

Modeling Workgroup Meeting Quarterly Review

Optimization update: Development of an Optimization API

*Kalyanmoy Deb, Pouyan Nejadhashemi,
Gregorio Toscano, Vahid Rafiei, and
Hoda Razavi.*

OCT 2022

MICHIGAN STATE UNIVERSITY



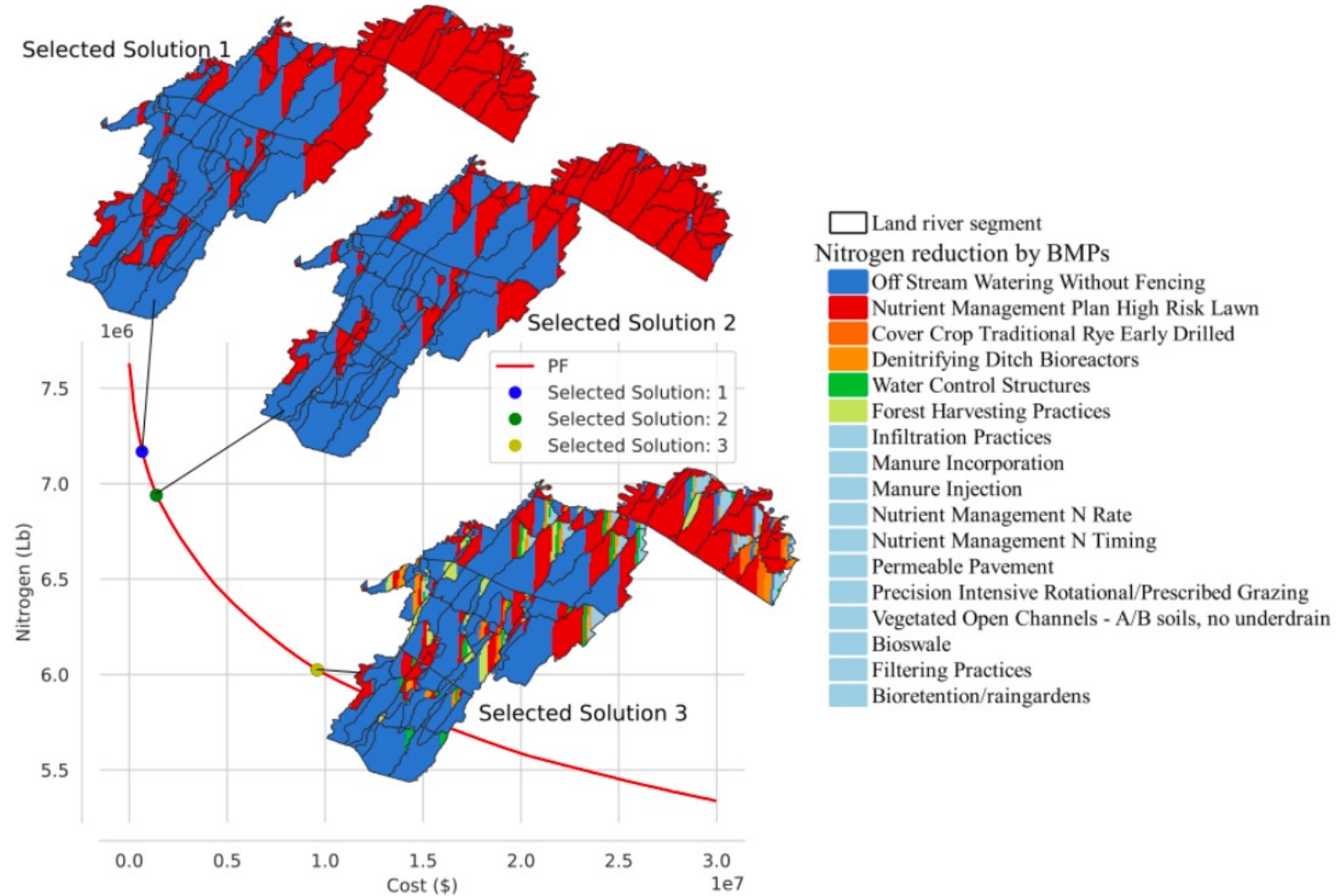
Agenda

- Objective 2: Development of Efficient Multi-objective Optimization Procedures
- Current Accomplishments
- Development of a RESTful API
- Next Steps

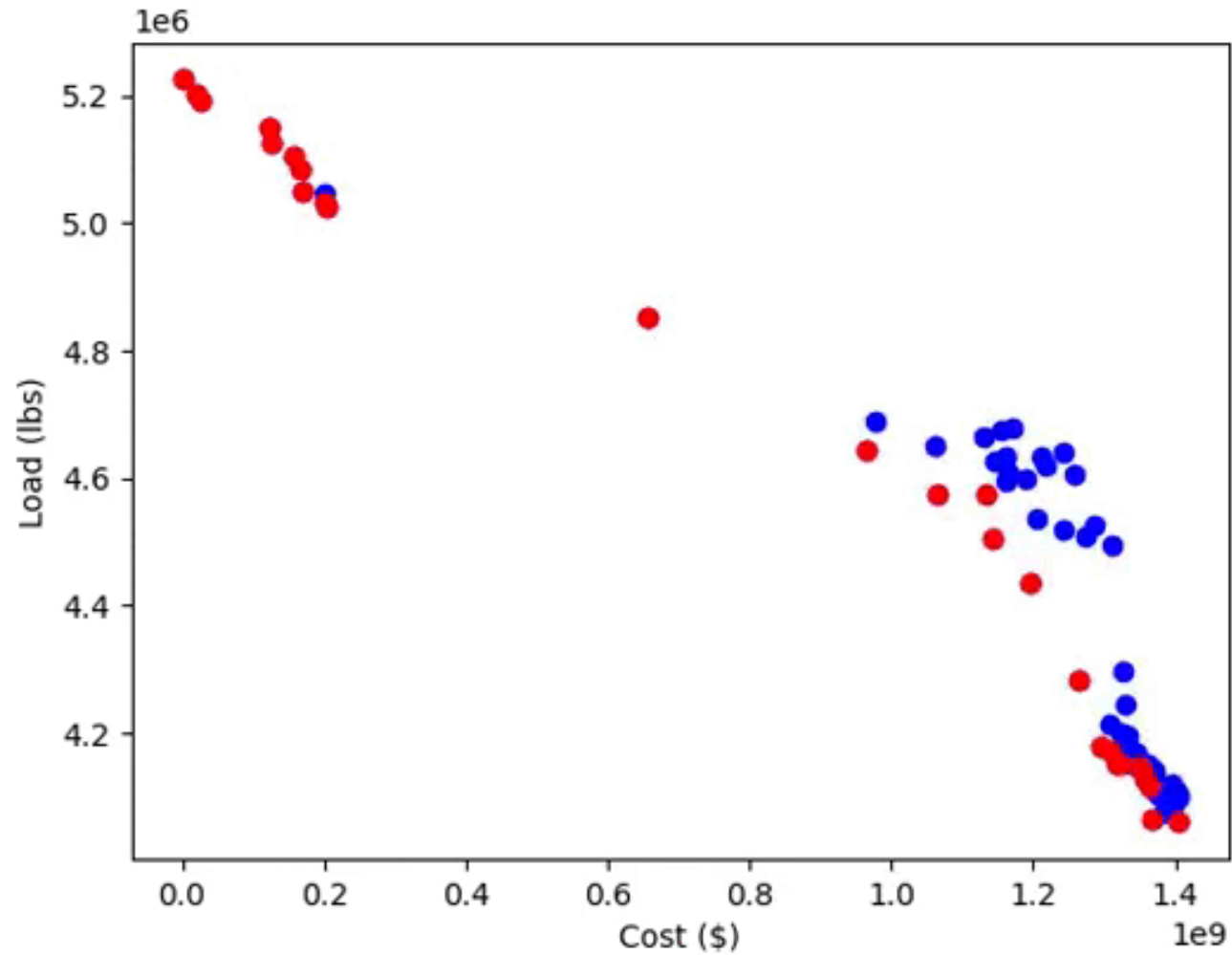
Benefits of multi-objective optimization

BMP Allocation

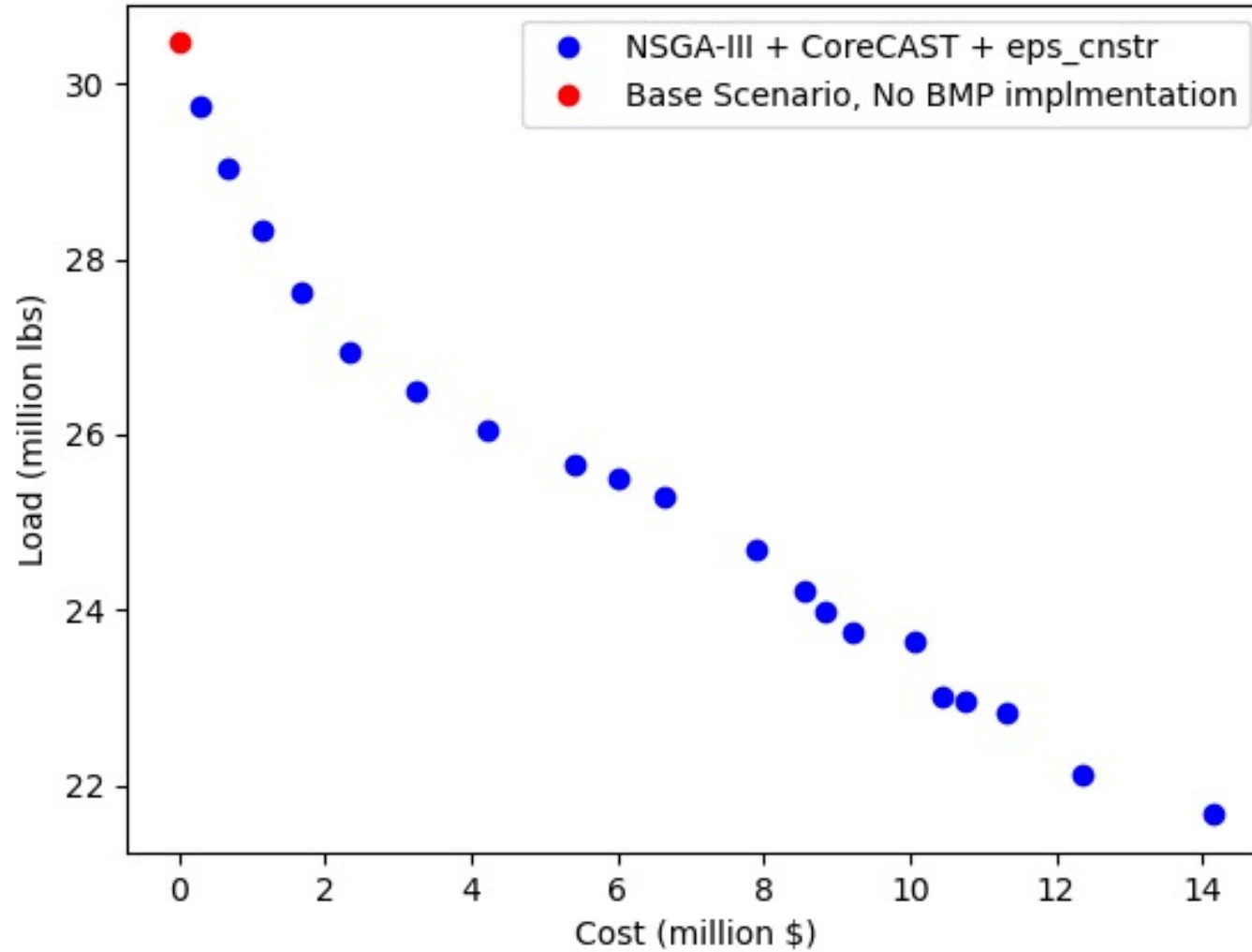
- Two objectives: Cost and Load
- Each point represents a solution with BMPs implementation required to achieve the referred Cost and Load.
- There is a compromise between the Cost and the Loads, we can achieve better load reductions but it will cost more.



