Presented to the MDO teaching team 2023
In Partial Fulfillment Of the Requirements for the Course
16.888 MDO; Multidisciplinary Design Optimization

Feburary - May 2023, Cambridge, MA

TERRA SPARKLING: MULTI-DISCIPLINARY DESIGN OPTIMIZATION FOR CARBON SEQUESTRATION

Warren Anderson[1], John Beilstein[1], Brooke DiMartino[1], Tanner Papenfuss[1], Stephen Tainter[1]

[1]Massachusetts Institute of Technology, Cambridge, MA

ABSTRACT

Carbon capture utilization and storage (CCUS) technology is crucial for mitigating greenhouse gas emissions and stemming global warming. Efficient optimization of CCUS projects involving multi-disciplinary and multi-objective problems is essential for a feasible and cost-effective approach. In this project, we present a multi-disciplinary optimization framework involving five essential modules: facilities, pipeline, wells, subsurface, and finance, each accounting for specific parameters and constraints. These modules are integrated using a single objective net present value (NPV) to provide a comprehensive representation of the CCUS system that ultimately helps decision-makers make strategic choices.

Initially, the framework employs a design of experiments to establish a baseline set of solutions. Subsequently, we incorporate a single objective particle swarm optimization (PSO) algorithm and a Sequential Least Squares Programming (SLSQP) optimization. These optimize the parameters of the combined systems and seeks to maximize NPV. PSO allows us to explore the broad problem space and SLSQP offers a quick optimization.

Lastly, we introduce a multi-objective optimization (MOO) algorithm to maximize NPV and minimize CO2 emissions, effectively balancing environmental and economic considerations. The MOO algorithm provides insight into the trade-offs between competing objectives and alternative design solutions. Decision-makers can easily navigate the Pareto front and select the best compromise solution considering their preferences and constraints. Applying simultaneous optimization also reveals possible synergies between the modules, enabling higher efficiency levels and coordination between the interconnected systems.

In conclusion, we utilized a multi-disciplinary optimization framework involving the DOE, PSO, SLSQP, and MOO algorithms. Our project proposes an integrated approach as a valuable tool to maximize the profitability of CCUS projects, minimize CO2 emissions, and maintain environmental sustainability. This structured and systematic analysis supports stakeholders in the CCUS industries to learn critical insights, optimal decisions, and effective management for technology deployment and infrastructure development.

Keywords: Multi-Disciplinary Design Optimization, CCUS, Carbon Sequestration

1. INTRODUCTION AND MOTIVATION

Carbon dioxide (CO_2) makes up 80% of greenhouse gas emissions in the US, and its accumulation in the atmosphere is a leading cause of global warming. CO_2 is the primary greenhouse gas emitted by anthropogenic sources and according to OurWorldInData.org, humanity produced 37.1 billion tons of carbon dioxide emissions in 2021. 46% of US CO_2 emissions are produced by electricity generation and industries that use fossil fuel or produce CO_2 through chemical reactions (i.e., cement, steel, and petrochemical plants). [1]

One promising solution to this problem is utilizing Point-Source Capture (PSC) and permanently storing it underground in a process called sequestration. This method captures up to 90% of CO_2 emitted in flue gas during combustion processes at large industrial plants. Unlike Direct Air Capture (DAC), this process does not remove CO_2 from the air, but instead captures it before it is released into the atmosphere in order to lower emissions associated with these industrial plants. The CO_2 concentration in these flue gasses is much higher than in the ambient air, making the process a more immediately commercial-ready opportunity. [2]

The technology of injecting CO_2 into the subsurface is nothing new; oil companies have injected CO_2 into depleted hydrocarbon reservoirs for enhanced recovery for decades. It is important to note that this enhanced recovery method does little to impact the overall CO_2 emissions into the atmosphere, but it does prove that permanent sequestration is technically possible. Two pilot programs in the US were primarily funded by the Department of Energy that tested aspects of CO_2 sequestration at a commercial scale. The Petra Nova project, located southwest of Houston, Texas, took CO_2 captured from an electric generation station and piped it 81 miles to the West Ranch Oil Field for enhanced oil

recovery [3]. The Illinois Basin Decatur Project is the second project, and it has successfully injected 1.3 million metric tons of CO_2 [4].

2. OPPORTUNITY

Between 2021 and 2022, an estimated 78 new US carbon capture utilization and storage (CCUS) projects were announced [5]. While the number of CCUS projects in the planning and evaluation phases has been rising recently, there has been a significant increase in the last year due to the Inflation Reduction Act of 2022 (IRA). Previously the 45Q tax code credited \$50/ton of CO_2 that was captured and stored using PSC. With the IRA, the credit has been raised to \$85/ton for PSC-based CCUS projects. This credit value increase has a lot of companies seriously looking at ways to make a commercial-scale CCUS plant viable [2].

Due to the capital investment that would be required, even with the IRA credit at \$85/ton of CO_2, the margins for making CCUS profitable are going to be narrow. Every function within a total PSC to CCUS system must be optimized for efficiency and cost. This paper uses a multidisciplinary approach to optimize the objectives through simulation and modeling, representing real-world scenarios as closely as possible.

In 2013 the USGS conducted a CO_2 storage resource assessment of the US for potential CCUS reservoirs [6]. According to their finding, the Coastal plains of Texas, Louisiana, Mississippi, Alabama, and the Florida panhandle have a potential storage capacity of 1,900 Gt of CO_2 or about 65% of all US capacity. This area is also densely populated with electric generation and petrochemical plants, so this region was a prime candidate for this PSC to CCUS optimization problem.

3. PROBLEM FORMULATION

The purpose of this project is to design a system that maximizes the project Net Present Value (NPV) and minimizes the amount of CO_2 generated by the point-source compression to supercritical conditions for transportation and injection into the subsurface. The model cumulative injection cannot exceed the reservoir volume capacity, nor can the injection pressure at the bottom of the wellbore exceed the maximum injection pressure set by the pore pressure fracture gradient. These problem objectives and constraints are summarized in equations (1) and (2) below, respectively:

$$min\ J(x) = \begin{pmatrix} -Net\ present\ value,\ NPV \\ CO2\ generated,\ tot\_CO2\_gen \end{pmatrix} \quad (1)$$

$$s.t.\ g(x) = \begin{pmatrix} q_{inj} * num\_wells * time - res\_volume \\ p_{wellbore} - p_{max\_injection} \end{pmatrix} \leq 0 \quad (2)$$

For the design, the team chose to focus on nine key variables. These are summarized in the Master Table, figure 1, along with the parameters used to build the model. Because all modules are being built from the ground up, there are many parameters included in our problem. It is worth noting that the problem has been simplified in several key ways:

1. The system assumes steady-state conditions, constant across the project's life.
2. The subsurface has been simplified to Darcy's Law, with no change to reservoir pressure over time.
3. All compression calculations assume natural gas as the fuel type (which causes CO_2 emissions for our system).
4. If there are multiple facility tie-ins, it assumes they all converge to one processing facility before entering our system.
5. No downtime is considered for the system over the 20-year period it is online.

4. MODELING AND SIMULATION

4.1 Modules

Each module was separately modeled per the simplified block diagram in figure 2, which is overlain on a simplified CO_2 capture and storage system for reference. For more detailed model descriptions, see the block diagram (Appendix A) and N2 diagram (Appendix B) contained in the appendix. Completing the system-level structure allowed the team to develop the models of each module in parallel and with some modeler autonomy in the equations and techniques used.

