Modular Framework for Simulation-Based Multi-objective Optimization of a Cryogenic Air Separation Unit

Bryan V. Piguave,* Santiago D. Salas,* Dany De Cecchis,* and José A. Romagnoli*

ABSTRACT: A framework to obtain optimal operating conditions is proposed for a cryogenic air separation unit case study. The optimization problem is formulated considering three objective functions, 11 decision variables, and two constraint setups. Different optimization algorithms simultaneously evaluate the conflicting objective functions: the annualized cash flow, the efficiency at the compression stage, and capital expenditures. The framework follows a modular approach, in which the process simulator PRO/II and a Python environment are combined. The results permit us to assess the applicability of the tested algorithms and to determine optimal operational windows based on the resultant 3-D Pareto fronts.

1. INTRODUCTION

The commodities derived from air separation units (ASUs) are essential for many supply chains worldwide. For instance, He et al.1 highlight the relevance of these units by reporting that they represent ̃5% the total power consumed in China. ASUs are an integral part of many processes and can be integrated into various systems such as natural gas systems2−4 and storage power plants.5,6 The commodities obtained from an ASU have important applications in the manufacture and health care sectors. Indeed, their production level has noticeably increased over the years, and it is projected to keep growing.7,8

Air is mainly composed of argon (Ar), nitrogen (N₂), and oxygen (O₂). Relevant uses of Ar can be found in welding applications such as tungsten inert gas (TIG) and gas tungsten arc welding (GTAW) because it generates an inert media, minimizing defects in final joints. Moreover, Ar is employed in fluorescent light bulbs,9 the production of silicon semiconductors (as a protective gas),10 and as a carrier in inductively coupled plasma spectroscopy.11 On the other hand, the uses of N₂ are mainly in the production of fertilizers, nitric acid, and nylon.12 It is utilized as an inert gas to suppress the risk of explosions within a combustible material.13 During the melting process of aluminum, N₂ works as a substitute for chlorine and Freon to prevent oxide formation.14 Finally, the main uses of O₂ entail support systems in metallurgy, polymer synthesis, and medicine. Applications in metallurgy include laser-oxygen cutting of mild steel.15 In addition, oxygen plasma treatment is applied to modify the surface of different polymers to improve their adhesive properties.16 Regarding medical applications, hyperbaric oxygen therapy contributes in the healing of open wounds,17 promoting an antimicrobial effect for suppressing bacterial growth on tissues.18 More recently, the demand for medical O₂ has increased during the COVID-19 outbreak in 2020−2021 because of its use in intermediate and intensive care units for mechanically ventilated patients.19

An ASU separates air into its principal components (Ar, N₂, and O₂) through a low-temperature liquefaction process. These products might vary in their composition, phase, and produced quantities. The selection of suitable operating conditions is crucial to guarantee the desired final product characteristics. Several contributions have performed simulation-based analyses of ASUs. For instance, Zhu et al.20 implemented a low-order dynamic model for a cryogenic distillation column, primarily applied in N₂ purification, using HYSYS as process simulator. Teague and Edgar21 suggested a predictive model of a small pressure-swing absorption (PSA) system of air separation to address technical needs and operation in military aircraft applications. Jiang et al.22 optimized a PSA separation system while tailoring optimization algorithms. Huang et al.23 proposed a nonlinear model predictive control (NMPC) to adjust the operating conditions of a PSA system to overcome variations in the product demand.

Received: November 24, 2021
Accepted: March 18, 2022
Published: April 2, 2022
Seeking to enhance the system’s design, Van Der Ham and Kjelstrup\textsuperscript{24} improved the heat integration of the distillation columns by moving the low-pressure column down along the high-pressure column. Such changes decreased $\sim$23% in the total entropy production. Manenti et al.\textsuperscript{25} studied the intensification of an ASU using the process simulator PRO/II. Main accomplishments included improving the $\text{O}_2$ purity, recycle of the rich Ar stream, and a feasibility analysis of energy generation. Aneke and Wang\textsuperscript{26} studied the incorporation of heat recovery cycles with different configurations at the compression stage to enhance the system’s energy efficiency. Fu et al.\textsuperscript{27} conducted a process analysis and optimization based on a complete equation-oriented approach. Tesch et al.\textsuperscript{28} evaluated the integration of LNG regasification into air separation processes to enhance the compression stage. An exergy and economic analysis was carried out using Aspen Plus. Finally, Young et al.\textsuperscript{29} performed a detailed design and economic evaluation of a cryogenic ASU, identifying the relevance of each stage when computing the capital expenditures. Although open source state-of-the-art platforms, such as FOQUS,\textsuperscript{30,31} allow the interaction with commonly used chemical engineering process modeling software (Aspen Plus, $\text{gPROMS}$, Thermoflex) for simulation-based optimization, the functionalities of such platforms are still under development. In this sense, a generic integration using multiobjective optimization (MOO) evolutionary algorithms and a high resolution ASU model in PRO/II has not being fully explored in the literature.

MOO algorithms permit to explore highly nonlinear systems with complex solution domains.\textsuperscript{32–34} A comparative study between different evolutionary algorithms demonstrated that the nondominated sorting genetic algorithm III (NSGA-III) and the unified nondominated sorting genetic algorithm III (UNSGA-III) can achieve competitive results when compared with other approaches.\textsuperscript{35} The performance of these algorithms was tested for an open-shop scheduling problem with resource constraints.\textsuperscript{36} On the other hand, the multiobjective evolutionary algorithm based on decomposition (MOEAD) demonstrated superior performance in benchmark problems, such as the traveling salesman,\textsuperscript{37} and in passive vehicle suspension optimization.\textsuperscript{38} Similarly, Gaspar-Cunha et al.\textsuperscript{39} applied a MOO methodology to minimize the cycle time, length of weld and warpage in an injection molding process using NSGA-III, UNSGA-III and MOEAD. A number of contributions have relied in these evolutionary optimization algorithms,\textsuperscript{40,41} and the Python package $\text{pymoo}$\textsuperscript{42} incorporates them for easy deployment.

In this work, three objective functions are optimized including the annualized cash flow (CF), the efficiency of the Rankine cycle (ef) at the compression stage, and the capital expenditures (CAPEX) of the facility. The first two objectives are maximized while the third is minimized. The achieved results provide guidance regarding the most adequate setup of operational variables. In a previous study, the reference vector-guided evolutionary algorithm (RVEA) was implemented to address the MOO of an ASU.\textsuperscript{43} In this context, major contributions of the present work are summarized next:

- A unified framework integrating the process simulator PRO/II and a Python environment for the MOO of an ASU, offering a variety of optimization algorithms, is proposed. It includes results handling and visualization.
- The reliability of the high-resolution simulation is assured by verifying and replacing the main simulation at each iteration step.
- A comparison under fair conditions between evolutionary algorithms for simulation-based MOO using quality indicators is detailed.
- Operational windows for flexible ASU operation based on Pareto front results are provided.

The remainder of this contribution is organized as follows. The ASU process and its simulation strategy are described in section 2. The MOO problem formulation, its objective functions, MOO algorithms, quality indicators, and the proposed framework are presented in section 3. Thereafter, we present the results and findings in section 4. Finally, conclusions and future work are outlined in section 5.

### 2. AIR SEPARATION UNIT AND COMPRESSION SYSTEM

Air separation can be performed with different technologies. Among them, the PSA, vacuum PSA, and cryogenic ASUs are the most relevant. In this contribution, a cryogenic ASU is considered. The proposed system incorporates air purification for the inlet stream, compression, and air separation. Cryogenic
The separation stage consists of two distillation towers: high pressure (HP) and low pressure (LP) columns. The HP column (T-201) operates at 600 kPa. The flow rate of the bottoms product contains approximately 35% of O₂, 1% Ar and processes about 60% of the main feed. It has a total condenser allowing a liquid reflux through the distillation column. A reboiler is not considered in the design as the air stream exiting the bottoms of the column exchanges heats with the upper stream before entering the Ar column.

