processes

MDPI

Article

Novel Module-Based Design Algorithm for Intensified Membrane Reactor Systems

Brent A. Bishop and Fernando V. Lima *

Department of Chemical and Biomedical Engineering, West Virginia University, Morgantown, WV 26506, USA; babishop@mix.wvu.edu

* Correspondence: fernando.lima@mail.wvu.edu

Abstract: The growing interest in intensified process units that improve efficiency by combining several phenomena into one unit, has led to a loss in degrees of freedom when addressing the control scheme of these units. Previous work demonstrated that a novel module-based design approach to membrane reactors could improve the operability index of membrane reactor systems. This approach sought to decouple the phenomena to regain some degrees of freedom for the control system. However, the computational time to determine such an optimal module design made this class of design problems intractable to solve in a reasonable amount of time. This work proposes a set of design heuristics for a new module-based design approach for membrane reactors. These heuristics are used in combination with a genetic algorithm formulation to produce a novel, two-staged algorithm for the design and control of membrane reactor systems. This algorithm is developed in Python and uses rigorous membrane reactor models built in AVEVA Process Simulation. The proposed algorithm solves the original non-polynomial (NP) complexity problem in polynomial time (P), while still being able to find the optimal designs discovered in previous work through exhaustive methods.

Keywords: process intensification; module-based design; membrane reactors

check for updates

Citation: Bishop, B.A.; Lima, F.V. Novel Module-Based Design Algorithm for Intensified Membrane Reactor Systems. *Processes* **2021**, *9*, 2165. https://doi.org/10.3390/pr9122165

Academic Editor: Enrico Drioli

Received: 2 November 2021
Accepted: 23 November 2021
Published: 1 December 2021

1. Introduction

In recent years, the chemical industry has been exploring a potential shift in its traditional paradigm for process design. This shift is towards the design of intensified process units that combine multiple unit operations that would traditionally happen sequentially with the desire to dramatically reduce the size and increase the efficiency of chemical processes. With intensified units, such as membrane reactors (which have shown potential to replace some energy-intense separation processes [1–3]) where heat transfer, reactions, and separations are occurring simultaneously, there is a tradeoff for this increased efficiency. Although these intensified process designs can achieve significant improvements in efficiency, they can be much more challenging to control. The specific challenge addressed through this work is pertaining to the degrees of freedom (DOF) problem in this area.

Traditionally, a unit operations-based approach is used for designing chemical processes. Generally, a chemical process consists of six main areas: reactor feed preparation, reactor, separator feed preparation, separator, recycle, and environmental control [4]. Each of these areas would also have its own control scheme in place to ensure that disturbances are rejected, and the proper operating conditions are maintained. This type of process has the freedom to scale up and down, as well, because there is the freedom to adjust the several control valves involved. Compare this, in contrast, to an intensified process, such as a membrane reactor. There is a significant reduction in the number of control valves available because there are physically less units and less streams to manipulate. This problem is then compounded with the fact that the few control valves that are present must attempt to control reactions, separations, and heat exchange phenomena simultaneously. This combination leads to a reduction in the degrees of freedom (DOF) available to a potential control scheme when designing an intensified process.

This DOF challenge was identified in the literature as early as 2003 [5] and continued to be researched to this day [6–12]. The main cause for the reduction in DOF has been identified as the coupling of phenomena that would have been traditionally designed in separate unit operations. However, the reason intensified processes are efficient is because of this coupling of phenomena. Therefore, there is a tradeoff between efficiency and controllability when switching from a traditional unit operation design paradigm to a process intensification one. Some have argued an entirely different approach to process synthesis is necessary where "process design, operation, and control should be considered simultaneously" to address the challenge [9].

Often in chemical engineering, engineers first consider the design of the equipment for a desired nominal operating point. Once the design has been optimized for that operating point, the control scheme and tuning are considered. Similarly, the process intensification research is now shifting to consider the control challenges of proposed intensified designs. Modular technology research in the process systems engineering community have identified solving the DOF challenge and other control challenges as one of the major steps required for advancing this area of research [7,8]. The majority of theoretical research on the specific DOF challenge has consisted primarily of empirical arguments at this point. It was not until recently (2017) that a more rigorous justification for the loss of DOF of intensified process units was provided [6]. Because of this, a useful next step is to develop heuristics, procedures, and design algorithms that consider this DOF challenge during the design phase of intensified modular systems. The term "modular plant" in the chemical industry is defined as a unit where "the process equipment, instrumentation, valves, piping components, and electrical wiring are mounted within a structural steel framework" [13]. Simply stated, a modular plant is a unit that contains a scaled-down version of a traditional process unit and is able to be transported on the back of a flatbed truck. This philosophy goes against the concept of economies of scale, where generally larger units are more profitable. Instead, modular plants aim to become profitable through improved efficiencies and mass production, i.e., economies of numbers. However, this concept of modular design differs from what "modularity" refers to in other industries. In other industries, such as the electronics industry, modularity refers to products that are highly customizable, not simply smaller [14–16]. This concept for the modularization of equipment is also the one considered in this work.

Some of the literature [17] has focused on the inclusion of flexibility as a part of economic optimization. This work is attempting to improve the design and control performance in the context of the DOF challenge using operability instead as the objective of the optimization. Operability is chosen because it considers the design and control problem simultaneously which literature [9] emphasized is necessary for addressing the DOF challenge. The operability of a given process unit can be quantified using a metric called the operability index (OI). In previous work [18], it was demonstrated that a modular design approach could significantly improve the operability index and, therefore, gave the unit more freedom for control than it previously had. Although this was a great result, there are two challenges for solving this type of problem. Even though the operability index is very useful for studying the DOF challenge, it is computationally expensive to calculate (as it will be shown in subsequent sections). In addition, the optimum design was determined through exhaustive methods, i.e., by checking every possible design. This combination of an exponentially growing problem and a computationally expensive objective makes it essentially intractable to solve in reasonable time once the number of modules in series exceeds four.

In this work, this challenge of computational time to find the optimal design is addressed. This is done through the development of heuristics for solving this classification of problems which are then applied to a novel design algorithm that drastically reduces the computational time for finding optimal modular designs that maximize the OI. To solve these large combinatorics/mixed integer design problems, this novel design algorithm for membrane reactor systems determines optimal designs for smaller versions of the larger problem by using a genetic algorithm. These solutions are then combined with heuristics that relate the smaller known optimal solutions with the larger, unknown optimal designs (whose design spaces are too large to search using genetic algorithms alone). The proposed approach results in drastic reductions in computational time with minimal reductions in solution accuracy.

The rest of this paper is organized as follows: First, background is provided about the module-based design approach developed in previous work for membrane reactor systems, the method used for simulating these systems, process operability concepts, and the operability performance optimization problem. The basis for the design heuristics is then explained to provide an intuition for the novel design algorithm. Lastly, the results of the algorithm and performance are compared to previous work before providing conclusions with an outline of potential directions for future work.

2. Module-Based Design, Simulation, and Process Operability Background

The following subsections are provided to offer background into the modeling approaches being used, process operability analyses, and previous work done in this area that will be useful to better understand the advancements made.

2.1. Modular Design Background

As the literature has shown, a significant contributor to the DOF problem is caused by the coupling of design parameters and physical phenomena. One way to think of it is that traditionally, one can design a control system for the reactor in the plant and then send the outlet of the reactor to the separations section where a control system could be designed separately for that unit. In the case of the membrane reactor, though, reactions and separations occur simultaneously, meaning any proposed control approach must simultaneously control the reaction and permeation rates of the unit, which are not independent of each other. This can be imagined as a spectrum where, on one end, there is a traditional approach with sequential unit operations that has more freedom for control but lacks efficiency, and, on the other end, there are intensified processes, such as membrane reactors, which are more efficient but lack the same freedom for control. Based on designs proposed in literature, it appears one solution might be to use a design paradigm that lies somewhere in the middle [17–19].

The compromise would be to propose a design that still performs membrane separation and reaction to get the benefits of the membrane reactor's efficiency, but where the unit's phenomena are decoupled to allow for more freedom for the control scheme. In previous work, the idea for modular design of membrane reactors was proposed as a potential option for this design approach [18]. This approach to designing intensified reactive distillation systems [17,20,21] and membrane reactor systems [22,23] using combinations of modules is becoming more popular in the intensification community. For this work, rather than designing one unit, the design can be thought of as the combination of multiple modular units in series. For example, one of these modules could be made by combining a reactor module, a membrane reactor module, and a membrane module, such as what is shown in Figure 1.

