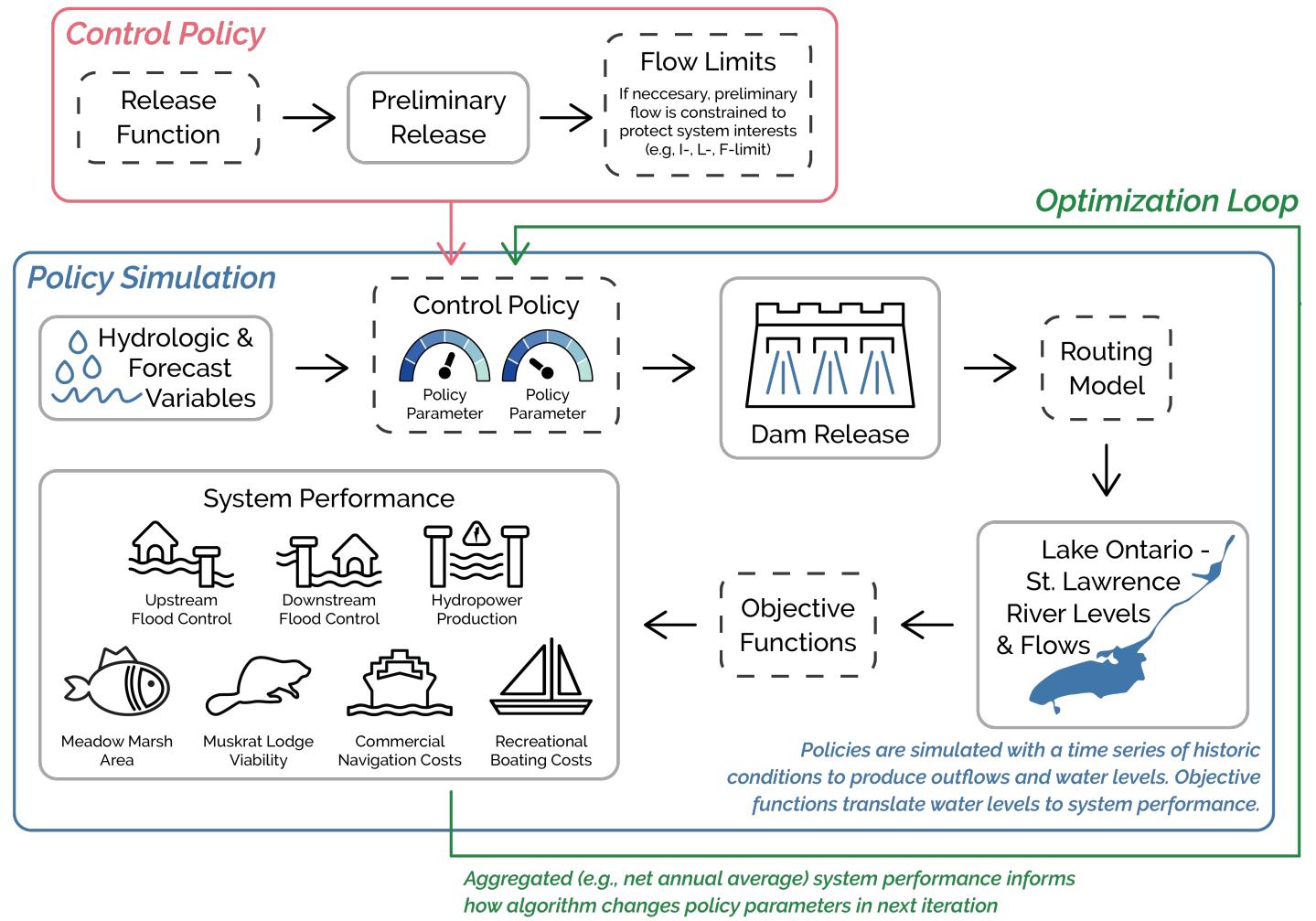


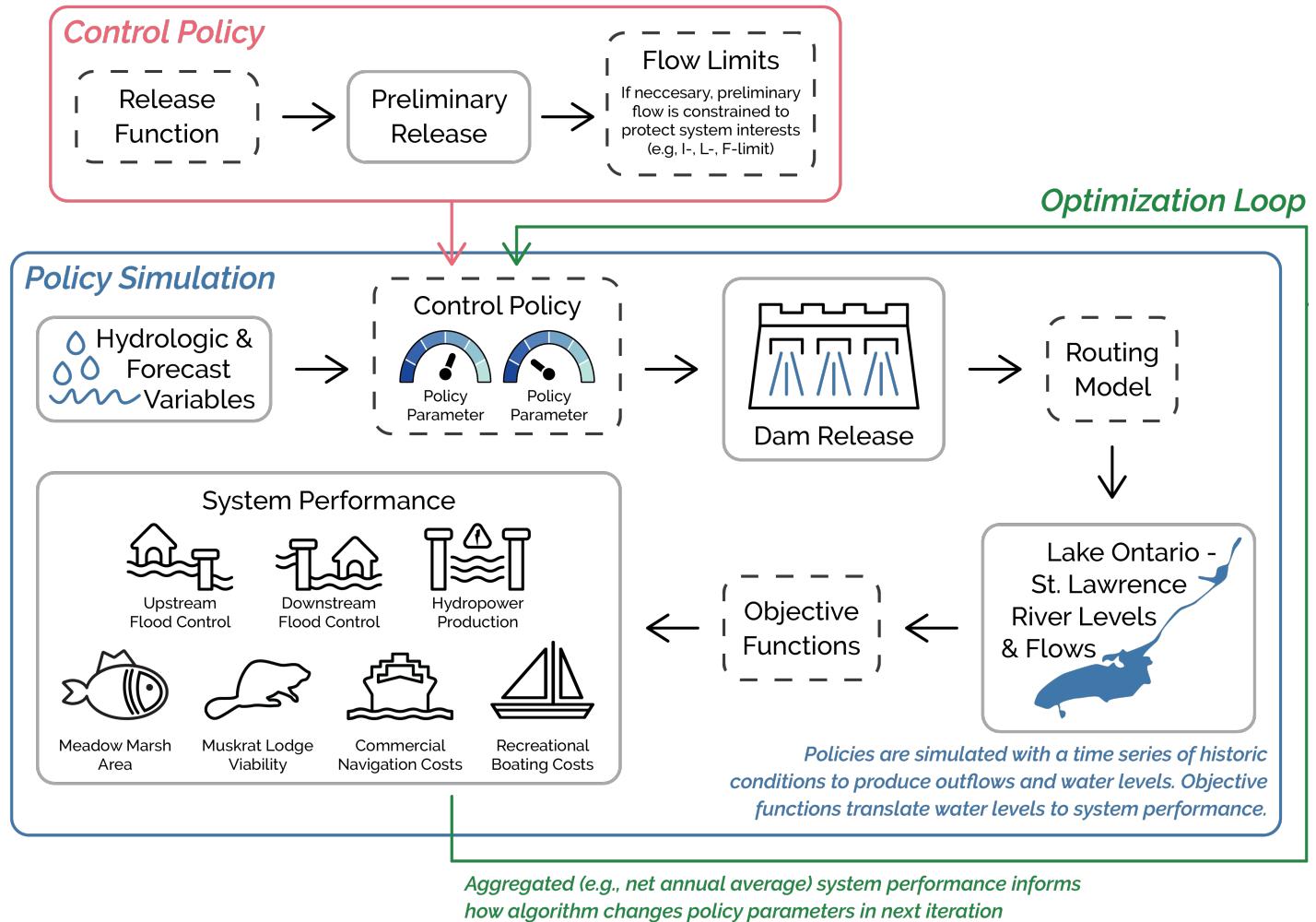
3 major components:

