for the operational model.

Overview

Treatment systems play a crucial role for achieving desired water quality for various purposes, such as recycling for hydraulic fracturing, beneficial reuse applications, and critical mineral recovery. Depending on the purpose and degree of treatment, the costs associated with treatment systems can be significant and greatly impact the investment cost in a management option. This makes it necessary to carefully consider the treatment models and their costs when evaluating produced water management strategies. Therefore, it is essential to integrate treatment models into the PARETO decision-making process. This will enable stakeholders to better understand the trade-offs between different management options and their associated costs, ultimately leading to more informed decisions.

Section
<i>Treatment model within PARETO network</i>
<i>Treatment Model Description</i>
<i>Treatment Efficiency (water recovery)</i>
<i>Treatment Component Removal Efficiency</i>
<i>Treatment Cost</i>
<i>Mechanical vapor compression model</i>

Treatment model within PARETO network

The PARETO network identifies treatment plants based on their location ($r \in R$), capacity ($j \in J$), and technology ($wt \in WT$). The streams that are piped or trucked to treatment plants are represented by arcs $((l, r) \in LRA \cup LRT)$, where l can be any location or node in PARETO network. The indices j and wt are employed in conjunction with a binary variable to install or expand a treatment plant with a specific capacity (for further details, please refer to *strategic water management*).

The following equation describes the flow balance at location r :

$$\sum_{l \in L | (l, r) \in LRA \cup LRT} F_{l, r, t} = F_{r, t}^{feed}$$

$$\sum_{l \in L | (l, r) \in LRA \cup LRT} F_{l, r, t} \cdot Q_{l, qc, t} = Q_{r, qc, t}^{feed} \cdot F_{r, t}^{feed}$$

$$LRA = \{(l, r) \in LLA \mid r \in R\}$$

$$LRT = \{(l, r) \in LLT \mid r \in R\}$$

where F and Q denotes the flow and quality (concentrations) of streams. The units of concentration are typically reported as mass/volume (mg/L, g/m³, kg/L, etc.) and the units of flow rate are reported in volume/time (e.g. bbl/week).

Treatment Model Description

Water treatment systems are modeled using overall water and constituent balances, treatment and removal efficiencies, and operating cost and capital cost values/equations. The schematic in Figure 1 depicts a treatment unit that processes a treatment feed with specific qualities, yielding two output streams: treated water and residual water. The treated water and residual water streams have distinct qualities, which vary depending on the specific treatment process employed. The overall water and constituent balance equations for water treatment systems are as follows:

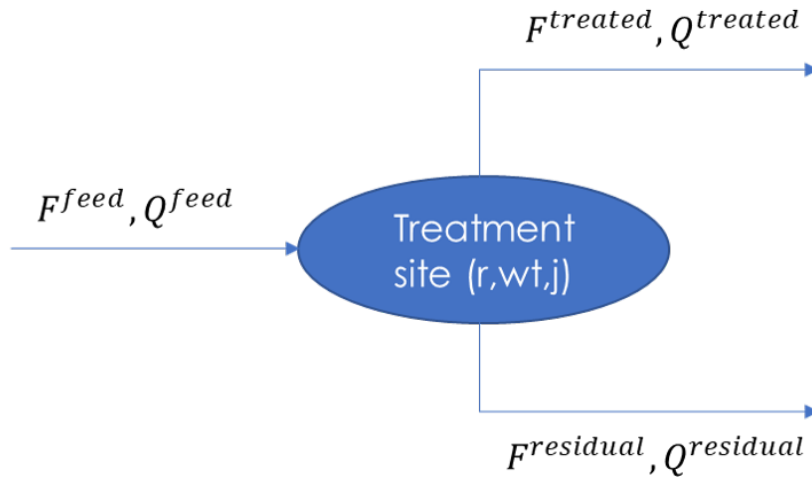


Fig. 4: Figure 1. Treatment plant schematic

- Overall water balance:

$$F^{feed} = F^{treated} + F^{residual}$$

- Overall constituent balance:

$$F^{feed} Q^{feed} = F^{treated} Q^{treated} + F^{residual} Q^{residual}$$

Treatment Efficiency (water recovery)

Treatment efficiency is defined as the ratio of the treated water volume to the ratio of the feed water volume to the treatment plant as follows:

$$\text{Treatment efficiency} = \frac{F^{treated}}{F^{feed}}$$

Note that treatment efficiency can also be expressed as a percentage by multiplying the above expression by 100.

Treatment Component Removal Efficiency

Removal efficiency is a measure of the overall reduction in the concentration or load of a constituent in a treatment plant, expressed as a percentage¹. The removal efficiency of a certain constituent is commonly calculated based on the influent (feed) concentration and the effluent (treated water) concentration as follows:

PARETO supports both formulations and gives the user the option to choose between the two methods based on their available data or the technology considered. The two options are expressed as `RemovalEfficiencyMethod.Concentration_based` and `RemovalEfficiencyMethod.Load_based` in PARETO configuration argument for removal efficiency.

Treatment Cost

The total cost of produced water treatment consist of capital costs and annual operating costs. Capital costs include the costs associated with the land purchase, construction, purchasing process equipment, and installation. Annual operating costs refer to the cost during plant operation such as cost of energy, equipment replacement, chemicals, labor, and maintenance. The sum of the unit operating costs and the unit annualized capital costs determines the total capital cost per unit volume of produced water.

Treatment costs can be incorporated into PARETO with three methods:

- 1) To begin, users can provide their own estimated capital and operating costs for each treatment technology. PARETO provides a treatment technology matrix (shown below) with data collected from available literature on various technologies such as membrane distillation, multi-effect distillation, mechanical vapor compression, and osmotically assisted reverse osmosis (for further detail regarding selected technologies and references please refer to the provided sheet: `treatment matrix`). The technologies considered in this matrix are capable of treating hypersaline produced water up to saturation limits. Users may use these values to evaluate treatment options using PARETO. However, we encourage users to provide their own cost data, obtained from treatment technology vendors, to enable better evaluation of management options. It is important to note that currently, PARETO incorporates treatment costs for discrete values of treatment capacity expansions. In other words, the treatment cost calculations are limited to specific capacity levels.

¹ Von Sperling, M., Verbyla, M. E., & Oliveira, S. M. (2020). Assessment of treatment plant performance and water quality data: a guide for students, researchers and practitioners. IWA publishing.

Treatment Technology	Pre-treatment	Multiefficient evaporation	Mechanical vapor compression (MVC)	Direct contact membrane distillation (DCM)	Air gap membrane distillation (AGMD)	Permeate gap membrane distillation (PGMD)	Conductive gap membrane distillation (CGMD)	Sweeping gas membrane distillation (SGMD)	Vacuum membrane distillation (VMD)	Osmotically assisted reverse osmosis (OARO)	Cascading osmotically mediated reverse osmosis (COMF)	Low-salt rejection reverse osmosis (LSRF)	Brine reflux OARO (BR-OARO)	Split feed counter-flow reverse osmosis (SF-OARO)	Consecutive loop OARO (CL-OARO)
CAPEX [\$ / (bbl feed/c)	60 - 90	726	1092	363-1148	511-589	534-749	461-645	1339	314-689	448-1432	1301	965	1389	1777	2181
OPEX [\$ / bbl feed]	0.04 - 1.50	1.25	0.34	0.61-1.51	0.43-0.62	1.28-3.80	0.53-1.15	1.27	0.45-1.77	0.066-0.32	0.47	0.43	0.51	0.55	0.68
total annualized cost [\$ / bbl feed]	0.07 - 1.40	1.57	0.82	0.79-1.83	0.56-0.73	1.44-3.92	0.67-1.25	1.56	0.60-1.84	0.12-0.54	0.83	0.7	0.82	0.94	1.15
Plant capacity [bbl feed/day]	3774	5661	5661	5079	5079	5079	5079	5079	5079	2944	2944	2944	5079	5079	5079
TDS operating limits [mg/l]	N/A	0-350,0	0-350,0	0-350,00	0-350,0	0-350,00	0-350,00	0-350,00	0-350,0	0-350,00	0-350,000	0-350,00	0-350,0	0-350,00	0-350,000
Energy type	Variable	Thermal	Electrical	Thermal	Thermal	Thermal	Thermal	Thermal	Thermal	Electrical	Electrical	Electrical	Electrical	Electrical	Electrical
Theoretical energy requirements [kWh]	Variable	200 kWh	20-30	182-359 kWh	117-167 kWh	395-1214 kWh	164-354 kWh	364 kWh	130-640 kWh	8-20	12.8	28.9	16.13	17.46	26.6

- 2) An alternative approach to incorporating treatment costs in PARETO is through the use of surrogate models. These models allow for linear or nonlinear approximations of treatment costs as a function of treatment capacity, feed quality, and recovery. This method is currently under development and not yet available in the current version of PARETO, and it is planned for inclusion in future updates.
- 3) The third method for incorporating treatment costs into PARETO is through the integration of rigorous technoeconomic optimization treatment models. These models allow for accurate capture of the effect of changes in input parameters on treatment plant performance and costs. Currently, a technoeconomic optimization-based modeling approach for single effect and multi-effect mechanical vapor compression (MVC) desalination systems is being developed for integration with PARETO. The following section will provide a detailed description of the MVC modeling effort.

Mechanical vapor compression model

Single effect evaporation and multi-effect evaporation has been studied for shale water desalination. Mechanical vapor compression, uses a compressor to utilize the heat from the evaporated vapor for further evaporation. As shown in the schematic in Figure (2), for a system with I effects, the produced water is fed into evaporator I . After evaporation, the brine from the i^{th} effect is sent to the $(i-1)^{\text{th}}$ effect and the vapor from the $(i-1)^{\text{th}}$ effect is sent to the i^{th} effect.