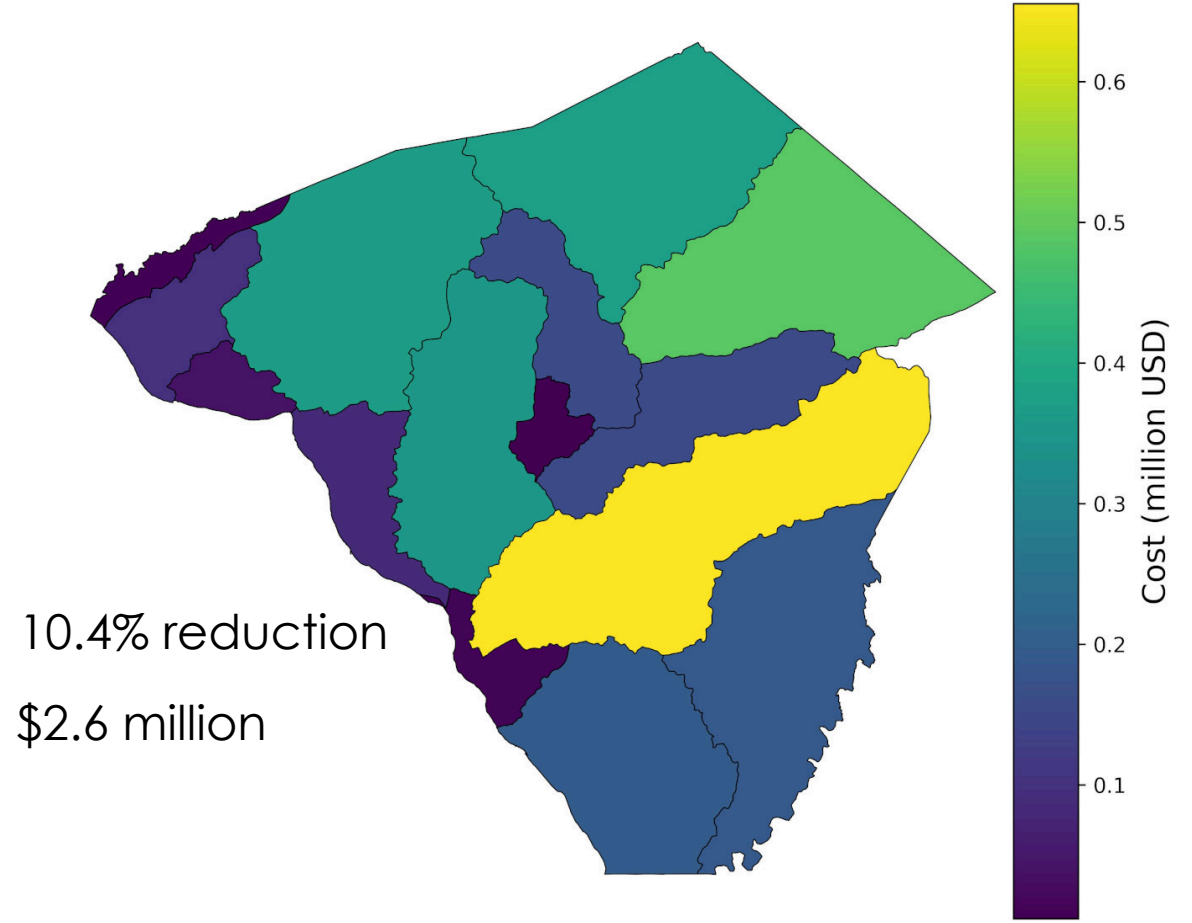
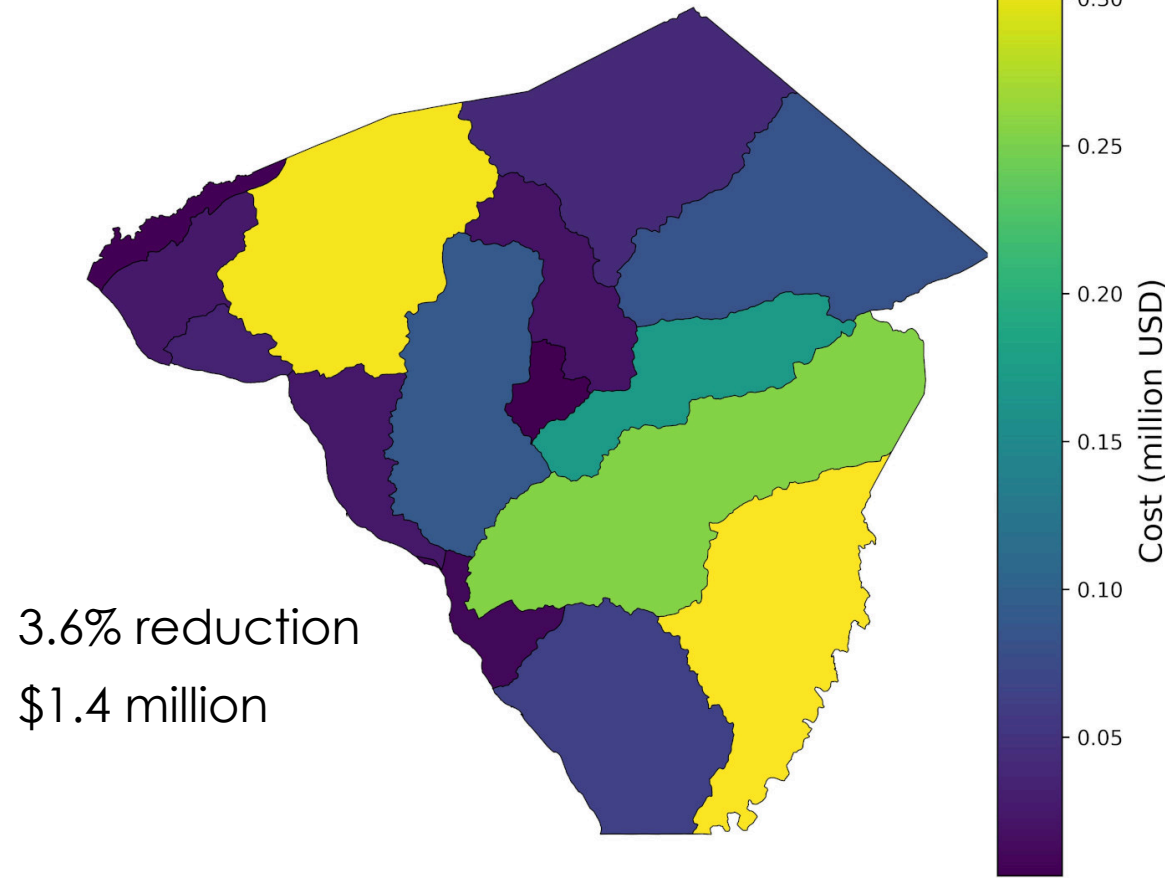
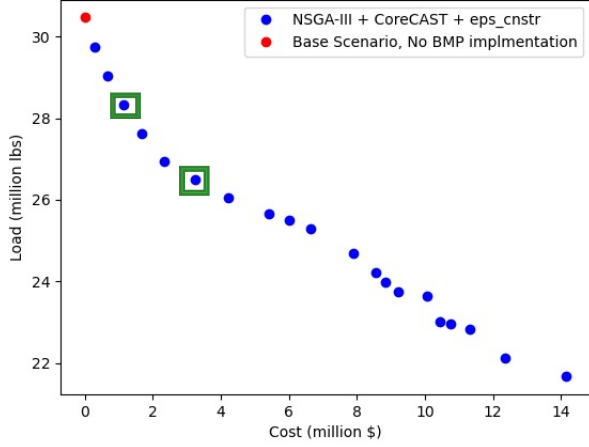
NSGA-III Convergence plot