4.1.1 Facilities. The Facilities module comprised of the release of CO_2 from the sorbent in a point-source capture system, to the outlet of the point-source physical footprint such that the CO_2 has reached a supercritical state before transport in the Pipeline Module. Based on the paper from Jackson et. Al 2018 [7], the team developed equations to represent the changes from the start point to end-state (super critical point). As seen in Figure 3 and 4 below, the design of the facilities section considers stages of compressors and heat exchangers to work up from the start point up to the supercritical point. The module uses the design variables to find the most 'energy efficient pathway' to the supercritical point. The advantage of operating a CCUS system in a supercritical state is that CO_2 has the density of a liquid, but retains the viscosity of a gas, which enables the system to process CO_2 in a more capitally efficient manner.

To model CO_2 behavior through a compressor, a polytropic behavior was assumed, and the equation (3) seen below was used. This equation calculates the work required to compress the CO_2 from an inlet state i to the outlet state j. The volume and temperature anticipated at the outlet can be calculated using the formula (4) and (5).

$$\dot{W}_{ij} = p_i \dot{v}_i \frac{\dot{v_j}^{1-n} - \dot{v_i}^{1-n}}{1-n} \quad (3)$$

$$\dot{v}_j = \left(\frac{p_i \dot{v_i}^n}{p_j}\right)^{\frac{1}{n}} \quad (4)$$

$$T_j = \left(\frac{p_j}{p_i}\right)^{\frac{n-1}{n}} \quad (5)$$

Next, per Figure 3, the CO_2 is cooled to 25°C before entering the next stage, while maintaining the same pressure from the outlet of the compressor. Thus, the heat required to cool the system can be calculated as follows.

	Module	Name	Symbol	Description	Upper Bound	Lower Bound	Nominal Initial Val	Unit of Measurement
Design Variables	Wells	Number of wells	num_wells	Number of injection wells in system	20	1	10	wells
	Pipeline	Number of connections	num_pc	Number of clients/sources we plan to bring in	50	1	3	diml
	Pipeline	Diameter	p_d	Inner diameter of pipe	20	4	8	in
	Pipeline	Length	p_l	Length from sales point to facility	55000	1000	10000	ft
	Facility	Outlet Pressure 4	p4	Outlet pressure for second stage compressor	1800	900	1300	psi
	Facility	Outlet Pressure 6	p6	Outlet pressure for third stage compressor	3200	2300	2800	psi
	Facility	Outlet Pressure 8	p8	Outlet pressure for fourth stage compressor	4800	3800	4400	psi
	Facility	Outlet Pressure 10	p10	Outlet pressure for fifth stage compressor	6000	5000	5500	psi
	Facility	Outlet Pressure 12	p12	Outlet pressure for sixth stage compressor	7000	6000	6500	psi
Parameters	Subsurface	Permeability	perm	Permeability of the reservoir rock	-	-	200	mD
	Subsurface	Reservoir capcity	Q_max	Frio Reservoir CO2 Storage Capacity (P5 Volume estimates)	-	-	87,000	megaton, Mt
	Subsurface	Skin	skin	Assumed skin around the wellbore due to completion	-	-	0	diml
	Subsurface	CO2 viscosity	visc_CO2SC	Viscosity of CO2 in the reservoir (assume temp 100 degF)	-	-	0.682	cP
	Subsurface	CO2 compressibility reservoir	Z_CO2SC	Compressibility of CO2 at reservoir conditions	-	-	0.992	diml
	Subsurface	Reservoir pressure	p_res	Initial pressure of the Frio reservoir	-	-	2234	psi
	Subsurface	CO2 density	density_CO2SC	Density of supercritical CO2	-	-	37.5	lb/ft^3
	Subsurface	Reservoir thickness	h_res	Frio reservoir thickness	-	-	150	ft
	Subsurface	Reservoir temperature	t_res	Frio reservoir temperature	-	-	135	deg F
	Subsurface	Maximum injection pressure	p_injmax	Maximum pressure to avoid fracturing Frio formation	-	-	3600	psi
	Wells	Relative roughness	rel_rough	Relative roughness of wellbore	-	-	0.0006	diml
	Wells	Well total depth	depth	Average depth of wells to be drilled	-	-	5500	ft
	Pipeline	Gravity	gravity	Gravity	-	-	32.2	ft/s^2
	Pipeline	Absolute Roughness	abs_rough	Absolute roughness	-	-	0.1	in
	Facility	CO2 Cp	Cp_CO2	Specific heat capacity of CO2 at constant pressure	-	-	0.844	KJ/(kgK)
	Facility	CO2 Cv	Cv_CO2	Specific heat capacity of CO2 at constant pressure	-	-	0.655	KJ/(kgK)
	Facility	Adiabatic Index	AI		-	-	1.289	diml
	Facility	CO2 Ideal Gas Constant	CO2_IDEAL_GAS	Ideal Gas Constant	-	-	8.3145	J/molK
	Facility	CO2 Compressibility Factor	CO2_compress	Compressibility Factor (non-ideal Gas)	-	-	Table	diml
	Facility	CO2 Enthalpy	h_CO2	Internal energy of CO2	-	-	Table	kJ/kg
	Facility	1st Compressor inlet Pressure	pin_1_comp	Pressure at inlet of 1st compressor	-	-	14.8	psi
	Facility	1st Compressor inlet_Temp	tin_1_comp	Temp at inlet of 1st compressor	-	-	25	C
	Facility	Elevation	z_fac	Sea-level, onshore, elevation of facility equipment	-	-	0	ft
	Finance	HX Fuel costs	hxfc	Cost of fuel needed to generate temperature difference for HX	-	-	2.50	$/mmbtu
	Finance	Comp Cost Coefficient	cpcc	The per-unit cost for running the compressor (electricity)	-	-	0.20	$/kWh
	Finance	Discount Rate	DR	Assumed discount rate for project	-	-	10	diml
	Finance	Well life	time	Assumed years of injection for each well	-	-	20	years
	Finance	Wellsite costs	c_site	Wellsite costs,includes leasing costs and other annual costs	-	-	23000	$2005/well
	Finance	CO2 tax credit	tax_credit	45Q tax credit for point source sequestered CO2	-	-	85	$/tonne
Dependent Variables	Subsurface	Injection rate	q_inj	Initial injection rate per well	-	-	-	mcf/d
	Facility	Final outlet pressure	p_out	Last compressor outlet pressure to transfer to pipelines	-	-	-	psi
	Facility	Facility CAPEX	CAPEX_fac	Capital costs associated with facilities	-	-	-	$
	Facility	Facility OPEX	OPEX_fac	Operating costs associated with facilities	-	-	-	$
	Pipeline	Pipeline CAPEX	CAPEX_pipe	Capital costs associated with pipelines	-	-	-	$
	Pipeline	Pipeline OPEX	OPEX_pipe	Operating costs associated with pipelines	-	-	-	$
	Pipeline	Velocity	vel	Velocity of supercritical CO2 leaving pipes into well(s)	-	-	-	ft/s
	Wells	Wellbore pressure	p_wf	Pressure at the bottom of the well at the formation face	-	-	-	psi
Constraints	Wells	Injection pressure limit	p_wf < p_injmax	Can't inject above fracture pressure of reservoir	-	-	-	psi
	Subsurface	Volumetric limits	q_inj*num_well* time < Q_max	Cannot inject more than the reservoir can hold	-	-	-	Mt
Objective Functions	Finance	Maximize NPV	NPV	Want to maximize NPV of the project	-	-	-	$
	Subsurface	Minimize CO2 generated	tot_CO2_gen	Want to maximize net CO2 injected	-	-	-	Mt