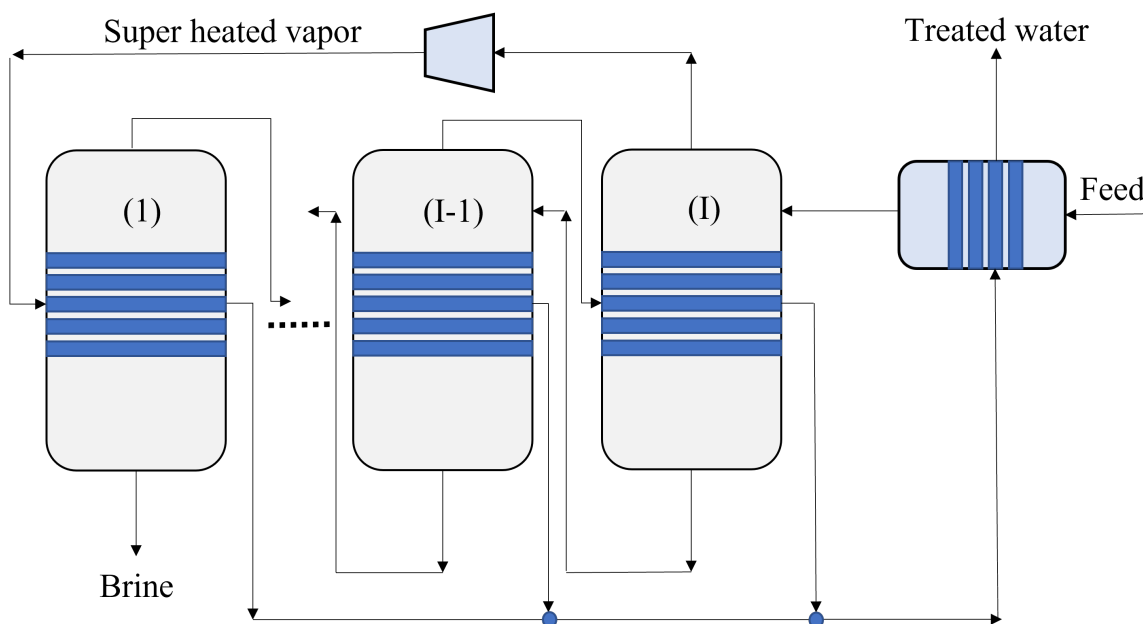


Fig. 5: Figure 2. MEE-SVC treatment flowsheet

The vapor from the i^{th} evaporator is sent to the compressor for compression. The superheated vapor from the compressor is then sent into the tubes of the 1^{st} evaporator to carry out the evaporation process. The condensate from all the evaporator effects is sent to the preheater where it preheats the feed and thus aids in heat integration.

Model Description

The multi-effect evaporator model is built to consider multiple evaporator effects. The user can specify the number of effects, feed flow rate, TDS concentration in feed and the minimum TDS specification in the brine. The model then calculates the capital costs, operating costs, compressor work, compressor capacity, evaporator heat exchange area and the preheater area. The user can also obtain the pressures, temperatures and concentrations of the individual streams. The model is built in Pyomo and is based on equations Onishi et al.'s study² on shale gas flowback water desalination.

Variable Definitions

Table 1: Variable definitions

Symbol	Doc	Units	Index sets
F_{in}	Flow rate of the feed	kg/s	None
F_{brine}	Flow rate of brine	kg/s	i
F_{vapor}	Flow rate of the vapor	kg/s	i
F_{spv}	Flow rate of super heated vapor	kg/s	None
$F_{externalwater}$	Flow rate of externally sourced water	kg/s	None
T_{feed}	Temperature of feed water	°C	None
T_{in}	Temperature of water entering the I^{th}	°C	None
T_{cond}	Temperature of condensate	°C	i
T_{brine}	Temperature of brine	°C	i
T_{vapor}	Temperature of vapor from the evaporators	°C	i
T_{spv}	Temperature of super heated vapor	°C	None
T_{ideal}	Ideal temperature in th evaporators	°C	i
T_{sv}	Temperature of saturated vapor	°C	i
T_{mix}^{out}	Temperature of mixer outlet	°C	None
$T_{externalwater}$	Temperature of externally sourced water	°C	None
BPE	Boiling point elevation	°C	i
H_{feed}	Enthalpy of the feed water	kJ/kg	None
H_{in}	Enthalpy of the water entering the I^{th} evaporator	kJ/kg	None
H_{brine}	Enthalpy of brine	kJ/kg	i
H_{vapor}	Enthalpy of vapor	kJ/kg	i
H_{spv}	Enthalpy of super heated vapor	kJ/kg	None
H_{cond}^{vap}	Enthalpy of condensate vapor	kJ/kg	None
H_{cond}	Enthalpy of condensate	kJ/kg	None
P_{vapor}	Vapor pressure in the evaporator	kPa	i
P_{sv}	Saturated vapor pressure	kPa	i
P_{spv}	Pressure of the super heated vapor	kPa	None
Q_{in}	TDS concentration of the feed	g/kg	None
Q_{brine}	TDS concentration in the brine	g/kg	i
Q_{spec}	TDS specification in the outlet brine stream	g/kg	None
X_s^{in}	Mass fraction of TDS in feed	(-)	None
X_s^{brine}	Mass fraction of TDS in brine	(-)	i
E	Heat flow in the evaporator	kW	i
W_{compr}	Work done by the compressor	kW	None

continues on next page

² Onishi, V. C., Carrero-Parreño, A., Reyes-Labarta, J. A., Ruiz-Femenia, R., Salcedo-Díaz, R., Fraga, E. S., & Caballero, J. A. (2017). Shale gas flowback water desalination: Single vs multiple-effect evaporation with vapor recompression cycle and thermal integration. *Desalination*, 404, 230-248.

Table 1 – continued from previous page

Symbol	Doc	Units	Index sets
C_{compr}	Capacity of the compressor	HP	None
η	Isentropic efficiency of the compressor	(-)	None
U_{evap}	Overall heat transfer coefficient of the evaporator (latent heat)	$\frac{kW}{m^2K}$	i
U_s	Overall heat transfer coefficient of the evaporator (Sensible heat)	$\frac{kW}{m^2K}$	None
U_{ph}	Overall heat transfer coefficient of the preheater	$\frac{kW}{m^2K}$	None
A_{evap}	Area of the evaporator	m^2	i
A_{ph}	Area of the preheater	m^2	None
C_p^{feed}	Specific heat capacity of the feed water	$kJ/(kg^\circ C)$	None
C_p^{vapor}	Specific heat capacity of vapor	$kJ/(kg^\circ C)$	None
C_p^{mix}	Specific heat capacity of water from the outlet of the mixer	$kJ/(kg^\circ C)$	None
ΔT	Approach temperatures	$^\circ C$	None
ΔP	Minimum pressure difference between evaporator stages	kPa	None
C_{elec}	Cost of electricity	$kUSD/kW year$	None
$CAPEX_{evap}$	Capital cost for all evaporators	$kUSD$	None
$CAPEX_{ph}$	Capital cost for Preheater	$kUSD$	None
$CAPEX_{compr}$	Capital cost for compressor	$kUSD$	None
$OPEX_{ann}$	Annualized operating cost	$kUSD/year$	None
$CAPEX_{ann}$	Annualized capital cost	$kUSD/year$	None

Modeling Equations

Evaporator model

Flow balance in the evaporators:

$$F_{in} = F_{brine}^{(I)} + F_{vapor}^{(I)}$$

$$F_{in}^{(i+1)} = F_{brine}^{(i)} + F_{vapor}^{(i)} \quad \forall i < I$$

Flow matching between super heated vapor and flow of vapor from the evaporator:

$$F_{spv} = F_{vapor}^{(I)}$$

Mass balance in the evaporators:

$$F_{in} Q_{in} = F_{brine}^{(I)} Q_{brine}^{(I)}$$

$$F_{brine}^{(i+1)} Q_{brine}^{(i+1)} = F_{brine}^{(i)} Q_{brine}^{(i)} \quad \forall i < I$$

Calculating mass fraction of salt from salt salinity:

$$X_s^{in} = 0.001 Q_{in}$$

$$X_s^{(i)} = 0.001 Q_{brine}^{(i)} \quad \forall i \in \{0, \dots, I\}$$

Energy balance in the evaporator:

$$\begin{aligned}
 E^{(I)} + F_{in}H_{in} &= F_{brine}^{(I)}H_{brine}^{(I)} + F_{vapor}^{(I)}H_{vapor}^{(I)} \\
 E^{(i)} + F_{brine}^{(i+1)}H_{brine}^{(i+1)} &= F_{brine}^{(i)}H_{brine}^{(i)} + F_{vapor}^{(i)}H_{vapor}^{(i)} \quad \forall i < I \\
 E^{(1)} &= F_{spv}C_p^{vapor}(T_{spv} - T_{cond}^{(1)}) + F_{spv}(H_{cond}^{vap(1)} - H_{cond}^{(1)}) \\
 E^{(i)} &= F_{vapor}^{(i-1)}\lambda^{(i)} \quad \forall i \in \{2, \dots, I\}
 \end{aligned}$$

Thermodynamic Relations

Relating pressures to temperatures using the Antoine equation:

$$\begin{aligned}
 \log(P_{vapor}^{(i)}) &= a + \frac{b}{T_{ideal}^{(i)} + c} \quad \forall i \in \{1, \dots, I\} \\
 \log(P_{spv}^{(1)}) &= a + \frac{b}{T_{cond}^{(1)} + c} \\
 \log(P_{sv}^{(i)}) &= a + \frac{b}{T_{sv}^{(i)} + c} \quad \forall i \in \{2, \dots, I\}
 \end{aligned}$$

Calculating elevation in boiling point due to TDS in the feed water:

$$BPE^{(i)} = 0.1581 + 2.769X_s^{(i)} - 0.002676T_{ideal}^{(i)} + 41.78X_s^{(i)2} + 0.134X_s^{(i)}T_{ideal}^{(i)}$$

Calculating temperature of brine from BPE and ideal temperature in the evaporator:

$$T_{brine}^{(i)} = T_{ideal}^{(i)} + BPE^{(i)}$$

Estimating the enthalpies:

$$\begin{aligned}
 H_{in} &= -15940 + 8787X_s^{in} + 3.557T_{in} \\
 H_{brine}^{(i)} &= -15940 + 8787X_s^{(i)} + 3.557T_{brine}^{(i)} \quad \forall i \in \{1, \dots, I\} \\
 H_{vapor}^{(i)} &= -13470 + 1.84T_{brine}^{(i)} \quad \forall i \in \{1, \dots, I\} \\
 H_{cond}^{(i) vap} &= -13470 + 1.84T_{cond}^{(i)} \quad \forall i \in \{1, \dots, I\} \\
 H_{cond}^{(i)} &= -15940 + 3.557T_{cond}^{(i)} \quad \forall i \in \{1, \dots, I\}
 \end{aligned}$$