The LP column (T-202) operates at 150 kPa to increase the relative volatility of O₂ and N₂. The composition of O₂ in the stream is 99% as the side draw removes the Ar. The liquid N₂ in T-201 supplies the reflux of T-202. The Ar column feed comes from the bottoms of the LP column. The draw rate is about 20% of the air feed rate, and 4% is removed as Ar product. The pressure drop available for this distillation is limited by the temperature of the expanded rich liquid from the HP column.43

2.2. Process Simulation Strategy. The selection of the most suitable equation-of-state (EOS) allows us to avoid inaccuracies in the thermodynamic calculations of the system. Skogestad suggested employing the Soave–Redlich–Kwong (SRK) EOS for processes with nonpolar components such as O₂ and N₂. On the other hand, Aneke and Wang utilized the Peng–Robinson EOS available in Aspen Plus to simulate an ASU. At low-density gas regions, both equations fit experimental pressure–temperature–volume data accurately and offer a similar reliability on the whole range of temperature.46 For the system under study, the SRK EOS handles the multicomponent vapor–liquid equilibrium. The simulation of the system starts at the end of the pretreatment section. The composition of the inlet air stream at the compression section is detailed in Table 1. The process parameters and variables considered in the simulation are detailed in the process description and are listed in Table 2.

The system is simulated using the commercial process simulator PRO/II in steady state. The high-resolution model incorporates quality constraints to ensure that products comply with the desired quality standards such as the degree of purity. In addition, two calculators are incorporated to simulate and include reference streams into the multistream heat exchanger E-202. In this sense, the path of each flow stream is traceable within the process, facilitating further analyses. Regarding the column trays efficiency, Biddulph suggests an efficiency of 70% for all trays in ternary mixtures such as air. However, Zhu et al assumed a tray efficiency of 100% for an ASU with crude Ar separation. In this work, an efficiency of 100% for all trays is considered. Finally, pseudostreams are included on each distillation column to simulate energy integration in the whole heat exchanger network.

3. SIMULATION-BASED MOO

3.1. Problem Formulation. The optimization problem considers three conflicting objectives; the annualized cash flow, efficiency of the Rankine cycle, and the capital expenditures. The

| component | molar fraction |
|-----------|---------------|
| N₂        | 78.09         |
| O₂        | 20.95         |
| Ar        | 0.93          |

Table 2. Process Parameters and Variables for the ASU Simulation in PRO/II

| parameter                              | value | units |
|----------------------------------------|-------|-------|
| δT approach of subcoolers              | 1     | K     |
| δT approach of condenser reboilers     | 1     | K     |
| trays in HP column                     | 44    |       |
| trays in LP column                     | 69    |       |
| trays in argon column                  | 55    |       |
| pressure drops in distillation columns | 10–15 | kPa   |

The temperature of each molecular sieve decreases over time, requiring regeneration. For this step, a stream of heated N₂ gas is injected in counter-flow, at approximately 523 K. Depending on the available N₂, regeneration could operate around a temperature of 407 K and pressure of 300 kPa. The regeneration gas is diffused uniformly over the surface of the bed for preventing accumulation of the residual components. Once the process concludes, a direct contact aftercooler (DCA) tower reduces the temperature of each molecular sieve.45

Thereafter, the pressure increases to 600 kPa before entering to the cooling section. This is achieved using three compressors (C-101/102/103) with an average compression ratio of 1.95:1 each. The heat generated is removed by inter coolers. Thus, the compressors operate close to isothermal conditions. The heat exchangers (E-101/102/103) reduce the air temperature to 313.15 K in the first two and to 303.15 K downstream of the heat exchanger E-103.

Heat is converted into electricity using a Rankine cycle (RC). The reduction of power consumption, obtained from the RC, is usually 10%. The efficiency of the compressors and the adiabatic expansion is assumed 80%.26 As the temperature of the discharged air raises, the amount of energy generated increases as well.46 Nevertheless, this configuration requires more power to attain the same pressure, limiting the reduction of energy consumption up to 0.2%. At the end of the compression stage, the air stream splits, and 10% of the flow rate goes to the turbine expander.

Table 3. Prices of Products and Raw Materials

| process stream | price | units |
|----------------|-------|-------|
| O₂             | 155.06 | $/Ton |
| N₂             | 86.51  | $/Ton |
| Ar             | 195.00 | $/Ton |
| air pretreatment | 10.00 | $/Ton |
The first two objective functions are maximized, \( f_1(x) \) and \( f_2(x) \), while the third, \( f_3(x) \), is minimized.

The optimization problem is written as follows:

\[
\begin{align*}
& \max_{x} \{ f_1(x), f_2(x) \}, \min_{x} f_3(x) \\
\text{subject to} & \\
& h(x) = 0, \\
& g(x) \leq 0, \\
& x_{LB} \leq x \leq x_{UB},
\end{align*}
\]

where the equality constraints are denoted by \( h(x) \) and are essentially incorporated by the process simulation (mass and energy balances) and techno-economic evaluation. The inequality constraints, \( g(x) \), include the convergence verification of the simulation, design, and feasibility constraints and other constraints incorporated to the objective functions. The vector \( x \in \mathbb{R}^3 \) represents the decision variables vector, and it is constrained between a lower and upper bound \( x_{LB} \) and \( x_{UB} \), respectively. The objective functions and the decision variables are described in more detail next.

The first objective function \( f_1(x) \) is the cash flow, \( CF(x) \), which is calculated by subtracting the operational expenditures \( OPEX(x) \) of the process from the generated revenues:

\[
CF(x) = \sum_{i=1}^{P} c_i F_i(x) - OPEX(x)
\]

The revenues are calculated by summing the product of the \( i \)th produced air constituent’s price \( c_i \) and its flow rate \( F_i(x) \), with \( i = 1, \ldots, P \). In this case, we consider three air constituents, \( O_2, N_2, \) and \( Ar \), i.e., \( P = 3 \). The \( OPEX(x) \) is defined in eq 6.

The ef at the compression stage, \( ef(x) \), is the second objective function \( f_2(x) \). The \( ef(x) \) is the ratio between the work generated by the expander \( W_{exp}(x) \) and the sum of the heat removed by the \( j \)th heat exchanger \( Q_{hex,j}(x) \) with \( j = 1, 2, \ldots, q \), in this case \( q = 3 \), at the compression

\[
ef(x) = \frac{W_{exp}(x)}{\sum_{j=1}^{q} Q_{hex,j}(x)}
\]

Finally, the capital expenditures, \( CAPEX(x) \), correspond to the third objective function \( f_3(x) \). It denotes the capital investment required for the facility. The \( CAPEX(x) \) is calculated as follows; the sum of the \( e \)th cost bare module \( C_{BM,e}(x) \) of each rotatory or static equipment \( e = 1, \ldots, E \). This total summation is multiplied by 1.18 times the ratio between the chemical

| Table 4: Utilities from CAPCOST |
|-----------------------------|
| utility                     | price | units   |
| low-temperature refrigeration | 8.49  | $/GJ   |
| electricity                 | 18.72 | $/GJ   |
| refrigeration moderated to low | 4.77  | $/GJ   |

| Table 5: Upper and Lower Bounds of the Decision Variables of the ASU System |
|-----------------------------|
| decision variables | \( x_{LB} \) | \( x_{UB} \) | units |
| \( T_{E,101} \)     | 288          | 303          | K     |
| \( T_{E,102} \)     | 288          | 303          | K     |
| \( T_{E,105} \)     | 288          | 303          | K     |
| \( Lx_{E1} \)       | 0.8          | 1.0          |       |
| \( Lx_{E2} \)       | 0.41         | 0.71         |       |
| \( \Delta T_{low} \) | 3.0          | 5.0          | K     |
| \( F_{mol} \)       | 4.0          | 6.0          | kmol/s|
| \( R_1 \)           | 0.1          | 0.4          |       |
| \( R_2 \)           | 0.1          | 0.4          |       |
| \( P_{C,104} \)     | 150          | 250          | kPa   |
| \( P_{P,101} \)     | 900          | 1100         | kPa   |