Figure 1. Example of a module-based design with a reactor module (R), membrane reactor (MR), and membrane separator (M).

The benefit of designing in this way is that the designer now has the freedom to choose the length in which each phenomenon occurs and where it occurs. Another benefit of this design approach is in terms of economics of manufacturing. Mass producing a single standard piece of equipment is almost always cheaper than manufacturing multiple, unique custom units. Hence why there are standard pipe sizes, relief valves sizes, etc. An exciting opportunity is that these individual reactor, membrane, and membrane reactor modules can be mass produced with standard sizes (thus cheaper manufacturing costs), but can be assembled differently to address the operational needs of a particular location (such as different well sites).

2.2. Block-Based Phenomena Simulation

Previous work has focused on the modeling and simulation of these types of units [24], for which a brief description is provided here. One of the major challenges to simulating these kinds of units is the combination of trying to solve a boundary value problem (BVP) with countercurrent flow while also making topological changes to the design. This particular problem is especially difficult when considering the multiple phenomena occurring (heat transfer, reactions, and permeation) and their interdependence. For example, if an engineer/user wanted to swap out a membrane reactor module for just a reactor module, the final solution to that design problem could be dramatically different as this will affect the reactions and permeation in other parts of the unit.

The easiest solution is to instead create a module that can represent all possible operations simultaneously and recognize that the presence or absence of certain phenomena will define the module. For example, a "reactor" module is a module where heat transfer and reactions occur, but not permeation. This relationship between the different modules and their respective phenomena are summarized in Figure 2.

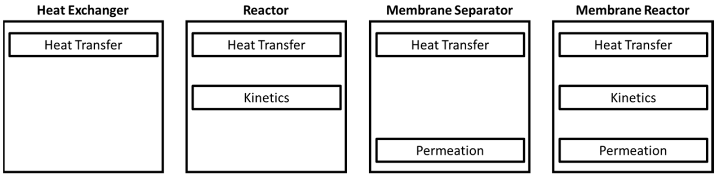


Figure 2. Each module can be defined by the phenomena that occurs within it.

To swap modules, the user can switch a specific phenomenon on or off using a variable that ensures homotopy between the two solutions. In other words, the "topological" change is implemented as a smooth, continuous change rather than a discrete change making the problem significantly easier to solve. A model developed using the model writing environment in the AVEVA Process Simulation Platform [25] is employed to simulate this module-swapping process. The developed membrane model is also now publicly available in the Membrane library on the AVEVA Process Simulation Platform.

2.3. Process Operability Background

Steady-state, set-point operability is used in this research to assess the performance of the modular equipment design, as defined in the operability analysis and concepts for square systems [26]. In the operability framework, a set of inputs ($u \in \mathbb{R}^m$) of a model (M) are mapped to their respective outputs ($y \in \mathbb{R}^p$) as follows:

$$M = \begin{cases} \dot{x} = f(x,u) \\ y = g(x,u) \\ h_1(\dot{x},x,y,\dot{u},u) = 0 \\ h_2(\dot{x},x,y,\dot{u},u) \geq 0 \end{cases}, \quad (1)$$

in which $x \in \mathbb{R}^n$ are the state variables, and h_1 and h_2 are equality and inequality process constraints, respectively. In addition, $\dot{x}$ and $\dot{u}$ represent time derivatives associated with x and u, respectively, and f and g are nonlinear process maps.

When performing an operability analysis, there are two spaces of concern: the input and output space. The input space can be defined by both design parameters, as well as operating parameters; however, only operating parameters are considered in this work. The output space is defined by the objectives the system is attempting to achieve. Within each of these spaces, there are three sets of points that are of interest and they are defined as follows:

Available Input Set (AIS): The set of all operational inputs or manipulated variables that are available to produce changes to the outputs of the process and is defined as:

$$\text{AIS} = \left\{ u \middle| u_i^{min} \le u_i \le \ u_i^{max}; 1 \le i \le m \right\}. \tag{2}$$

Because the AIS would include the set of every combination of possible control moves that an undetermined control structure could impose on the system, this analysis is independent of the defined control structure. This feature makes operability especially appealing for the design of modular membrane systems for addressing the DOF reduction challenge.

Desired Output Set (DOS): This set represents the region of operation that is desired for a given process and is defined as:

$$\text{DOS} = \left\{ y \middle| y_i^{min} \le y_i \le \ y_i^{max}; 1 \le i \le p \right\}. \tag{3}$$

Achievable Output Set (AOS): This set consists of all possible outputs that can be achieved, given the available input set and is mathematically defined as:

$$\text{AOS}_u = \{ y | M(u); \forall u \in \text{AIS} \}. \tag{4}$$

Any steady-state operating points that lie outside of the AOS for a given design are, by definition, unachievable regardless of the control system that is ultimately selected.

Servo-OI: Without regulatory control, the servo operability index (s-OI) for this analysis is given by:

$$\text{s} - \text{OI} = \frac{\mu(\text{AOS} \cap \text{DOS})}{\mu(\text{DOS})} \tag{5}$$

Here, μ represents a measure of the size of the space for instance length, area, volume, and hypervolume for their respective dimensions. The servo-OI is a way of quantifying what fraction of the DOS can be achieved by a given design. An OI of 1 would mean the given design can achieve any steady-state operating point in the DOS, whereas an OI of 0 would mean the given design can achieve no steady-state operating points in the DOS. Figure 3 summarizes the operability concept with the green region representing the region of the DOS that can be achieved by a given design considering a schematic for the water-gas shift reaction example that will be addressed in this article [19,24].

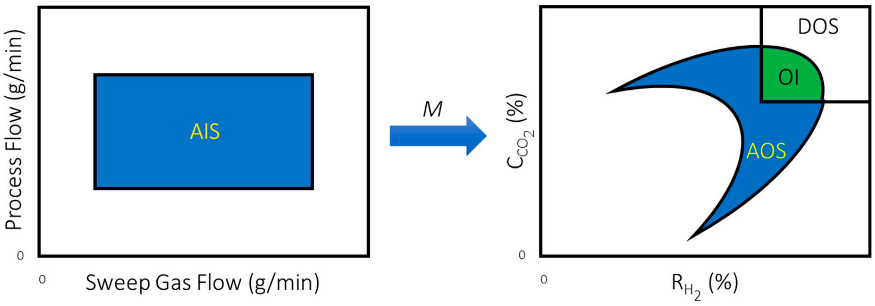


Figure 3. Schematic of the process operability concept. The fraction of the DOS covered by the green region would correspond to the operability index (OI) for the design.

3. Module-Based Design Effect on Process Operability

The operability framework provides a great measure for determining the best modular design for addressing the degrees of freedom challenge as an increase in the servo-OI implies that a future control system will be able to achieve more points in the DOS. To demonstrate that this modular design approach could lead to an improvement in the servo-OI, a water-gas shift membrane reactor (WGS-MR) case study was explored in previous work [18]. This membrane reactor consisted of a shell and tube arrangement with a polybenzimidazole (PBI) membrane that is selective to hydrogen and highly selective to water. Syngas was fed into the tube side of the WGS-MR where the water-gas shift reaction would occur, producing hydrogen. This hydrogen would then permeate through the PBI membrane and be captured by the sweep gas (steam) flowing countercurrently, allowing for further reaction to occur. A schematic for the WGS-MR is shown in Figure 4.

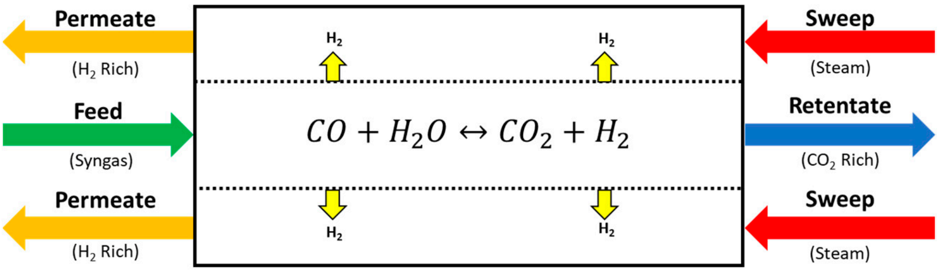


Figure 4. Schematic of a shell and tube WGS-MR for hydrogen recovery and carbon capture.