Calculating LMTD:

$$\begin{aligned}
 \theta_1^{(i)} &= T_{spv} - T_{brine}^{(i)} \quad \text{for } i = 1 \\
 \theta_1^{(i)} &= T_{sv}^{(i)} - T_{brine}^{(i)} \quad \forall i > 1 \\
 \theta_2^{(i)} &= T_{cond}^{(i)} - T_{brine}^{(i+1)} \quad \text{for } i = 1 \\
 \theta_2^{(i)} &= T_{sv}^{(i)} - T_{brine}^{(i+1)} \quad \forall i \in \{2, \dots, I-1\} \\
 \theta_2^{(i)} &= T_{sv}^{(i)} - T_{in} \quad \text{for } i = I \\
 LMTD^{(i)} &= (0.5\theta_1^{(i)}\theta_2^{(i)}(\theta_1^{(i)} + \theta_2^{(i)}))^{1/3}
 \end{aligned}$$

Heat transfer coefficient for evaporator:

$$U_{evap}^i = 0.001(1939.4 + 1.40562T_{brine}^{(i)} - 0.002T_{brine}^{(i)2} + 0.0023T_{brine}^{(i)3})$$

Design Equations

Area of first evaporator calculation:

$$A_{evap}^{(1)} = F_{spv} C_p^{vapor} \frac{(T_{spv} - T_{cond}^{(1)})}{U_s(LMTD^{(1)})} + F_{spv} \frac{H_{cond}^{(1)vap} - H_{cond}^{(1)}}{U_{evap}^{(1)}(T_{cond}^{(1)} - T_{brine}^{(1)})}$$

Total Evaporator Area:

$$A_{evap}^{total} = \sum_{i=1}^I \frac{E^{(i)}}{U_{evap}^{(i)} LMTD^{(i)}}$$

Compressor Model

Thermodynamic Relations

Isentropic temperature calculation:

$$T_{is} = (T_{brine}^{(I)} + 273.5) \left(\frac{P_{spv}}{P_{vapor}^I} \right)^{\frac{\gamma-1}{\gamma}} - 273.5$$

Temperature of the super heated vapor can be calculated as:

$$T_{spv} = T_{brine}^{(I)} + \frac{1}{\eta} (T_{is} - T_{brine}^{(I)})$$

The enthalpy of the super heated vapor can be estimated by:

$$H_{spv} = -13470 + 1.84 T_{spv}$$

Design Equations

The compression work is given by:

$$W_{compr} = F_{spv} (H_{spv} - H_{vap}^{(I)})$$

The compressor capacity in horse power is given by:

$$\mathcal{C}_{compr} = W_{compr} \times 1.34$$

Mixer Model

Mass balance in the mixer:

$$F_{externalwater} = \sum_{i=1}^I F_{vapor}^i$$

Energy balance in the mixer:

$$T_{mix}^{out} = F_{vapor}^I T_{cond}^1 + \sum_{i=1}^I F_{vapor}^{(i-1)} T_{cond}^i$$

Preheater Model

Energy balance in the preheater:

$$F_{externalwater}C_p^{mix}(T_{mix}^{out} - T_{externalwater}) = F_{in}C_p^{feed}(T_{in} - T_{feed})$$

Thermodynamic Relations

Estimating specific heat capacities:

$$C_p^{feed} = 0.001(4206.8 - 6.6197Q_s^{in} + 1.22 \times 10^{-2}Q_s^{in^2} + (-1.262 + 5.418 \times 10^{-2}Q_s^{in})T_{feed})$$
$$C_p^{mix} = 0.001(4206.8 - 1.1262T_{mix}^{out})$$

Preheater heat transfer coefficient calculation:

$$U_{ph} = 0.001(1939.4 + 1.40562T_{mix}^{out} - 0.002T_{mix}^{out^2} + 0.0023T_{mix}^{out^3})$$

Preheater LMTD calculation:

$$\theta_{1ph} = T_{mix}^{out} - T_{in}$$
$$\theta_{2ph} = T_{externalwater} - T_{feed}$$
$$LMTD_{ph} = (0.5\theta_{1ph}\theta_{2ph}(\theta_{1ph} + \theta_{2ph}))^{1/3}$$

Design Equations

Preheater area calculation:

$$A_{ph} = \frac{F_{externalwater}C_p^{mix}(T_{mix}^{out} - T_{externalwater})}{U_{ph}LMTD_{ph}}$$

Bounds for feasible operation:

$$T_{spv} \geq T_{cond}^{(1)} + \Delta T_1^{min}$$
$$T_{brine}^{(I)} \geq T_{in} + \Delta T_2^{min}$$
$$T_{cond}^{(I)} \geq T_{in} + \Delta T^{min}$$
$$T_{brine}^{(i-1)} \geq T_{cond}^{(i)} + \Delta T_1^{min} \quad \forall i \in \{2, \dots, I\}$$
$$T_{brine}^{(i-1)} \geq T_{brine}^{(i)} + \Delta T_{stage}^{min} \quad \forall i \in \{2, \dots, I\}$$
$$T_{cond}^{(i)} \geq T_{brine}^{(i+1)} + \Delta T^{min} \quad \forall i \in \{1, \dots, I-1\}$$
$$T_{cond}^{(i)} \geq T_{brine}^{(i)} + \Delta T^{min} \quad \forall i \in \{1, \dots, I\}$$
$$T_{sv}^{(i)} \geq T_{brine}^{(i)} + \Delta T^{min} \quad \forall i \in \{1, \dots, I\}$$
$$P_{vapor}^{(i)} \geq P_{vapor}^{i+1} + \Delta P^{min} \quad \forall i \in \{1, \dots, I-1\}$$
$$CR_{max}P_{vapor}^{(I)} \geq P_{spv} \geq P_{vapor}^{(I)}$$
$$S_{brine}^{(1)} \geq S_{spec}$$

Objective function

The goal is to minimize the total annualized cost (TAC) of the treatment unit. CAPEX of the equipments were calculated using empirical relations from Couper et. al. Assuming the evaporator is a falling film evaporator made of nickel steel to avoid corrosion, the annualized CAPEX in kUSD is given by:

$$CAPEX_{evap} = \frac{CEPCI_{2022}}{CEPCI_{base}} \times f_m \times 1.218 \sum_{i=1}^I \exp \left[3.2362 - 0.0126 \log(\bar{A}_{evap}^{(i)} \times 10.64) + 0.0244 \log(\bar{A}_{evap}^{(i)} \times 10.64)^2 \right]$$

CAPEX of centrifugal compressor made of carbon steel in kUSD is given by:

$$CAPEX_{compr} = \frac{CEPCI_{2022}}{CEPCI_{base}} \sum_{i=1}^{N_{compr}} 7.9 \mathcal{C}_{compr}^{0.62}$$

Assuming the preheater is a U-tube heat exchanger, the CAPEX of the preheater is given by:

$$CAPEX_{ph} = \frac{CEPCI_{2022}}{CEPCI_{base}} \frac{1.218}{1000} \left(\exp \left[-0.9816 + 0.0830 \log(\bar{A}^{ph} \times 10.64) \right] \times \exp \left[8.821 - 0.308 \log(\bar{A}^{ph} \times 10.64) + 0.0681 \log(\bar{A}^{ph} \times 10.64)^2 \right] \right)$$

Note: For CAPEX calculation, the areas have to be in sq. ft.

The total CAPEX is given by:

$$CAPEX = CAPEX_{evap} + CAPEX_{compr} + CAPEX_{ph}$$

Annualization factor: The annualization factor for CAPEX is calculated based on fractional interest rate $r = 0.1$ per year and amortization period $y = 10$ years.

$$fac = \frac{r(1+r)^y}{(1+r)^y - 1}$$

The annualized CAPEX is calculated by:

$$CAPEX_{ann} = fac \times CAPEX$$

The cost of operating the treatment unit comes from operating the compressor using electricity.

$$OPEX_{ann} = C_{elec} \times W_{compr}$$

The total annualized cost is therefore given by:

$$TAC = CAPEX_{ann} + OPEX_{ann}$$

This is our objective function which we'll minimize.

Sensitivity Analysis

To demonstrate the effect of the feed salinity on the TAC, we consider a single effect evaporator without heat integration using a preheater. The feed flow rate is fixed to 10 kg/s and the outlet brine TDS concentration needs to be above 250 g/kg. The salt concentration in the feed is varied from 70 g/kg to 190 g/kg. A plot of feed salinity vs TAC is generated as shown in Figure 3:

For the same conditions, the sensitivity analysis for a multi-effect evaporator with two stages and heat integration using a preheater is shown in Figure 4:

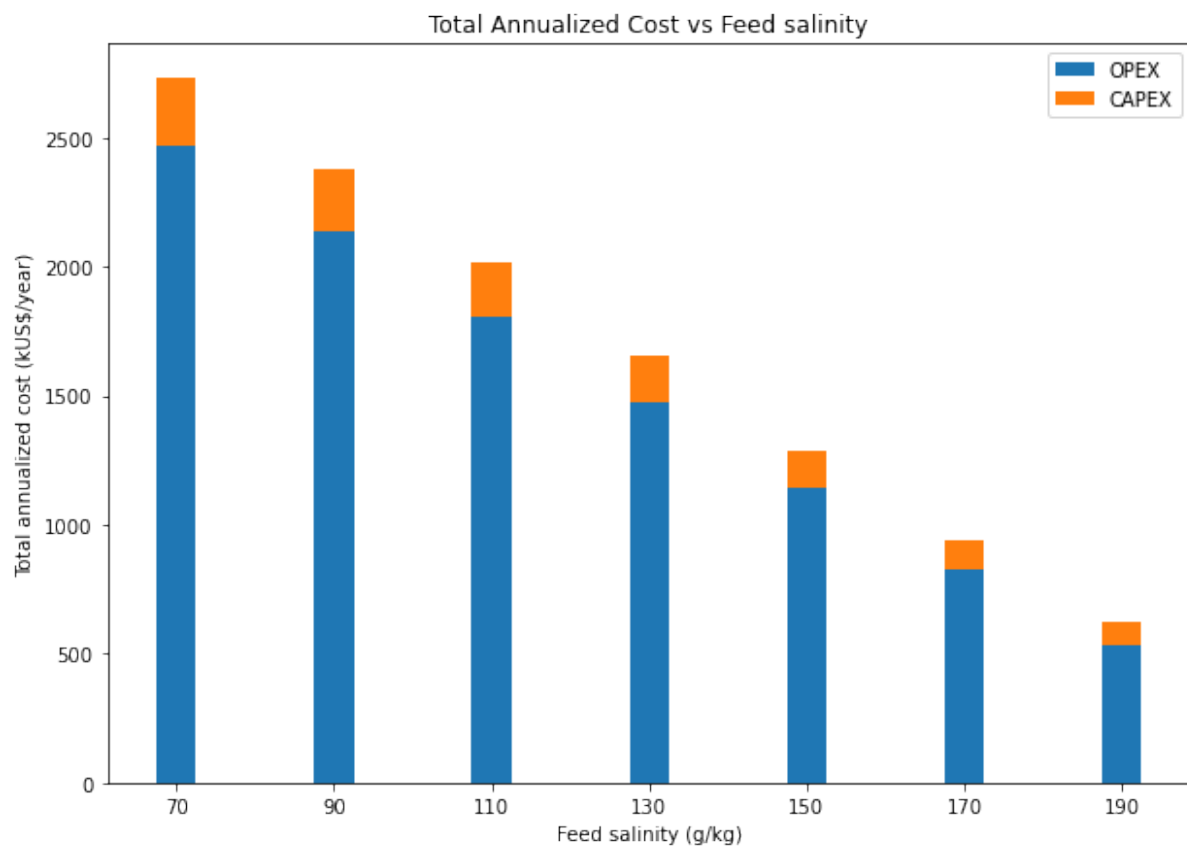


Fig. 6: Figure 3. TAC vs feed salinity for a single effect evaporator

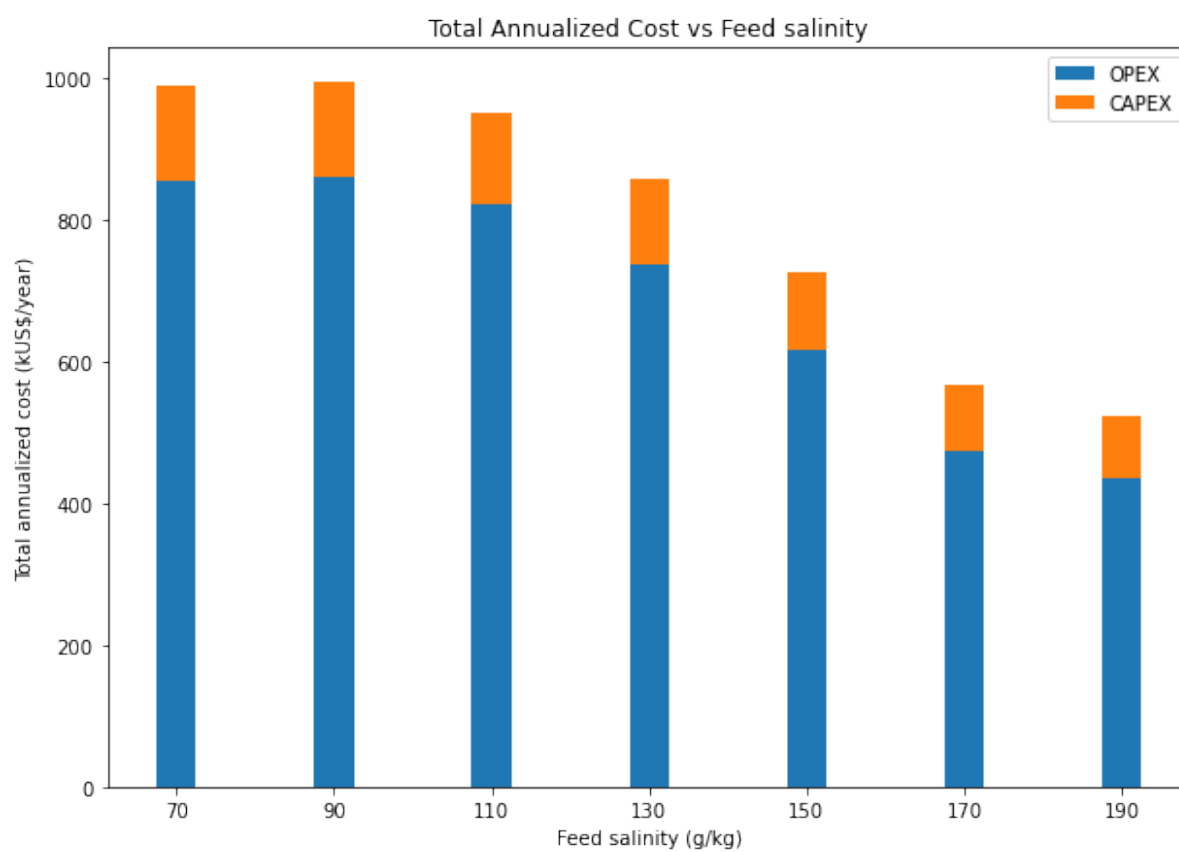


Fig. 7: Figure 4. TAC vs feed salinity for a two effect evaporator with heat integration

References

Pipeline Hydraulics

Note: The hydraulics module presented below is currently only added as an extension to the strategic model.

Overview

Produced water (PW) network operations are highly affected by the pipeline designs and geographical terrain because of pressure losses due to friction and elevation changes. PARETO hydraulics module is an extension to the strategic model that allows the user to compute pressures at every node in the network enabling the user to add operational constraints such as maximum allowable operating pressures (MAOP) and minimum required pressure at injection facilities or at 3rd party offtake points. For this, the PARETO model explicitly considers pipeline details (i.e., length, diameter, material, etc.) and geographic elevations in the model to compute time varying pressures at each node. The hydraulics module is an extension to the PARETO strategic model and can be accessed through the following options in the config argument:

- a) `Hydraulics.false`: This option allows the user to skip the hydraulics computations in the PARETO model.
- b) `Hydraulics.post_process`: In this method, the basic PARETO strategic model is solved as step 1, and then using the optimal flows and network design, the hydraulics block containing constraints for pressure balances and losses is solved as step 2. In this case, as only the hydraulics model block solved for the objective of minimizing cost, the optimal values for variables included in the main strategic model and obtained from step 1 remain unaffected.
- c) `Hydraulics.co_optimize`: In this method, the hydraulics model block is solved together with the strategic model. However, as the flow and diameter are variables in the strategic model, the addition of hydraulics block makes the model a mixed integer nonlinear programming (MINLP) model. In order to solve this MINLP model, the strategic model without the hydraulics constraints is solved as step 1 to determine a good initial state for all variables and constraints.
- d) `Hydraulics.co_optimize_linearized`: This is a linearized approximated model for the `co_optimize` method designed to give solutions quickly in a realistic time frame.

Note: The MINLP as currently implemented requires the following MINLP solvers: SCIP and BARON. The model is first solved using BARON (if available) to determine a feasible solution and then using SCIP. Some subtle differences in model components such as in the definition of variables and parameters have been made to avoid nonlinearities and allow the user to use the same solver for solving the post-process method as used for the strategic model. These differences will be shown in the description of mathematical notation and formulation below.

Section
<i>Hydraulics Model Mathematical Notation</i>
<i>Hydraulics Model Formulation</i>

Hydraulics Model Mathematical Notation

Similar to the strategic model, following color coding has been implemented in describing the model notation. All input *parameters* are in *green* and all model *variables* are in *red*.

Parameters

ζ_l = Geographical elevation of a Node

$\rho \cdot g$ = Product of water density and acceleration due to gravity

k^{CHW} = Constant for pipeline material in Hazen-Williams equation

v^{Pump} = Fixed cost of installing a pump

$v^{Electricity}$ = Electricity price

η^{Pump} = Efficiency of the pump

η^{Motor} = Efficiency of the motor for pump

$\xi^{Minimum}$ = Minimum allowable operating pressure in a pipeline

$\xi^{Maximum}$ = Maximum allowable operating pressure in a pipeline (MAOP)

$D_{l,i}^{Existing}$ = Diameter of an existing pipeline

$v_{p,t}^{Maximum}$ = Well pressure at production or completions pad

Binary Variables

$y_{l,i,t}^{Pump}$ = New pump installation, 1 if a new pump is installed, 0 otherwise

Continuous Variables

$H_{l,i,t}^{Pump}$ = Pump head added in the direction of flow

$H_{l,i,t}^{Valve}$ = Valve head removed in the direction of flow

$C_{l,i}^{Pump}$ = Total cost of pumping between given nodes

$P_{l,t}$ = Node pressure

$Z^{Hydraulics}$ = Total cost of all pumps, Objective function variable

Notations specific to the post_processing method:

$D_{l,i}^{Effective}$ = Effective pipeline diameter

$H_{l,i,t}^{Friction,HW}$ = Head loss due to friction (using Hazen-Williams equation)

Notations specific to the co_optimize method:

$D_{l,i}^{Effective}$ = Effective pipeline diameter

$H_{l,i,t}^{Friction,HW}$ = Head loss due to friction (using Hazen-Williams equation)

Hydraulics Model Formulation

Objective:

Total cost of pumping in the pipeline network.

$$\min Z^{Hydraulics} = \sum_{(l,\tilde{l}) \in LLA} C_{l,\tilde{l}}^{Pump}$$

Max allowable pressure rule: $\forall l \in L, t \in T$

Limits the maximum operating pressure in a pipeline.

$$P_{l,t} \leq \zeta^{Maximum}$$

Pump head rule: $\forall l, \tilde{l} \in LLA, t \in T$

Allows pumping only if a pump exists in a pipeline.

$$H_{l,\tilde{l},t}^{Pump} \leq M^{Flow} \cdot y_{l,\tilde{l},[t]}^{Pump}$$

Equations/constraints specific to the post_process method

Effective diameter calculation: $\forall l, \tilde{l} \in LLA$

Aggregate diameters for all existing pipelines between any 2 locations.