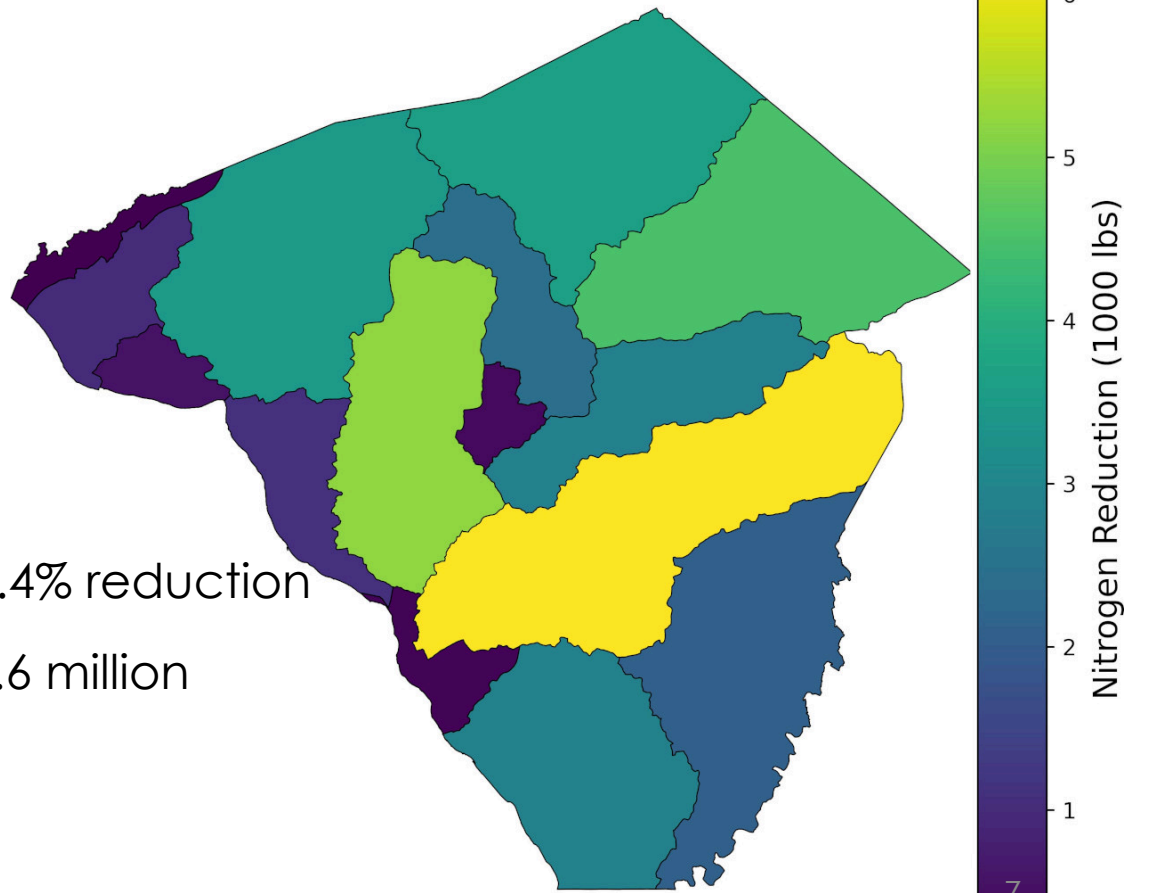
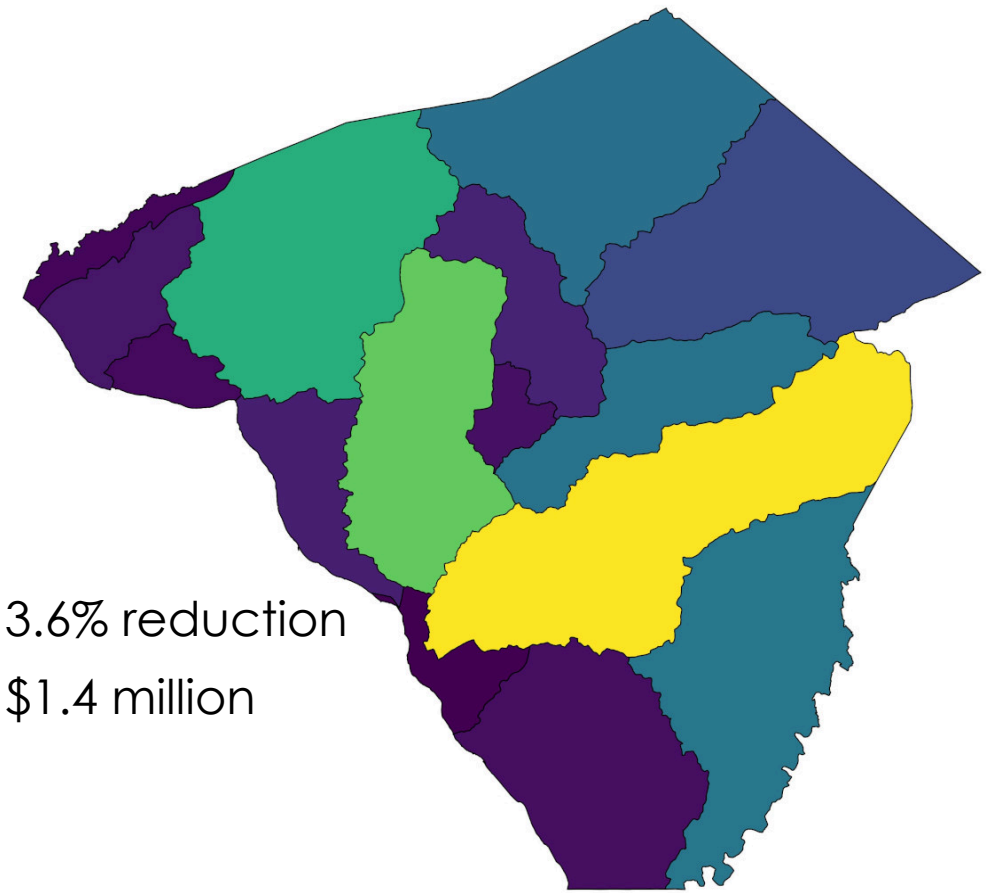
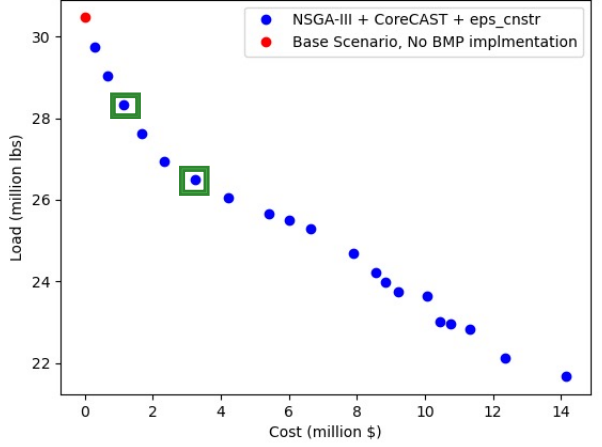
Lancaster, PA