- Simulating a control policy
- Evaluating a control policy's performance
- Optimizing control policy parameters



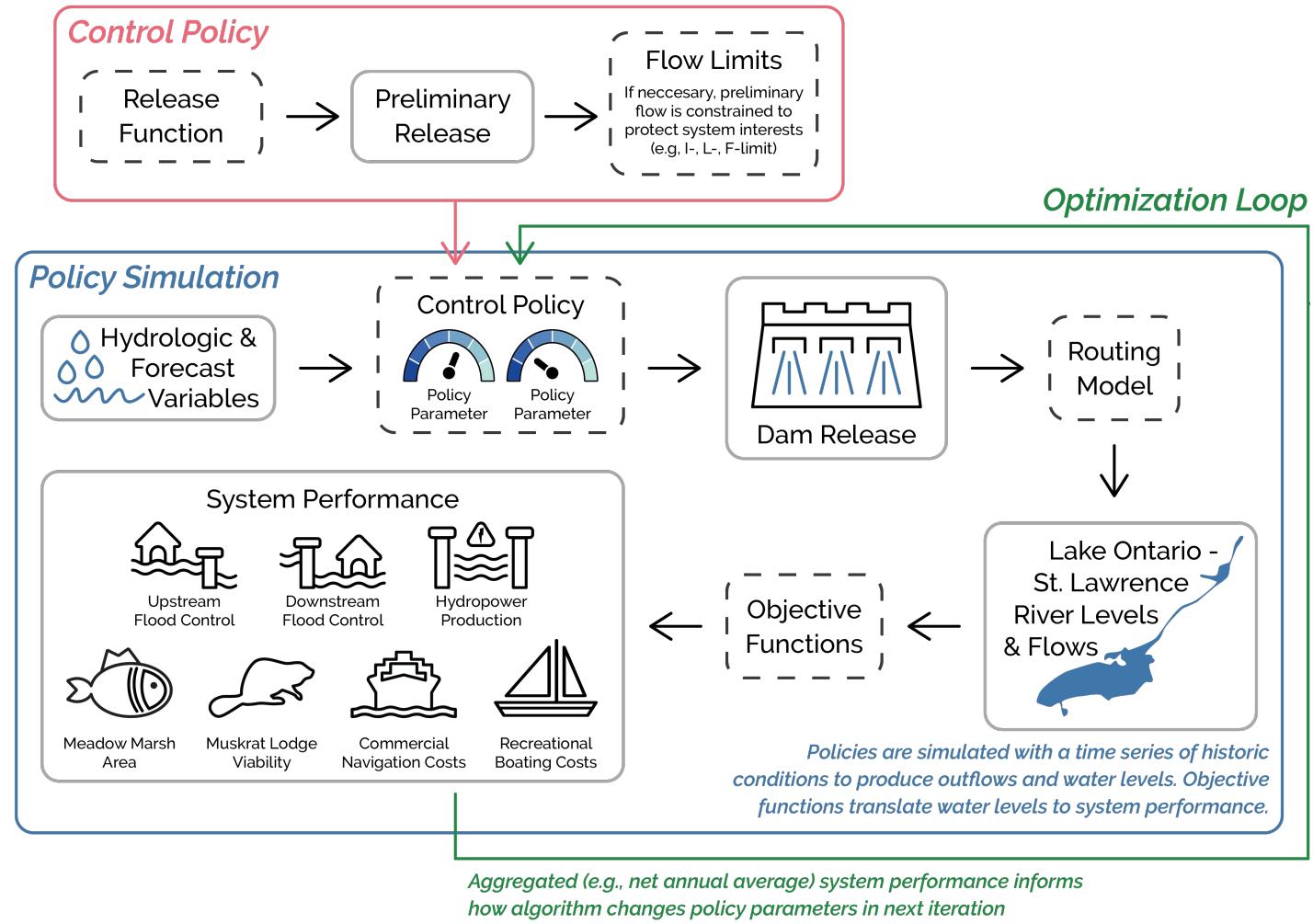
Simulating a control policy

- What is a control policy?
 - A **release function** calculates a preliminary outflow
 - Flow **limit function** adjusts the preliminary outflow (*if necessary*) and returns the final outflow
- Final outflow is **routed** through the system