Figure 2. Framework with the main elements for the evolutionary MOO of an ASU simulation using the coupling of Python and PRO/II simulator.
Engineering plant cost index (CEPCI) of 2019 and a base year (2001):

$$\sum_{i=1}^{E} C_{BMi}(x) \cdot \frac{\text{CEPCI}_{2019}}{\text{CEPCI}_{base}}$$

The operational expenditures OPEX(x), mentioned in eq 3, consist of 18% of the CAPEX(x), plus 1.23 times the raw material costs (RMC) and the utility costs (UC(x)), which is

$$\text{OPEX}(x) = 0.18 \cdot \text{CAPEX}(x) + 1.23 \cdot (\text{RMC} + \text{UC}(x))$$

Table 3 enlists the prices of the air constituent products and the cost of the pretreatment for air conditioning. Table 4 shows the utilities cost. The UC(x) and the parameters for computing the CAPEX(x) of each equipment were obtained from Turton et al.51

The vector $x \in \mathbb{R}^{11}$ gathers the 11 decision variables considered in this work, which also correspond to the degrees of freedom (DOF) of the problem (11 DOF). Table 5 depict them with their respective lower $x^{\text{LB}}$ and upper bounds $x^{\text{UB}}$, and the physical units, when it applies. Here, $T_{E-101/102/103}$ corresponds to the outlet hot product temperature of the heat exchangers E-101, E-102, and E-103, respectively. $L_{E1/2}$ is the liquid fraction of the streams entering at stages 1 and 28, respectively, in the column T-202. $\Delta T_{dew}$ is the hot product temperature increase above the dew point of the stream entering at stage 33 in the column T-202. $F_{cool}$ is the flow rate of the coolant at the compression stage. $R_{1/2}$ is the ratio of the coolant flows that split before entering the heat exchangers E-101 and E-102, respectively. Since $R_3$ is defined by the equation $R_3 = 1 - R_1 - R_2$, it is not considered as a decision variable. $P_{C-104}$ is the discharge pressure of compressor C-104, and $P_{P-101}$ is the discharge pressure of pump P-101.

### Optimization Definition

```python
class MyProblem(Problem):
    def __init__(self):
        # define lower and upper bounds
        xl = amp.zeros(11)
        xu = amp.ones(11)
        super().__init__(n_var=11, n_obj=3, n_constr=0,
                        xl=xl, xu=xu, ... )

    def _evaluate(self, x, out, *args, **kwargs):
        ...
        f = ASU(db_path, x)
        out["f"] = f

    # create the algorithm object
    algorithm = NSGA3(pop_size=78, ... )

    # execute the optimization
    res = minimize(problem, algorithm, seed=1, ... )
```

3.2. Evolutionary MOO Algorithms. MOO addresses the balanced evaluation between conflicting objectives. In practice, it is not possible to attain a single optimal point capable of simultaneously optimizing all objective functions under study. In this sense, a set of optimal points offers a broader and flexible result. These optimal points could be selected on the basis of specific needs. In contrast, a common approach to solve multiple objective problems is to build a single function in the fashion of a weighted sum. However, the magnitude of these weights depends on an expert opinion who could provide subjective importance to each objective, introducing bias.

As an alternative, a set of optimal solutions permits us to overcome such limitations and to have a broader picture of the
problem solution. This set of solutions is referred to as the Pareto-optima points, corresponding to nondominated solutions. Such a characteristic implies that no objective can improve without penalizing at least another one. The Pareto-optima points delimit the Pareto front (PF), which is the boundary between feasible and unfeasible solutions.\textsuperscript{34}

The use of evolutionary multiobjective algorithms incorporates parallelization features such as vectorized evaluation, threaded-loop evaluation, and distributed evaluation. Such features contribute to debottlenecking the optimization routine. When considering a Python-scientific computing environment, the package \textit{pymoo} shows competitiveness when interconnected to simulation-based systems. The framework offers updated single- and MOO algorithms and different features related to such optimizations following an object oriented programming strategy. Additionally, visualization and decision making are included in the package.\textsuperscript{42}

In this work, the resultant PFs are illustrated as 3D surfaces because of the three evaluated objective functions. The following optimization algorithms are taken into consideration:

3.2.1. Nondominated Sorting Genetic Algorithm III. The Nondominated Sorting Genetic Algorithm III (NSGA-III) employs a mating-restriction scheme and establishes a multiple targeted search in advance. This feature ensures the diversity of the solutions. The NSGA-III characterizes by the combination of the parent and offspring populations. Thus, it allows us to conserve the fittest members of the previous generation. More details of this algorithm are provided by Deb and Jain.\textsuperscript{52} The

```
def ASU( db_path, opt_var):
    decision_var = diff_bound*opt_var + Lower_bound

    # Open the Simulation
    pro2, pro2db = COMconnect(db_path)
    pro2check = pro2db.CheckData
    Set_DecisionVars( decision_var, ... )

    # Run the simulation
    pro2check = pro2db.CheckData
    pro2check = pro2db.DbsSaveDb
    pro2db = None
    pro2run = pro2.RunCalcul(db_path)
    print("******** Simulation executed ******\n")

    if pro2run > 2: # Sim with errors
        y1, y2, y3 = penalties
    else:
        pro2db = pro2.OpenDatabase(db_path)
        y1 = Calculate_CF( ... )
        y2 = Calculate_Ef( ... )
        y3 = Calculate_CAPEX( ... )
        obj = np.array([-y1, -y2, y3])

    # Close simulation
    COMdisconnect(pro2, pro2db)
    pro2db = None
    return obj
```
selected parameters for testing this algorithm are a population size of 78 individuals and 24 generations. To verify its accuracy, three iterations are carried out.

### 3.2.2. Unified Nondominated Sorting Genetic Algorithm III

The unified approach of NSGA-III (UNSGA-III) holds more flexibility in terms of the number of objective functions in the optimization problem. The normalization operators and niching-based selection procedures automatically defunct for mono-objective problems and become active for multi- and multiobjective problems. The selected parameters for testing this algorithm are the same as those in NSGA-III.

### 3.2.3. Multiobjective Evolutionary Algorithm Based on Decomposition

Finally, the multiobjective evolutionary algorithm based on decomposition (MOEA/D) keeps the diversity of the trade-off solutions by setting weight vectors randomly. Each population member in the resulting subproblem is correlated with a weight vector. Zhang and Li proposed two metrics referred as Penalised-Based Intersection (PBI) and Tchebycheff (TCH). The PBI metric is the measurement of the distance between a point and the ideal point produced by the weighted sum of perpendicular distance $d_j$. Results are obtained by adding the distance $d_j$ along the reference direction as depicted in the following equation

$$\text{PBI}(x, w) = d_1 + \theta d_2$$  \hspace{1cm} (7)

The TCH metric considers an ideal point denoted as $z^*$ and the weight vector $w$. The objective function of the $j$th subproblem is described as

$$\text{TCH}(x, w, z^*) = g^j(xw^j, z^*) = \max_i w^j_i (x_i - z_i^*).$$  \hspace{1cm} (8)

The selected parameters for this algorithm are 20 neighbors, 0.7 probability of neighbor mating, and 24 generations per iteration. To verify its accuracy three iterations are carried out. These settings are selected for matching the total simulations/evaluations performed by the other tested algorithms and to compare them under fair conditions.

### 3.3. Quality Indicators

Four quality indicators (QIs) are selected to illustrate the performance of each optimization algorithm. These indicators include convergence, spread, uniformity, and cardinality.