$$D_{l,\tilde{l}}^{Effective} = D_{l,\tilde{l}}^{Existing} + \sum_{d \in D} \delta_d^{Pipeline} \cdot y_{l,\tilde{l},d,[t]}^{Pipeline}$$

Hazen-Williams based frictional head loss calculation: $\forall l, \tilde{l} \in LLA, t \in T$

Calculate head loss using Hazen-Williams equation. Note that units for all terms in this equation are in SI units so, appropriate conversion factors must be added.

$$H_{l,\tilde{l},t}^{Friction,HW} \cdot (D_{l,\tilde{l}}^{Effective})^{4.87} = 10.704 \cdot (F_{l,\tilde{l},t}^{Piped} / \lambda_{l,\tilde{l}}^{Pipeline})^{1.85} \cdot \lambda_{l,\tilde{l}}^{Pipeline}$$

Node pressure rule: $\forall l, \tilde{l} \in LLA, t \in T$

Pressure constraint based on Bernoulli's energy balance equation.

$$P_{l,t} + \zeta_l \cdot \rho \cdot g = P_{\tilde{l},t} + \zeta_{\tilde{l}} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Friction,HW} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Pump} \cdot \rho \cdot g - H_{l,\tilde{l},t}^{Valve} \cdot \rho \cdot g$$

Pump cost rule: $\forall l, \tilde{l} \in LLA$

Allows pumping only if a pump exists in a pipeline.

$$C_{l,\tilde{l}}^{Pump} = v^{Pump} \cdot y_{l,\tilde{l},[t]}^{Pump} + v^{Electricity} \cdot \rho \cdot g \cdot \sum_{t \in T} H_{l,\tilde{l},t}^{Pump} \cdot F_{l,\tilde{l},t}^{Piped}$$

Equations/constraints specific to the co_optimize method

Effective diameter rule: $\forall l, \tilde{l} \in LLA$

Aggregate diameters for all existing pipelines between any 2 locations.

$$D_{l,\tilde{l}}^{Effective} = D_{l,\tilde{l}}^{Existing} + \sum_{d \in D} \delta_d^{Pipeline} \cdot y_{l,\tilde{l},d,[t]}^{Pipeline}$$

Hazen-Williams based frictional head loss calculation: $\forall l, \tilde{l} \in LLA, t \in T$

Calculate head loss using Hazen-Williams equation. Note that units for all terms in this equation are in SI units so, appropriate conversion factors must be added.

$$H_{l,\tilde{l},t}^{Friction,HW} \cdot (D_{l,\tilde{l}}^{Effective})^{4.87} = 10.704 \cdot (F_{l,\tilde{l},t}^{Piped} / i^{CHW})^{1.85} \cdot \lambda_{l,\tilde{l}}^{Pipeline}$$

Node pressure rule: $\forall l, \tilde{l} \in LLA, t \in T$

Pressure constraint based on Bernoulli's energy balance equation.

$$\begin{aligned} \tilde{l}, l \in LLA : P_{l,t} + \zeta_l \cdot \rho \cdot g &= P_{\tilde{l},t} + \zeta_{\tilde{l}} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Friction,HW} \cdot \rho \cdot g - H_{l,\tilde{l},t}^{Friction,HW} \cdot \rho \cdot g - H_{l,\tilde{l},t}^{Pump} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Valve} \cdot \rho \cdot g \\ \tilde{l}, l \notin LLA : P_{l,t} + \zeta_l \cdot \rho \cdot g &= P_{\tilde{l},t} + \zeta_{\tilde{l}} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Friction,HW} \cdot \rho \cdot g - H_{l,\tilde{l},t}^{Pump} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Valve} \cdot \rho \cdot g \end{aligned}$$

Pump cost rule: $\forall l, \tilde{l} \in LLA$

Allows pumping only if a pump exists in a pipeline.

$$C_{l,\tilde{l}}^{Pump} = v^{Pump} \cdot y_{l,\tilde{l},t}^{Pump} + v^{Electricity} \cdot \rho \cdot g \cdot \sum_{t \in T} H_{l,\tilde{l},t}^{Pump} \cdot F_{l,\tilde{l},t}^{Piped}$$

Linearizing the Equations/constraints specific to the co_optimize method

Parameters Δ_I = Length of interval of flows for building piecewise linear model M = Large enough constant

Binary Variables

$z_{l,\tilde{l},t,i}$, $i \in \{0, 1, 2, \dots, [70000/\Delta_I] - 1\}$ = Intermediate binary variables to determine the section of the piecewise linear graph.

Continuous Variables

$\lambda_{l,\tilde{l},t,i}$, $i \in \{0, 1, 2, \dots, [70000/\Delta_I]\}$ = Intermediate continuous variables, convex combination multipliers.

$term_{l,\tilde{l},t}$ = Continuous variables for flow raised to the power 1.85 (in RHS of Hazen-Williams).

$term2_{l,\tilde{l},t,d}$ = Continuous variables for the product of pressure drop and binary to select diameter (in LHS of Hazen-Williams).

$ec_{l,\tilde{l},t}$ = Node pressure

$Z^{Hydraulics}$ = Continuous variables for electricity cost for a particular pump at particular time period.

Effective diameter rule: $\forall l, \tilde{l} \in LLA$

We remove this constraint. When effective diameter is required, we would use the expression of binaries.

Setting term in RHS: $\forall l, \tilde{l} \in LLA, t \in T$

Set the right hand side term flow to the power 1.85 in Hazen-Williams equation.

$$\begin{aligned} F_{l,\tilde{l},t}^{Piped} &= \sum_{i \in \{0,1,2,\dots,[70000/\Delta_I]\}} i \cdot \Delta_I \cdot \lambda_{l,\tilde{l},t,i} \\ term_{l,\tilde{l},t} &= \sum_{i \in \{0,1,2,\dots,[70000/\Delta_I]\}} (1.84 \times 10^{-6} \cdot i \cdot \Delta_I)^{1.85} \cdot \lambda_{l,\tilde{l},t,i} \end{aligned}$$

Sum of convex multipliers is one $\forall l, \tilde{l} \in LLA, t \in T$

$$\sum_{i \in \{0,1,2,\dots,[70000/\Delta_I]-1\}} z_{l,\tilde{l},t,i} = 1$$

Sum of binaries determining the section of piecewise linear is one $\forall l, \tilde{l} \in LLA, t \in T$

$$\sum_{i \in \{0,1,2,\dots,[70000/\Delta_I]\}} \lambda_{l,\tilde{l},t,i} = 1$$

Only two convex multipliers are non-zero $\forall l, \tilde{l} \in LLA, t \in T, j \in \{0, 1, 2, \dots, [70000/\Delta_I] - 1\}$

$$\lambda_{l,\tilde{l},t,1} \leq z_1$$

$$\lambda_{l,\tilde{l},t,i} \leq z_i + z_{i-1} \forall i \in \{2, 3, \dots, [70000/\Delta_I] - 1\}$$

$$\lambda_{l,\tilde{l},t,[70000/\Delta_I]} \leq z_{[70000/\Delta_I]-1}$$

Setting term in LHS: $\forall l, \tilde{l} \in LLA, t \in T, d \in D$

Set the product of pressure change and binary diameter selection variables by the following set of constraints.

$$term2_{l,\tilde{l},t,d} \leq H_{l,\tilde{l},t}^{Friction,HW}$$

$$term2_{l,\tilde{l},t,d} \leq M \cdot y_{l,\tilde{l},d}^{Pipeline}$$

$$term2_{l,\tilde{l},t,d} \geq H_{l,\tilde{l},t}^{Friction,HW} - M \cdot (1 - y_{l,\tilde{l},d}^{Pipeline})$$

Hazen-Williams based frictional head loss calculation: $\forall l, \tilde{l} \in LLA, t \in T$

Calculate head loss using Hazen-Williams equation. Note that units for all terms in this equation are in SI units so, appropriate conversion factors must be added.

$$\sum_{d \in D} term2_{l,\tilde{l},t,d} \cdot (D_{l,\tilde{l}}^{Existing} + \delta_d^{Pipeline})^{4.87} = 10.704 \cdot term_{l,\tilde{l},t}^{Piped} / (l^{CHW})^{1.85} \cdot \lambda_{l,\tilde{l}}^{Pipeline}$$

Node pressure rule: $\forall l, \tilde{l} \in LLA, t \in T$

Pressure constraint based on Bernoulli's energy balance equation.

$$\tilde{l}, l \in LLA : P_{l,t} + \zeta_l \cdot \rho \cdot g = P_{\tilde{l},t} + \zeta_{\tilde{l}} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Friction,HW} \cdot \rho \cdot g - H_{l,\tilde{l},t}^{Friction,HW} \cdot \rho \cdot g - H_{l,\tilde{l},t}^{Pump} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Valve} \cdot \rho \cdot g$$

$$\tilde{l}, l \notin LLA : P_{l,t} + \zeta_l \cdot \rho \cdot g = P_{\tilde{l},t} + \zeta_{\tilde{l}} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Friction,HW} \cdot \rho \cdot g - H_{l,\tilde{l},t}^{Pump} \cdot \rho \cdot g + H_{l,\tilde{l},t}^{Valve} \cdot \rho \cdot g$$

Electricity cost rule: $\forall l, \tilde{l} \in LLA, j \in \{0, 1, 2, \dots, [70000/\Delta_I] - 1\}$

Calculate electricity (variable) cost of pump at every time period.

$$ec_{l,\tilde{l},t} \geq v^{Electricity} \cdot \rho \cdot g \cdot j \cdot \Delta_I \cdot H_{l,\tilde{l},t}^{Pump} - M(1 - z_j)$$

Pump cost rule: $\forall l, \tilde{l} \in LLA$

Allows pumping only if a pump exists in a pipeline.

$$C_{l,\tilde{l}}^{Pump} = v^{Pump} \cdot y_{l,\tilde{l},[t]}^{Pump} + \sum_{t \in T} ec_{l,\tilde{l},t}$$