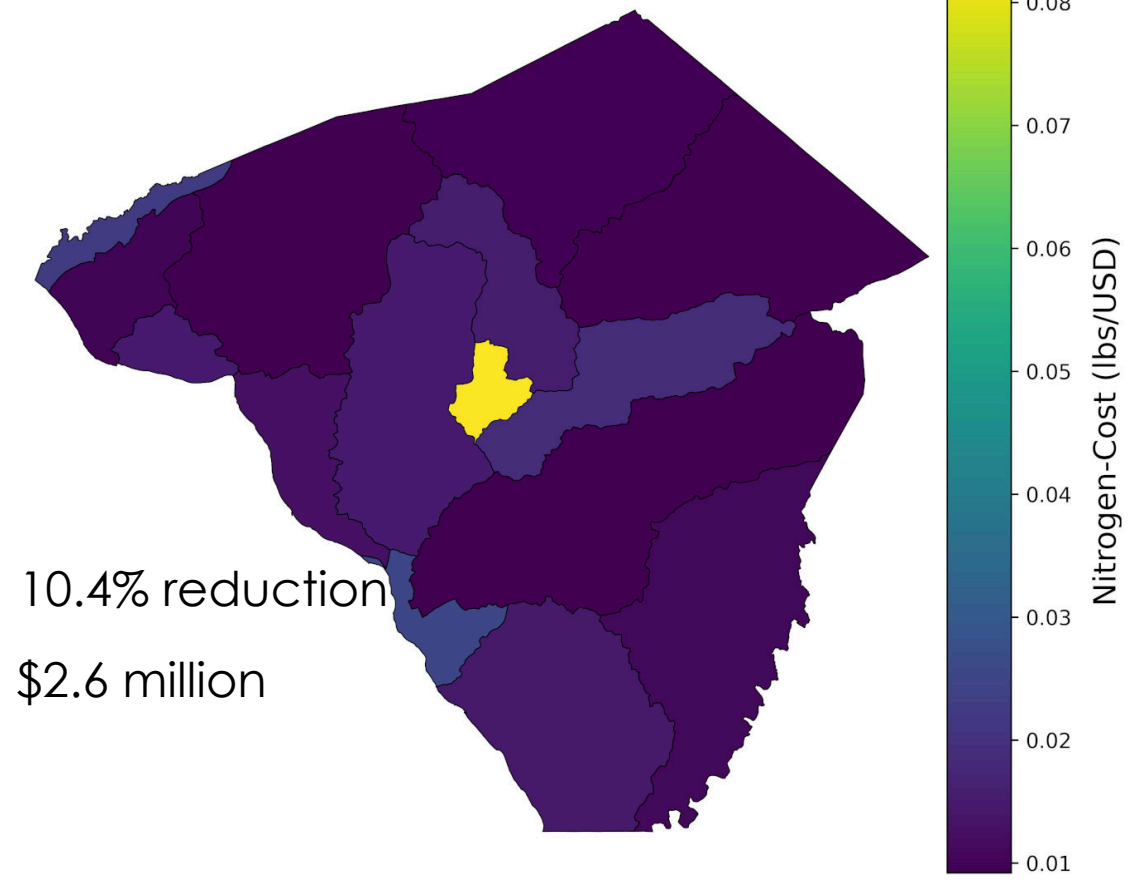
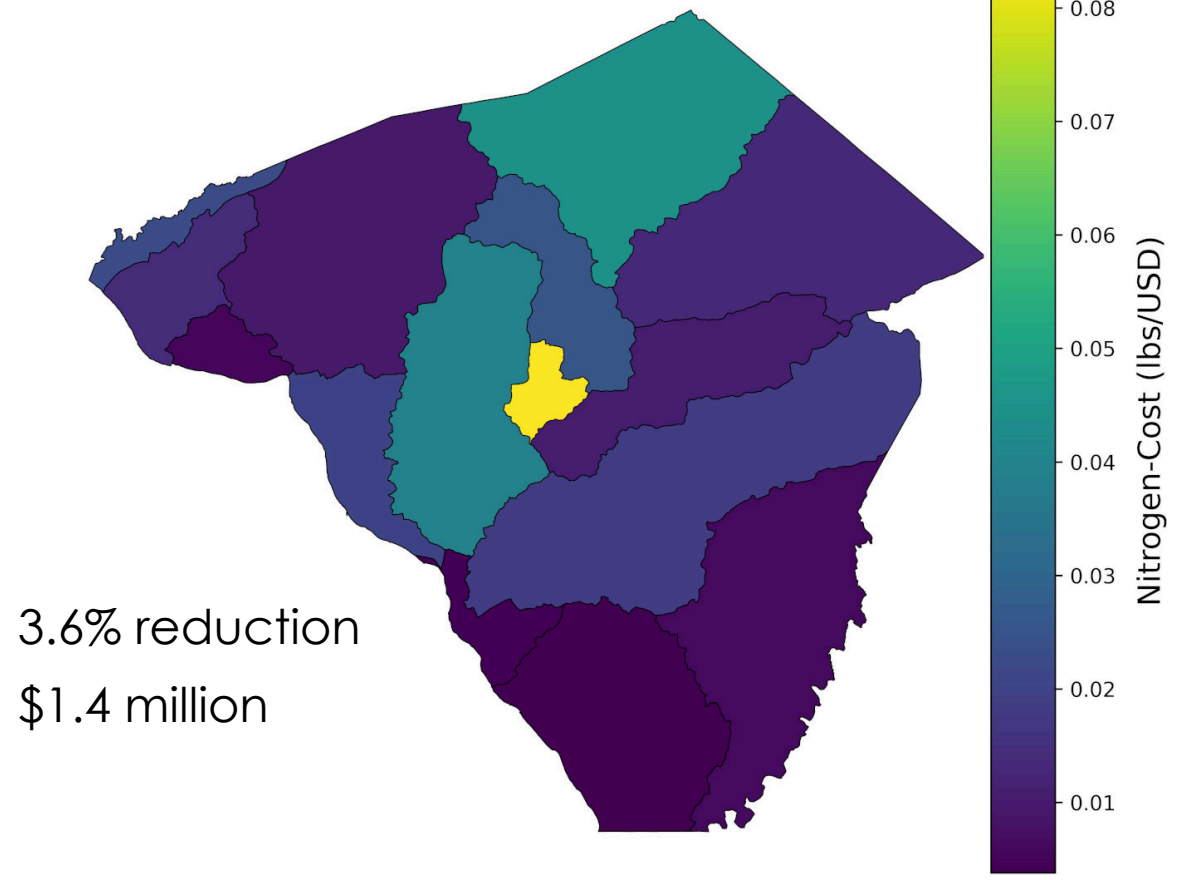
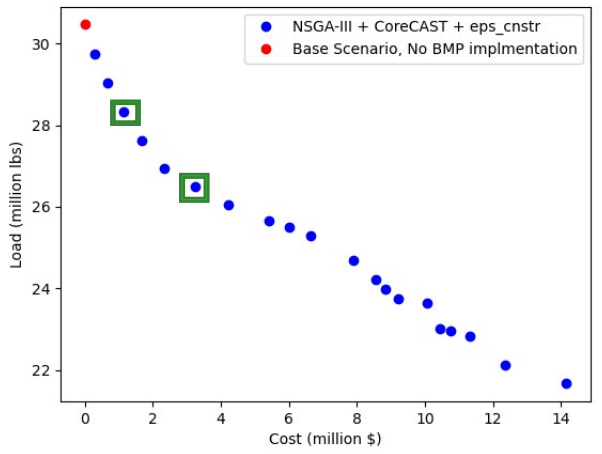
Cost



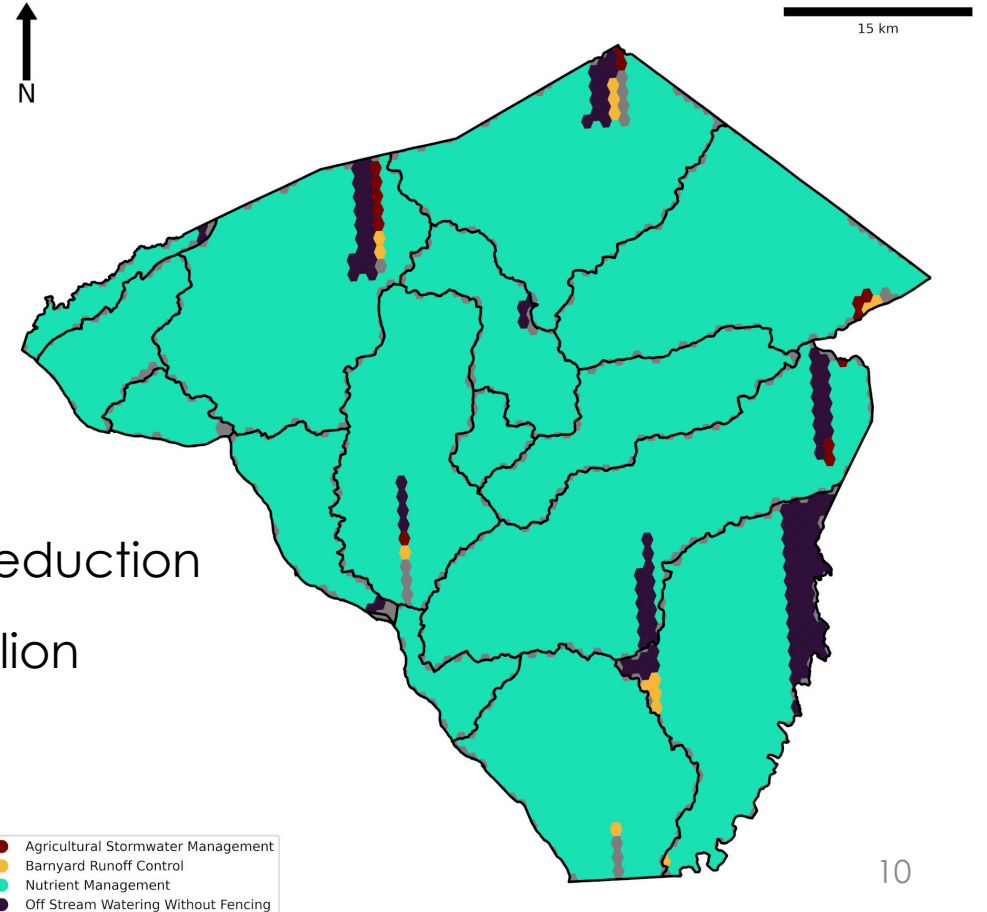
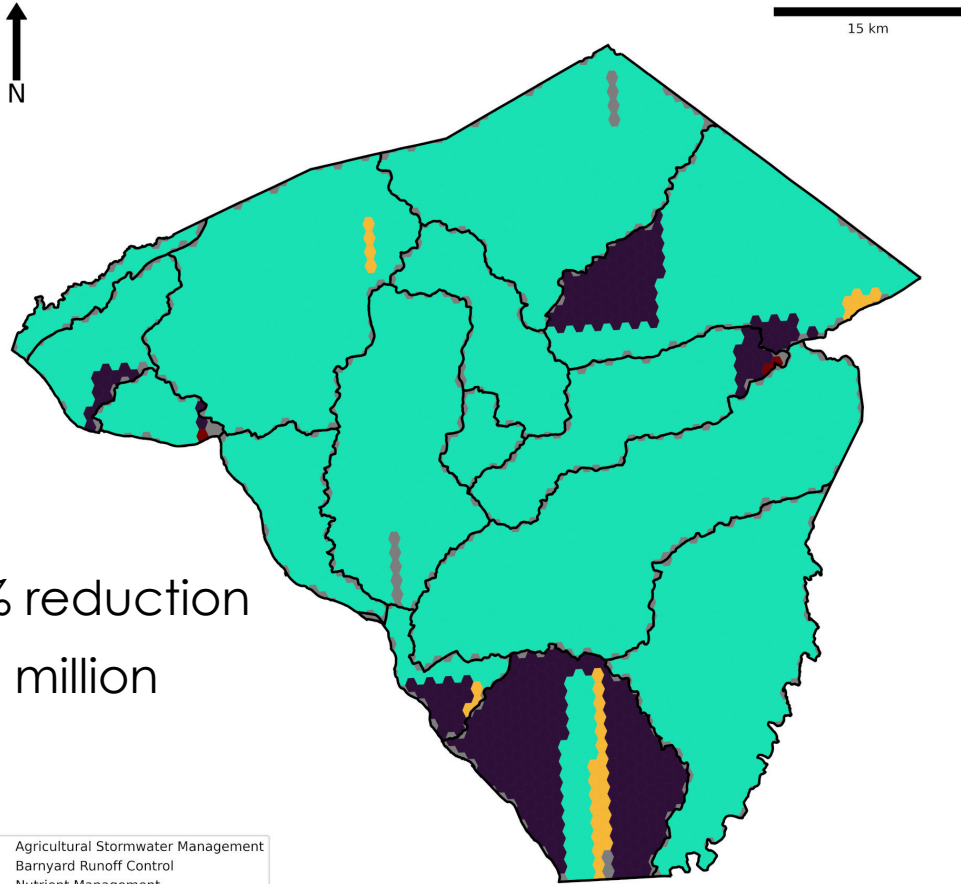
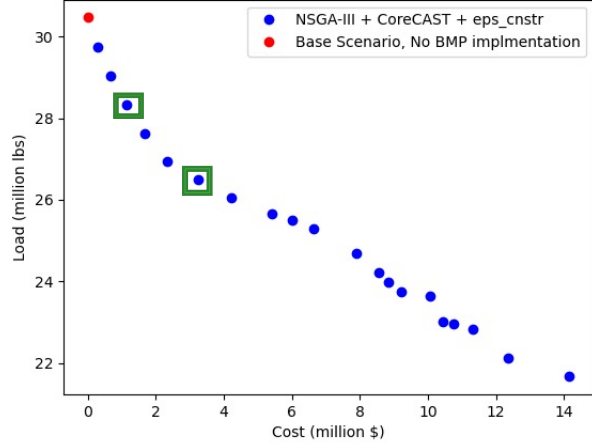
Nitrogen Reduction