Two main approaches to quantify convergence consider the PF availability or not. If available, the distance to the PF (or some designated points) to a solution set are quantified. When the PF is not available, the convergence is estimated through the Pareto dominance between solution sets. The alternative strategy applied in this work is based on the Pareto dominance relation provided by the solutions, which is known as the C-indicator.

$$C(\mathbf{A}, \mathbf{B}) = \frac{|\mathbf{b} \in \mathbf{B} \cap \mathbf{a} \in \mathbf{A} |}{|\mathbf{B}|}$$  \hspace{1cm} (9)

where $\mathbf{A}$ and $\mathbf{B}$ correspond to the compared solution sets, and $a \leq b$ means that a weakly Pareto dominates $b$. In summary, the C-indicator gives a ratio of the number of elements in $\mathbf{B}$ that are weakly dominated by an element in $\mathbf{A}$. As this quality indicator does not follow a commutative behavior, both $C(\mathbf{A}, \mathbf{B})$ and $C(\mathbf{B}, \mathbf{A})$ are calculated.

Uniformity is a QI that considers the even spacing between elements of a solution set. The uniformity of a solution set describes the distribution and its ability to represent appropriately a PF. The most common uniformity indicator of a solution set $\mathbf{A} = \{a_1, a_2, ..., a_N\}$ is known as spacing (SP), and it is defined as follows

$$SP(\mathbf{A}) = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (\bar{d} - d(\mathbf{a}_i, \mathbf{A} / \mathbf{a}_i))^2}$$  \hspace{1cm} (10)

where the distance $d(\mathbf{a}_i, \mathbf{A} / \mathbf{a}_i)$ is the minimum 1-norm of the element $\mathbf{a}_i \in \mathbb{R}^n$ to the rest of the elements in $\mathbf{A}$. $m$ is the number of objective functions as shown

$$d(\mathbf{a}_i, \mathbf{A} / \mathbf{a}_i) = \min_{\mathbf{a} \in \mathbf{A} / \mathbf{a}_i} \sum_{j=1}^{m} |a_j - \mathbf{a}_j|$$  \hspace{1cm} (11)

and $\bar{d}$ is the mean of all $d(\mathbf{a}_i, \mathbf{A} / \mathbf{a}_i)$, with $i = 1, ..., n$.

The spread of a solution refers to the cover of a solution set. The maximum spread (MS) provides a measure of the range obtained in a solution set

$$MS(\mathbf{A}) = \sqrt{\sum_{i=1}^{m} \max_{\mathbf{a} \neq \mathbf{a}_i} (a_j - \mathbf{a}_j)^2}$$  \hspace{1cm} (12)

with $\mathbf{a}_j, \mathbf{a}_i \in \mathbb{R}^n$, and $m$ as the number of objective functions. Thus, the spread is directly proportional to the maximum extent of each objective function.

Finally, cardinality refers the number of elements in a solution set. The cardinality of solution is defined by the number of nondominated solutions in the set ($N$). The error ratio (ER) indicator proposed by Van Veldhuizen establishes the proportion as

$$ER(\mathbf{A}) = \sum_{\mathbf{a} \in \mathbf{A}} \frac{e(\mathbf{a})}{N}$$  \hspace{1cm} (13)

where $\mathbf{a}$ is an element of the solution set $\mathbf{A}$ and the function $e(\mathbf{a})$ is defined as follows

$$e(\mathbf{a}) = \begin{cases} 0 & \text{if } \mathbf{a} \in \text{PF} \\ 1 & \text{otherwise} \end{cases}$$  \hspace{1cm} (14)

In this case, the expression $\mathbf{a} \in \text{PF}$ is assumed to be the nondominated solutions.

### 3.4. Framework Architecture

The proposed framework uses a first-principles high-resolution commercial simulator to obtain data from a process simulation. Thereafter, within a Python ecosystem further analyses take place. Building simulations with this kind of software are a common practice for process engineers. However, with the advent of more powerful computers and the availability of more and cutting edge scientific computing tools, it is possible to obtain better insights. Currently, it is possible to run simulators recurrently with Python programs using all kinds of optimization, statistical, or data science functionalities. This allows us to build more flexible and sophisticated designs and architectures considering more complex and realistic scenarios, which is an extraordinary basis for decision-making.

The interaction between PRO/II and the Python ecosystem occurs by using the Component Object Model (COM) implementation. The COM is a technology developed by Microsoft for building an interface in which different software components can interact. Certain industry standards use this technology, like CAPE-OPEN for process simulation technologies. The PRO/II simulator provides a Simulator COM Server that allows full read/write access to the PRO/II simulation database. Thus, any language or application that supports COM
can use these functionalities provided by the PRO/II COM Server. The Python for Windows extensions package, or pywin32, allows accessing Window’s COM to control and interact with other applications. Once the communication is established, the PRO/II COM Server grants access to read and write in the objects and streams contained in the process simulation. The ability of manipulating a simulation model from Python permits either concurrently or iteratively evaluate the high-resolution model. The implemented framework in this work is illustrated in Figure 2. Similar frameworks for solving other PRO/II process simulation problems have been proposed by Jones et al.59 and the authors.34,43,60

Because the PRO/II simulations are manipulated from Python, it is important to guarantee a well-developed process simulation as described in section 2.2. PRO/II provides a graphical user interface to comply with this task. Once the base simulation is set, from a Python program the decision variables (operating conditions) are set accordingly to the MOO algorithm. The Python code consists of two main functional blocks, the evaluation of the objective functions and the main program. In the main program, the hyper-cube domain in $\mathbb{R}^3$ is set. Following the documentation of pymoo, a MyProblem class is defined from the base class Problem. In the MyProblem class, the \textit{init} reserved method of certain dimension (i.e., 11) defines upper and lower bounds of the decision variables (i.e., between 0 and 1), number of objective functions, number of constraints, and other settings. In addition, it is required to define a method written as \texttt{evaluate}, which implements the mechanism of evaluating the cost function to return a vector with the cost values. In these methods, we use the function \texttt{ASU} that implements the multiobjective function, which is detailed later. Thereafter, the program creates several objects, problem of the class MyProblem, an algorithm object algorithm defining the optimization method (i.e., NSGA-III, UNSGA-III or MOEAD). Finally, the function \texttt{minimize} is called with problem and algorithm as part of the arguments. This runs the solver of the minimization problem and returns the resultant PF. A code snippet of the main block is depicted in Chart 1.

The interaction between PRO/II and Python occurs in the objective function \texttt{ASU}. A snippet of the objective function \texttt{ASU} is provided in Chart 2. The function receives a vector of decision variables $x$ and returns a vector of values of the objective functions $y \in \mathbb{R}^3$.

The program connects through the PRO/II Server to the simulation database. Then the 11 values given in the vector $x$ are set into the simulation database, and the changes are committed. At this point, the database is closed and the simulation runs. Once the simulation finishes, the database is opened again and the resulting values, required to calculate the objective functions, are retrieved from the PRO/II database. Finally, the output vector of the functions $y = [f_1, f_2, f_3]$ is calculated and returned to the optimization algorithm. This function is iteratively executed by the function \texttt{minimize} until the stop criteria is met.

Sometimes an input can produce an unfeasible simulation setup, corrupting the simulation file. To avoid this problem and to evaluate the system under fair conditions a replacement mechanism places the original simulation before each iteration. Moreover, this technique ensures the independence of each simulation during the optimization process and prevents the original file from being unusable. At the end of the process, the results are stored in data frames using \texttt{pandas}, to be used later for postprocessing and visualization.