5.3 Case Studies

5.3.1 Network schematics

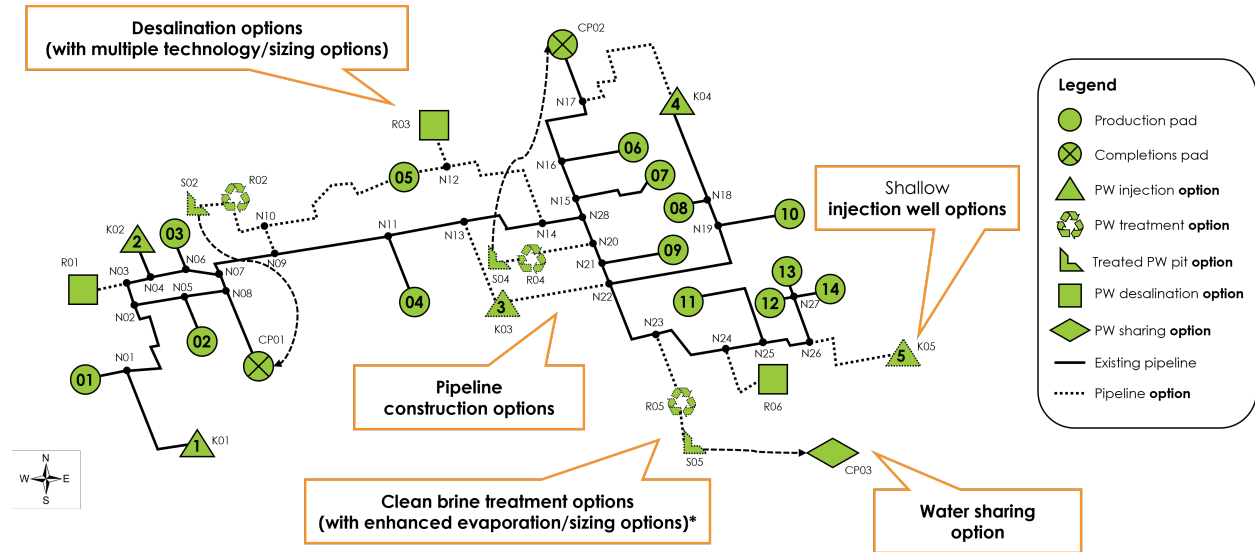


Fig. 8: Strategic Permian demo network.

Strategic toy case study

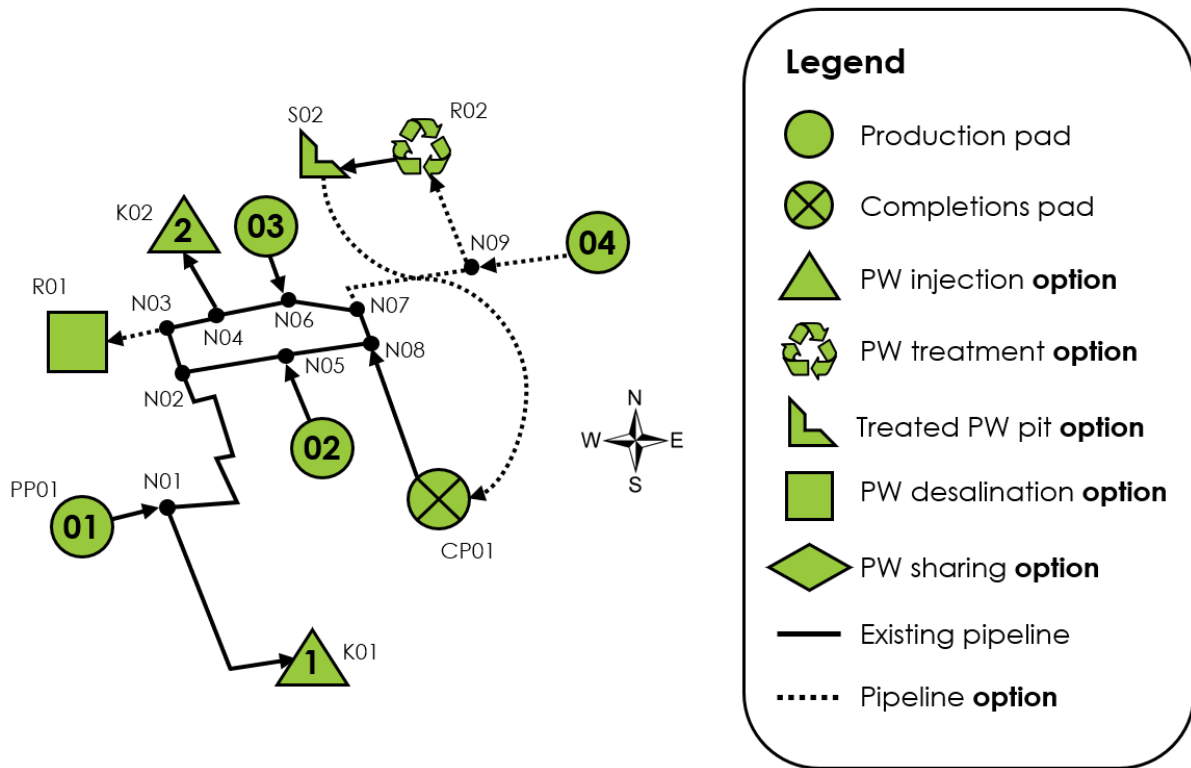


Fig. 9: Strategic toy case study network.

Strategic small case study

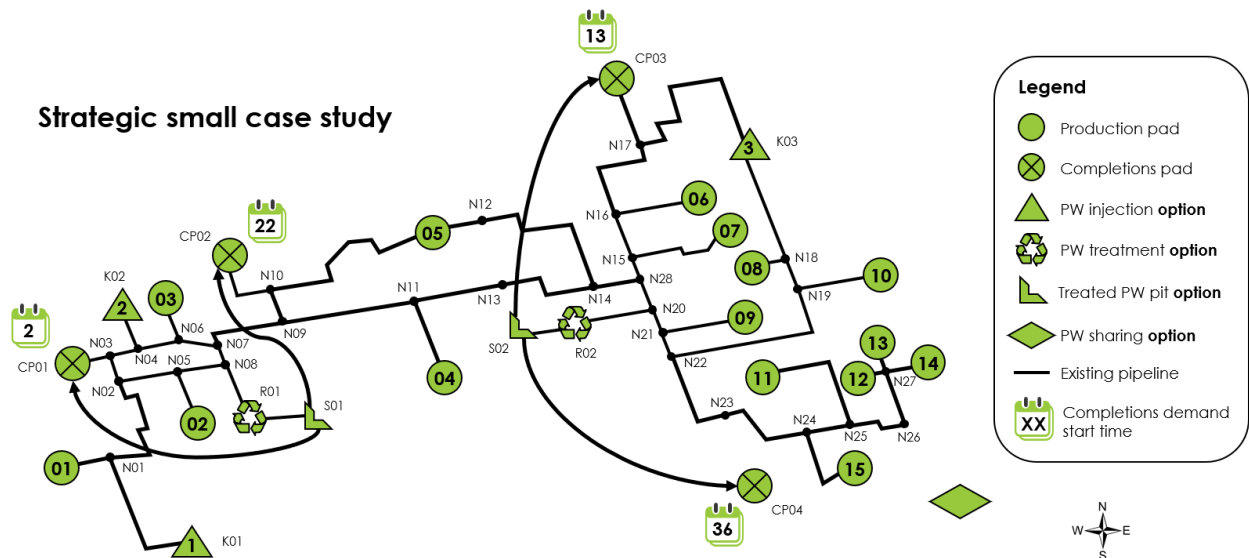


Fig. 10: Strategic small case study network.

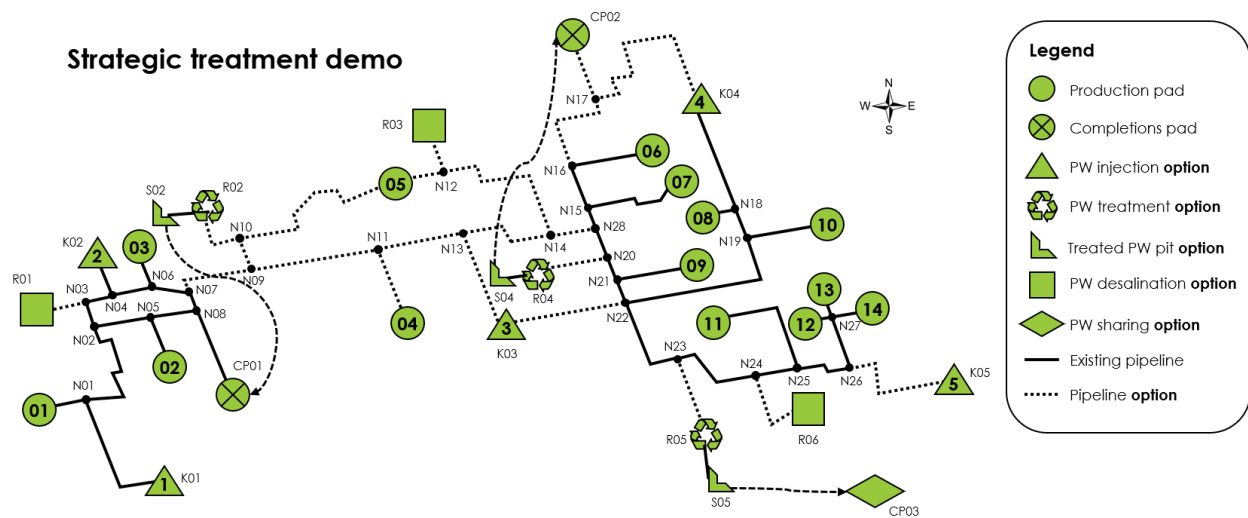


Fig. 11: Strategic treatment demo network.