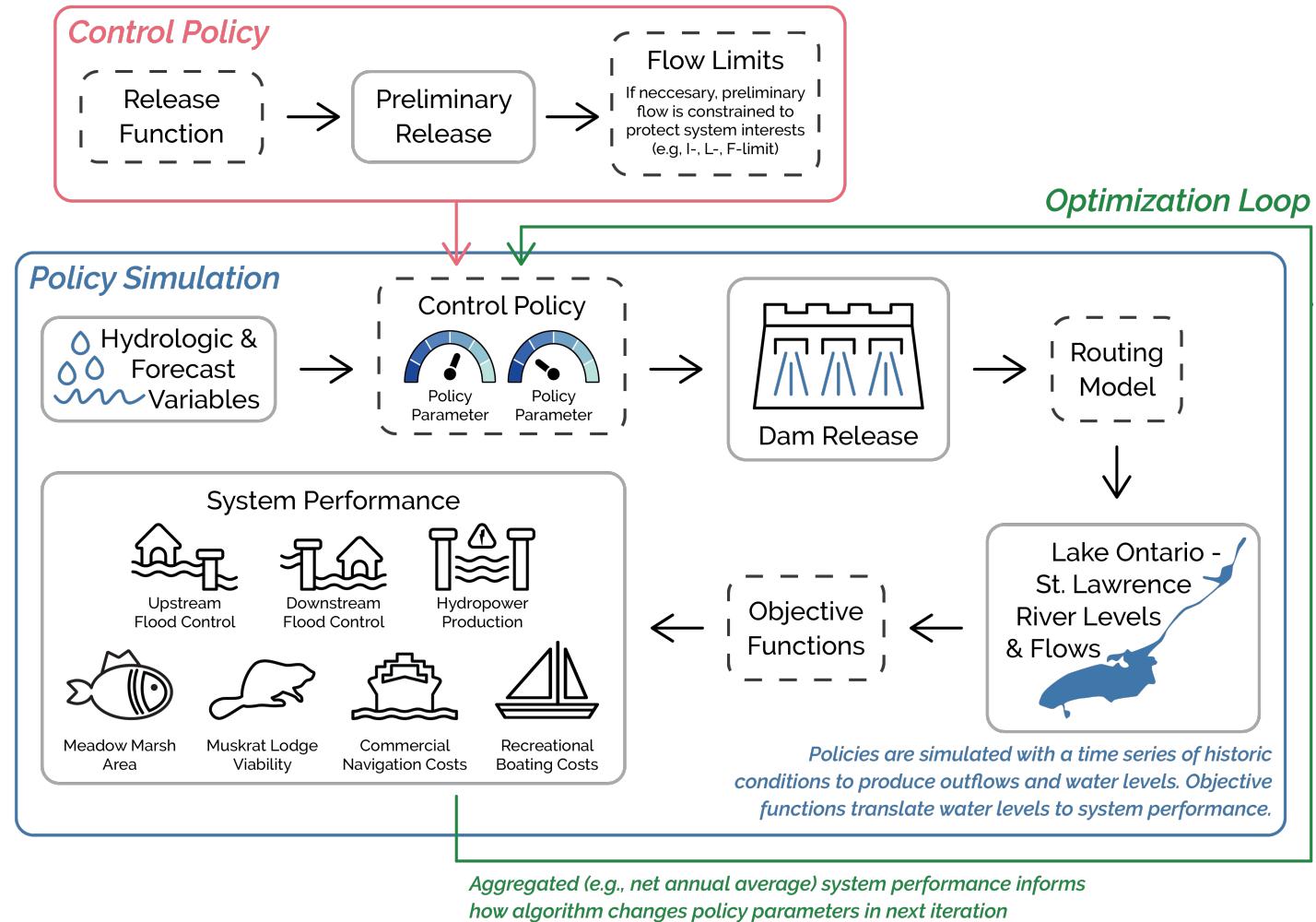
Evaluating a control policy's performance

- Control policy returns a time series of quarter-monthly water levels and flows
- Performance indicator models translate levels and flows into a **time series of system impacts**



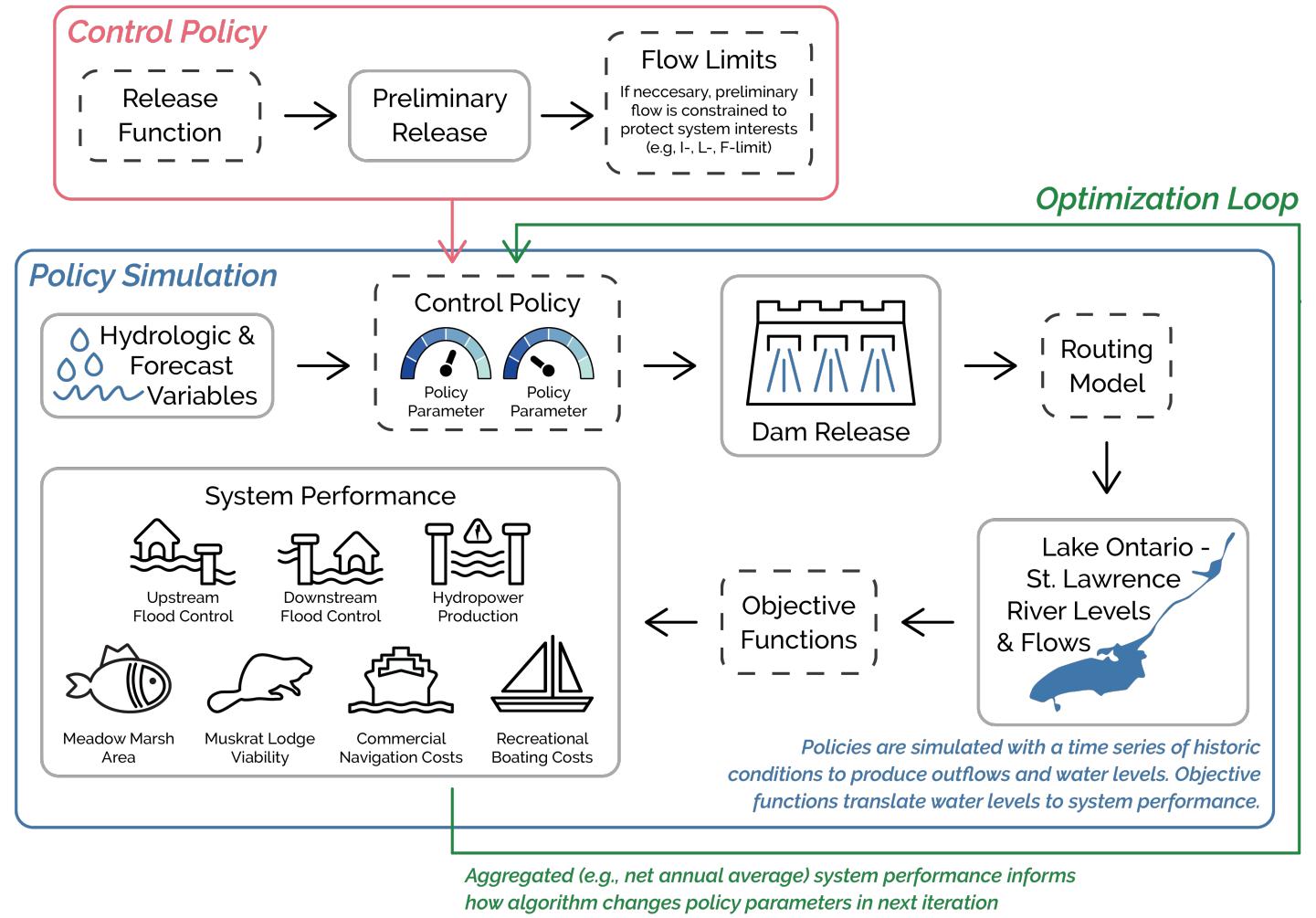
Optimizing control policy parameters

- The time series of system impacts is aggregated to the optimization algorithm
- For upstream flooding, the total number of homes flooded in each year is summed, then the average across all years is returned (i.e. **net annual average**)