The reaction kinetics for this system were proposed by Choi and Stenger [27] and is described by Equation (6):

$$r_{CO} = k\left(P_{CO}P_{H_2O} - \frac{P_{CO_2}P_{H_2}}{K_P}\right), \tag{6}$$

where r_{CO} is the rate of consumption of carbon monoxide, k is the reaction coefficient, P_i is the partial pressure of component i, and K_P is the equilibrium coefficient for the water-gas shift reaction. The PBI polymer membrane's permeation is modeled using a Fickian diffusion model as shown in Equation (7).

$$J_i = \frac{Q_{i,o}}{\delta}(p_{i,t} - p_{i,s}), \tag{7}$$

where $Q_{i,o}$ is the permeance of component i, δ is the membrane thickness, and $p_{i,t}$ and $p_{i,s}$ are the component partial pressures on the tube and shell sides, respectively. For this work, the values of permeance come from Reference [19] and are summarized in Table 1.

Table 1. Component permeance ($Q_{i,o}$) values used reported in gas permeation units (GPU).

Component (*i*)	**$Q_{i,o}$ (GPU)**
H_2	250.0
CO_2	8.9
H_2O	750.0
CO	2.5
N_2	2.5

Because a polymer membrane is used, some undesired species are also able to permeate through the membrane from the process side, namely carbon dioxide and carbon monoxide. Another desired objective of this unit is to produce and capture as much hydrogen as possible. Therefore, the performance objectives for this system are described by Equations (8) and (9):

$$R_{H_2} = \frac{H_2 \; in \; permeate}{(H_2 + CO) in \; feed} = \frac{F_{H_2,p}}{F_{H_2,f} + F_{CO,f}}, \tag{8}$$

$$C_{CO_2} = \frac{carbon \; in \; retentate}{carbon \; in \; feed} = \frac{F_{CO,r} + F_{CO_2,r}}{F_{CO,f} + F_{CO_2,f}}, \tag{9}$$

in which R_{H_2} and C_{CO_2} are the hydrogen recovery and carbon capture fractions, respectively. These two objectives are not independent of each other, and they are actively competing objectives. Changing the process to capture more hydrogen might ultimately lead to more carbon dioxide and carbon monoxide escaping the process and decreasing the carbon capture, for example, if the membrane permeance is increased substantially.

To see if the proposed modular design approach could improve the operability of the membrane reactor, an optimum membrane reactor design and nominal operating point was first determined. This was done using the following objective function:

$$f(L, D, N_t, F_{steam}) = \left(1 - R_{H_2}\right)^2 + \left(1 - C_{CO_2}\right)^2, \tag{10}$$

where L is reactor length, D is the shell diameter, N_t are the number of tubes, and F_{steam} is the nominal steam sweep gas flow. A utopian, however unachievable, design and operating point would lead to 100% hydrogen recovery and 100% carbon capture. This objective function effectively leads to the design and nominal operating point that results in the minimum Euclidean distance to the utopian solution and, for this work, is considered the "best tradeoff" between hydrogen recovery and carbon capture. This optimization was performed using a sequential quadrative programming (SQP) optimization tool built into the AVEVA Process Simulation Platform and assumed 500 kg/h of syngas must be processed under normal operation. The results of this optimization are summarized in Table 2.

Table 2. Optimal membrane reactor design that produces the "best tradeoff" between hydrogen recovery and carbon capture according to the defined objective in Equation (10).

Parameter	Optimal Value
L	4.9 m
D (shell)	3.3 m
Nt	53
Fsteam	1088 kg/h
R_{H_2}	0.949
C_{CO_2}	0.870

However, as discussed above, although membrane reactors are highly efficient, they tend to face challenges when operating away from the operating point they were designed for. For example, if a company was interested in using a modular membrane reactor for multiple well sites, it would likely struggle to maintain high hydrogen recoveries at well sites with larger load than 500 kg/h, as well as high carbon capture at sites with less load than 500 kg/h. Figure 5 shows the assumed layout for the study.

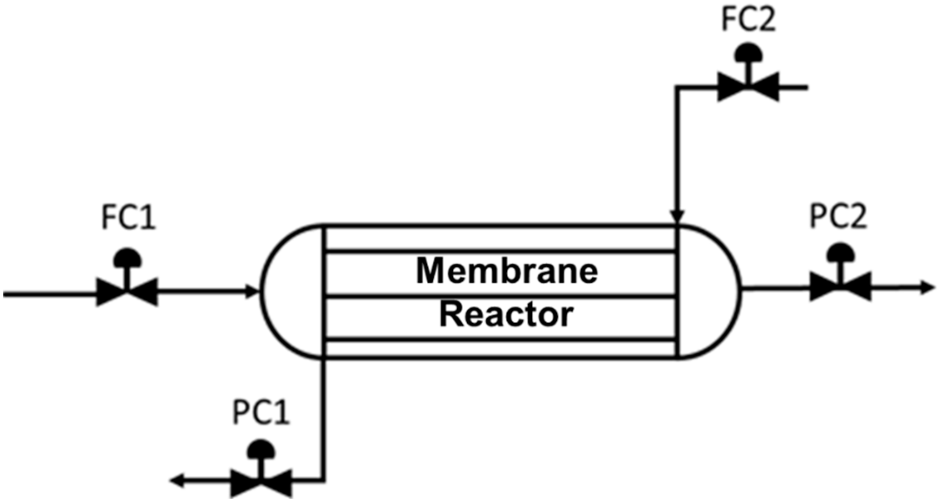
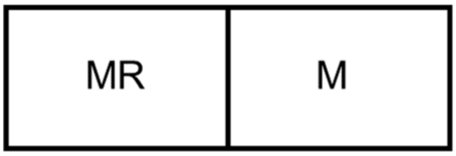
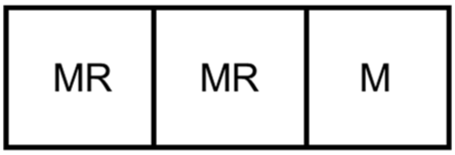
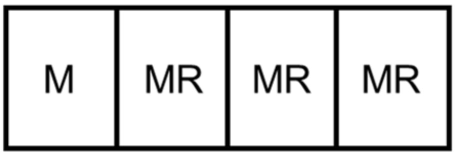


Figure 5. Image of the countercurrent WGS-MR setup where the syngas process flow control valve (FC1) and the sweep gas control valve (FC2) manual positions are changed to generate the AIS.

The purpose of this previous study was to determine if a combination of N modules, rather than a full membrane reactor, could improve the servo-OI of the original membrane reactor and, therefore, improve the unit's ability to service a wider range of syngas flowrates into the process side of the membrane reactor. For the purpose of this research, determining the design that leads to the largest improvement in the servo-OI given using a combination of N modules is referred to as the N-mod problem. Since this initial study was to gain insights into the N-mod problem, it was decided to use an exhaustive algorithm. This means an operability analysis was conducted for every possible modular design for a given N-mod problem. Although this is extremely inefficient, the knowledge gained through doing this would eventually lead to the novel design algorithm that is proposed in this work for solving general N-mod problems for membrane reactor systems. This previous study found global optima for the 2, 3, and 4-mod problems for the WGS-MR case, which are summarized in Table 3.

Table 3 shows that, by switching to a modular design instead (especially in the 4-mod case), significant improvements in operability are achievable. The obtained solutions also make sense as to why they would improve the operability. In all of these cases, the incoming syngas has some hydrogen which ultimately inhibits the forward reaction for WGS. In the 2 and 3-mod cases, the membrane reactor modules are used to produce and capture hydrogen and the final membrane module can only remove hydrogen. This means the front end is used to produce hydrogen, and then the control system can use the sweep gas flow to determine the amount of hydrogen removed at the end of the unit, ultimately giving it more control over the hydrogen recovery and, therefore, a larger servo-OI.

Table 3. Summary of the operability and optimization analyses for maximizing the servo-OI.

N	Optimum Design (For Maximizing the Servo-OI)	Increase in the OI Compared to the Membrane Reactor Base Case
2		12%
3		9%
4		37%

An interesting switch happens though when 4 modules can be used. Instead, the membrane is moved to the front and the servo-OI is significantly improved. This solution also makes sense. In this approach, the control system now has the freedom to use the sweep gas to determine the amount of the initial hydrogen that is removed which will determine the initial reaction rates when the syngas finally enters the membrane reactor modules. In addition, since reactions do not occur at the beginning and the reactant CO has a lower permeance than the product CO_2, this also ends up giving it some freedom to change the carbon capture of the unit (because in the 2 and 3-mod solutions, most of the CO will be converted to CO_2 in the membrane reactor modules before reaching the membrane module at the end of the unit).