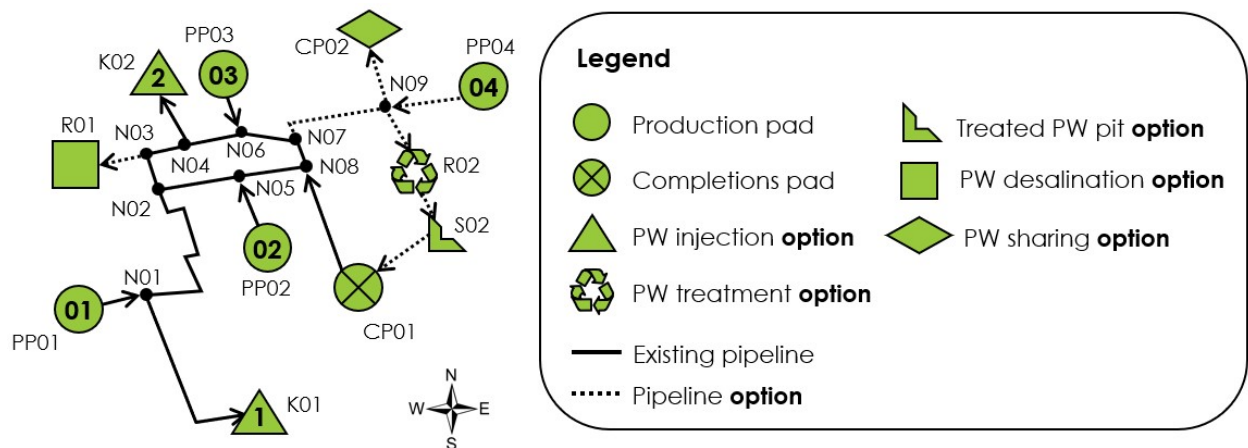


Fig. 12: Strategic model network developed for use in PARETO workshops.

5.3.2 Comparison table

	Strategic model Permian demo	Strategic model toy case study	Strategic model small case study	Strategic model treatment demo	Strategic model baseline	Strategic model workshop	Operational model case study
Input file	strategic_model_permian_demo.xlsx	strategic_model_toy_case_study.xlsx	strategic_model_small_case_study.xlsx	strategic_model_treatment_demo.xlsx	workshop_baseline_all.xlsx	operational_generic_case_study.xlsx	
Mod type	Strategic	Strategic	Strategic	Strategic	Strategic	Strategic	Operational
Description	A representative example of a Permian system. Nearly identical to treatment demo, but with reduced CAPEX options.	A very small, toy-sized network. Useful for testing and debugging.	Larger network, but “small” in the sense that disposal and pipeline expansion are not allowed, so the model solves quickly.	Larger network, and disposal and pipeline expansion are allowed. Takes a bit longer to solve. This can be seen as the “default” case study for the strategic model.	Network developed for use in PARETO workshops. Solves quickly.		Generic case study for the operational model. Note that this case study cannot currently be run in PARETO UI - it can only be run using the Python command line interface.
Decision period	Week	Week	Week	Week	Week	Week	Day
Decision horizon	52 weeks	52 weeks	52 weeks	52 weeks	52 weeks	52 weeks	5 days
Network node	28	9	28	28	9	9	0
Production pads	14	4	15	14	4	4	5
Production tank	N/A	N/A	N/A	N/A	N/A	N/A	14
Completion pads	3	1	4	3	2	2	1
External completion pads	1 (CP03)	0	0	1 (CP03)	1 (CP02)	1 (CP02)	N/A
5.3. Case Studies							
Disposal	5	2	3	5	2	2	2

Abbreviations

- CB: Clean brine treatment
- CB-EV: Clean brine treatment with enhanced evaporation
- FF: Falling film evaporation
- HDH: Humidification-dehumidification
- MD: Membrane distillation
- MVC: Mechanical vapor compression
- OARO: Osmotically assisted reverse osmosis
- SWD: Salt water disposal

5.4 Tutorials and Examples

PARETO currently has the following Jupyter notebook demonstrations available:

1. [Strategic model demo](#)
2. [Strategic model - treatment module and sensitivity analysis demo](#)
3. Visualization utility demonstrations:
 - [Bar charts](#)
 - [Scatter plots](#)
 - [Sankey diagrams](#)

Users interested in utilizing PARETO programmatically by writing Python code can download and run these Jupyter notebooks. Additional Jupyter notebook tutorials and examples are currently under development and will be added to the [examples directory](#) of the PARETO repository when they are available. The above list will be updated as well.

PARETO is an open source project in collaboration with the [IDAES](#) and [WaterTAP](#) projects. As such, the [IDAES tutorials](#) and [WaterTAP tutorials](#) are good ancillary learning materials.

New users of PARETO will also benefit from familiarity with Python and Pyomo. PARETO, IDAES, and WaterTAP are all built with Python and Pyomo, so we refer users unfamiliar with Python or Pyomo to the following tutorials:

- [Short Introduction to Python and Pyomo](#)
- [Longer Introduction to Python and Pyomo](#)

5.5 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.

¹ In the strategic model, external completions pads can be used to model opportunities for water sharing outside of the main network.

² In the strategic model, disposal capacity expansion is only allowed for SWD sites for which the initial disposal capacity is 0.

5.5.1 Get Data Functions

The following functions are used to conveniently acquire data.

Function	Section
get_data	<i>Get Data</i>
set_consistency_check	<i>Set Consistency Check</i>
od_matrix	<i>Origin/Destination Matrix</i>

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:

	A	B	C	D
1	List of all Production Pad Identifiers [-]			
2	PP01			
3	PP02			
4	PP03			
5	PP04			
6	PP05			
7				

Fig. 13: Figure 1. get_data Set Setup Format

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.

Table format

	A	B	C	D
1	Table of Production Rate Forecasts by Tanks and Pads [bbl/day]			
2				
3	ProductionPads	NODES	T1	T2
4	PP01	A01	1058	1029
5	PP01	A02	1058	1029
6	PP02	A03	466	460
7	PP02	A04	466	460

Column format

	A	B	C	D
1	Table of Initial Disposal Capacity [bbl/day]			
2	INDEX	Value		
3	K01	5000		
4	K02	10000		
5				

Fig. 14: Figure 2. get_data Parameter Setup Format

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:

```
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 aforementioned 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 accepts 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

5.5.2 Results Functions

The following functions are used to conveniently display and analyze data.

Function	Section
<code>generate_report</code>	<i>Generate Report</i>
<code>generate_sankey</code>	<i>Generate Sankey</i>
<code>plot_sankey</code>	<i>Plot Sankey</i>
<code>plot_bars</code>	<i>Plot Bars</i>
<code>plot_scatter</code>	<i>Plot Scatter</i>
<code>is_feasible</code>	<i>Is Feasible</i>

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.

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