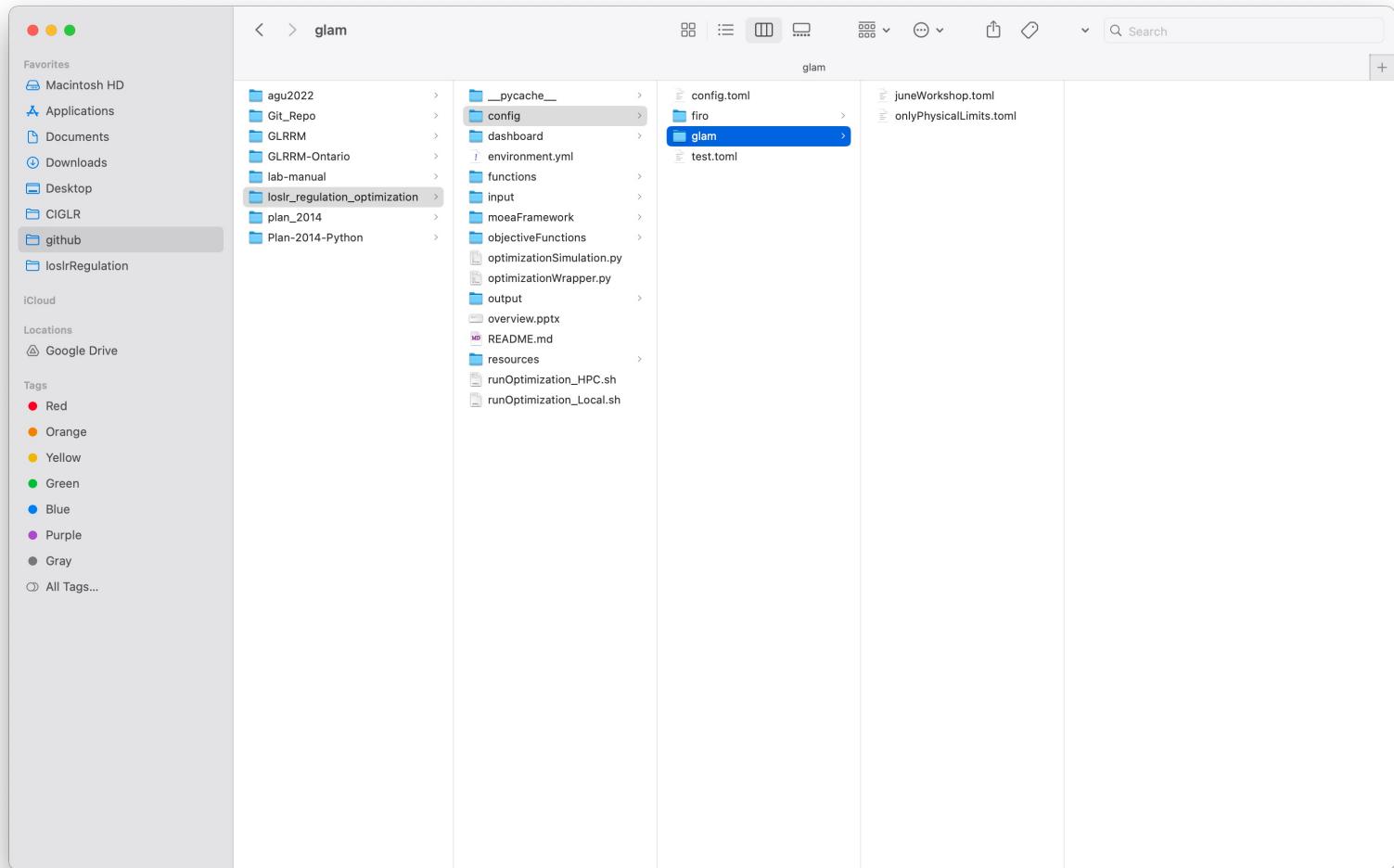
Optimizing control policy parameters

- Optimization algorithm sees 1 value for each objective
- Algorithm uses performance to inform how it changes the dials in the next iteration or function evaluation



Running an experiment:

- Configuration files setup the experiment and are written in TOML
 - Specify the release function, flow limit function, routing scheme, and objective functions
 - Setup the optimization parameters (e.g. number of function evaluations)



A screenshot of a Mac desktop interface showing a terminal window, a code editor, and a file browser.

The terminal window at the bottom left shows the command `dps*`.

The code editor window on the left displays a `juneWorkshop.toml` configuration file. A red box highlights the following section:

```
[experimentalDesign]
releaseFunction = "ruleCurve" # release.function.script.name
septemberRule = "off" # september.rule:"off" or september.rule.script.name
limitType = "Bv7_glamUpdates" # flow.limits.script.name
stlawRouting = "stlaw" # st.lawrence.routing.script.name
histMatch = "false" # false means the hydrologists that contain input.data.that.is.being.optimized.over
```

The file browser window on the right shows a directory structure under `release`:

- agu2022
- __pycache__
- config
- dashboard
- environment.yml
- functions
- input
- moeaFramework
- objectiveFunctions
- optimizationSimulation.py
- output
- overview.pptx
- README.md
- resources
- runOptimization_HPC.sh
- runOptimization_Local.sh

The `functions` folder is selected.

The sidebar on the left includes:

- Favorites: Macintosh HD, Applications, Documents, Downloads, Desktop, CIGLR, github, loslrRegulation.
- iCloud: Google Drive.
- Tags: Red, Orange, Yellow, Green, Blue, Purple, Gray, All Tags...

The screenshot shows a macOS desktop environment with several windows open:

- Terminal Window:** The title bar says "juneWorkshop.toml M X". The content of the terminal is:

```
[experimentalDesign]
releaseFunction="ruleCurve".....#.release.function.script.name
septemberRule="off".....#.september.rule:"off".or.september.rule.script.name
limitType="Bv7_glamUpdates".....#.flow.limits.script.name
stlawRouting="stlaw".....#.st.lawrence.routing.script.name
hydro_Histovich".....#.follows one of the hydrologists that contains input.data.that.is.being.optimized.over

inputFile="1961_2020/12month_sqAR".....#.path.and.file.name.of.the.input.data.that.is.being.optimized.over

[optimizationParameters]
numDV=9.....#.number.of.decision.variables
numObj=7.....#.number.of.objectives
numCon=0.....#.number.of.constraints
nfe=75000.....#.number.of.function.evaluations
popSize=100.....#.initial.population.size
metFreq=100.....#.frequency.of.function.evaluations

[decisionVariables]
dvName=[.....#.decision.variables.name
"wetIndicatorThreshold",
"wetConfidenceThreshold",
"C1",
"addC1",
"P1",
"C2",
"P2",
"dryLevelThreshold",
"dryFlowAdjustment"]

lowerBounds=[-5500.0,-0.0,-0.0,-0.0,-0.0,-0.5,-0.5,-74.0,-50.0].....
upperBounds=[-10000.0, 250.0, -1000.0, -250.0, 2.0, -300.0, -2.0, -76.5, -50.0]
normalized="False".....#.optimize.normalized
normalizedRange=[]

[performanceIndicators]
objectiveFunction="legacyPIs".....#.objective.function.script.name
metricWeighting="netAnnualAverage".....#.weighting.function
objectiveModels=[.....#.script.names.of.PI.models
"upstreamCoastalImpacts",
"downstreamCoastalImpacts",
"commercialNavigation",
"hydropowerProduction",
"meadowMarsh",
"muskratLodgeViability",
"recreationalBoating"
]

piName=[.....#.names.of.performance.indicators
"Coastal_Impacts", "Upstream_Buildings_Impacted", "#"]
```
- Code Editor:** A code editor window titled "juneWorkshop.toml M X" showing the same configuration file content as the terminal.
- File Browser:** A Finder window titled "limits" showing the directory structure of a project folder. The "limits" folder is highlighted with a red box. The contents of the "limits" folder include: agu2022, __pycache__, config, dashboard, environment.yml, functions, input, moeaFramework, objectiveFunctions, optimizationSimulation.py, optimizationWrapper.py, output, overview.pptx, README.md, resources, runOptimization_HPC.sh, and runOptimization_Local.sh.