4. RESULTS
The computational tools to test the performance of the proposed framework are the process simulator PRO/II Process Simulation. Table 6. Optimization Overview for Unconstrained and Constrained Evaluations

| algorithm   | NSGA-III | UNSGA-III | MOEAD | units          |
|-------------|----------|-----------|-------|---------------|
| comp. time  | 3.45     | 3.49      | 3.54  | hours per iter|
| feasible sim. | 1586    | 1580     | 1608  | -             |
| CF          | 15.082   | 15.082   | 17.716| MM USD/year   |
| eff         | 6.69     | 6.75     | 6.495 | %             |
| CAPEX       | 42.707   | 42.686   | 42.967| MM USD        |
| constrained comp. time | 3.51   | 3.50     | 3.55  | hours per iter|
| feasible sim. | 1422   | 1465     | 1608  | -             |
| CF          | 16.390   | 16.208   | 17.716| MM USD/year   |
| eff         | 6.70     | 6.65     | 6.49  | %             |
| CAPEX       | 42.804   | 42.780   | 42.967| MM USD        |

Table 7. Percentage (%) of Variation from the Ideal Optimal Points for Selection

| algorithm   | NSGA-III | UNSGA-III | MOEAD |
|-------------|----------|-----------|-------|
| unconstrained results | 10 | 5 | 0.5 | 6.70 6.65 6.49 % |
| constrained results | 6 | 5 | 0.5 | 6.69 6.75 6.495 % |

Figure 3. PF points using NSGA-III (blue pentagon), UNSGA-III (gold star), and MOEA/D (purple square). The violet semiplane delimits a boundary between the negative and the positive CF values.
PF and their correlation. The three previously described MOO algorithms are evaluated under fair conditions. Table 6 shows the computational time spent by the algorithms per iteration, the number of feasible solutions verified by a conditional which evaluates its status (in each iteration a total of 1872 runs take place), and the ideal optimal points for the unconstrained and constrained results. From a computational time perspective there is not significant difference between algorithms. However, the MOEAD shows a slightly higher computational time. The addition of the constraint does not impact the evaluation time, yet it improves the ideal points specially for the NSGA-III and UNSGA-III results.

The best results for the optimal set for each evolutionary algorithm are selected based on their closeness to the ideal optimal point, which is an unfeasible point holding the best values of the optimization objectives. The percentage of variation (equivalent to an uncertainty index) of the selected Pareto optimal points to their ideal optimal point are listed in Table 7. The idea behind this selection is to provide an Euclidean-like distance from the ideal optimal, with a reasonable variation. The selected points aim to establish operational windows. However, these operating conditions require further dynamic evaluations and an operability assessment to guarantee its final applicability.

4.1. Unconstrained Results. The points observed in Figure 3 represent the PF solutions obtained by the MOO algorithms. The blue pentagons, gold stars, and purple squares represent the PF set achieved by the NSGA-III, UNSGA-III, and MOEAD algorithms, respectively. The colored squared surface denotes the separation between positive and negative CF. In the case of the NSGA-III and UNSGA-III, a broader exploration in the solution domain is observed. These results have a higher diversity and draw a clear surface. On the other hand, the MOEAD results favor the CF. Indeed, the results show only positive and higher values if compared with the results achieved by the other MOO genetic algorithms. Moreover, 2-D projections are illustrated in Figure 4. These projections depict a nonlinear relationship between the CAPEX and the ef, which might be influenced by the heat recovery unit in the compression section. A strong concavity point is observed when the ef reaches approximately 6.4%. The other two projections CF vs ef, and CF vs CAPEX appear to be scattered without a solid trend.

Even though MOEAD does not provide a wide range of CF, CAPEX and ef, it clearly improves the CF. Such results define highly profitable operating conditions. On the other hand, this algorithm fails to minimize the CAPEX and maximize the ef when compared with the other two algorithms. The NSGA-III and UNSGA-III generate a set of results with a wider
Distribution and diversity, which achieve similar optimal results for CAPEX and ef. Nevertheless, these algorithms fail to improve the CF when compared to the MOEAD.

Tables 8−10 list the selected PF for each MOO algorithm, including the decision variables. Table 8 corresponds to the unconstrained results of NSGA-III. The minimum CAPEX is 43.08 MM $, followed by a maximum ef of 6.4% and CF of 15.08 MM $/Y. The UNSGA-III unconstrained results are in Table 9. The highest ef, 6.73%, is obtained. In contrast, a higher CAPEX is observed. If comparing it to NSGA-III, the results are similar for CF (15.08 MM $/Y). The top-5 results of MOEAD are in Table 10. An annualized CF of 17.66 MM $/year, an ef of 6.47%, and a minimum CAPEX of 43.06 MM $ are observed. Notice that in all cases the decision variables $L_X_1/X_2$ tend to reach $x_{UB}$, and $F_{cool}$ and $\Delta T_{dew}$ are closer to $x_{LB}$.

Moreover, $P_{C−104}$ and $P_{P−101}$ show a comparable range. On the other hand, $R_{1/2}$ vary depending on the algorithm.

The QIs for the unconstrained evaluation are shown in Table 11. In terms of cardinality, MOEAD exhibits the highest error ratio as some of the obtained solutions are not part of the overall nondominated solutions. On the other hand, the uniformity of this solution set has the lowest spacing score, showing a high uniformity of results among the MOO algorithms. NSGA-III and UNSGA-III have high spread meaning that offers a broad range for the objectives. These assertions are observed in the 2D projections of the PFs when comparing each solution set.

Table 12 shows the performance of the MOO genetic algorithms in terms of the C-indicator. As observed, the elements of the NSGA-III and UNSGA-III solution sets fail to dominate the results of MOEAD. In addition, 22% of NSGA-III results and 26% of UNSGA-III results are dominated by MOEAD. Regarding to the UNSGA-III C-indicator, it shows a dominance of 26% over the solution set from NSGA-III. On the other hand, 3% of NSGA-III set dominates UNSGA-III solution.

4.2. Constrained Results. The constrained PF solutions correspond to the points in Figure 5. As in the previous section, the algorithms NSGA-III, UNSGA-III, and MOEAD hold the same visual representation. The outcomes achieved in this setup favor the CF results because only profitable points (positive) are considered. NSGA-III and UNSGA-III demonstrate a broader exploration of the solution domain and improve all optimization objectives.

Influenced by the constraint, an overall lower bound CF of 8 MM $/Y was obtained by NSGA-III. The overall maximum CF is 17.66 MM $/year, and it is obtained by MOEAD. However, this result is achieved at a CAPEX of 43.07 MM $, meaning that higher investments will be required. In general, MOEAD offers a narrow set of optimal solutions in the Pareto surface, and it fails to explore lower CAPEX and higher ef. On the other hand, both

Table 10. Selected Unconstrained Results from MOEAD

| objectives | case 1 | case 2 | case 3 | case 4 | case 5 |
|------------|-------|-------|-------|-------|-------|
| CF         | 17.66 | 17.66 | 17.66 | 17.65 | 17.65 |
| ef         | 6.47  | 6.47  | 6.47  | 6.47  | 6.49  |
| CAPEX      | 43.06 | 43.06 | 43.06 | 43.06 | 43.07 |

| variables | case 1 | case 2 | case 3 | case 4 | case 5 |
|-----------|-------|-------|-------|-------|-------|
| $T_{E−101}$ | 288.00 | 288.00 | 288.00 | 288.00 | 288.00 |
| $T_{E−102}$ | 299.90 | 299.90 | 299.90 | 300.10 | 300.10 |
| $T_{E−105}$ | 288.00 | 288.10 | 288.00 | 288.00 | 288.20 |
| $L_X_1$ | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| $L_X_2$ | 0.71 | 0.71 | 0.71 | 0.71 | 0.71 |
| $\Delta T_{dew}$ | 3.05 | 3.05 | 3.05 | 3.12 | 3.02 |
| $F_{cool}$ | 4.00 | 4.00 | 4.00 | 4.00 | 4.00 |
| $R_1$ | 0.21 | 0.20 | 0.21 | 0.21 | 0.20 |
| $R_2$ | 0.21 | 0.21 | 0.21 | 0.22 | 0.22 |
| $P_{C−104}$ | 152.00 | 152.00 | 151.60 | 151.50 | 151.50 |
| $P_{P−101}$ | 1070.20 | 1070.20 | 1070.20 | 1065.00 | 1097.20 |