FIGURE 1: MASTER TABLE OF DESIGN VARIABLES, PARAMETERS, DEPENDENT VARIABLES, CONSTRAINTS, AND OBJECTIVES

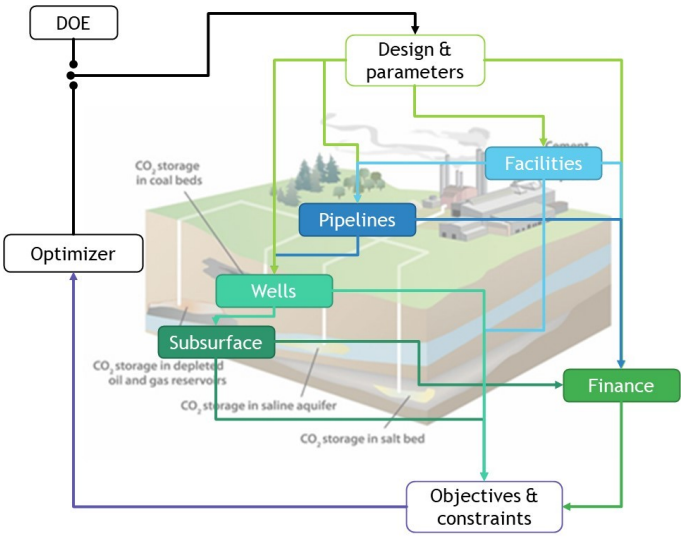


FIGURE 2: SIMPLIFIED BLOCK DIAGRAM OF SYSTEM

$$\dot{Q}_{ij} = \dot{m}(h_j - h_i) \tag{6}$$

$$\Delta h_{ij} = c_p(T_j - T_i) \tag{7}$$

Once the energy requirements have been determined for one stage, this is repeated in series seven times per Jackson et. Al 2018 [7], until the CO_2 reaches a supercritical state. After calculating the energy required for each stage, the costs for delivering that energy can be modeled as well. To do so, each of the unitized energy calculations must be multiplied by the mass flow rate of the system. The team utilized various research papers that parameterized capital (CAPEX) and operating (OPEX) costs for compressors and heat exchangers. Referencing Darrow et. Al 2015 [8], the capital and operating costs for a compressor can be seen below:

$$W_{ij_{CAPEX}} = \frac{\$2020.6}{kWh} W_{ij} \tag{8}$$

$$W_{ij_{OPEX}} = \frac{365days}{year}\left(\frac{\$3.06}{MMBTU} * \frac{1MMBTU}{293.07kWh} + \frac{24hrs}{day} * \frac{\$0.01108}{kW}\right)W_{ij} \tag{9}$$

In the OPEX equation, it is important to note that there are fuel costs, and O&M costs.

Similarly for the heat exchanger, the CAPEX and OPEX equations originated from Bautista 2014 [9].

$$Q_{ij_{CAPEX}} = \frac{\$247.35}{kWh} Q_{ij} \tag{10}$$

$$Q_{ij_{OPEX}} = Q_{elec_{OPEX}} + Q_{refrig_{OPEX}} + Q_{water_{OPEX}} \tag{11}$$

$$Q_{elec_{OPEX}} = \frac{\$75.66/kWh_{elec}}{kWh_{cooled}} Q_{ij} \tag{12}$$

$$Q_{refrig_{OPEX}} = \frac{\$0.009/kWh_{ref}}{kWh_{cooled}} Q_{ij} \tag{13}$$

$$Q_{water_{OPEX}} = \frac{\$0.0097/kWh_{m^3}}{kWh_{cooled}} Q_{ij} \tag{14}$$

In addition to calculating costs, the emissions generated with the compression (fuel emissions) and cooling (electricity grid intensity) were calculated to understand how much CO_2 was generated in this process.

$$CO_{2_{W_{ij}}} = CO_{2_{fuel}} W_{ij} * 24 * 365 \tag{15}$$

$$CO_{2_{HX_{ij}}} = \frac{Elec_{HX_{ij}} * CO_{2_{TX_{grid}}}}{1000} * 24 * 365 \tag{16}$$

$$Elec_{HX_{ij}} = \frac{757.28 * CO_{2_{TX_{grid}}}}{1000} * 24 * 365 \tag{17}$$

$$CO_{2_{TX_{grid}}} = \frac{941lb}{MWh} * \frac{0.453kg}{lb} \tag{18}$$

Each cost and emission value from each stage were then aggregated to calculate the total costs and emissions across the entire and routed to the Finance module.

4.1.2 Pipelines. The pipeline module takes the final supercritical pressure state from the Facilities module and calculates the pressure drop across the distance of the pipeline. When CO_2 drops below the supercritical state, a booster pump is installed to return the CO_2 to a supercritical state.

Pressure drop is determined by multiple factors such as pipe length and diameter, but it also leverages the fanning coefficient, which calculates the estimated frictional effects using pipe diameter and the Reynolds number (Neutrium). Since the viscosity used for the Reynolds number can change with state conditions of the CO_2, the viscosity is a dynamic formula based on temperature changes (Nave).

$$\Delta p = 2f\rho_{new}v^2 \frac{length_{pipe}}{diameter_{pipe}} \tag{19}$$

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left(\frac{\epsilon/D}{3.7065} - \frac{5.0452}{Re}\log_{10}\left(\frac{\epsilon/D^{1.1098}}{2.8257} + \frac{5.8506}{Re^{0.8981}}\right)\right) \tag{20}$$

$$v = v_0\left(\frac{a}{b}\right)\left[\frac{T}{T_0}\right] \tag{21}$$

$$\begin{aligned} a &= 0.555T_0 + C \\ b &= 0.555T + C \end{aligned} \tag{22}$$

Costs and emissions associated with the pipeline and compressors leverage the equations below and equations used in the Facilities module [10].

$$CAPEX_{pipeline} = \left(32.086p_l n_{pc}^{-0.033p_d}\right)p_l n_{pc} \tag{23}$$