juneworkshop.toml M X

Users > kylasemmendinger > Documents > github > losr_regression_optimization > config > glam > juneworkshop.toml > {} releaseFunction > [] inputVars

```
1 [experimentalDesign]
2 releaseFunction="ruleCurve".....#.release.function.script.name
3 septemberRule="off".....#.september.rule:"off".or.september.rule.script.name
4 limitType="Bv7_glamUpdates".....#.flow.limits.script.name
5 tlm_Pcting="t10".....#.t10.limits.script.name
6 trace="historic".....#.folder.name.of.the.hydrologic.trace.that.contains.input.data.that.is.being.optimized.over
7 inputFile="1961_2020/12month_sqAR".....#.path.and.file.name.of.the.input.data.that.is.being.optimized.over
```

[optimizationParameters]

```
10 numDV=9.....#.number.of.decision.variables
11 numObj=7.....#.number.of.objectives
12 numCon=0.....#.number.of.constraints
13 nfe=75000.....#.number.of.function.evaluations
14 popSize=100.....#.initial.population.size
15 metFreq=100.....#.frequency.of.function.evaluations
```

[decisionVariables]

```
17 dvName=[.....#.decision.variables.name
18 "wetIndicatorThreshold",
19 "wetConfidenceThreshold",
20 "C1",
21 "addC1",
22 "P1",
23 "C2",
24 "P2",
25 "dryLevelThreshold",
26 "dryFlowAdjustment"]
```

```
27 lowerBounds=[-5500.0,-0.0,-0.0,-0.0,-0.0,-0.5,-0.5,-74.0,-50.0].....
28 upperBounds=[-10000.0,250.0,-1000.0,-250.0,-2.0,-300.0,-2.0,-76.5,-50.0].....
29 normalized="False".....#.optimize.normalized
30 normalizedRange=[]
```

[performanceIndicators]

```
34 objectiveFunction="legacyPIs".....#.objective.function.script.name
35 metricWeighting="netAnnualAverage".....#.weighting.function
36 objectiveModels=[.....#.script.names.of.PI.models
37 "upstreamCoastalImpacts",
38 "downstreamCoastalImpacts",
39 "commercialNavigation",
40 "hydropowerProduction",
41 "meadowMarsh",
42 "muskratLodgeViability",
43 "recreationalBoating"
44 ]
45 piName=[.....#.names.of.performance.indicators
46 "Coastal_Impacts", "Upstream_Buildings_Impacted", "#"].....
```

dps* ⌂ 0 △ 0 ⌂ 0

The screenshot shows a Mac OS X desktop environment. On the left is a vertical dock with various icons. In the center is a terminal window titled 'juneworkshop.toml M X' containing configuration code for 'glam'. A red box highlights the section of the code where 'trace' is set to 'historic'. To the right of the terminal is a file browser window showing a directory structure for '1961_2020' data. The file browser has a sidebar with 'Favorites' (Macintosh HD, Applications, Documents, Downloads, Desktop, CIGLR, github, losrRegulation) and 'Tags' (Red, Orange, Yellow, Green, Blue, Purple, Gray, All Tags...). The main pane shows folders like agu2022, __pycache__, config, dashboard, environment.yml, functions, input, inputRetrieval.py, 1900_2008, 1900_2020, 1961_2020, historic, README.md, optimizationSimulation.py, optimizationWrapper.py, output, overview.pptx, README.md, resources, runOptimization_HPC.sh, runOptimization_Local.sh. A file named '12month_sqAR.txt' is selected in the sidebar. The right side of the file browser shows detailed information for '12month_sqAR.txt' and a preview of its contents. The status bar at the bottom shows 'dps*' and other system icons.

juneworkshop.toml M

Users > kylasemmendinger > Documents > github > loslr_regression_optimization > config > glam > juneworkshop.toml > {} releaseFunction > [] inputVars

```
1 [experimentalDesign]
2 releaseFunction = "ruleCurve" #.release.function.script.name
3 septemberRule = "off" #.september.rule:"off".or.september.rule.script.name
4 limitType = "Bv7_glamUpdates" #.flow.limits.script.name
5 stlawRouting = "stlaw" #.st.lawrence.routing.script.name
6 trace = "historic" #.folder.name.of.the.hydrologic.trace.that.contains.input.data.that.is.being.optimized.over
7 inputFile = "1961_2020/12month_sqAR" #.path.and.file.name.of.the.input.data.that.is.being.optimized.over
8
9 [optimizationParameters]
10 numDV = 9 #.number.of.decision.variables
11 numObj = 7 #.number.of.objectives
12 numCon = 0 #.number.of.constraints
13 nfe = 75000 #.number.of.function.evaluations
14 popSize = 100 #.initial.population.size
15 metFreq = 100 #.frequency.of.function.evaluations.to.report.metrics
16
17 [decisionVariables]
18 dvName = [ #.decision.variables.names
19   "wetIndicatorThreshold",
20   "wetConfidenceThreshold",
21   "C1",
22   "addC1",
23   "P1",
24   "C2",
25   "P2",
26   "dryLevelThreshold",
27   "dryFlowAdjustment"]
28 lowerBounds = [-5500.0, -0.0, -0.0, -0.0, -0.5, -0.5, -74.0, -50.0] #.lower.bounds.of.decision.variables
29 upperBounds = [10000.0, 250.0, 1000.0, 250.0, 2.0, 300.0, 2.0, 76.5, 50.0] #.upper.bounds.of.decision.variables
30 normalized = "False" #.optimize.normalized("True").or.actual.value.ranges("False")
31 normalizedRange = []
32
33 [performanceIndicators]
34 objectiveFunction = "LegacyPIs" #.objective.function.script.name
35 metricWeighting = "netAnnualAverage" #.weighting.function
36 objectiveModels = [ #.script.names.of.PI.models.to.run.in.`functions/
37   "upstreamCoastalImpacts",
38   "downstreamCoastalImpacts",
39   "commercialNavigation",
40   "hydropowerProduction",
41   "meadowMarsh",
42   "muskratLodgeViability",
43   "recreationalBoating"
44 ]
45 piName = [ #.names.of.performance.indicators
46   "Coastal_Impacts", "Upstream_Buildings_Impacted_(#)"
```