Table 11. Quality Indicators from Unconstrained Solution Sets

| quality indicator | NSGA-III | UNSGA-III | MOEAD |
|-------------------|----------|-----------|-------|
| maximum spread    | 51.14    | 44.08     | 0.51  |
| error ratio       | 0        | 0         | 0.33  |
| spacing           | 2.12     | 3.57      | 0.02  |

Table 12. C-Indicators from Solution Sets

| algorithms | NSGA-III | MOEAD | UNSGA-III |
|------------|----------|-------|-----------|
| NSGA-III   | 1        | 0     | 0.03      |
| MOEAD      | 0.22     | 1     | 0.26      |
| UNSGA-III  | 0.26     | 0     | 1         |

Figure 5. Pareto points obtained by the evolutionary algorithms NSGA-III (blue pentagon), UNSGA-III (yellow star), and MOEAD (purple square) with positive cash flow constrain.
UNSGA-III and NSGA-III generate a set of more diverse points. Figure 5 illustrates more detailed solutions with positive CF. In addition, a similar exploration for both algorithms is observed. The UNSGA-III optimal set achieves the highest ef (6.55%) of the heat recovery cycle, and the NSGA-III provides the lowest CAPEX (43.09$ MM).

Furthermore, the 2-D projections of the Pareto surface are shown in Figure 6. No correlation is observed between ef vs CF and CAPEX vs CF. However, the CAPEX vs ef subplot shows two inflection points at 5.6% and 6.4%. The conflict between these variables, and its nonlinear relationship, remains in the constrained evaluation. When comparing these results with the unconstrained, a gap between 5.8% and 6.1% is observed. A reason for this empty range might be that unfeasible operating conditions are neglected during the optimization routine. The possibility of these solutions not satisfying the CF restriction is also considered.

Tables 13—Table 15 list the constrained results of each MOO algorithm. The top 3 results of NSGA-III are in Table 13. A minimum CAPEX of 43.09 MM $, and a maximum efficiency of 6.53% and CF of 15.96 MM$/Y are observed. In contrast to the NSGA-III unconstrained results, higher values of CF are achieved along with a higher CAPEX. Table 14 has the top 2 constrained results of UNSGA-III, denoting an efficiency of 6.55%, a CF of 15.78 MM$/Y, and a CAPEX of 43.14 MM $. Regardless the CF constrain, UNSGA-III has a similar trend for $L_XE_1$ and $\Delta T_{dew}$ as the previous results, $L_XE_1$ approximates to $x_{UB}$ and $\Delta T_{dew}$ approaches to $x_{LB}$. When compared to the results of NSGA-III, UNSGA-III offers a narrower operational window for the pump P-101, the heat exchangers E-101/102, and the ratio of coolant flow $R_1$. Table 15 depicts the PF of MOEAD. It achieves a CF of 17.66 MM$/Y, an ef of 6.47%, and a minimum CAPEX of 43.06 MM $. As observed, MOEAD has the narrowest operational window for P-101 and the lowest $R_1$ when compared to the previous algorithms. The decision variables and the objective functions are bounded by the same operational windows as the unconstrained results, meaning that the constraint does not have a strong effect on MOEAD.

Even though the MOEAD results for CF show the highest profitable scenarios, the set of decision variables construct a narrow operating window which is not appropriate in practical terms. These results get trapped and do not offer a broader perspective of the problem nor show an extensive exploration of the solution domain. Such characteristic could be overcome by re adjusting the hyper-parameters of the optimization algorithm. However, default setups were considered in this work, and the mentioned analysis could be considered for future work.

A broader range of operating conditions, as those attained by the NSGA-III, will permit process adjustments while maintaining the ASU profitability above its threshold. This characteristic...
orcid.org/0000-0001-

III results dominate the solution set from NSGA-III. However, 29% of UNSGA-III performance of UNSGA-III and NSGA-III, 10% of NSGA-III dominance is presented in the other way. Regarding the accuracy of UNSGA-III solution sets are dominated by MOEAD. No results of better allocating computational resources for MOO with only feasible scenarios. An advantage of the proposed framework is that other objective functions, such as products composition, greenhouse gas emissions, and water use, could be easily incorporated. Such formulation could bring broader insights regarding the compromise from operational, techno-economic, and environmental perspectives. For future work, the study of fine-tuning the parameters of the evolutionary MOO algorithms could be considered. These parameters control several functional aspects of the algorithms, such as the probability of the mutation process, alternative crossover techniques, and the number of required parents to create offspring.

| Table 15. Selected Constrained Results from MOEAD |
|--------------------------------------------------|
| objectives | case 1 | case 2 | case 3 | case 4 | case 5 | units |
| CF | 17.66 | 17.66 | 17.66 | 17.65 | 17.65 | MM $/Y |
| ef | 6.47 | 6.47 | 6.47 | 6.47 | 6.49 | - |
| CAPEX | 43.06 | 43.06 | 43.06 | 43.06 | 43.07 | MM $ |

| variables | case 1 | case 2 | case 3 | case 4 | case 5 | units |
|----------|--------|--------|--------|--------|--------|-------|
| $T_{E,101}$ | 288.00 | 288.00 | 288.00 | 288.00 | 288.00 | K |
| $T_{E,102}$ | 299.90 | 299.90 | 299.90 | 300.10 | 300.10 | K |
| $T_{E,103}$ | 288.00 | 288.10 | 288.00 | 288.00 | 288.20 | K |
| $L_{E}$ | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | - |
| $L_{Eg2}$ | 0.71 | 0.71 | 0.71 | 0.71 | 0.71 | - |
| $\Delta T_{av}$ | 3.05 | 3.05 | 3.05 | 3.12 | 3.02 | K |
| $F_{cool}$ | 4.00 | 4.00 | 4.00 | 4.00 | 4.00 | kmol/s |
| $R_1$ | 0.21 | 0.20 | 0.21 | 0.21 | 0.20 | - |
| $R_2$ | 0.21 | 0.21 | 0.21 | 0.22 | 0.22 | - |
| $P_{C,101}$ | 152.06 | 152.06 | 152.06 | 151.60 | 151.50 | kPa |
| $P_{P,101}$ | 1070.20 | 1070.20 | 1070.20 | 1065.00 | 1097.20 | kPa |

| Table 16. Quality Indicators |
|-------------------------------|
| quality indicator | NSGA-III | UNSGA-III | MOEAD |
| maximum spread | 8.51 | 7.98 | 0.51 |
| error ratio | 0 | 0 | 0.53 |
| spacing | 0.37 | 0.35 | 0.02 |

| Table 17. C-Indicators from Solution Sets |
|-------------------------------------------|
| algorithms | NSGA-III | MOEAD | UNSGA-III |
| NSGA-III | 1 | 0.0 | 0.10 |
| MOEAD | 0.40 | 1 | 0.32 |
| UNSGA-III | 0.29 | 0.2 | 1 |

will offer the flexibility of changing the evaluated decision variables avoiding failure and other detrimental process conditions.

The QI for each MOO evolutionary algorithm are described in Table 16. In this section, negative values of the cash flow objective function are neglected, leading to a considerable reduction of the spacing and spread values. Nevertheless, since MOEAD does not provide negative CF values in the previous section, its QIs are not affected by this change. In addition, the error ratio of all the solution sets remain the same.