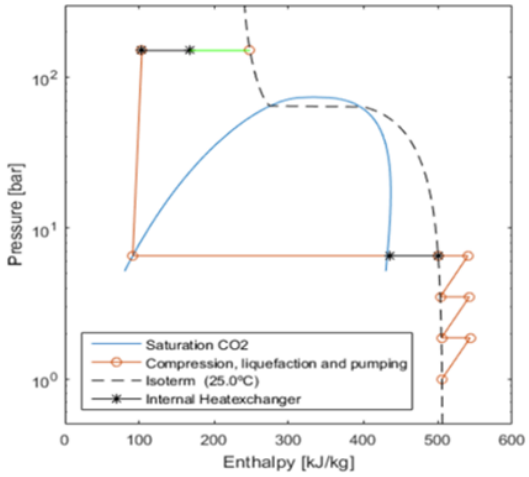


FIGURE 3: COMPRESS AND COOL CO_2 TO A SUPERCRITICAL STATE

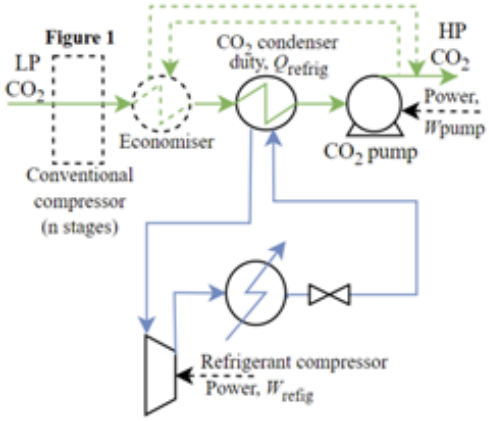


FIGURE 4: COMPRESSOR AND HEAT EXCHANGER LAYOUT FOR EACH STAGE IN FIGURE 3

4.1.3 Wells. Once the CO_2 has been delivered from the point source to wellhead, the Wells module calculates the pressure at the reservoir face at the end of the well on a per-well basis. Similar to the pipeline module, the well transports the CO_2 vertically into the ground through tubing which can be modeled as a pipe [11] [12] [13].

$$p_{wf}(t) = p_{wh} - \Delta p_f + \rho g D \tag{24}$$

$$f = 0.0055\left(1 + \left(2 * 10^4 \frac{k}{D} + \frac{10^6}{Re}\right)^{\frac{1}{3}}\right) \tag{25}$$

$$\Delta P_f = \frac{2 f \rho v^2 L}{g_c D} \tag{26}$$

$$Re = \frac{1488 \rho V D}{\mu} \tag{27}$$

The cost per well was based on an empirical study of drilling costs for CO_2 injection wells based on well depth as seen in equation 28 below [10]:

$$CAPEX_{well} = (-3.9 * 10^{-8} D^3 + 4.00 * 10^{-4} D^2 - 0.84 D + 903) D \tag{28}$$

4.1.4 Subsurface. The Subsurface module then determines the volumetric flow rate of CO_2 entering the reservoir given the bottom hole pressure derived from the Wells module, and the static reservoir pressure. The formula used below is derived from reservoir engineering principles for a gas injection well [14].

$$q_g = \frac{0.703 k h (p_R^2 - p_{wf}^2)}{T \mu_g Z [ln(\frac{r_e}{r_w}) - 0.75 + s]} \tag{29}$$

While this module was simplified in terms of fluid dynamics and geomechanics, given the time constraints of the class, a constraint was applied to ensure that the total volume injected does not exceed the total volume of the target reservoir. The Frio Formation along the Coastal Plains of Texas was selected as the target reservoir for this simulation. According to the USGS National Assessment of Geologic Carbon Dioxide Storage Resources, the Frio Formation has a P50 storage probability estimate of 87,000 Megatons of CO_2 [6]. An additional constraint was added to this module to ensure that the pressure at the bottom of the well did not exceed the maximum injection pressure. The maximum injection pressure is determined by the reservoir pore pressure fracture gradient. Again using the Frio Formation as the target,

we used data from a reservoir model generated by Ghomian et al. 2006, for the pore pressure fracture gradient, formation depth, and maximum injection pressure parameter [15].

4.1.5 Finance. The finance module ultimately provides the primary objective output: NPV. It calculates the revenue, total CAPEX, annual CAPEX, and annual OPEX to keep the system online over its lifetime. From this, it can calculate the upfront capital spend at time 0 along with the annual cashflow. The annual cashflow is then discounted and summed over the life of the project to generate the net present value. The critical equation for this module is as follows:

$$NPV = \sum_{t=1}^{T} \frac{cashflow_t}{(1 + discount\_rate)^t} - CAPEX_0 \qquad (30)$$

4.2 Validation

The model was initially validated by checking the deterministic output with a manual excel model and with research papers for similar orders of magnitudes of results. The initial design vector x_0 was pulled from our design-of-experiments (DoE).

TABLE 1: DESIGN VECTOR X_0

Design Variable	Value
Num of Wells	5
Num of Connections	2
Pipeline Diameter (in)	12
Pipeline Length (m)	28000
P2 (kPa)	700
P4 (kPa)	1800
P6 (kPa)	3200
P8 (kPa)	4800
P10 (kPa)	5500
P12 (kPa)	7000

TABLE 2: OUTPUT FROM DESIGN VECTOR X_0

Output Variable	Value
Tot CO_2 gen MMT/yr	4.37
Net CO2 MMT/yr	4.24
Wellbore inj Press (psi)	1471.6
Total Injection VOlume (MMscf/day)	19.55
NPV ($)	4989
Pipe Press Out (kPa)	350
Pipe Velocity output (m/s)	60.25
CO_2 entering plant (kg/yr)	126144000

Additionally, we ran validation checks against the constraints and bounds of the optimization problem against our DoE to ensure the design space was correctly upholding bounds and constraints of the problem. This was relatively easy to do as the computational time was >10ms.

5. DESIGN OF EXPERIMENTS

Upon completing the assembly of our model and ensuring its functionality, we established a Design of Experiments. Initially, we defined the bounds for each variable; see table 3. This was defined based on the outcomes of our problem formulation analysis. From our bounds, we created a list of options for each parameter, as seen in figure 5. Subsequently, we employed the pycubeDOE [1] Python function to develop a comprehensive Design of Experiments seen in figure 6. Finally, we executed our entire Design of Experiments via our model, generating a set of initial baseline responses that served as a starting point for further optimization.

TABLE 3: BOUNDS

Option	Lower Bound	Upper Bound
Number of Wells	5	20
Number of Connections	1	3
Pipeline Diameter	4	20
Pipeline Length	1000	55000
p2	500	700
p4	900	1800
p6	2300	3200
p8	3800	4800
p10	5000	6000
p12	6000	7000

5.1 Key Results

Best options:

DOE Option 4: NPV –> $30.69M
DOE Option 6: NPV -> $24.44M

Worst option:

DOE Option 20: NPV -> -$6,170M

5.2 Insights

The optimal choices entail fewer connections and a shorter pipeline length. Conversely, the suboptimal alternative involves a longer pipeline length combined with more connections.

6. SINGLE OBJECTIVE OPTIMIZATION

In this section, we will detail two efforts to optimize our model. First, we evaluated different gradient-based optimization algorithms. We ended up selecting a local derivative-free method due to the constraints of our model. Next, we tested several heuristic-based optimizations. Ultimately, we selected SLSQP and PSO for their respective advantages.