This was a great result; however, it was determined using an exhaustive approach of checking every design. As most combinatorics problems do, this exhaustive approach grows at a rate that, at best, is exponential according to the following equation determined in previous work [18]:

$$N_{designs} = 4^{N_{modules}} - 2 \times 2^{N_{modules}} + 1. \tag{11}$$

The objective of this research is to take the lessons learned from previous work to develop heuristics for solving these N-mod problems and propose a novel design algorithm that can reproduce these results, while also significantly reducing the size and complexity of the problem.

4. Design Algorithm Theory and Development

The previous work provided many useful insights into the relationships between the various N-mod problems and their optimal solutions. In this section, the focus changes to the development of a novel design algorithm for solving this class of problems. This is done by using previous observations to develop heuristics for the design of module-based membrane reactor systems. It should be noted (for improved clarity) that the algorithm is developed by assuming a global optimum exists for the N $\rightarrow \infty$ problem and then work back to the 3-mod problem. Therefore, for clarity of presentation, the following sections are in reverse order of how the complete algorithm would run.

4.1. Shortcut Method for Large Values of N

The greatest challenge to solving the general N-mod problem efficiently is the number of possible combinations of modules as the value of N increases. This section provides the thought process behind a proposed novel approach to addressing this challenge. The first step is to think about the global solutions for some N-mod problem, where N is very large, say, the 30-mod problem. Assume Figure 6 is the global optimum to the 30-mod problem, in which reactor modules are colored red, membrane modules are colored blue, and membrane reactor modules are colored purple.

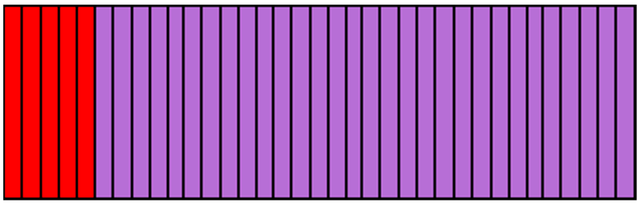


Figure 6. A representation of the solution to a hypothetical 30-mod problem with reactor modules (red) and membrane reactor modules (purple).

The next step is to take one step back and ask the question: What is the global optimum of the 29-mod problem? Note: The total length of the entire unit remains constant throughout this process. Figure 7 shows two potential designs that could be the global optimum for the 29-mod problem.

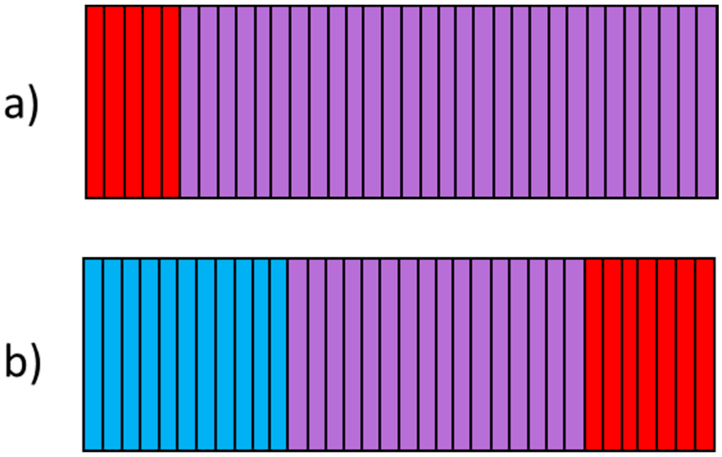


Figure 7. Two hypothetical 29-mod designs with reactor modules (red), membrane modules (blue), and membrane reactor modules (purple). Here, (**a**) is a 29-mod design that is very similar to the 30-mod design in Figure 6, whereas (**b**) represents a very different 29-mod design.

Knowing that the global optimum to the 30-mod problem is the one shown in Figure 6, intuition would suggest that the first design shown in Figure 7a is much more likely to be the global optimum to the 29-mod problem than the second design is. This means that, if the solution to one N-mod problem is known, then this provides insight into what, say, the (N + 1)-mod problem solution could be. Because of this, an algorithm is developed to determine the "best guess" for the (N + 1)-mod problem given the solution to the N-mod problem.

As shown in Figure 8, the (N + 1) guess can be generated by taking a given solution to the N-mod problem and recutting it as if it had one additional module. The algorithm then calculates which phenomena block occupies the most length within the new module and then assigns it as such. For example, in the middle two modules, although there is some red (reactor) and some blue (membrane), purple (membrane reactor) occupies the majority of the new module; therefore, it is assigned as a membrane reactor in the guess. As observed in previous work [18], as N gets larger (tends to infinity), the difference between the best guess and the actual global optimum may only differ by one module at most. This gives the first heuristic for the design of these membrane-based systems:

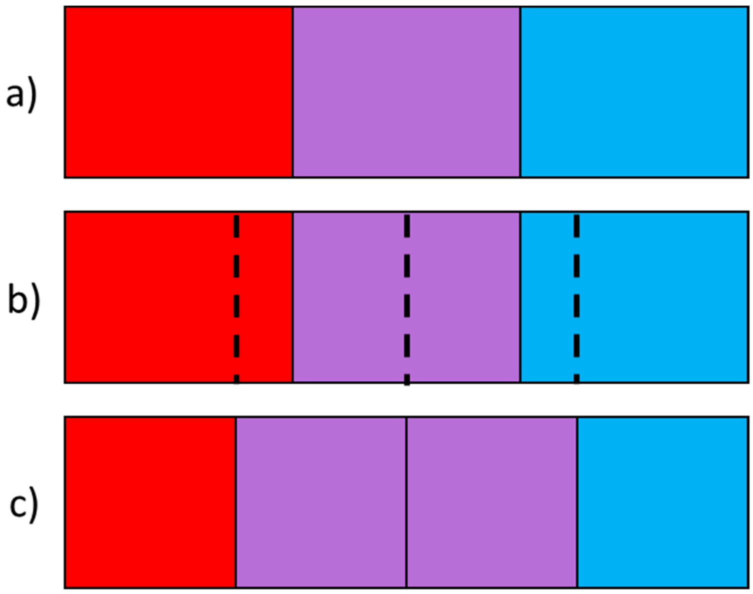


Figure 8. Example of a 3-mod solution generating the "best guess" to the solution of the 4-mod problem. (**a**) represents the hypothetical solution to a 3-mod problem, (**b**) is the 3-mod solution arranged as a pseudo-4-mod design, and (**c**) is the 4-mod design that most closely resembles (**b**), i.e., the best 4-mod guess.

Heuristic 1. *As N becomes sufficiently large* $(N \to \infty)$, *the actual global optimum for the N-mod problem will only differ from a guess generated from the known global optimum of the* $(N-1)$*-mod problem by exactly one module.*

This heuristic holds because this specific problem is solving for the fixed total length version of the problem. As N approaches infinity, the ratio of the length of the modules in the N and N + 1 problems approaches unity. This means that as N gets larger, the modules tend to become the same length despite the value of N increasing by one and ultimately result in identical solutions. Although the optimal solutions of the N-mod and (N + 1)-mod problems become identical in functionality as N goes to infinity, one still has one more module than the other. This means the only necessary work at this stage would be to find the correct module (membrane separator, reactor, membrane reactor, or heat exchanger) for the one new module that is added when going from the N to (N + 1) problem.

So, once a guess is generated for the N-mod problem, one only needs to simply swap each of the N modules of the guess with the other two candidate modules (for example, swap a membrane reactor module with a reactor or membrane module) and check the OI of that design and the global optimum will be the design with the largest OI in that group of designs. This process is almost identical to performing a grid search and constantly refining the grid as one approaches the optimum. This results in a massive reduction in the size of the problem for larger values of N as the number of designs that must be checked in this approach is bounded quadratically rather than exponentially as follows:

$$N_{designs} \leq \frac{3}{2}n^2{}_{modules} + \frac{5}{2}n_{modules} - 4. \tag{12}$$

A block flow diagram is provided in Figure 9 to help visualize the algorithm for N-mod problems with sufficiently large values of N which will now be referred to as the "Shortcut Method." Note that this Shortcut Method assumes the design fed into it initially is optimal and then finds the optimal designs for higher values of N using it as an initial guess. Therefore, feeding a suboptimal design will likely lead to finding suboptimal solutions to the larger problems.