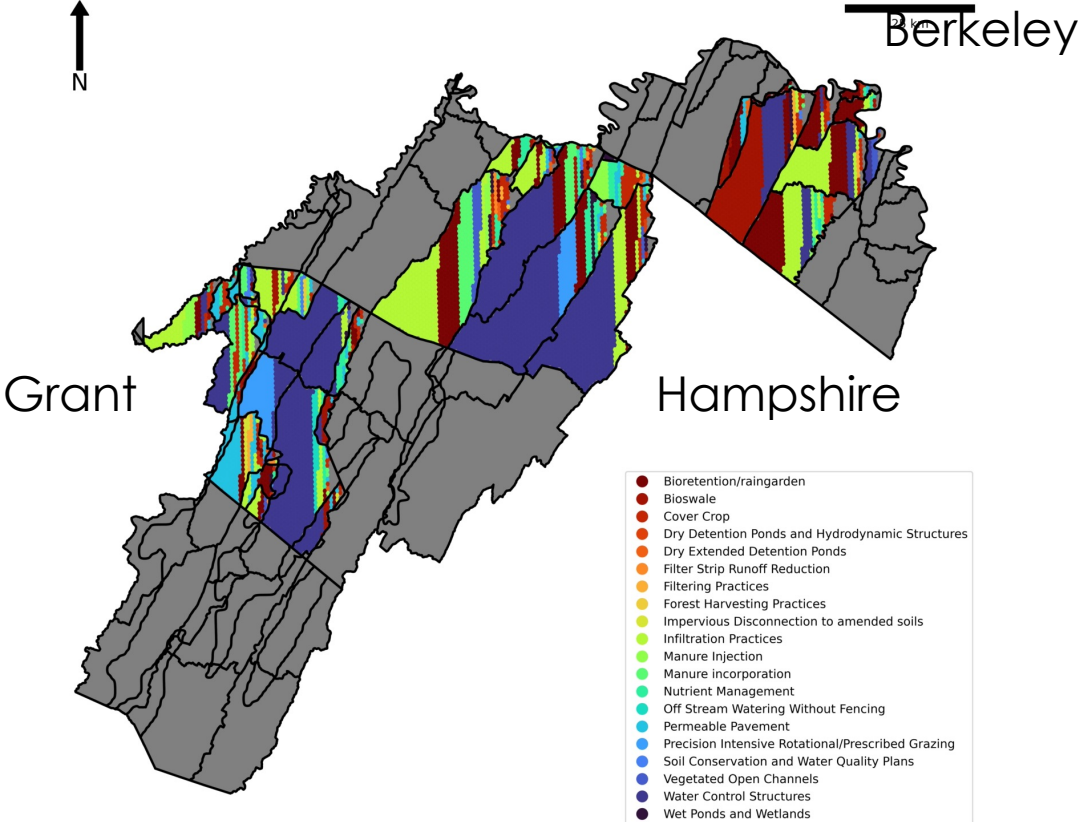
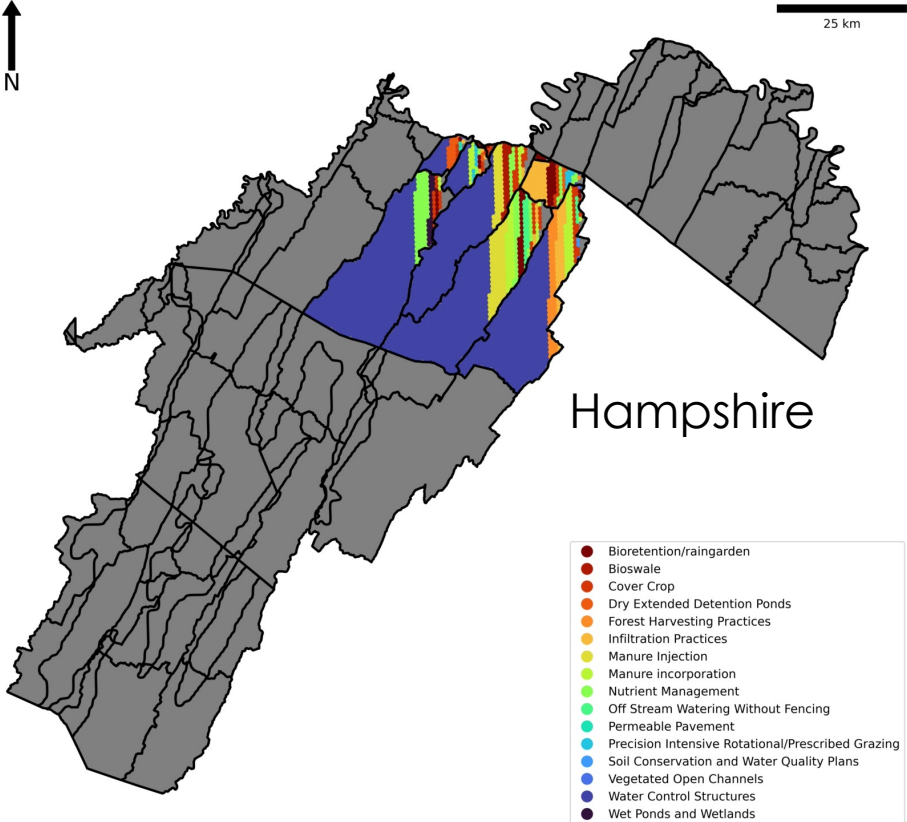
Nitrogen/Cost