The C-indicators denoted in Table 17 illustrate the augmentation of MOEAD dominance over the other solution sets. In this context, 40% of the elements of NSGA-III and 32% of UNSGA-III solution sets are dominated by MOEAD. No dominance is presented in the other way. Regarding the performance of UNSGA-III and NSGA-III, 10% of NSGA-III solution set dominates UNSGA-III. However, 29% of UNSGA-III results dominate the solution set from NSGA-III.

5. CONCLUSIONS

In this work, a modular multifunctional framework was proposed for the MOO of an ASU. The integration between the simulation package PRO/II and a Python environment was performed using the Python COM Interface. The continuous monitoring of the simulation status and the replacement of the original simulation before each iteration evaluation assured reliability of the results. Three objective functions were implemented and optimized. The ef of the heat recovery unit and the annualized CF were maximized while the CAPEX minimized. The results displayed a well-defined 3D Pareto surface, in which the trade-off between the CAPEX and the ef was remarked.

Although the results carried out by the MOEAD algorithm achieved the best values in terms of the annualized CF, the NSGA-III and UNSGA-III offered more diversified solutions, confirmed by the spread and spacing QIs. Based on the obtained QIs, the UNSGA-III showed the overall most competitive performance (maximum spread, error ratio, spacing, and dominance).

The unconstrained results showed that some combinations of decision variables lead to unprofitable results (negative CF) at the PF. For this reason, the constrained formulation explored only solutions with a positive annualized CF. The non-dominated solutions for the constrained formulation exhibited improvements in some objective functions at the expense of the spread and spacing QIs, which resulted in a poorer performance. However, the overall benefits of better allocating computational resources for MOO with only feasible scenarios. An advantage of the proposed framework is that other objective functions, such as products composition, greenhouse gas emissions, and water use, could be easily incorporated. Such formulation could bring broader insights regarding the compromise from operational, techno-economic, and environmental perspectives.

For future work, the study of fine-tuning the parameters of the evolutionary MOO algorithms could be considered. These parameters control several functional aspects of the algorithms, such as the probability of the mutation process, alternative crossover techniques, and the number of required parents to create offspring.

**AUTHOR INFORMATION**

**Corresponding Authors**

Bryan V. Piguave — Escuela Superior Politécnica del Litoral, ESPOL, Facultad de Ciencias Naturales y Matemáticas, Guayaquil 09015863, Ecuador; Email: bpgiguave@espol.edu.ec

Santiago D. Salas — Escuela Superior Politécnica del Litoral, ESPOL, Facultad de Ciencias Naturales y Matemáticas, Guayaquil 09015863, Ecuador; orcid: 0000-0001-7731-0275; Email: sdsalas@espol.edu.ec

Dany De Cecchis — Escuela Superior Politécnica del Litoral, ESPOL, Facultad de Ciencias Naturales y Matemáticas, Guayaquil 09015863, Ecuador; Email: dany@espol.edu.ec
Complete contact information is available at:
https://pubs.acs.org/10.1021/acsomega.1c06669

Notes
The authors declare no competing financial interest.

ACKNOWLEDGMENTS
The authors acknowledge AVEVA for facilitating the process simulator PRO/II 10.2, under the academic program license, for its use at the Polytechnic University of the Coast (ESPOL), Guayaquil, Ecuador. This work is part of the project FCNM-54-2020, registered at ESPOL and jointly supported by PSE@LSU and PSE@ESPOL.

REFERENCES
(1) He, X.; Liu, Y.; Rehman, A.; Wang, L. A novel air separation unit with energy storage and generation and its energy efficiency and economy analysis. Appl. Energy 2021, 281, 115976.
(2) Ebrahimi, A.; Ziaibasharlaghi, M. Optimal design and integration of a cryogenic Air Separation Unit (ASU) with Liquefied Natural Gas (LNG) as heat sink, thermodynamic and economic analyses. Energy 2017, 126, 868–885.
(3) Kim, D.; Gianetta, R. E.; Gundersen, T. Optimal use of liquefied natural gas (LNG) cold energy in air separation units. Ind. Eng. Chem. Res. 2018, 57, 5914–5923.
(4) Liu, R.; Xiong, Y.; Ke, L.; Liang, J.; Chen, D.; Zhao, Z.; Li, Y. Integration of LNG regasification process in natural gas-fired power system with oxy-fuel combustion. J. Therm. Sci. 2020, 29, 1–16.
(5) Rieger, M.; Pardemann, R.; Rauchfuss, H.; Meyer, B. Carbon capture and storage power plants. Effects of ASU integration on IGCC performance and gas turbine operation. VGB PowerTech 2008, 88, 102–107.
(6) Zamani-Gargari, M.; Kalavani, P.; Abapour, M.; Mohammadi-Ivatloo, B. Reliability assessment of generating systems containing wind power and air separation unit with cryogenic energy storage. J. Energy Storage 2018, 16, 116–124.
(7) IDataResearch. Oxygen Concentrator Market Size, Share and COVID-19 Impact Analysis 2021–2027. 2021, https://idatresearch.com/product/global-oxygen-concentrator-market-analysis/.
(8) Verified Market Research, Global Industrial Nitrogen Market Size By Form, By Technology, By Transportation/Distribution, By End-User, By Geographic Scope and Forecast. 2021; https://www. verifiedmarketresearch.com/product/industrial-nitrogen-market/.
(9) Häussinger, P.; Glatthaar, R.; Rhode, W.; Kick, H.; Benkmann, C.; Weber, J.; Wunschel, H.-J.; Stenke, V.; Leicht, E.; Stenger, H. Ullmann’s Encyclopedia of Industrial Chemistry; John Wiley & Sons, Ltd, 2001; pp 1003–1008.
(10) Strudwick, A. J.; Marrows, C. H. Argon annealing procedure for producing an atomically terraced 4H-SiC (0001) substrate and subsequent graphene growth. J. Mater. Res. 2013, 28, 1–6.
(11) Sheppard, B. S.; Shen, W.-L.; Davidson, T. M.; Caruso, J. A. Helium-argon inductively coupled plasma for plasma source mass spectrometry. J. Anal. At. Spectrom. 1990, 5, 697–700.
(12) Böcker, N.; Grahl, M.; Tota, A.; Häussinger, P.; Leitgeb, P.; Schmucker, B. Ullmann’s Encyclopedia of Industrial Chemistry; John Wiley & Sons, Ltd., 2013; pp 1–27.
(13) Watson, J. T. R. Viscosity of Gases in Metric Units; HM Stationery Office, 1972; p 47.
(14) D’yachkov, L. G.; Zhilyakov, L. A.; Kostanovskii, A. V. Melting of aluminum nitride at atmospheric nitrogen pressure. Technol. Phys. 2000, 45, 928–930.
(15) Powell, J.; Petring, D.; Kumar, R.; Al-Mashhiki, S.; Kaplan, A.; Voiyis, K. Laser-oxygen cutting of mild steel: the thermodynamics of the oxidation reaction. J. Phys. D: Appl. Phys. 2009, 42, 015504.
(16) Mrosic, I.; Bäuerle, T.; Ulitzsch, S.; Lorenz, G.; Rebrner, K.; Kandelbauer, A.; Chassé, T. Oxygen plasma surface treatment of polymer films-Pellethane 5SDE and EPR-g-VTMS. Appl. Surf. Sci. 2021, 536, 147782.
(17) Kivisaari, J.; Nuuinkoski, J. Effects of hyperbaric oxygenation and prolonged hypoxia on the healing of open wounds. Acta Chir. Scand. 1975, 141, 14–19.
(18) Kaide, C. G.; Khandelwal, S. Hyperbaric oxygen: applications in infectious disease. Emerg. Med. Clin. N. Am. 2008, 26, 571–595.
(19) Whittle, J. S.; Pavlov, I.; Sacchetti, A. D.; Atwood, C.; Rosenberg, M. S. Respiratory support for adult patients with COVID-19. J. Am. Coll. Emerg. Physicians Open 2020, 1, 95–101.
(20) Zhu, G.-Y.; Henson, M. A.; Megan, L. Low-order dynamic modeling of cryogenic distillation columns based on nonlinear wave phenomena. Sep. Purif. Technol. 2001, 24, 467–487.
(21) Teague, K. G.; Edgar, T. F. Predictive dynamic model of a small pressure swing adsorption air separation unit. Ind. Eng. Chem. Res. 1999, 38, 3761–3775.
(22) Jiang, L.; Bieglcr, L. T.; Fox, V. G. Simulation and optimization of pressure-swing adsorption systems for air separation. AIChE J. 2003, 49, 1140–1157.
(23) Huang, R.; Zavala, V. M.; Bieglcr, L. T. Advanced step nonlinear model predictive control for air separation units. J. Process Control 2009, 19, 678–685.
(24) Van Der Ham, L. V.; Kjelstrup, S. Improving the heat integration of distillation columns in a cryogenic air separation unit. Ind. Eng. Chem. Res. 2011, 50, 9324–9338.
(25) Manenti, F.; Rossi, F.; Croce, G.; Grottolli, M. G.; Altavilla, M. Intensifying air separation units. Chem. Eng. Trans. 2013, 35, 1249–1254.
(26) Aneke, M.; Wang, M. Potential for improving the energy efficiency of cryogenic air separation unit (ASU) using binary heat recovery cycles. Appl. Therm. Eng. 2015, 81, 223–231.
(27) Fu, Q.; Zhu, L.; Chen, X. Complete equation-oriented approach for process analysis and optimization of a cryogenic air separation unit. Ind. Eng. Chem. Res. 2015, 54, 12096–12107.
(28) Tesch, S.; Morosuk, T.; Tsatsaronis, G. Exergetic and economic evaluation of safety-related concepts for the regasification of LNG integrated into air separation processes. Energy 2017, 141, 2458–2469.
(29) Young, A. F.; Villardi, H. G. D.; Araujo, L. S.; Raptopoulos, L. S. C.; Dutra, M. S. Detailed design and economic evaluation of a cryogenic air separation unit with recent literature solutions. Ind. Eng. Chem. Res. 2021, 60, 14830–14844.
(30) Edick, J. C.; Ng, B.; Gao, Q.; Tong, C. H.; Sahinidis, N. V.; Miller, D. C. A framework for optimization and quantification of uncertainty and sensitivity for developing carbon capture systems. Energy Procedia 2014, 63, 1055–1063.
(31) Soepyan, F. B.; Anderson-Cook, C. M.; Morgan, J. C.; Tong, C. H.; Bhattacharyya, D.; Omell, B. P.; Matuszewski, M. S.; Bhat, K. S.; Zamarripa, M. A.; Edick, J. C. et al. Computer-Aided Chemical Engineering: Elsevier; 2018; Vol. 44; pp 283–288.
(32) Beykal, B.; Boukouvala, F.; Floudas, C. A.; Pistikopoulos, E. N. Optimal design of energy systems using constrained grey-box multi-objective optimization. Comput. Chem. Eng. 2018, 116, 488–502.
(33) Shaju, S.; Sukhwani, M.; Kala, A. Optimizing furnace efficiency for factory of future using cooperative games. arXiv preprint arXiv:2104.08586v2 2021, 1–6.
(34) Salas, S. D.; Contreras-Salas, L.; Rubio-Deúeñas, P.; Chebeir, J.; Romagnoli, J. A. A multi-objective evolutionary optimization framework for a natural gas liquids recovery unit. Comput. Chem. Eng. 2021, 151, 107363.
(35) Yanninbelli, V.; Pacini, E.; Monge, D.; Mateos, C.; Rodríguez, G. A comparative analysis of NSGA-II and NSGA-III for autoscaling parameter sweep experiments in the cloud. Sci. Program. 2020, 2020, 1–17.
(36) Campos Ciro, G.; Dugardin, F.; Yalaoui, F.; Kelly, R. A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraints. IFAC-PapersOnLine 2016, 49, 1272–1277.