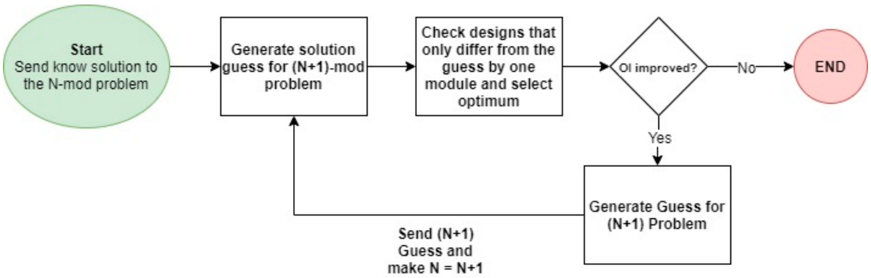

Start
Send know solution to the N-mod problem
Generate solution guess for (N+1)-mod problem
Check designs that only differ from the guess by one module and select optimum
OI improved?
No
END
Yes
Generate Guess for (N+1) Problem
Send (N+1) Guess and make N = N+1

Figure 9. Algorithm for finding the optimum design for an N-mod problem given a known solution to a smaller (N − 1)-mod problem and that N is sufficiently large. This algorithm is referred to as the "Shortcut Method".

4.2. Determining When N Is "Sufficiently Large"

This algorithm in the previous section leaves two natural, unanswered questions: (1) When is N "sufficiently large" to use this approach, and (2) what should be done instead when N is deemed too small? This leads to the second heuristic for the algorithm:

Heuristic 2. *N is considered sufficiently large when the difference in performance between the optimum N-mod solution and the (N + 1) guess generated from it differs by a pre-specified tolerance level, based on the flux and reaction rates occurring in each module.*

This heuristic is conceptually similar to the guessing algorithm, but in this case there is a comparison of the flux and reaction rates occurring in each module rather than just the identities of the modules. Figure 10 is provided to demonstrate how such a comparison may be performed.

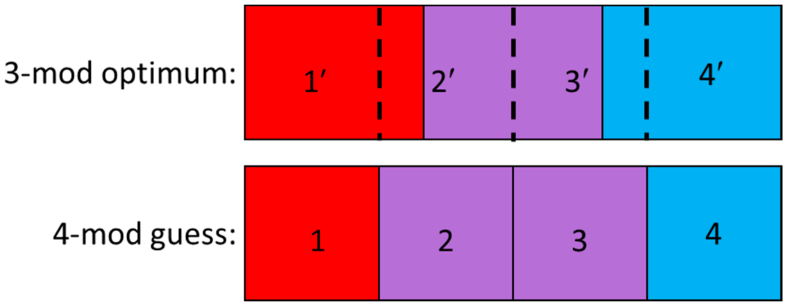


Figure 10. An example of how a one-to-one comparison can be performed to determine the similarity of performances between an optimum solution and the guess generated by it.

In the Figure 10 example, a 4-mod guess is first generated from the 3-mod optimum solution. The 3-mod optimum is then redivided into 4 pseudo-modules denoted with the prime notation. This allows for a one-to-one comparison between the two designs. Heuristic 2 essentially says that, if the differences in flux and reaction rates between module 1 and 1′, module 2 and 2′, etc., are very similar, and it is assumed that the 3-mod design here is the optimal design, then, the 4-mod guess is likely the optimal design or very close to the optimum (i.e., one module away) for the 4-mod problem. Heuristic 2 in practice is the same as Heuristic 1, except it takes the specific system into account, whereas Heuristic 1 is not dependent on the specific system.

There are likely many ways to assess how similar two designs are, but for this work, the following steps are taken to determine the similarities:

1. Generate the (N + 1) guess from the N-mod optimum solution.

2. Divide the N-mod solution into (N + 1) pseudo modules.
3. Calculate the average flux and reaction rates in each module of the (N + 1) guess and in each pseudo-module of the N-mod solution.
4. Calculate the error in flux and reaction rates between 1 and 1′, 2 and 2′, etc.
5. Choose the largest of these error values as the error between the two designs.
6. If the error does not exceed the user's predefined tolerance (e.g., 10%), then, N is sufficiently large and the shortcut method can be used.

For this research, the relative error between two modules is defined using Equation (13):

$$E_{B,i} = \max\left\{\frac{|B_i - B_{i'}|}{\max\{B\}}\right\}, \tag{13}$$

where *i* and *i*′ refers to the module and pseudo-module being compared, B_i is the average of the quantity being compared (in this case flux and reaction rates) in module *i* or *i*′, and max{B} is the maximum value of quantity B of every module in the analysis. The max{B} quantity is used to convert the absolute errors into relative errors. For this work, an error of less than 10% is considered acceptable; however, in the final form of the algorithm, this tolerance is ultimately up to the user.

4.3. Solving N-Mod Problems When N Is Too Small

The only remaining case to consider is how to solve the problem when N is too small to use the Shortcut Method. One option would be to go back and use the exhaustive method of checking every design used in previous work [18]. This would be fine for maybe the 2-mod problem which, according to Equation (11), only has 9 designs to check or the 3-mod problem with 49 to check, but what happens if the error conditions are not met by then? The 4-mod problem takes several hours to calculate through exhaustive checks due to having 255 valid combinations to check. Instead, an approach, such as a genetic algorithm, is a better candidate as it searches more intelligently through the viable space of designs rather than simply checking every single one. Genetic algorithms are also very useful for mixed integer-based optimizations, such as the one in this work. First, a system is developed for encoding these modular designs as strings of digits of ones and zeros, as shown in Figure 11.

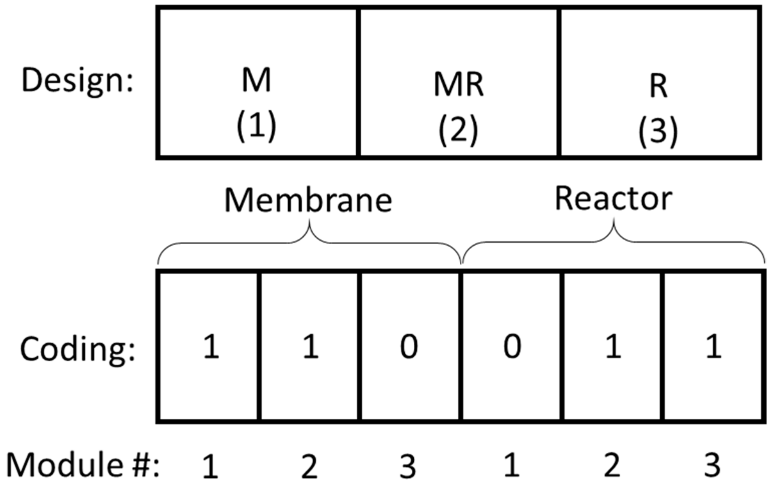


Figure 11. An example of the encoding process to convert a given modular design to its coded form for the genetic algorithm.

This coding system is then implemented using the genetic algorithm2 library [28] available in Python. However, simply applying a genetic algorithm to this problem leads to many problems. One way to better understand this is to look at the design spaces for these N-mod problems. Figure 12 shows the design space for the 2-mod problem in the form of a graph.