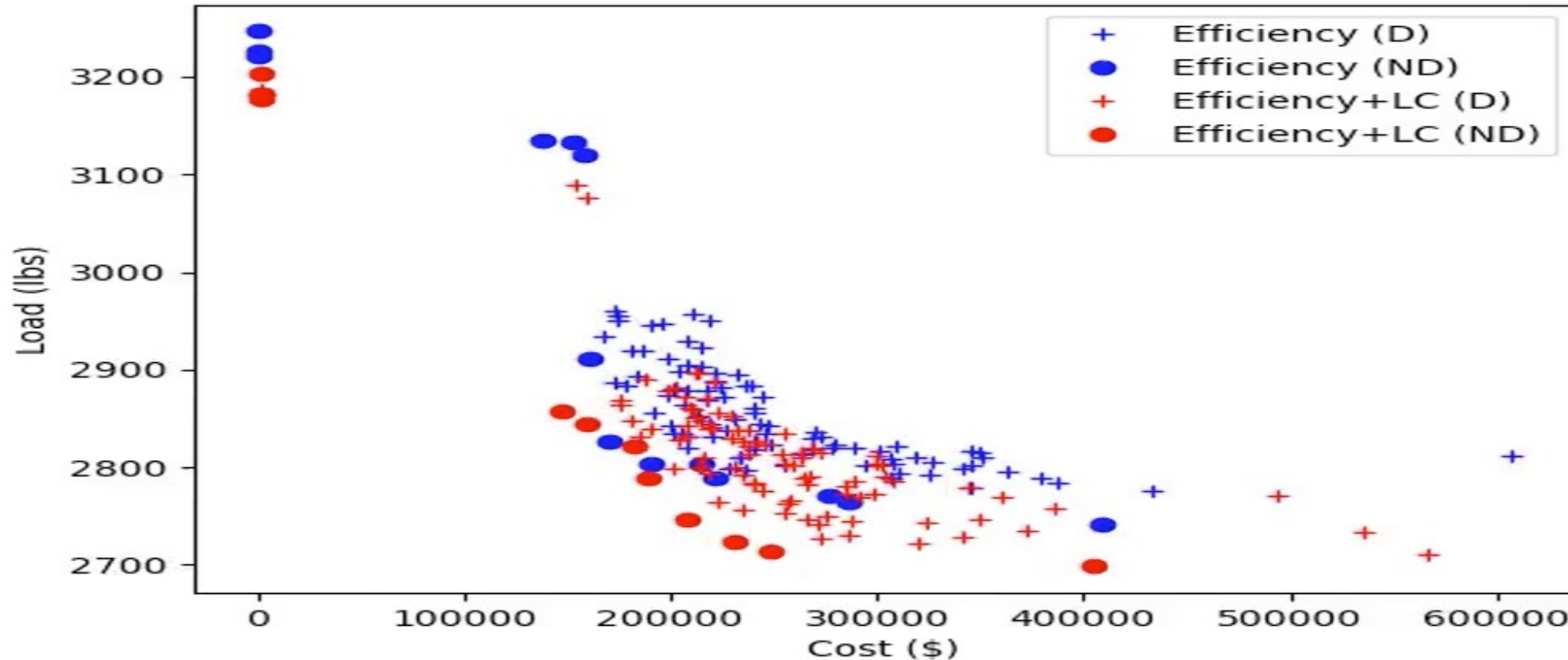
BMP Implementation



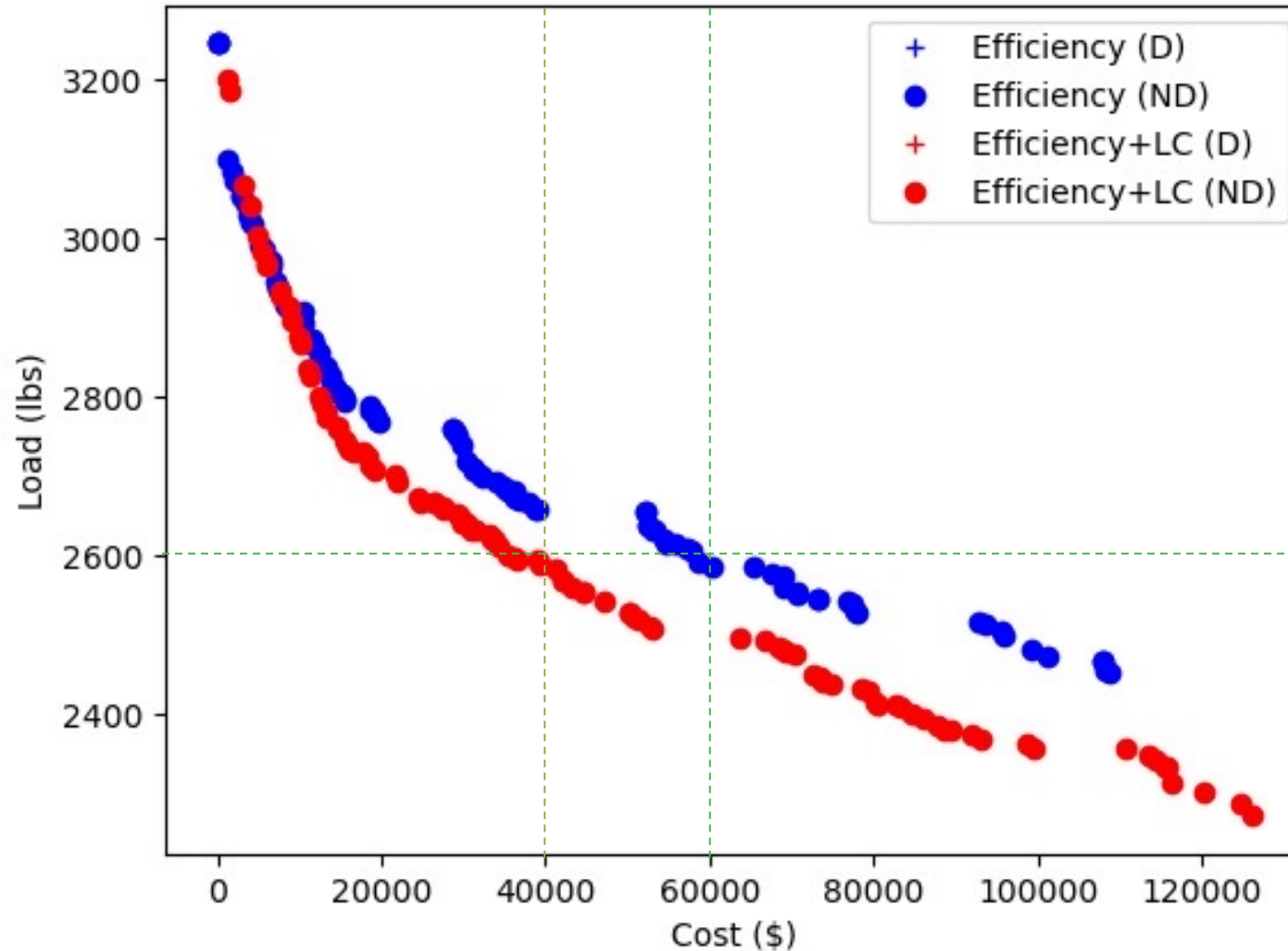
From previous studies the optimization can benefit when optimizing more counties

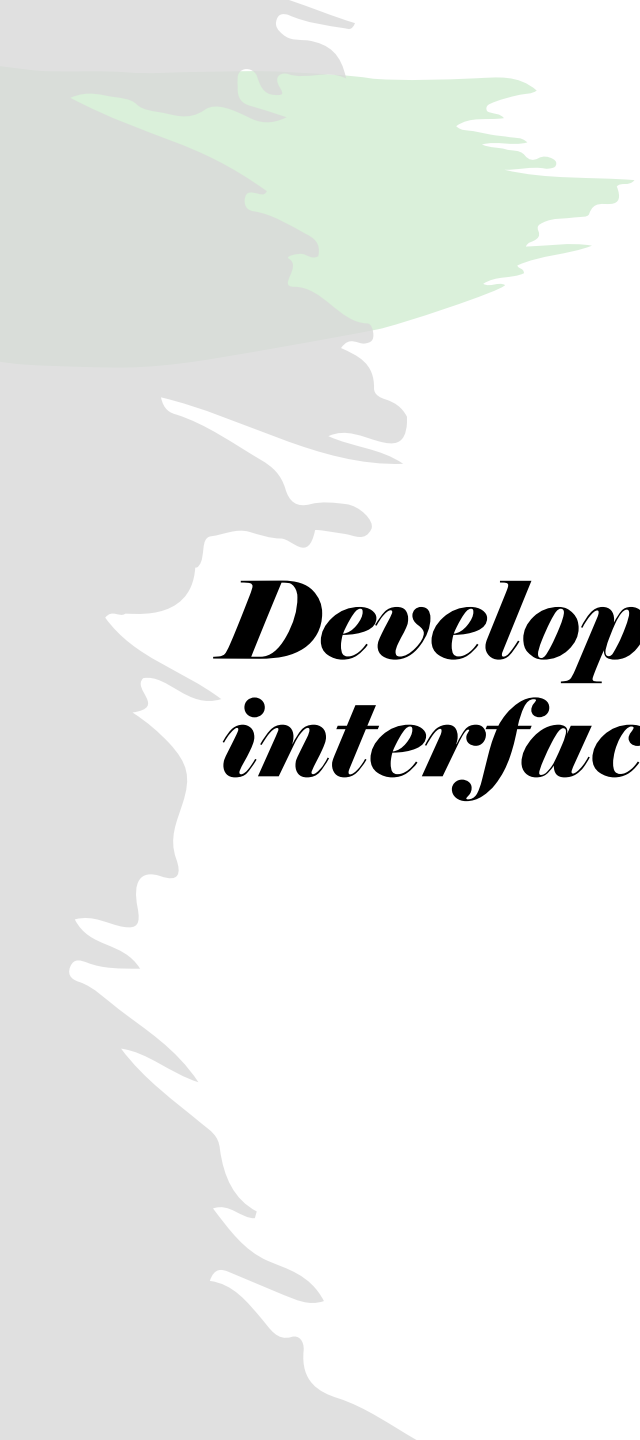


Comparison run: Efficiency vs Efficiency + Land Conversion (CoreCAST)



Comparison run on Tucker county: E vs $E + LC$ (CoreCAST)





1

Scenario Info

2

Loads

3

BMP Selection

4

Update Costs

5

BMP Constraint

6

**BMP Constraint
Advanced**

BMP Constraint (Advanced)

Add your preferences:

- Developed
- Septic
- Natural
- Manure Treatment
- Animal**
- Land Conversion
- Agriculture

Animal BMPs

BMP

UNIT

ANIMAL GROUP

LOAD SOURCE

LOWER BOUND

UPPER BOUND

#	BMP	Animal Group	Load Source	Lower Bound	Upper Bound	
1	Animal Waste Management System	beef	Permitted Feeding Space	5	10	

PREVIOUS

OPTIMIZE

Developed interface

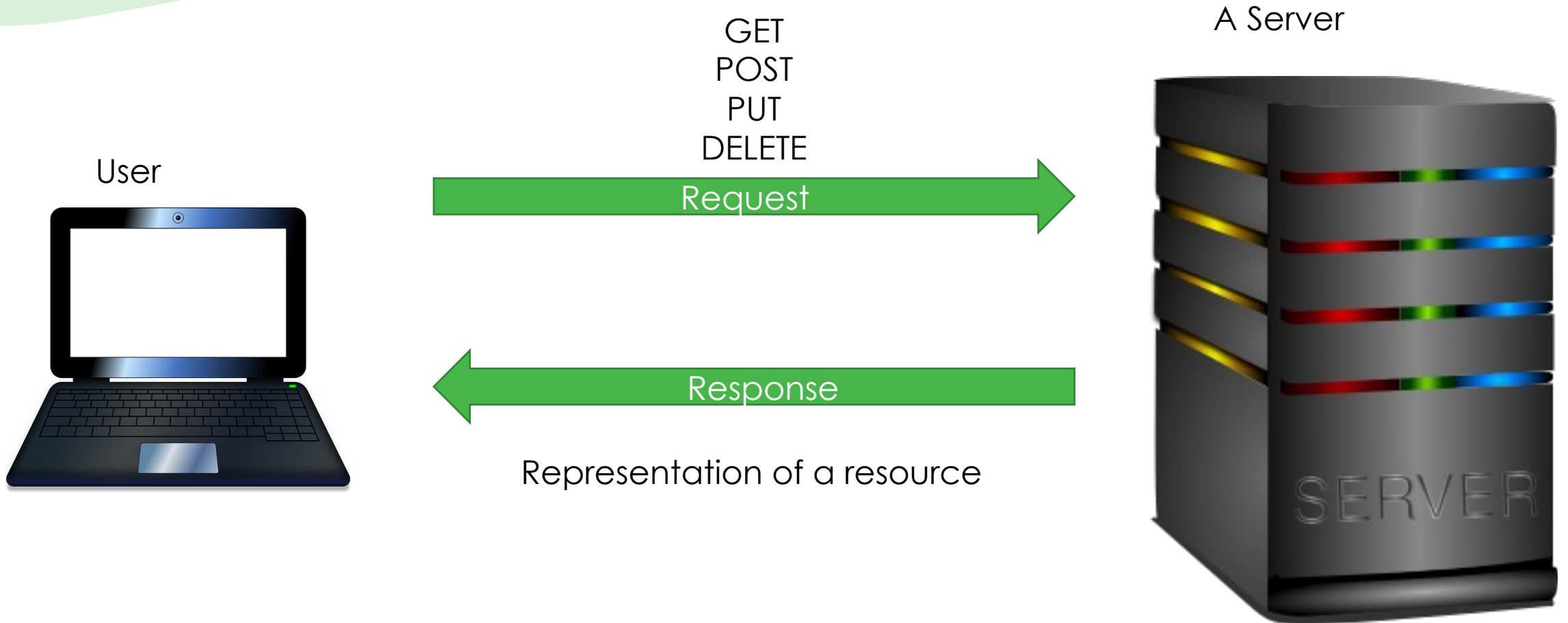
Web Interface

- We have an interface web interface that lets set up the users' preferences.
- We placed our efforts in producing an easy to use and self-content platform
- Users can review the optimization results directly on the platform
- The MSU team has new members that are analyzing the results.
- However, web interfaces are usually time-consuming for massive experimentation and validation