juneworkshop.toml M

Users > kylasemmendinger > Documents > github > loslr_regression_optimization > config > glam > juneworkshop.toml > {} releaseFunction > [] inputVars

```
1 [experimentalDesign]
2 releaseFunction = "ruleCurve" .....#.release.function.script.name
3 septemberRule = "off" .....#.september.rule:"off".or.september.rule.script.name
4 limitType = "Bv7_glamUpdates" .....#.flow.limits.script.name
5 stlawRouting = "stlaw" .....#.st.lawrence.routing.script.name
6 trace = "historic" .....#.folder.name.of.the.hydrologic.trace.that.contains.input.data.that.is.being.optimized.over
7 inputFile = "1961_2020/12month_sqAR" .....#.path.and.file.name.of.the.input.data.that.is.being.optimized.over
8
9 [optimizationParameters]
10 numDV = 9 .....#.number.of.decision.variables
11 numObj = 7 .....#.number.of.objectives
12 numCon = 0 .....#.number.of.constraints
13 nfe = 75000 .....#.number.of.function.evaluations
14 popSize = 100 .....#.initial.population.size
15 metFreq = 100 .....#.frequency.of.function.evaluations.to.report.metrics
16
17 [decisionVariables]
18 dvName = [ .....#.decision.variables.names
19 .... "wetIndicatorThreshold",
20 .... "wetConfidenceThreshold",
21 .... "C1",
22 .... "addC1",
23 .... "P1",
24 .... "C2",
25 .... "P2",
26 .... "dryLevelThreshold",
27 .... "dryFlowAdjustment"]
28 lowerBounds = [-5500.0, -0.0, -0.0, -0.0, -0.5, -0.5, -74.0, -50.0] .....#.lower.bounds.of.decision.variables
29 upperBounds = [10000.0, 250.0, 1000.0, 250.0, 2.0, 300.0, 2.0, 76.5, 50.0] .....#.upper.bounds.of.decision.variables
30 normalized = "False" .....#.optimize.normalized("True").or.actual.value.ranges("False")
31 normalizedRange = []
32
33 [performanceIndicators]
34 objectiveFunction = "LegacyPIs" .....#.objective.function.script.name
35 metricWeighting = "netAnnualAverage" .....#.weighting.function
36 objectiveModels = [ .....#.script.names.of.PI.models.to.run.in.`functions/
37 .... "upstreamCoastalImpacts",
38 .... "downstreamCoastalImpacts",
39 .... "commercialNavigation",
40 .... "hydropowerProduction",
41 .... "meadowMarsh",
42 .... "muskratLodgeViability",
43 .... "recreationalBoating"
44 ]
45 piName = [ .....#.names.of.performance.indicators
46 .... "Coastal_Impacts", "Upstream_Buildings_Impacted_(#)"
```

The screenshot shows a macOS desktop environment with several open windows:

- Terminal Window:** The title bar says "juneWorkshop.toml M X". The content of the terminal is a TOML configuration file named "juneWorkshop.toml". A red box highlights the section from line 33 to line 45, which defines performance indicators.
- File Browser:** The title bar says "functions". The sidebar shows "Favorites" including Macintosh HD, Applications, Documents, Downloads, Desktop, CIGLR, and GitHub. The GitHub folder is selected. The main pane shows a directory structure under "functions": agu2022, __pycache__, config, dashboard, GLRRM, GLRRM-Ontario, lab-manual, legacyPIs, __pycache__, README.md, functions, input, moeaFramework, objectiveFunctions, optimizationSimulation.py, optimizationWrapper.py, output, overview.pptx, README.md, resources, runOptimization_HPC.sh, and runOptimization_Local.sh.
- Code Editor:** The title bar says "juneWorkshop.toml M X". The content is the same TOML configuration file. The section from line 33 to line 45 is highlighted with a red box.

```
[experimentalDesign]
releaseFunction = "ruleCurve" # release.function.script.name
septemberRule = "off" # september.rule.off # flow.limits.script.name
limitType = "Bv7_glamUpdates" # st.lawrence.routing.script.name
stlawRouting = "stlaw" # st.lawrence.routing.script.name
trace = "historic" # folder.name.of.the.hydrology.log
inputFile = "1961_2020/12month_sqAR" # path.and.file.name.of.the.inputfile

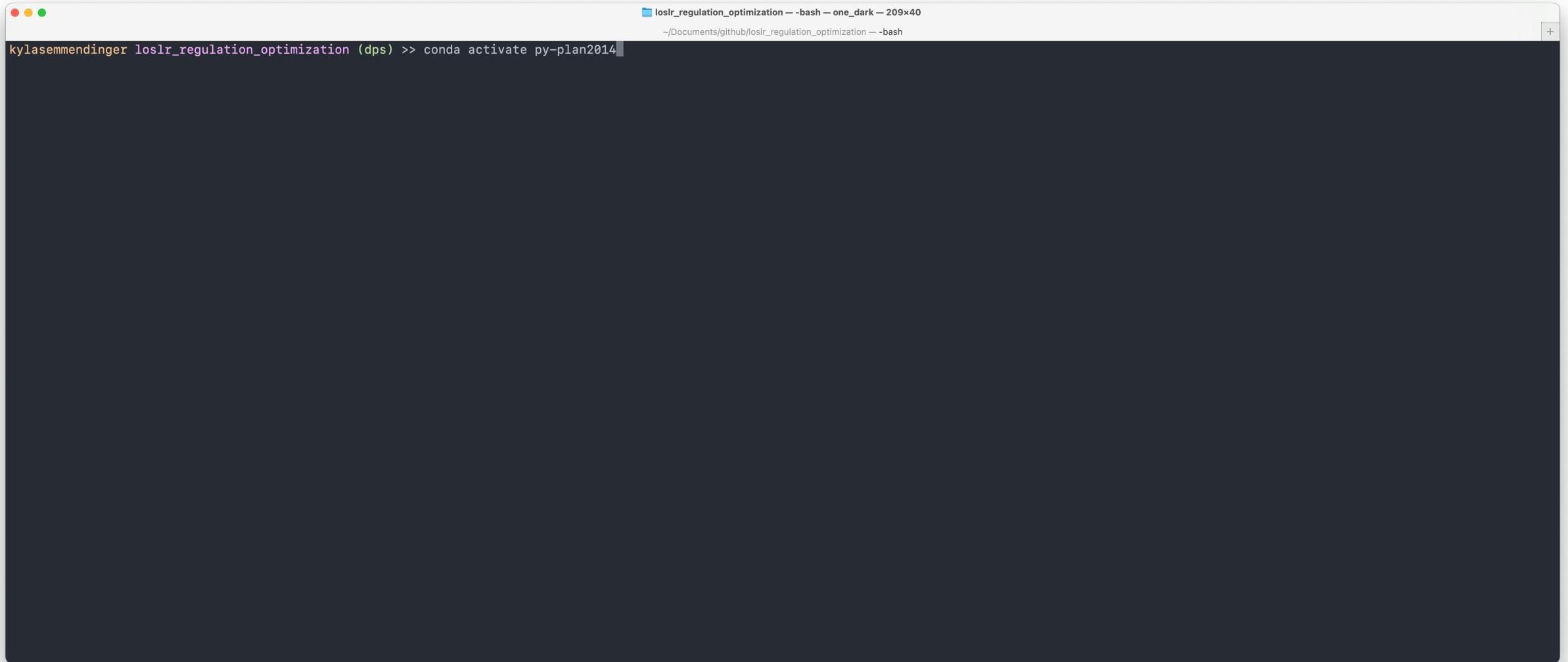
[optimizationParameters]
numDV = 9 # number.of.decision.variables
numObj = 7 # number.of.objectives
numCon = 0 # number.of.constraints
nfe = 75000 # number.of.function.evaluations
popSize = 100 # initial.population.size
metFreq = 100 # frequency.of.function.evaluations

[decisionVariables]
dvName = [ # decision.variables.name
    "wetIndicatorThreshold",
    "wetConfidenceThreshold",
    "C1",
    "addC1",
    "P1",
    "C2",
    "P2",
    "dryLevelThreshold",
    "dryFlowAdjustment"
]
lowerBounds = [-5500.0, 0.0, -0.0, -0.0, -0.0, -0.5, -0.5, -74.0, -50.0] # lower bounds
upperBounds = [10000.0, 250.0, 1000.0, 250.0, 2.0, 300.0, 2.0, 76.5, 50.0] # upper bounds
normalized = "False" # optimize.normalized.(true)
normalizedRange = []

[performanceIndicators]
objectiveFunction = "legacyPIs" # objective.function.script.name
metricWeighting = "netAnnualAverage" # weighting.function
objectiveModels = [ # script.names.of.PI.models.to.run.in.`functions/` 
    "upstreamCoastalImpacts",
    "downstreamCoastalImpacts",
    "commercialNavigation",
    "hydropowerProduction",
    "meadowMarsh",
    "muskratLodgeViability",
    "recreationalBoating"
]
piName = [ # names.of.performance.indicators
    "Coastal_Impacts_Unterstream_Buildings_Impacted_(#)"
```