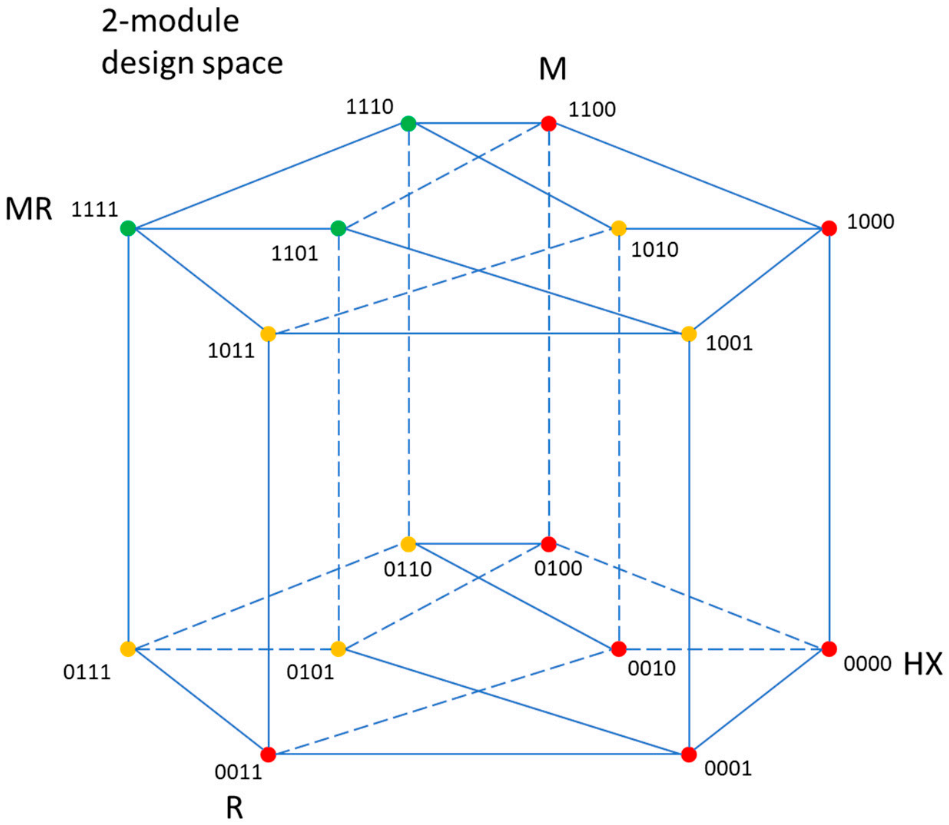


Figure 12. A graphical representation of the hypercube that defines the design space for the 2-mod problem with invalid designs (in red), valid designs with an OI of zero (in orange), and valid designs with a non-zero OI (in green). The pure heat exchanger (HX), membrane separator (M), reactor (R), and membrane reactor (MR) are labeled for reference.

In Figure 12, every N-mod design can be represented as a vertex on a 2N hypercube. For the 2-mod problem, there are three potential outcomes for a given design: (1) the design could have either no membrane or no reaction zones and, therefore, is an invalid design because it defeats the purpose of doing process intensification in the first place (red points); (2) the design could be valid but has an OI of zero because none of the inputs map to a desired output (orange points); and (3) the design could be valid and have a nonzero value for the OI (green points). Most genetic algorithms will produce a random initial population to initialize the algorithm. Since there are 16 total possibilities, in this case there is interest in doing two iterations with generations of size four using a genetic algorithm to cut the number of checks in half from the exhaustive approach. Since 13 out of 16 designs produce an OI of zero or are designs that are not worth the computational expense to check, there is a 57% chance that every design in the initial population has an OI of zero. This would give the optimizer no information toward learning how to improve the OI; therefore, additional iterations need to be added to ensure an optimum solution is determined.

The solution is to provide a population to initialize the genetic algorithm rather than letting it randomly pick. One point to remember is that the goal of this algorithm is to improve the operability of an intensified membrane reactor system. That objective also suggests how an initial population could be generated. The initial population should consist of the original membrane reactor, as well as designs that are random mutations of the membrane reactor. This effectively searches the subspace of designs around the membrane reactor instead of the entire design space during the first iteration of the genetic algorithm. The important fact is that the genetic algorithm should get examples of designs that have a nonzero OI so that it can learn what increases or decreases the OI so that the following iterations are attempting to find a better design than the membrane reactor, not just trying again to find a design that works.

The first run of the genetic algorithm is the only time over the course of the entire algorithm that the membrane reactor is used as the main blueprint for the population initialization. For example, the membrane reactor may be used to initialize the population of the genetic algorithm to solve the 3-mod problem. After it finishes running, it will return an optimum design for that problem and a guess will be generated (as described earlier) and this guess will be checked to see if N is large enough to use the Shortcut Method. If the error is too great, it means the genetic algorithm must be used again for the 4-mod problem. Instead of just using the membrane reactor to initialize the population for this problem, both the membrane reactor and the 4-mod guess that is generated are used. This is for two reasons: (1) As discussed through much of this work, the solution to the 3-mod problem gives insight into how the 4-mod optimal design might look and, therefore, is a good candidate for basing the initial population after; (2) including the membrane reactor again adds some diversity to the initial population of this problem and helps to reduce the chance of being pulled into a local minimum found by the previous N-mod optimization.

In summary, when N is small, a genetic algorithm is used to search the space and find optimum designs. This process continues by feeding the optimum design back into the genetic algorithm until it is determined that the value of N is now sufficiently large. When this occurs, the algorithm switches to using the Shortcut Method until the OI can no longer be improved. This algorithm is summarized in detail using a flowchart in Figure 13.

To better understand the concept behind the algorithm, an analogy is provided now. One way to think about this is when one is presented a very distorted, pixelated image and is asked to figure out what the image is. When the problem starts, there are many images it could be, and it is hard to say what the answer is. Although the image is not random, it is a distorted version (small value of N) of a very clear image (solution as $N \to \infty$). In this case, the genetic algorithm is the initial process to try and refine the image to be clearer and easier to guess. However, anyone who has played one of these "guess the pixelated image" games can normally guess the image at some point despite it still being pixelated. This moment when one can accurately guess the image despite the pixelation is synonymous to the tolerance check that switches the algorithm from the genetic algorithm to the shortcut method. Therefore, the Shortcut Method is the stage after the player recognizes the final image and simply needs to correct some of the pixels to get to the final solution. This framework is implemented using the following three-part software infrastructure: (1) The WGS-MR model and all simulation calculations are performed in the AVEVA Process Simulation (APS) platform; (2) the design algorithm, specifically the population initialization method, genetic algorithm, operability analyses, guess-generation method, and the shortcut method, are all written in Python; and, (3) to allow the methods in Python to interact with the model in APS, the APS Python Scripting Interface (developed by AVEVA) is utilized. This is a Python library consisting of several methods that a Python user can incorporate into their codes for controlling the simulator using Python. The actual output of this Python/APS infrastructure is discussed further in the subsequent Results section.

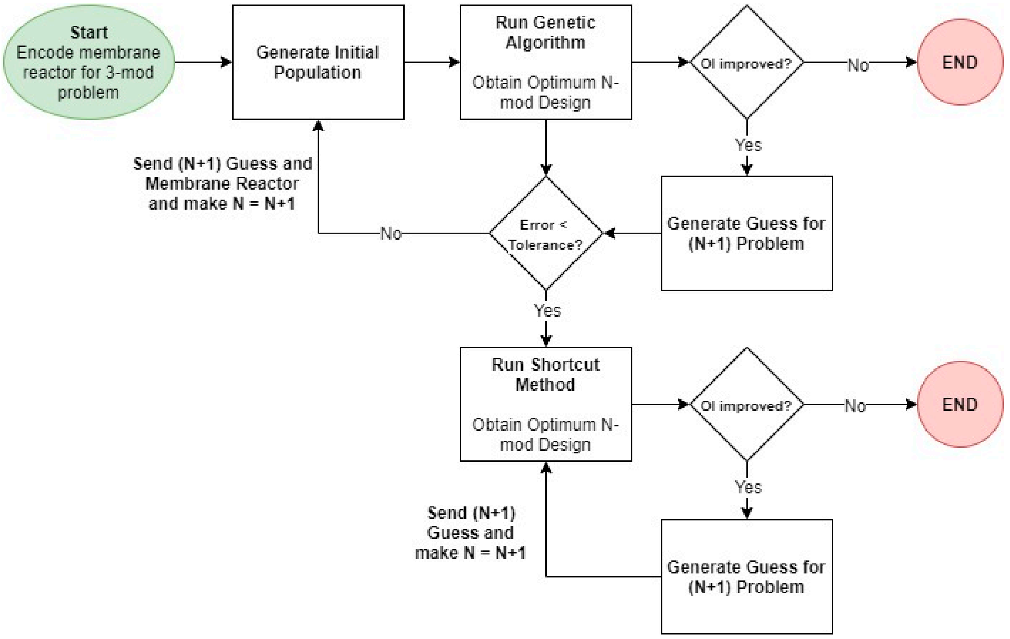

Start
Encode membrane reactor for 3-mod problem
Generate Initial Population
Run Genetic Algorithm
Obtain Optimum N-mod Design
OI improved?
No
END
Yes
Generate Guess for (N+1) Problem
Error < Tolerance?
No
Send (N+1) Guess and Membrane Reactor and make N = N+1
Yes
Run Shortcut Method
Obtain Optimum N-mod Design
OI improved?
No
END
Yes
Generate Guess for (N+1) Problem
Send (N+1) Guess and make N = N+1

Figure 13. The proposed algorithm for the design of modular membrane systems for improved operability performance.