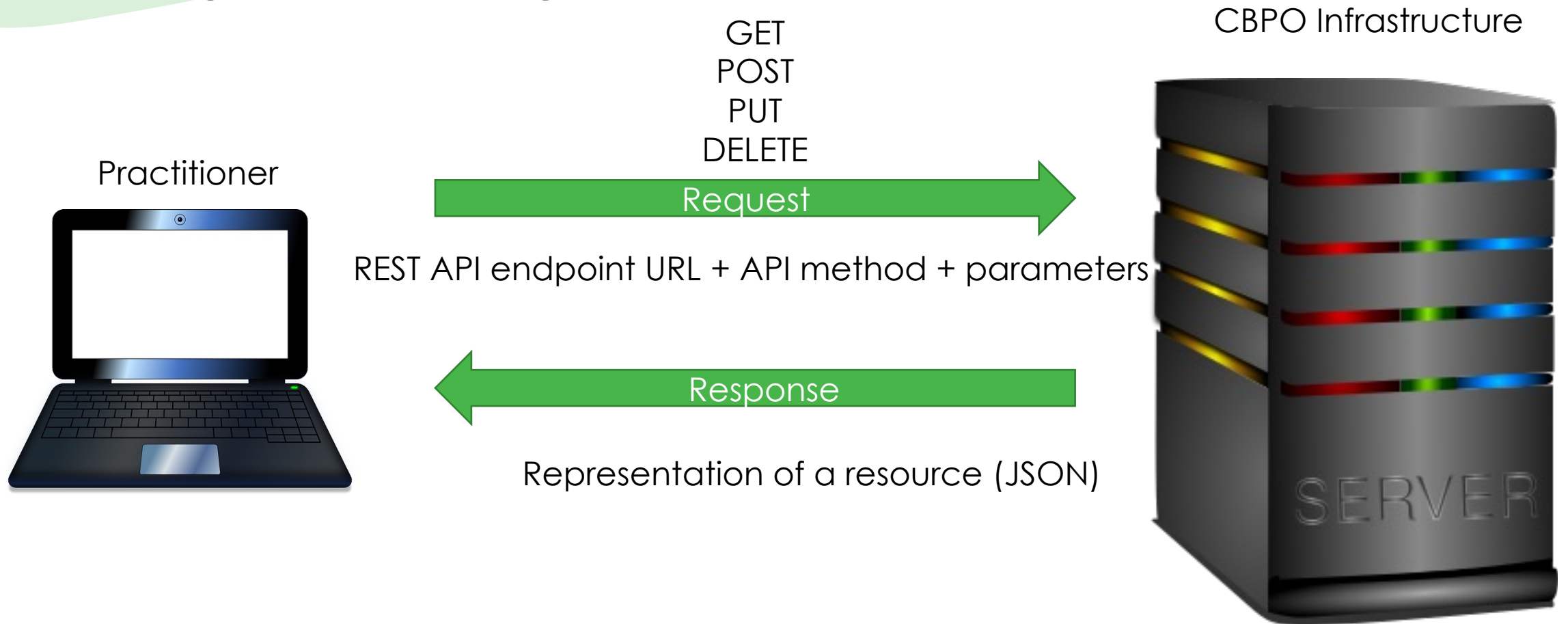
Solved problems

- Call CoreCAST
- Communication errors
- Synchronization errors
- We have executed CoreCAST thousands of times, geographically distant, many possible problems in these circumstances have appeared.
- Optimization working
- Plotting

HTTP



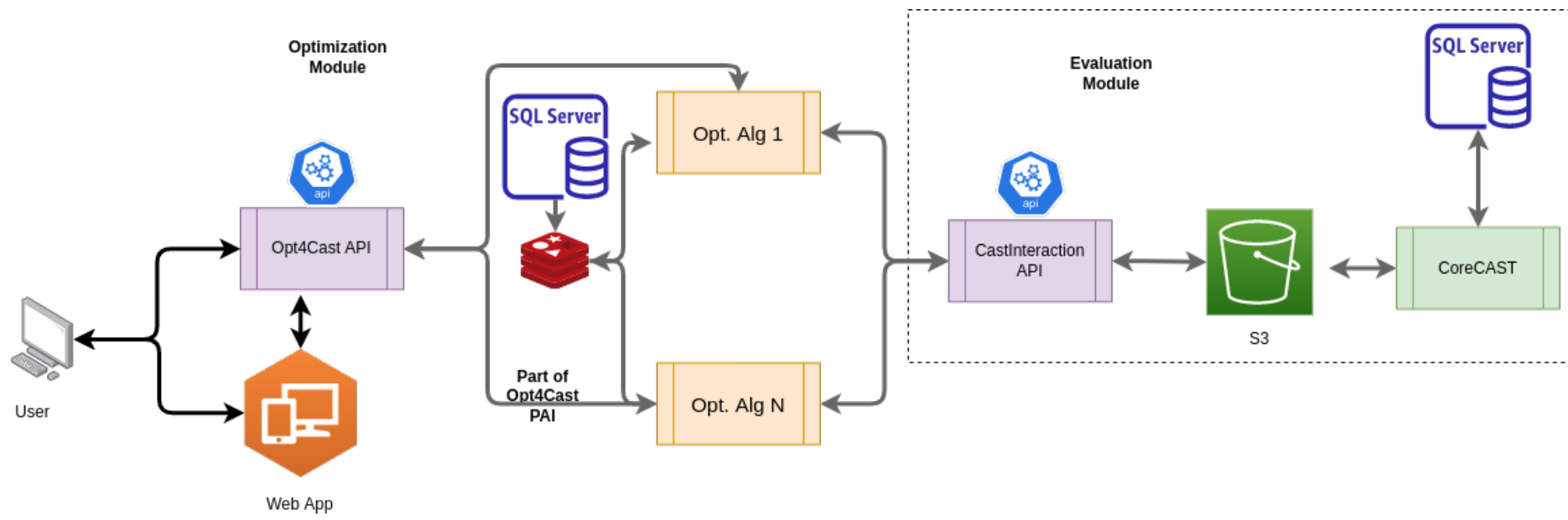
REpresentational State Transfer API *(Roy Fielding)*



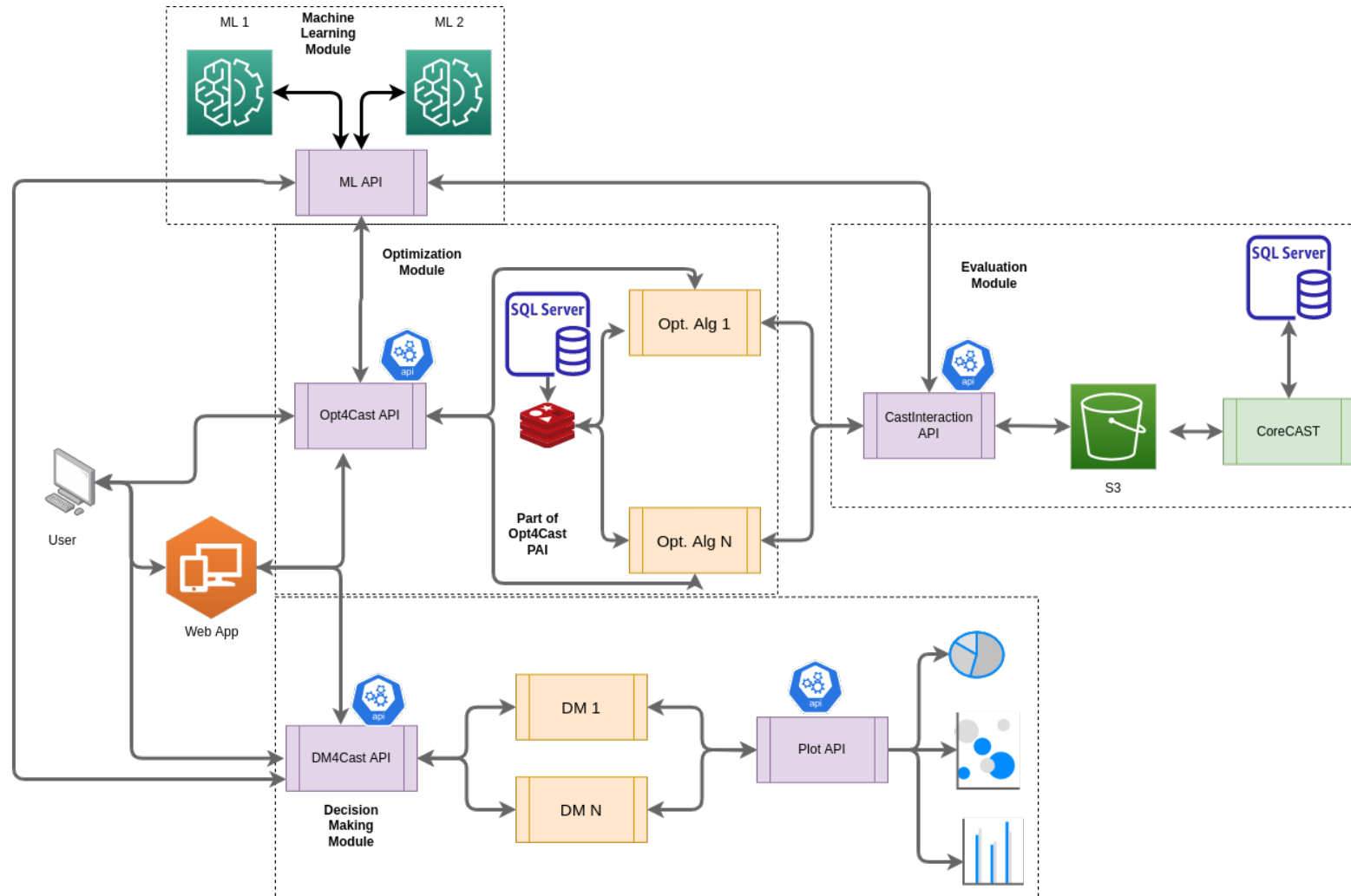
Optimization API (API4Opt)

- 3 different layers: base scenarios, scenario evaluation, optimization execution
 - i.e., users can set up a base scenario, evaluate the reduction of certain BMPs, or trigger a specific algorithm. Users can also download results.
- HTTP request let users use their preferred language to call our API and to manage the results.
- Users can access the results and data any where through the web.