6.1 Local derivative-free optimization

Upon assessing the gradient-based optimization techniques we've learned, we concluded that our model's sheer number of equations would make it impractical to utilize such methods. Consequently, we explored the possibility of applying the "direct" approach for local derivative-free optimization. Unfortunately, we encountered issues implementing the algorithm using Python and could not achieve convergence toward a solution. Eventually, we opted for the Nelder-Mead method for local derivative-free optimization as a viable alternative. Figure 7 and 8 show the

[1] https://pypi.org/project/pycubedoe/

Design of Experiments - Options									
Number of Wells	Number of Connections	Pipeline Diameter	Pipeline Length	p2	p4	p6	p8	p10	p12
10	2	4	28000	500	900	2300	3800	5000	6000
20	3	12	1000	900	1800	3200	4800	6000	7000
5	1	20	55000	700	1300	2800	4400	5500	6500

FIGURE 5: DOE OPTIONS

Full Design of Experiments (DOE)

Option	Number of Wells	Number of Connections	Pipeline Diameter	Pipeline Length	p2	p4	p6	p8	p10	p12
1	5	2	12	28000	700	1800	3200	4800	5500	7000
2	5	1	4	1000	900	900	3200	4800	5500	7000
3	5	3	20	28000	500	1800	3200	3800	6000	6500
4	20	1	20	1000	700	900	2800	3800	6000	6000
5	5	2	12	28000	900	1800	3200	4800	5000	6000
6	5	1	12	1000	900	900	2300	4400	5000	7000
7	20	3	20	1000	500	1800	3200	4400	5500	6000
8	20	3	20	1000	700	1800	2300	4400	6000	6500
9	20	3	4	1000	900	1800	2300	3800	6000	7000
10	20	3	12	1000	500	1800	2300	4800	5500	6000
11	20	2	12	55000	900	900	2300	3800	6000	7000
12	5	3	12	55000	900	1300	3200	4800	5000	6000
13	20	2	4	1000	900	900	2800	4400	6000	7000
14	5	3	12	55000	500	1800	2800	4800	6000	6500
15	20	2	20	55000	900	900	3200	4800	6000	7000
16	5	3	12	55000	700	1300	3200	4800	5500	7000
17	20	3	12	1000	900	1800	3200	4800	6000	7000
18	10	1	12	55000	500	1800	3200	4800	5000	7000
19	10	2	20	1000	900	1300	3200	4800	5000	7000
20	10	3	4	55000	700	1800	3200	4400	6000	6000
21	20	2	4	1000	500	1300	2300	4400	6000	6500
22	10	1	12	55000	900	1800	3200	4800	5500	6500
23	10	2	12	1000	900	1300	2800	3800	5500	7000
24	20	3	4	1000	700	1800	3200	3800	5000	6500
25	20	3	4	1000	500	1800	2800	3800	6000	6000
26	20	3	20	1000	900	1800	2800	4400	6000	7000
27	20	3	12	1000	700	1800	2800	4800	5000	6500
28	20	1	12	28000	900	1300	2800	4400	6000	7000
29	10	3	12	28000	900	900	3200	4800	5500	6500
30	20	1	20	1000	900	1300	2300	3800	6000	7000
31	10	3	12	28000	700	1800	2300	4800	6000	6000
32	20	1	4	28000	900	1300	3200	4800	6000	7000
33	10	3	12	28000	500	900	3200	4800	5000	7000

FIGURE 6: FULL DOE TABLE

TABLE 4: NELDER-MEAD CONVERGENCE AND NPV

Starting point	Number of iterations to convergence	NPV
DOE #4	75	$40.3M
DOE #6	150	$41.6M

number of iterations for Nelder-Mead method convergence starting from DOE option 4 and option 6, respectively. Table 4 shows the number of iterations and final NPV values for the same options.

6.2 SLSQP and PSO

After executing the Nelder-Mead algorithm, we experimented with various heuristic-based optimization algorithms. We selected two specific approaches: SLSQP (sequential least squares programming) and PSO (particle swarm optimization). Our rationale for choosing SLSQP was due to its speed and relative ease of implementation. On the other hand, we opted for PSO as it provided a more comprehensive set of tuning parameters for us to utilize. Figure 9 and 10 show the results of PSO and SLSQP starting at DOE options 4 and 20.

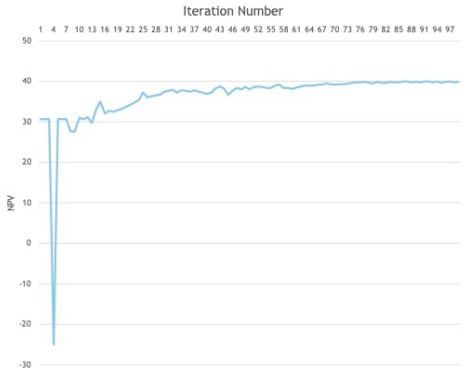


FIGURE 7: NELDER-MEAD CONVERGENCE PLOT STARTING AT DOE OPTION 4

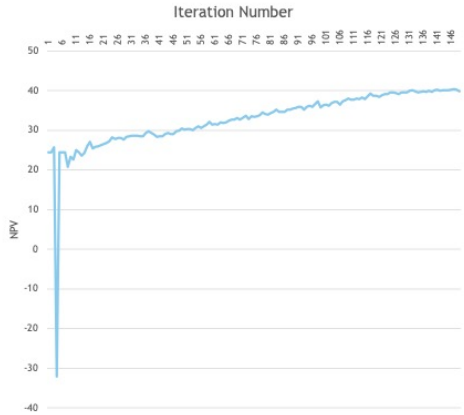


FIGURE 8: NELDER-MEAD CONVERGENCE PLOT STARTING AT DOE OPTION 6

Optimization Technique	Optimized NPV ($MM)	# of Wells	# of Connect-ions	Pipeline Diameter (in)	Pipeline Length (ft)	P2 (psi)	P4 (psi)	P6 (psi)	P8 (psi)	P10 (psi)	P12 (psi)	Time to complete
PSO	$42.85	20	1	20	3984	575	900	2300	3800	5000	6077	3.58s
SLSQP	$42.67	20	1	20	1000	700	900	2300	3800	5013	6402	0.11s

FIGURE 9: PSO AND SLSQP TABLE STARTING AT DOE OPTION 4

Optimization Technique	Optimized NPV ($MM)	# of Wells	# of Connect-ions	Pipeline Diameter (in)	Pipeline Length (ft)	P2 (psi)	P4 (psi)	P6 (psi)	P8 (psi)	P10 (psi)	P12 (psi)	Time to complete
PSO	$42.82	20	1	20	1000	509	900	2300	3800	5000	6000	3.28s
SLSQP	$41.58	20	1	20	55000	700	900	2300	3800	5560	6262	0.16s

FIGURE 10: PSO AND SLSQP TABLE STARTING AT DOE OPTION 20

PSO tuning parameters

The PSO algorithm has several tuning parameters. The two we focused on were the number of iterations and swarm size. Figures 9 and 10 were both run with a swarm size of 100 and iterations of 100. Table 5 shows additional tuning parameters and their effect on the overall NPV.

TABLE 5: PSO TUNING

Swarm Size	Number of Iterations	NPV
100	100	$43M
5	10	-$24M
5	100	$42M
100	5	$40M

Constraints

Our constraints are represented in our problem formulation, equation (2) above. For all our optimizations, we elected to use a penalty function for the constraints. So, every time a constraint is not met, we penalize the NPV by subtracting $100M. Our Python implementation is shown in listing 1.

Listing 1: Constraint's in Python

```
# q_const is the q_inj constraint in
# scf/day converted to mcf/year *
# 365 days/year * time * num_wells
# This whole thing is divided by 19.64
# to convert from mcf to tonnes.
# Then converted to MMT/year.
if(q_constraint > variables.Q_max):
    NPV = -100000000 + NPV

# The pressure at the bottom of the
# well cannot exceed max inj pressure
# which is determined by reservoir pore
# pressure fracture gradient, PWH > Pwf
if(p_wf_t > variables.p_injmax):
    NPV = -100000000 + NPV
```

Confidence

We have a strong belief in the effectiveness of our Particle Swarm Optimization (PSO) model. Its rapid processing ability, along with its comprehensive particle coverage, guarantees that the entire solution space is examined thoroughly. Nevertheless, we are also interested in examining possibilities beyond the predefined constraints of our current parameters to gain additional insights and potential improvements.