5. Results

Now, the algorithm is applied to the water-gas shift membrane reactor (WGS-MR) system from previous work to assess its accuracy and computational time. This system was introduced in Figure 5 above.

In this figure, there are four control valves available as inputs to the WGS-MR system. FC1 and FC2 are the process syngas flow control valve (tube side) and the steam sweep gas control valve (shell side), respectively. PC1 and PC2 are pressure control valves for the shell and tube sides, respectively. Although the transport across the membrane is heavily dependent upon the pressure drop across it, these pressure control valves have little effect on the system. This is because the pressure drop across the membrane is around 20 bar and these control valves negligibly affect this driving force. For this reason, only the two flow controllers are considered in this analysis.

The membrane reactor design parameters were determined through an optimization of the nominal operating point to produce the highest hydrogen recovery and carbon capture percentage that the pure membrane reactor could achieve. As stated in the Introduction, this improvement in efficiency through process intensification also comes with the tradeoff that the two flow controllers have less control over the system. So, the operability analysis should focus on how the control moves of these two controllers are mapped to the output space.

For this case study, the available input set (AIS) is defined as the combination of every possible valve position the two control valves (FC1 and FC2) could assume (in this case this range is from 10–100% open for both valves). To perform the operability analysis, each of these combinations is simulated for a given module-based design to find the set of all achievable hydrogen recovery and carbon capture percentages (AOS). Lastly, a definition of what is a "desired" output (or DOS) is needed for the analysis. For this work, a desired operating point is defined as one that achieves higher than 85% hydrogen recovery and higher than 85% carbon capture. Figure 14 is provided here to assist in visualizing the problem:

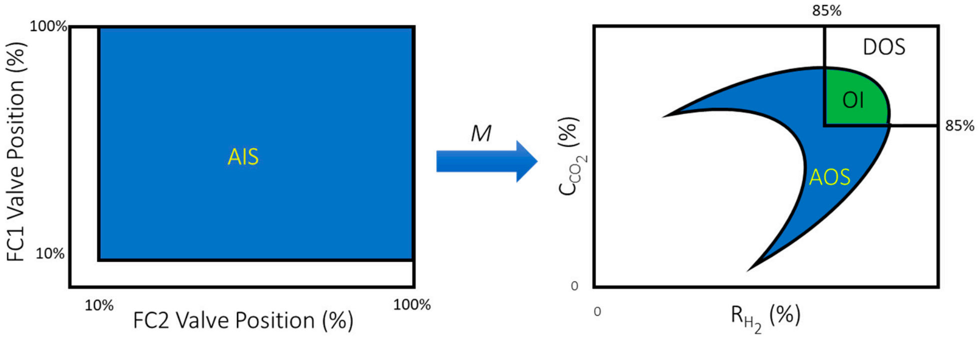


Figure 14. A general visualization for the WGS-MR case being studied. Here, the AIS is defined as the set of all control valve positions available and the DOS is defined as the set of all operating points with greater than 85% for both hydrogen recovery and carbon capture.

Referring to Figure 14, the proposed algorithm is attempting to find an optimal *M* (model) for this problem. *M* in this case represents some module-based membrane reactor design for the WGS-MR system. The optimal *M* for this system is the one that, when it maps the AIS to the output space, covers the largest fraction of the DOS. Therefore, the input set remains fixed throughout this process and the algorithm only modifies *M*. Described qualitatively, the AIS defines everything a hypothetical control system (whether that be PID controllers, model predictive control, etc.) has available to it. Then, the algorithm is making changes to *M* to find the design that allows the control system to access more of the desired output set than the original membrane reactor alone was able to achieve. This is the benefit of using operability and operability analysis for the DOF challenge as it simultaneously considers both the design and control of the system.

To understand how the results are determined, the outputs of the algorithm are discussed sequentially here. First, since the best-known design at the start of the problem is the membrane reactor (as no other designs have been considered), then the initial population of the genetic algorithm is generated by randomly mutating the membrane reactor. Then, additional members are created by randomly selecting a member of the current population and randomly mutating again. It should be noted that the mutation process described here for the population initialization should not be confused with the one done by the genetic algorithm on subsequent generations using the traditional approach (where there is a probability of mutation or crossover happening). Figure 15 shows an example of the output of this initialization algorithm using the colored-coded blocks established earlier.

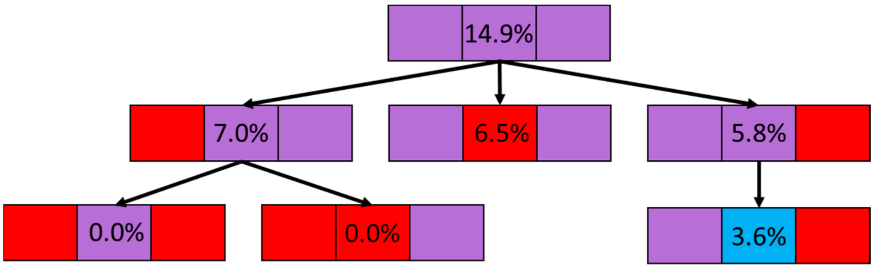


Figure 15. An example of how members of the initial population are randomly generated from the membrane reactor where red is for reactors, blue are membrane separators, and purple are membrane reactors (note that not all ten members are shown due to a lack of space). The % values shown inside the middle blocks are each design's respective OI.

As one can see in Figure 15, this randomized approach led to some reactor-heavy candidates for the genetic algorithm to initially test. When each of the designs are checked, the genetic algorithm determines that the original membrane reactor is the best design for maximizing the OI. The genetic algorithm also finds that as more membrane is removed, the OI seems to decrease. This is why, for the following generation, the genetic algorithm generates a population that removes more catalyst rather than membrane and determines the optimal design to be the one shown in Figure 16.

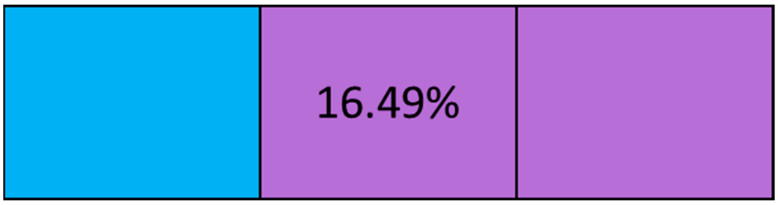


Figure 16. The optimum solution for the 3-mod problem (OI = 16.49%) according to the genetic algorithm with one membrane separator module (blue) and two membrane reactor modules (purple).

The algorithm then checks to see if the OI has been improved from the previous optimal design (pure membrane reactor). In this case, the OI was improved by about 10.7% to an OI of 16.5%, so, the algorithm should continue to improve the design by attempting the 4-mod problem. A guess for the 4-mod problem is generated and compared to the 3-mod optimal design to see if they are similar enough (within the 10% tolerance) to use the Shortcut Method or to continue with the genetic algorithm. Figure 17 shows the visual comparison between the 3-mod optimum and its 4-mod guess.

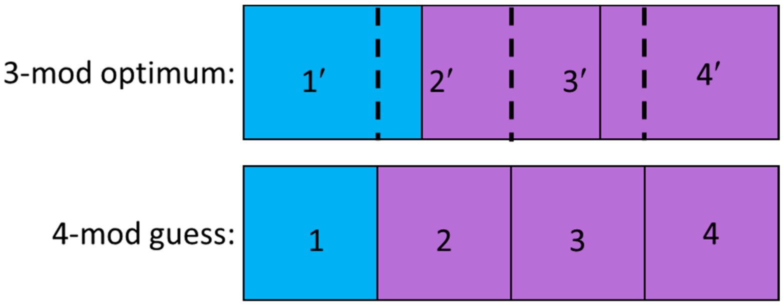


Figure 17. Comparison of the 3-mod optimal design and the 4-mod guess generated from it. The 3-mod design is split into 4 pseudo-modules to compare their relative permeation and reaction rates.