Terminal status bar: dps* ⌂ ⌂ 0 △ 0 ⌂ 0 Ln 58, Col 40 Spaces: 4 UTF-8 LF { TOML no schema selected Formatting: ✓

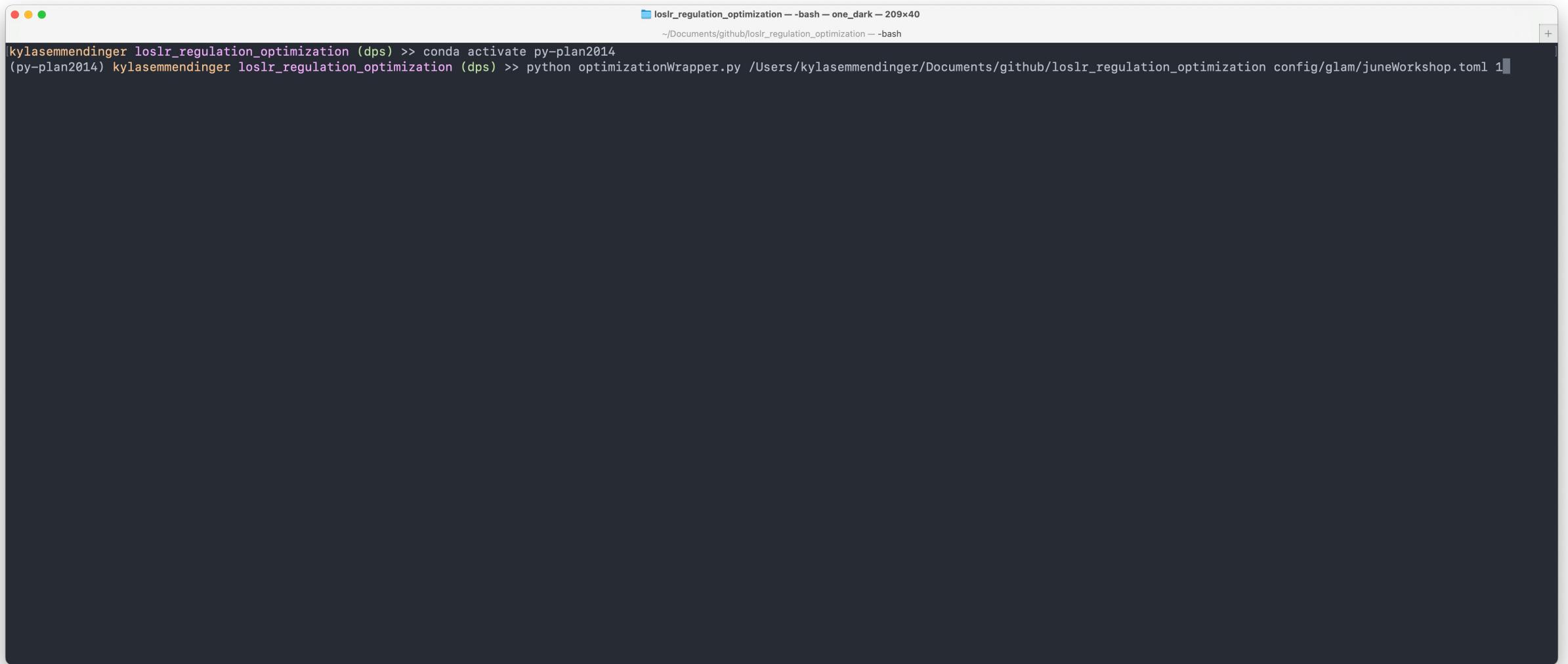
Running the optimization



A screenshot of a macOS terminal window. The window title is "loslr_regulation_optimization — bash — one_dark — 209x40" and the path is "~Documents/github/loslr_regulation_optimization — bash". The terminal prompt shows the user "kylasemmendinger" and the command "loslr_regulation_optimization (dps) >> conda activate py-plan2014". The terminal has a dark background and standard macOS window controls.

```
kylasemmendinger loslr_regulation_optimization (dps) >> conda activate py-plan2014
```

Running the optimization



A screenshot of a macOS terminal window. The window title is "loslr_regulation_optimization -- bash -- one_dark -- 209x40" and the path is "~/Documents/github/loslr_regulation_optimization -- bash". The terminal contains the following command:

```
kylasemmendinger loslr_regulation_optimization (dps) >> conda activate py-plan2014  
(py-plan2014) kylasemmendinger loslr_regulation_optimization (dps) >> python optimizationWrapper.py /Users/kylasemmendinger/Documents/github/loslr_regulation_optimization config/glam/juneWorkshop.toml 1
```