(37) Peng, W.; Zhang, Q.; Li, H. Multi-objective memetic algorithms; Springer, 2009; pp 309–324.

(38) Yuen, T. J.; Ramli, R. Comparison of computational efficiency of MOEA/D and NSGA-II for passive vehicle suspension optimization. Proceedings 24th European Conference on Modelling and Simulation. Kuala Lumpur, Proceedings 24th European Conference on Modelling and Simulation. Kuala Lumpur, 2010, 219–225.

(39) Gaspar-Cunha, A.; Antunes, C. H.; Coello, C. C. Evolutionary Multi-Criterion Optimization: 6th International Conference, EMO 2015, Guimarães, Portugal, March 29 -April 1, 2015. Proceedings, Part II; Springer International Publishing, 2015; p 591.

(40) De Rainville, F. M.; Fortin, F. A.; Gardner, M. A.; Parizeau, M.; Gagné, C. DEAP: A Python framework for Evolutionary Computation. GECCO’12- Proceedings of the 14th International Conference on Genetic and Evolutionary Computation Companion 2012, 85–92.

(41) Hart, W. E.; Laird, C. D.; Watson, J.-P.; Woodruff, D. L.; Hackebeil, G. A.; Nicholson, B. L.; Sirola, J. D. et al. Pyomo - Optimization Modeling in Python, 2nd ed.; Springer, 2017; Vol. 67; p 277.

(42) Blank, J.; Deb, K. Pyomo: Multi-objective optimization in python. IEEE Access 2020, 8, 89497–89509.

(43) Salas, S. D.; De Cecchis, D.; Piguave, B. V.; Romagnoli, J. A. Many-objective simulation-based optimization of an air separation unit. IFAC-PapersOnLine 2021, 54, 522–527.

(44) Castle, W. Air separation and liquefaction: recent developments and prospects for the beginning of the new millennium. Int. J. Refrig. 2002, 25, 158–172.

(45) Kerry, F. G. Industrial gas handbook: Gas separation and purification 2007, 70–113.

(46) Singla, R.; Chowdhury, K. Comparisons of thermodynamic and economic performances of cryogenic air separation plants designed for external and internal compression of oxygen. Appl. Therm. Eng. 2019, 160, 114052.

(47) Skogestad, S. Chemical and Energy Process Engineering, 1st ed.; CRC Press, 2008; p 442.

(48) Bisotti, F.; Galeazzi, A.; Galatìoto, L.; Masserottì, F.; Bigì, A.; Grittì, P.; Manenti, F. Implementing robust thermodynamic model for reliable bubble/dew problem solution in cryogenic distillation of air separation units. Int. J. Thermofluids 2021, 10, 100083.

(49) Biddulph, M. W. Multicomponent distillation simulation-distillation of air. AIChE J. 1975, 21, 327–335.

(50) Zhu, Y.; Legg, S.; Laird, C. D. Optimal design of cryogenic air separation columns under uncertainty. Comput. Chem. Eng. 2010, 34, 1377–1384.

(51) Turton, R.; Shaeiwitz, J.; Bhattacharyya, D.; Whiting, W. Analysis, Synthesis, and Design of Chemical Processes; International Series in the Physical and Chemical Engineering Sciences; Pearson Education, 2018.

(52) Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans. Evol. Comput. 2014, 18, 577–601.

(53) Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. 2007, 11, 712–731.

(54) Li, M.; Yao, X. Quality evaluation of solution sets in multiobjective optimization: A survey. ACM Comput. Surv. 2020, 52, 1–38.

(55) Zitzler, E.; Thiele, L. Multiobjective optimization using evolutionary algorithms—a comparative case study. International conference on parallel problem solving from nature 1998, 292–301.

(56) Schott, J. R. Fault tolerant design using single and multicriteria genetic algorithm optimization. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, 1995.

(57) Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. 2000, 8, 173–195.