7. MULTI-OBJECTIVE OPTIMIZATION

To implement the Multi-Objective Optimization (MOO) capabilities into our project we started off by investing several MOO frameworks that have been built in Python. We looked at a number of optimizer tools but ultimately settled on using PyMOO [16]. We started off incorporating PyMOO by first creating a copy of both our Single Objective Optimizer (Optimzer_ldfo.py) and our experimentation module (Main.py). We then modified the Optimizer to incorporate a function call to the PyMOO plugin. We then modified the Main module (now called Main_MOO.py) to return two outputs (NPV and CAPEX). The new Main_MOO module was further updated to append to a .csv file specific information about each iteration through the optimizer. This information included the design vector requested by the optimizer, as well as the resulting NPV, CAPEX, CO_2 generated, and CO_2 injected. This allowed us to track the progress of the optimizer as it marched from an initial design vector starting point provided by the team towards its ultimate "optimal solution".

Once the modules were updated to incorporate the PyMOO framework we decided to look at two methods of performing MOO's. The first was the Non-Dominated Sorting Genetic Algorithm (NSGA-2) and the second was the S-Metric Evolutionary Multi-Objective Optimization Algorithms (SMS-EMOA).

As mentioned, the first method we used to analyze our problem set was the NSGA-2 method. We chose to use a heuristic algorithm as our model was not well suited for a gradient-based approach. Using NSGA-2 involved importing the algorithm from the PyMOO framework and providing execution parameters for the algorithm to adhere to. Our team performed a small sensitivity analysis to better understand how changing these parameters affected the algorithm's optimal solution. We ultimately decided to investigate the solution results and number of iterations required to perform the algorithm based upon: population size, mutation rate, cutoff, and number of offspring. The results from this investigation can be seen in Figure 11. After running the initial sensitivity analysis we decided to take the two best results, and combine them to see if the resulting parameters would yield a beneficial result. It turns out in this case that combining mutation rate and number of offspring did yield a better end result than changing any one single parameter. The number of iterations required to perform these calculations ranged between 1380 and

TABLE 6: SMS EMOA VS. NSGA-2

Description	NPV (USD)	CAPEX (USD)
SMS – EMOA	$42.86M	$786.68M
NSGA-2	$42.04M	$788.36M

2550 iterations.

Next, we looked at the SMS-EMOA algorithm to see how it performed compared to the NSGA-2 and single objective optimization methods. In this case we ran the algorithm with the default parameters, while using the same initial starting design vector as was used with NSGA-2. In this case the SMS-EMOA method found a better MOO optimum than the NSGA-2 method t. It's worth noting, however, that the SMS-EMOA required over 21 times the number of iterations to complete the optimization process. The end result and number of iterations through the optimizer can be seen in Table 6. It's also worth noting that the results of the SMS-EMOA method include very small changes in the design vector fed back into the experimentation module. These steps are extraordinarily small and from a practicality standpoint may be overcome by system noise or other limitations such as manufacturing quality, sensor limitations, or control system limitations. If factors such as these are not applicable to the given problem, then SMS-EMOA may be a desirable algorithm to use. The number of iterations required to perform this optimization method was 60,051 iterations. This was a staggering 23.5 times the worst-case iterations required for the NSGA-2 algorithm.

A comparison of the two algorithm's Pareto Fronts can be seen in Figure 12. Both algorithms trended towards an optimal solution with only a single connection in the design vector as this minimized pressure loss experienced through the Pipeline module. Pressure loss requires the system to re-pressurize the CO2 before it can be injected so the higher the number of connections, the more pressure will be lost, and the more compressors will be required by the system. As compressors are expensive to purchase and generate CO2, the algorithms favor fewer connections with one connection being the lowest achievable value. Ultimately, both algorithms chose shorter pipe lengths, larger pipe diameters, and the minimum number of connections. These three choices reduce the overall pressure loss experienced by the system.

8. FINAL RECOMMENDATION

Focusing on the CAPEX positive side of the results, there are many design options along the Pareto Front (Figure 14). There are tradeoffs to consider when selecting the optimal solution because the system's efficiency is in tension with the maximum value. Looking at the total efficiency of the system's ability is critical, balancing CO_2 injection as well as CO_2 generation. Table 13, illustrates that for the lower CO_2 generated per year from Option 3, there is a decrease in CAPEX of $12.45 MM. Focusing on CAPEX alone and selecting Option 1 will increase the CO_2 generated per year by 73 MMT. Neither is an ideal solution, but Option 2 is a fair compromise that balances the system options.

More importantly, the optimization model is a valuable tool that can assist with evaluating, prospecting, and planning a commercial-scale CCUS plant. Leveraging this tool will allow businesses to assess commercial agreements, permitting request needs, and contract negotiations. Once that preparation work has been done, the tool can also be used as a communication device between the multiple functions that will plan and execute the building, construction, designing, drilling, and monitoring, by demonstrating how design choices in each of these functions will impact the results. Abstractly, this optimization model is similar to a digital twin of a CCUS plant; it can be utilized to align the stakeholder network towards a common goal.

9. LEARNINGS AND FUTURE WORK

Reaching the end of this project, the team has greatly enhanced their understanding of multidisciplinary optimization, both from a coding perspective and problem formulation. The following sections summarize the key learnings from the team and future work the group would like to pursue to deepen the value of the project.

9.1 Learnings

Working through this project has been an incredible learning opportunity for the team. Not only did it allow us to practice what we were learning in class with a real-world application, but it also gave us a much deeper understanding of the CCUS ecosystem. The project team had varying levels of adjacent experience in this space, but it was generally fairly new to us all.

One key learning we had on the CCUS front was the value of compressing the CO_2 at the point-source capture facility before transporting it to the well and injecting it. The first iteration of our model did not have this sequence and instead transported it at ambient conditions and compressed it at the well site. However, once we compressed it before transporting it, we found the supercritical fluid properties (density and viscosity in particular) were much more ideal for transport, and the NPV of the system improved accordingly. Upon additional research, we found that this is the preferred way of transporting CO_2 per the literature.

Another important learning for the team was the value of designing for flexibility within the code itself. One team member recommended early on that we create separate files and functions for each module to allow for better collaborative coding and maximum flexibility if we needed to change things throughout the project. When we learned about the supercritical CO_2 properties described in the previous paragraph halfway through the semester, this structure in our code made it very simple to switch the sequence of the pipelines and facilities modules within our model. We imagine in real life things like this could come up regularly or there may be a desire to test things like this, and a model architecture that considers this upfront can be very beneficial.

Finally, we found that some optimization methods (i.e. SMS-EMOA) can generate very precise solutions but at very high computational costs. For some applications, these solutions may be so precise that they ultimately can be infeasible from an execution standpoint. As an example, maintaining a system pressure within fractions of a PSI (pounds per square inch) may be infeasible for some systems.