A screenshot of a terminal window titled "loslr_regulation_optimization -- bash -- one_dark -- 209x40" running on a Mac OS X system. The window shows a command-line interface with a purple header bar containing standard OS X window controls (minimize, maximize, close) and a search bar. The main area displays a bash script named "runOptimization_Local.sh".

```
$ runOptimization_Local.sh x
Users > kylasemmendinger > Documents > github > loslr_regression_optimization > $ runOptimization_Local.sh
1 #!/bin/bash
2
3 loc="$1".....# home directory for the optimization code
4 config="$2".....# path to config file from home directory
5 nseeds="$3".....# number of random seeds to run optimization
6
7 # array of unique seeds based on number of seeds specified
8 SEEDS=$(seq 1 ${nseeds})
9
10 # for loop to run optimizationWrapper.py for each seed
11 for S in ${SEEDS};
12
13 do
14 python optimizationWrapper.py ${loc} ${config} ${S}
15 done
16
17 # tells shell script to wait for all jobs to finish
18 wait <<-(jobs -p).
```

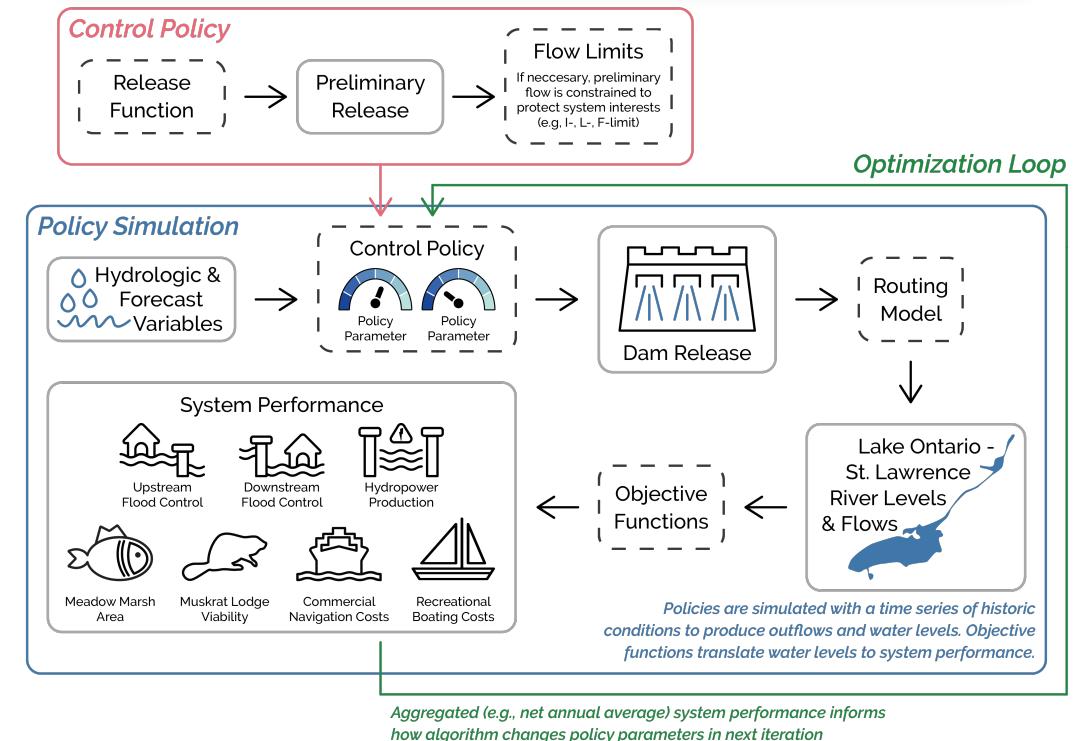
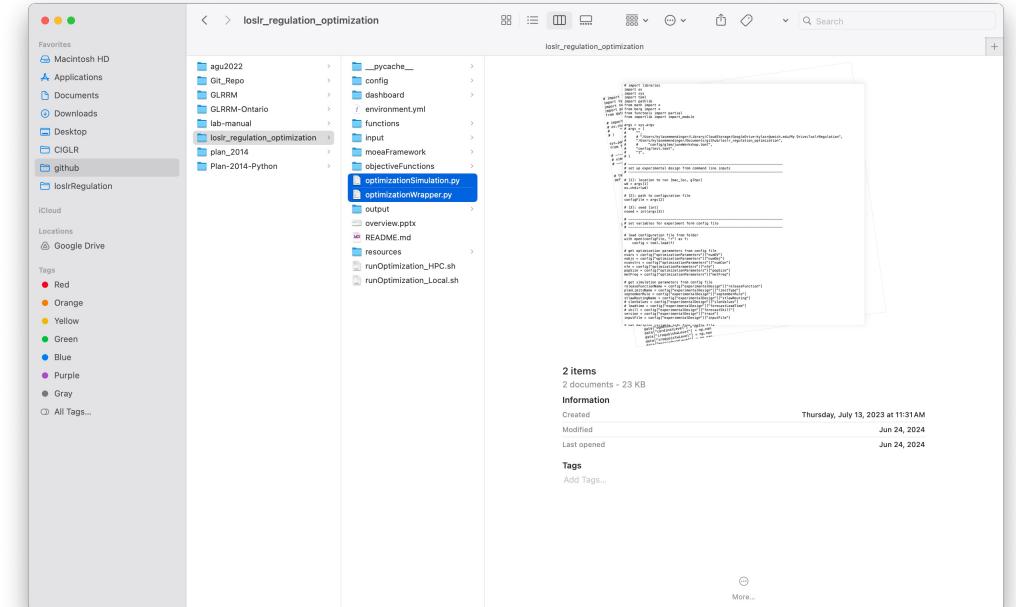
The terminal output shows the command being run:

```
kylasemmendinger loslr_regression_optimization (dps) >> conda activate py-plan2014
(py-plan2014) kylasemmendinger loslr_regression_optimization (dps) >> ./runOptimization_Local.sh /Users/kylasemmendinger/Documents/github/loslr_regression_optimization config/glam/juneWorkshop.toml 5
```

At the bottom of the terminal window, there are several small icons: a red square, a yellow circle, a green triangle, a red cross, a white square, a white triangle, a white circle, and a white square with a diagonal line.

Main scripts

- `optimizationWrapper` sets up the experiment in Borg and calls `optimizationSimulation`
 - Green optimization loop
- `optimizationSimulation` simulates the time series of water levels and system impacts and returns aggregated metrics to Borg
 - Blue simulation loop
- Modules are defined in the `functions/` directory
 - Pink control policy loop (and routing)



Other things to note:

- Syntax might look funky... code has been optimized for speed!
- Have not tested on Windows
 - Pathing might be an issue, just be aware!
- Templates are included to help you write new functions
 - Required outputs are described in each function
- Setting up the many-objective evolutionary algorithm, Borg
 - Documentation is linked in the repo
- Compiling the MOEAFramework (to calculate Borg metrics)
 - Documentation is linked in the repo