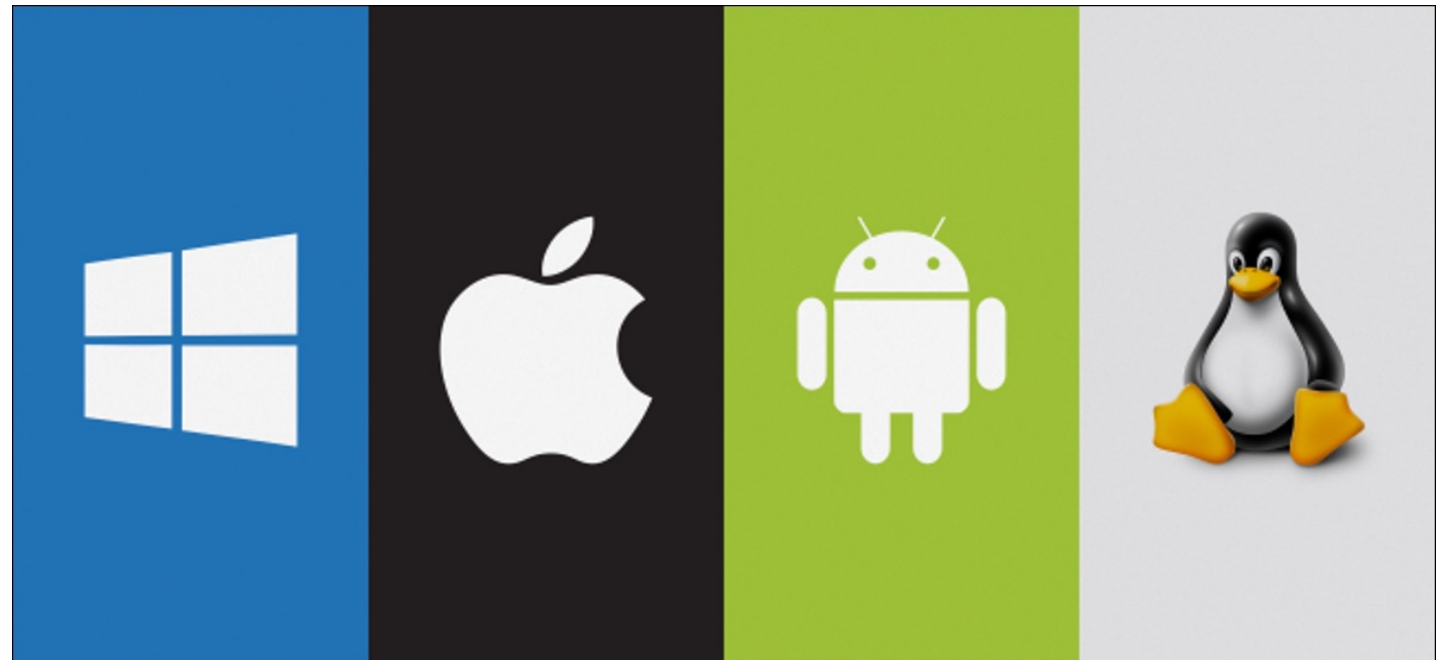
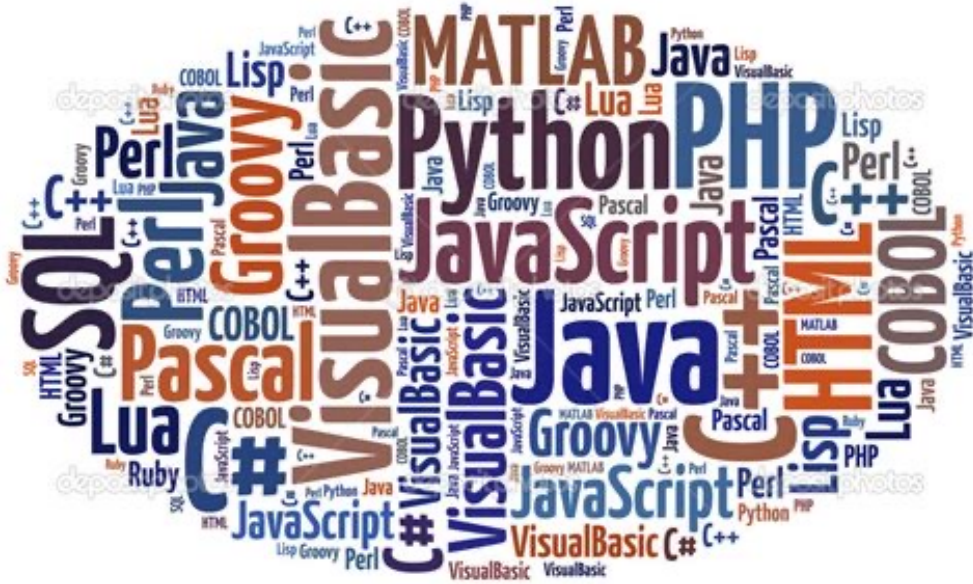
API4Opt = Opt4Cast + CASTInteraction APIs



API4Opt = Opt4Cast + CastInteraction APIs



Independence



RESTful API (Web interface)

Api Root / Evaluation List

Evaluation List

Filters OPTIONS GET

GET /api4opt/evaluations/?format=api

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
{
  "count": 0,
  "next": null,
  "previous": null,
  "results": []
}
```

Raw data HTML form

Title	<input type="text"/>
Base	LANCASTER(Gregorio) <input type="button" value="v"/>
Source	MathModel <input type="button" value="v"/>
Scenario	<input type="text" value="null"/>
Notify	<input type="checkbox"/>
Delete at	<input type="text" value="mm / dd / yyyy"/>

Django REST framework gtp

Api Root / Execution List

Execution List

Filters OPTIONS GET

GET /api4opt/executions/?format=api

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
{
  "count": 0,
  "next": null,
  "previous": null,
  "results": []
}
```

Raw data HTML form

Media type: application/json

Content:

```
{
  "title": "",
  "base": null,
  "algorithm": null,
  "parameters": null,
  "notify": false,
  "delete_at": null,
  "status": null
}
```

POST

RESTful API (Code using http requests)

```
import time

data = {"title": "LANCASTER(Gregorio)",
        "atm_dep_data_set_id": 0,
        "back_out_scenario_id": 6611,
        "base_condition_id": 256,
        "base_load_id": 6,
        "cost_profile_id": 4,
        "climate_change_data_set_id": 59,
        "historical_crop_need_scenario_id": 6608,
        "point_source_data_set_id": 36,
        "scenario_type_id": 2,
        "soil_p_data_set_id": 31,
        "source_data_revision_id": 8,
        "counties": [332],
    }

url = url_base+'bases/'
response = requests.post(url=url, json=data, headers=head)
if response.status_code == 200:
    cid = json.loads(response.text)['id']
    print('ID: {}'.format(cid))
```


RESTful API (Code using http requests)

Send a scenario to evaluate and wait for its execution

Using CoreCAST

To use CoreCAST to retrieve the information, please set "source" to "C".

By default, source is set to "C" (CoreCAST)

```
import time

base = 3
scenario_lst = [[998, 9, 3, 137, 1.0], [999, 9, 3, 137, 1.0], [999, 9, 6, 137, 1.0], [998, 9, 6, 137, 1.0]]
scenario = json.dumps(scenario_lst)
print (scenario)
data = {
    "source": "C",
    "title" : "Using Corecast",
    "base" : base,
    "scenario" : scenario,
    "notify" : False
}
# Land River Segment, Agency, Load Source, BMP, percentage [0-1]

url = url_base+'evaluations/'
response = requests.post(url=url, json=data, headers=head)

if response.status_code == 200:
    cid = json.loads(response.text)['id']
    print('ID: {}'.format(cid))
```

RESTful API (Admin interface)

Base Scenarios

The screenshot displays the 'API4Opt Admin' interface. The top navigation bar includes 'Home', 'Api', and 'Bases'. A sidebar on the left contains a search bar and a menu with categories like 'API', 'AUTHENTICATION AND AUTHORIZATION', 'CORE', and 'TAGS'. The main content area is titled 'Select base to change' and features a search bar and a table of base scenarios. The table has columns for ID, TITLE, COUNTRIES, STATUS, CREATED, COMPLETED, and STATUS. Five rows are visible, all with a status of 'Complete'. A 'FILTER' sidebar on the right offers options to filter by 'created' date and 'status'.

ID	TITLE	COUNTRIES	STATUS	CREATED	COMPLETED	STATUS
16	LANCASTER(Gregorio)	[332]	Complete	Sept. 26, 2022, 3:05 p.m.	Sept. 26, 2022, 3:06 p.m.	Complete
15	LANCASTER(Gregorio)	[332]	Complete	Sept. 26, 2022, 11:35 a.m.	Sept. 26, 2022, 11:35 a.m.	Complete
14	LANCASTER(Gregorio)	[332]	Complete	Sept. 26, 2022, 10:18 a.m.	Sept. 26, 2022, 10:18 a.m.	Complete
13	LANCASTER(Gregorio)	[332]	Complete	Sept. 26, 2022, 10:16 a.m.	Sept. 26, 2022, 10:17 a.m.	Complete
3	LANCASTER(Gregorio)	[332]	Complete	Sept. 26, 2022, 10:09 a.m.	Sept. 22, 2022, 2:08 p.m.	Complete

Conclusions and Future Work

- We have developed a RESTful API that provides an additional interface to developers and testers of our optimization approaches.
- The implemented RESTful API provides a mechanism to manage base scenarios, evaluate scenarios and execute optimization algorithms.
- Such a tool will help us reduce the time to evaluate and analyze our optimization algorithms.

- Add this work to the web interface (future work).
- Incorporation of remaining BMPs (future work).
- Improve the decision-making process (future work).



Thank you