Inputs - Using the Non-Dominated Sorting Genetic Algorithm (NSGA-2)					Outputs	
Description	Population Size	Number of Offspring	Crossover Probability	Mutation	NPV ($ million)	CAPEX ($ million)
Baseline	70	10	.9	20	35.10	792.07
Low Pop	36	10	.9	20	34.57	792.26
Hi Pop	140	10	.9	20	-8.10	817.61
Low Offspring	70	5	.9	20	20.77	800.22
Hi Offspring	70	20	.9	20	41.33	786.99
Low Crossover	70	10	.85	20	36.74	790.85
Hi Crossover	70	10	.95	20	34.84	790.39
Low Mutation	70	10	.9	15	40.09	787.56
Hi Mutation	70	10	.9	25	35.47	791.14
Optimized	70	20	.9	15	42.04	788.36

FIGURE 11: NON-DOMINATED SORTING GENETIC ALGORITHM (NSGA-2) PARAMETER EXPLORATION AND OPTIMIZATION RESULTS

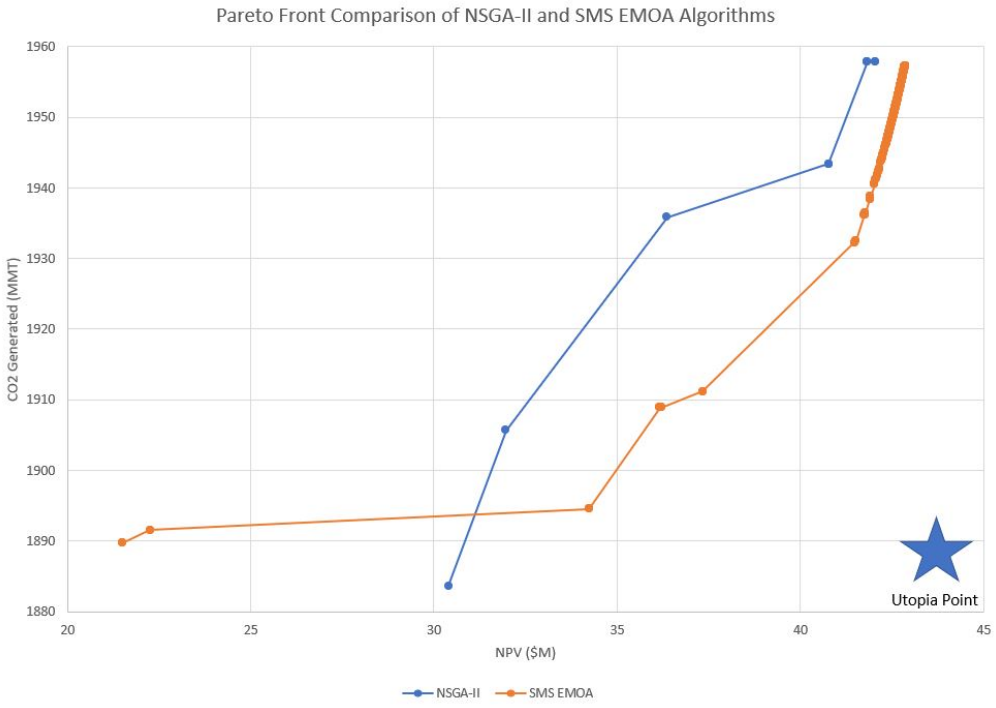


FIGURE 12: COMPARISON OF PARETO FRONTS BETWEEN THE NON-DOMINATED SORTING GENETIC ALGORITHM (NSGA-2) AND S-METRIC EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS (SMS-EMOA)

ID	# of Wells	# of Connections	Pipeline Diameter (in)	Pipeline Length (ft)	NPV ($ MM)	CO2 Gen (MMT/yr)	CAPEX ($MM)
1	20.0	1.0	20.0	1000.0	42.856	1957.33	786.68
2	20.0	1.0	20.0	1012.5	41.524	1932.56	787.62
3	10.0	1.0	17.0	21812.0	30.405	1883.61	793.41

FIGURE 13: TABLE RECOMMENDATION OPTIONS

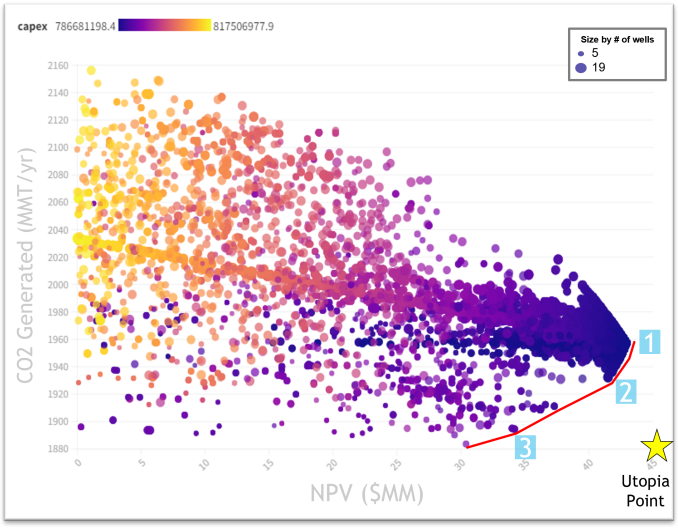


FIGURE 14: FINAL RECOMMENDATION OPTIONS SHOWING THE PARETO FRONT AND UTOPIA POINT

9.2 Future Work

Due to the short time frame of the course, the team had to utilize simplified versions of several modules. As discussed in the problem formulation section, this meant making many upfront assumptions to keep the project scope manageable for a classroom setting. However, if given more time the model could be greatly improved by incorporating these more rigorous techniques.

One such improvement would be to connect our optimizer to a dynamic reservoir simulation model rather than the simplified Darcy's Law equation for the subsurface module. This would allow us to model changes to the pressure in the reservoir as the CO_2 is injected and more accurately predict backpressure on the system that would impact injection rates. In reality, we would expect injection rates to decline at some rate over time due to this phenomenon that is currently not accounted for in our system.

The next future improvement would be to model the point-source capture facility distances and CO_2 capture rates on a facility-by-facility basis, as opposed to the regional average assumptions we included in this iteration of the model. This would enable us to test each facility discretely in our system to optimize which one(s) we would want to partner with.

Another important future improvement would be to test the incorporation of renewable energy sources into the model to minimize CO_2 emissions during compression. The current version of the model indicates that we would emit more CO_2 during compression than we would actually store, which defeats the purpose of the system. Some of this may be due to the assumptions made in our facilities module but is certainly worth investigating further with experts regardless. An example worth investigating would be electrifying compression, the predominant emission source in our model. One could tentatively use the average grid emission intensity and re-evaluate the ability to make a net-negative system.

Finally, expanding some of our bounds for our decision variables could lead to better results. We made the decision to cap the number of wells for this study to twenty to keep CAPEX in a reasonable range. However, it might be useful for the decision-maker to know what the true optimum would be in the system if CAPEX were not a constraint. Another version of the study that greatly expands the number of wells allowed but includes minimizing CAPEX as a third objective could help visualize this tension on the tradespace. It also could enhance the value of tie-ing in multiple facilities, since all of our high NPV results currently only included one facility connection.

We are encouraged by this first multidisciplinary optimization problem and look forward to the opportunity to enhance it further in the future with these and other improvements. Based on the findings, there is much value to be had in a project like this at the current tax incentive rates.