```

===== VB_Y_DISPOSAL =====
('Disposal Site', 'Injection Capacity', 'Disposal')
vb_y_Disposal('K01', 'I0') = 1.0
vb_y_Disposal('K02', 'I0') = 1.0
vb_y_Disposal('K03', 'I0') = 1.0
vb_y_Disposal('K04', 'I0') = 1.0
vb_y_Disposal('K05', 'I0') = 1.0

```

Fig. 15: Figure 1. Example of Terminal Output

	A	B	C	D
1				
2	Disposal Site	Injection Capacity	Disposal	
3	K01	I0	1	
4	K02	I1	1	
5	K03	I0	1	
6	K04	I0	1	
7	K05	I1	1	
8				
9				
10				
11				
12				

Excel sheet tabs: v_C_Storage, v_L_Storage, v_L_PadStorage, vb_y_Pipeline, **vb_y_Disposal**, vb_y_Storage

Same data as seen in
terminal output

Notice the variable names
as the sheet names

Fig. 16: 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 separate 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

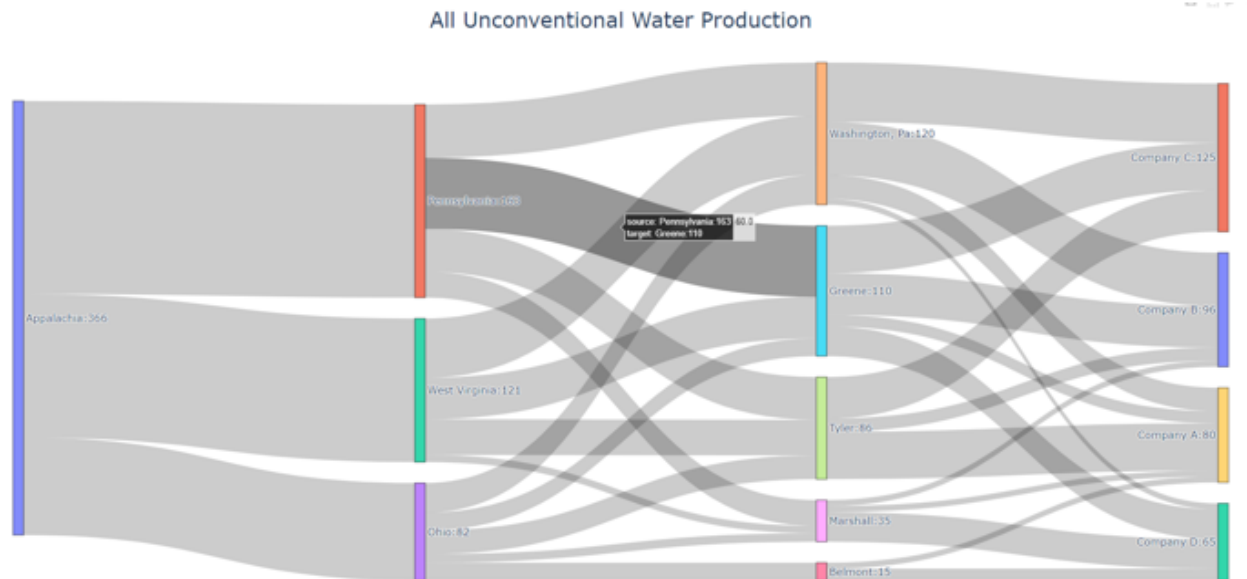


Fig. 17: Figure 3. Example of Sankey Diagram Showing Water Production Flows

“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).

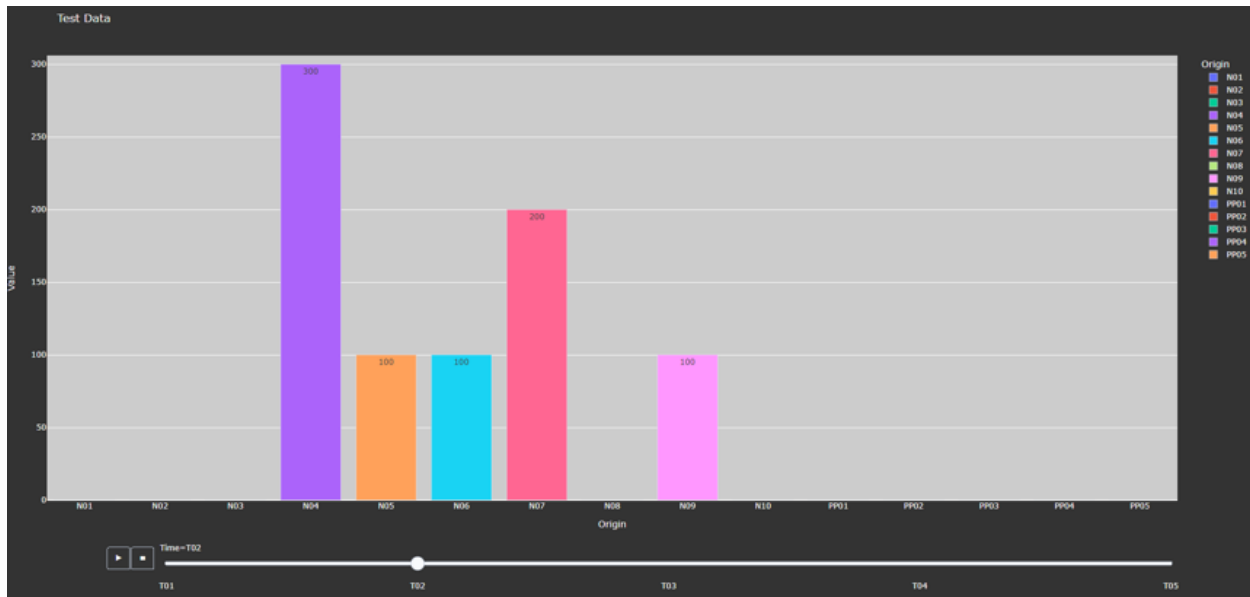


Fig. 18: Figure 4. Animated Bar Chart

“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")] }

plotBars(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.

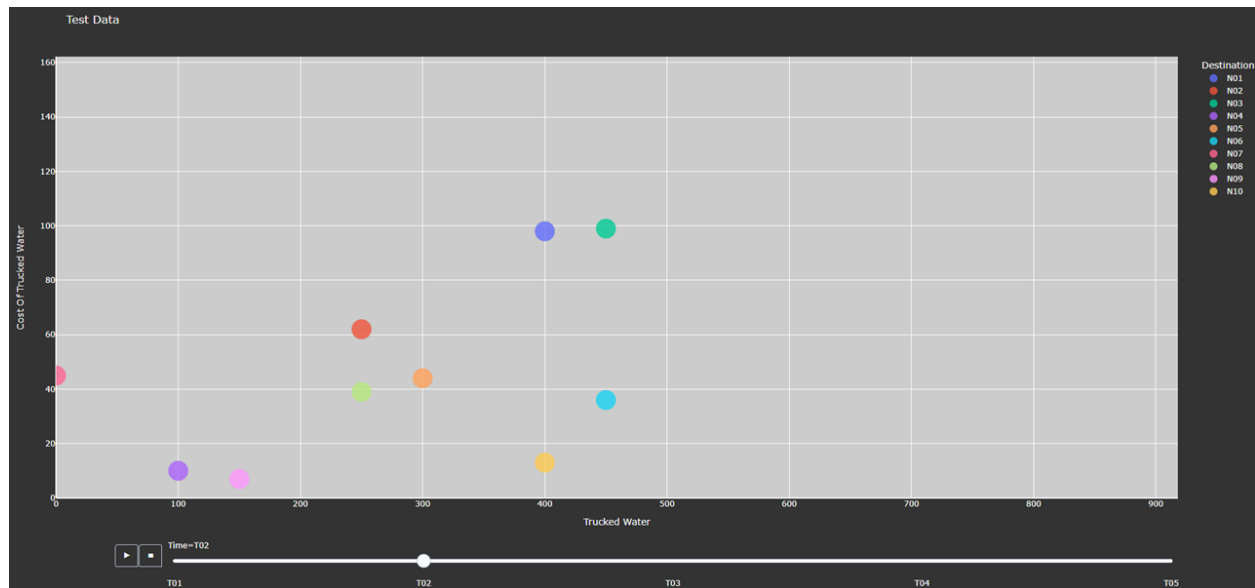


Fig. 19: Figure 5. Animated Scatter Chart. Notice the time period slider at the bottom.

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


```

"size": df_parameters["v_F_Trucked"],
"size": 'x/y',
"size": 'y/x',

```

Fig. 20: Figure 6. Options for specifying the bubbles size.

need to be 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. 21: 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.

	A	B
1		
2	Node	Category
3	PP01	1
4	PP02	1
5	CP01	2
6	CP02	2
7	CP03	2
8	S01	3
9	R01	4
10	R02	4

Fig. 22: 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)
```

Is Feasible

Method Description

Verifies the solution contained in a Pyomo model object is feasible. This requires iterating through all variables and constraints and ensuring that the constraint and variable bounds are satisfied at the solution present in the model.

How to Use

The method requires that a Pyomo model be passed as the first argument. Two other keyword arguments are optional:

1. “*bound_tol*” - Violation tolerance for bounds. Default value is 1e-3.
2. “*cons_tol*” - Violation tolerance for constraints. Default value is 1e-3.

Example of how this method can be used:

```
from pareto.utilities.results import is_feasible, nostdout

# Assume model is any Pyomo model that has already been created and solved

# Use with statement to suppress messages from the call to is_feasible
with nostdout():
    feasibility_status = is_feasible(model)

if not feasibility_status:
    print("\nModel results are not feasible and should not be trusted\n" + "-" * 60)
else:
    print("\nModel results validated and found to pass feasibility tests\n" + "-" * 60)
```

5.5.3 Earthquake Distance Function

The following function is used to conveniently acquire earthquake data for subsurface risk analysis.

Function	Section
calculate_earthquake_distances	<i>Calculate Earthquake Distances</i>

Calculate Earthquake Distances

Method Description:

This method uses the [API](#) of the [USGS Earthquake Hazards Program](#) and the [RESTful API](#) of the [TexNet Earthquake Catalog](#) to collect earthquake data and calculate distances from input saltwater disposal (SWD) well coordinates. The method accepts the following input arguments:

- **swd_latlons:**
REQUIRED. Defines a list of dictionaries with a SWD well ID (*swd_id*), its decimal latitude (*lat*) and longitude (*lon*). For example:

```
[
    {"swd_id": 1, "lat": 32.262, "lon": -101.931},
    {"swd_id": 2, "lat": 31.682, "lon": -104.401},
]
```

The SWD well IDs will be reported in the output list.

- **api:**
OPTIONAL. Specifies either the USGS or TexNet API.
 - "usgs": Use the USGS API (default)
 - "texnet": Use the TexNet API
- **max_radius_mi:**
OPTIONAL. Specifies the maximum radius in miles from a SWD well for earthquake data collection. By default, it is 5.59 miles.
- **min_magnitude:**
OPTIONAL. Specifies the minimum earthquake magnitude for earthquake data collection. By default, it is 3.
- **min_date:**
OPTIONAL. Specifies the minimum date in YYYY-MM-DD for earthquake data collection. By default, it is None, which indicates no limit.
- **max_date:**
OPTIONAL. Specifies the maximum date in YYYY-MM-DD for earthquake data collection. By default, it is None, which indicates no limit.
- **save:**
OPTIONAL. Specifies an output filename in *.csv or *.xlsx. If this argument is specified, the output will be saved as a file. It only supports the CSV and XLSX formats. The XLSX format requires the pandas and openpyxl modules. By default, it is None, which means no output file.
- **overwrite:**
OPTIONAL. Specifies whether or not to overwrite the save output file. If a file given in the save argument exists, passing True to this argument will overwrite the file. By default, it is False.

The output of this method is a list of dictionaries containing the input SWD well ID (swd_id), earthquake ID (eq_id), event time (time), distance in miles to the SWD well (distance_mi), and its magnitude (magnitude). For example:

```
[
  {
    "swd_id": 1,
    "eq_id": "tx2024fukh",
    "time": "2024-03-23 01:36:02",
    "distance_mi": 0.9231348790234829,
    "magnitude": 3
  },
  {
    "swd_id": 2,
    "eq_id": "tx2024emvg",
    "time": "2024-03-04 15:31:07",
    "distance_mi": 2.200703681215228,
    "magnitude": 3.3
  }
]
```

How to Use

Example of how this method is used:

```
swd_latlons = [
    {"swd_id": 1, "lat": 32.251, "lon": -101.940},
    {"swd_id": 2, "lat": 31.651, "lon": -104.410},
```

(continues on next page)

(continued from previous page)

```

]

earthquake_distances = calculate_earthquake_distances(
    swd_latlons, "texnet", save="eq_dist_texnet_results.csv", overwrite=True
)
earthquake_distances = calculate_earthquake_distances(
    swd_latlons, "texnet", save="eq_dist_texnet_results.xlsx", overwrite=True
)
print("# TexNet API\n", json.dumps(earthquake_distances, indent=1))

earthquake_distances = calculate_earthquake_distances(swd_latlons)
print("# USGS API\n", json.dumps(earthquake_distances, indent=1))

earthquake_distances = calculate_earthquake_distances(
    swd_latlons, min_date="2024-03-20"
)
print("# USGS API\n", json.dumps(earthquake_distances, indent=1))

earthquake_distances = calculate_earthquake_distances(
    swd_latlons, max_date="2024-03-20"
)
print("# USGS API\n", json.dumps(earthquake_distances, indent=1))

earthquake_distances = calculate_earthquake_distances(
    swd_latlons, min_date="2024-03-23", max_date="2024-03-23"
)
print("# USGS API\n", json.dumps(earthquake_distances, indent=1))

```

5.6 Frequently Asked Questions

5.6.1 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/operational_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 the [support email list](#). For more specific technical questions, we recommend our new [PARETO discussions board on Github](#).

5.6.2 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
```

5.7 License Agreement

PARETO Copyright (c) 2021-2024, 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 IS 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.

5.8 Copyright Notice

PARETO was produced under the DOE Produced Water Application for Beneficial Reuse Environmental Impact and Treatment Optimization (PARETO), and is copyright (c) 2021-2024 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`

INDEX

H

Home
 pareto, 1

P

pareto
 Home, 1