REFERENCES

[1] US EPA, OAR. "Inventory of U.S. Greenhouse Gas Emissions and Sinks." Technical report no. 2017. URL https://www.epa.gov/ghgemissions/inventory-us-greenhouse-gas-emissions-and-sinks.

[2] Hunt, Kara. "The Inflation Reduction Act creates a whole new market for carbon capture." *Clean Air Task Force* (2022)URL https://www.catf.us/2022/08/the-inflation-reduction-act-creates-a-whole-new-market-for-carbon

[3] Smyth, J. "Petra Nova carbon capture project stalls with cheap oil." *Energy and Policy Institute.* (2020)URL https://www.energyandpolicy.org/petra-nova/.

[4] N.A. "Illinois Basin – Decatur Project (IBDP)." Technical report no. 2017. URL https://netl.doe.gov/coal/carbon-storage/atlas/mgsc/phase-III/ibdp.

[5] Emehelu, Ike and Ojike, Chinelo. "Notable US Carbon Capture And Storage Projects - Oil, Gas Electricity - United States." (2022)URL https://www.mondaq.com/unitedstates/oil-gas--electricity/1267524/notable-us-carbon-capture-and-storage-projects.

[6] USGS. "National Assessment of Geologic Carbon Dioxide Storage Resources—Results." Technical report no. 2013. URL https://pubs.usgs.gov/circ/1386/.

[7] S Jackson, E Brodal. "A Comparison of the Energy Consumption for CO2 Compression Process Alternatives." *IOP Conference Series: Earth and Environmental Science* Vol. 167 (2018): p. 012031. DOI 10.1088/1755-1315/167/1/012031.

[8] US EPA, Ken Darrow. "Section 3. Technology Characterization - Combustion Turbines." Technical report no. 2015. URL https://19january2017snapshot.epa.gov/sites/production/files/2015-07/documents/catalog_of_chp_technologies_section_3._technology_characterization_-_combustion_turbines.pdf.

[9] Bautista, Juan Pablo Vargas. "Heat Recovery System in an Industrial Furnace to Generate Air Conditioning through an Absorption Chiller." *INVESTIGACION DESARROLLO* Vol. 14 (2014): pp. 117–134. DOI 10.23881/idupbo.014.1-7i.

[10] J. Ogden, N. Johnson. *Developments and Innovation in Carbon Dioxide (CO2) Capture and Storage Technology*. Vol. 1. Woodhead Publishing (2010). DOI https://doi.org/10.1533/9781845699574.1.27.

[11] Sheng, James J. "Chapter 5 - Polymer Flooding." Sheng, James J. (ed.). *Modern Chemical Enhanced Oil Recovery*. Gulf Professional Publishing, Boston (2011): pp. 101–206. DOI https://doi.org/10.1016/B978-1-85617-745-0.00005-X. URL https://www.sciencedirect.com/science/article/pii/B978185617745000005X.

[12] "Friction factors for pipe flow." (1943).

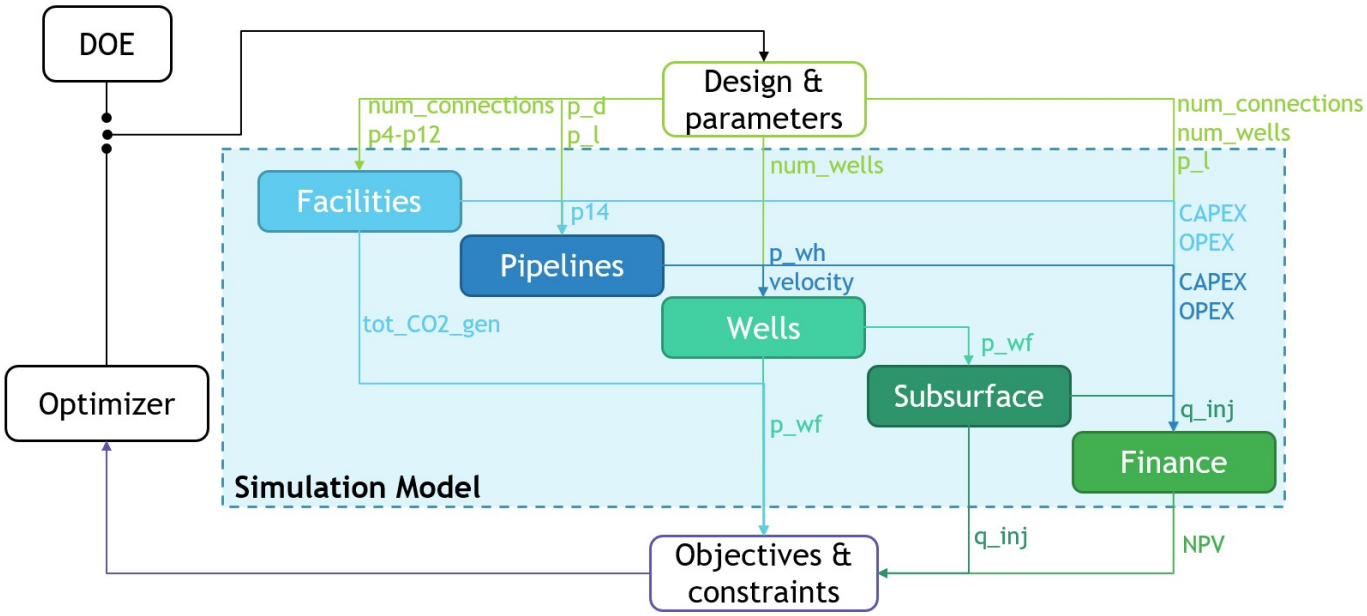
[13] NA. "Pressure Loss Correlations - Multiphase Flow Theory." IHS Markit Ltd. (2021). Accessed May 5, 2023, URL https://www.ihsenergy.ca/support/documentation_ca/Piper/2018_1/Content/HTML_Files/Reference/20materials/Pressure_loss_correlations/c-te-pressure.htm.

[14] Morton-Thompson, Diane and Woods, Arnold M. *Development Geology Reference Manual ()*, 3rd ed. Vol. 10 of *AAPG Methods in Exploration series*. AAPG (1994).

[15] Ghomian, Yousef, Pope, Gary A. and Sepehrnoori, Kamy. "Reservoir simulation of CO2 sequestration pilot in Frio brine formation, USA Gulf Coast." *Energy* Vol. 33 No. 7 (2008): pp. 1055–1067. DOI https://doi.org/10.1016/j.energy.2008.02.011. URL https://www.sciencedirect.com/science/article/pii/S0360544208000509.

[16] J. Blank, Kalyanmoy Deb. "Pymoo: Multi-Objective Optimization in Python." *Institute for Electronic and Electrical Engineers* (2020)URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9078759/.

APPENDIX A. BLOCK DIAGRAM



APPENDIX B. N2 DIAGRAM

(x,p)	num_connections, p4-p12	p_d, p_l	num_wells		num_wells, p_l num_connections	J(x,p), g(x,p)
	Facilities	p_{out}			CAPEX, OPEX	CO_2 generated
		Pipeline	$p_{wellhead}$, velocity		CAPEX, OPEX	
			Wells	$P_{wellbore}$		$P_{wellbore}$
				Subsurface	q_{inject}	q_{inject}
					Finance	NPV