In this case, the algorithm determines the relative error as defined by Equation (13) for the reaction rates is too high in the second module (about 15%). This means there is a chance the actual 4-mod optimal design could differ from this guess by more than one module; therefore, the genetic algorithm is preferred over the Shortcut Method. However, this guess (4-mod) is still based on a design (3-mod) that is known to be better than the pure membrane reactor; therefore, the next population initialization for the genetic algorithm should be done using the pure membrane reactor (as in in Figure 15) but also should include the 4-mod guess. This combination allows the genetic algorithm to both attempt to improve on the currently best-known design, as well as search other alternatives to the membrane reactor that could be better than what is currently known.

Upon completion of the next iteration, the genetic algorithm determines that the 4-mod guess shown in Figure 17 is in fact the optimal design for the 4-mod problem. The operability index is compared to the 3-mod optimal design and the OI has been improved to 17.97% with an improvement of 8.9% from its 3-mod counterpart. This means the algorithm should be continued to see if there is a 5-module design that could further improve the OI. A 5-mod guess (shown in Figure 18) is generated and compared to the 4-mod optimal design. In this case, the relative error is now below the 10% tolerance which means the two designs are similar in functionality and the Shortcut Method can now be employed.

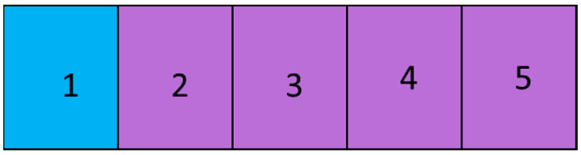


Figure 18. The 5-mod guess based on the 4-mod optimal design. One membrane separator (blue), followed by four membrane reactors (purple).

For the Shortcut Method, first, the OI is determined for the 5-mod guess that is generated. Then, the first module is swapped for a reactor and the OI is calculated. Next it is swapped with a membrane reactor and the OI is determined. This procedure is followed for the second module, then third, and so on, until all neighboring designs are checked. Because of Heuristic 1, the best design out of this set is the assumed to be the 5-mod optimal design. In this case, the 5-mod guess shown in Figure 18 is the 5-mod optimal design.

The OI is checked and is once again improved from the 4-mod optimal design, however, by only about 0.6% to an OI of 18.09%. A 6-mod guess is generated, and the Shortcut Method is run again to determine the 6-mod optimal design. This time, however, the solution that is found leads to a decrease in the OI and the algorithm is terminated. Because the 6-mod optimal design is worse than its 5-mod counterpart, the 5-mod optimal design (which is identical to the guess shown in Figure 18) is selected as the optimal design for the original WGS-MR. This design improved the OI by 21% when compared to the original membrane reactor. Most notably, the proposed algorithm not only found an optimal design for the 3-mod, 4-mod, 5-mod, and 6-mod problems, it also did it significantly faster than using the exhaustive method. Using an Intel® Core ™ i7-4790 CPU @ 3.60 GHz processor, the algorithm found these four optimal designs in 1 h and 36 min. For comparison, to determine that the 5-mod design is in fact the optimal design, exhaustive methods would have taken approximately 5.8 days to do the same. This amount in the reduction in problem computational time means, even if the user was worried the algorithm may have missed the global optimum, they could run it an additional 86 times before the exhaustive method would finally finish. Likewise, they could increase the population sizes or number of generations in the genetic algorithm and still finish in significantly less time.

6. Conclusions

In previous work, it was shown that the OI of a pure membrane reactor could be improved through the use of a module-based design approach. It was also shown that the global optimum solutions to the 2-mod, 3-mod, and 4-mod problems tended to share many similarities, i.e., they were related to each other. However, these global optima had to be determined by checking every possible design for each problem. Although this gives one the confidence that the solution is in fact the global optimum, the time it takes to determine this grows exponentially and eventually becomes intractable to solve anything larger than the 4-mod problem.

The work presented here aimed to address this challenge through the development of a novel design algorithm. This algorithm used a genetic algorithm paired with a population initializing algorithm to determine optimal solutions to the N-mod problem for smaller values of N where the design space is much smaller and easier to search. When the value of N is determined to be sufficiently large, the algorithm utilizes two heuristics for the design of module-based membrane reactor systems to find the optimal designs to larger N-mod problems. This leads to significant reductions in the complexity of the overall problem. Although only one case study is presented here, the justifications for the stated heuristics do not assume a specific membrane reactor system and, therefore, are expected to have application potential to many other systems. Future work will investigate the application of this design algorithm to other possible membrane reactor systems, such as steam-methane reforming to test the generalization of the algorithm and associated heuristics.

The motivation for this work was to find module-based designs that addressed the degrees of freedom challenge by improving the operability index of a membrane reactor system. For the case of the WGS-MR, the 5-mod design from Figure 18 (with one membrane module followed by 4 membrane reactor modules) was determined to be the optimum design and led to a 21% improvement in the operability index as compared to the original membrane reactor. This means that a control system applied to this design can achieve 21% more of the desired output set than one applied to the original membrane reactor. In addition, this novel design algorithm grows at a rate according to polynomial time, whereas the exhaustive method grew exponentially. This means that this work reduced the size of the N-mod problem from NP-complexity to P-complexity, leading to a reduction in computational time by a factor of 87. Future work will focus on expanding the heuristics and algorithm to consider parametric changes, as well, such as changing the length, number of tubes, etc., while simultaneously solving the combinatorics problem.

Author Contributions: This paper is a collaborative work among the authors. B.A.B. developed the algorithm and wrote the manuscript. F.V.L. helped with the paper writing and oversaw all technical aspects of the research work. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the financial support from AVEVA.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors thank the AVEVA Research and Development team for their technical guidance and advice.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Burns, R.L.; Koros, W.J. Defining the challenges for C3H6/C3H8 separation using polymeric membranes. *J. Membr. Sci.* **2003**, *211*, 299–309. [CrossRef]
2. Khalilpour, R.; Mumford, K.; Zhai, H.; Abbas, A.; Stevens, G.; Rubin, E.S. Membrane-based carbon capture from flue gas: A review. *J. Clean. Prod.* **2015**, *103*, 286–300. [CrossRef]
3. Scholes, C.A.; Stevens, G.; Kentish, S. Membrane gas separation applications in natural gas processing. *Fuel* **2012**, *96*, 15–28. [CrossRef]
4. Turton, R.; Bailie, R.; Whiting, W.; Shaeiwitz, J.; Bhattacharyya, D. *Analysis, Synthesis, and Design of Chemical Processes*, 4th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2012.
5. Schembecker, G.; Tlatlik, S. Process synthesis for reactive separations. *Chem. Eng. Process. Process Intensif.* **2003**, *42*, 179–189. [CrossRef]
6. Baldea, M. From process integration to process intensification. *Comput. Chem. Eng.* **2015**, *81*, 104–114. [CrossRef]
7. The Department of Energy. *Advanced Manufacturing Office: Process Intensification Workshop. Report of Conclusions from a Meeting Sponsored by the Department of Energy*; Department of Energy: Alexandria, VA, USA, 2015. Available online: https://www.energy.gov/sites/prod/files/2016/03/f30/2015%20DOE%20AMO%20Process%20Intensification%20Workshop%20Report_2.pdf (accessed on 24 June 2019).
8. Vlachos, D.; Ierapertritou, M.; Dauenhauer, P.; Hock, A. *Modular Manufacturing Workshop. Report of Conclusions from a Meeting Sponsored by the National Science Foundation and the Department of Energy*; National Science Foundation and the Department of Energy: Arlington, VA, USA, 2017. Available online: https://cpb-us-w2.wpmucdn.com/sites.udel.edu/dist/d/1086/files/2018/07/MMW_Report-2o9jwun.pdf (accessed on 24 June 2019).
9. Nikačević, N.M.; Huesman, A.E.; Hof, P.M.V.D.; Stankiewicz, A.I. Opportunities and challenges for process control in process intensification. *Chem. Eng. Process. Process Intensif.* **2012**, *52*, 1–15. [CrossRef]
10. Baldea, M. Multum in parvo: A process intensification retrospective and outlook. In Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design, Cle Elum, WA, USA, 13–17 July 2014; Volume 34, pp. 15–24.
11. Tian, Y.; Pistikopoulos, E.N. Operability and safety considerations in process intensification. *IFAC Pap. Online* **2020**, *53*, 11434–11439. [CrossRef]
12. Pistikopoulos, E.N.; Tian, Y.; Bindlish, R. Operability and control in process intensification and modular design: Challenges and opportunities. *AIChE J.* **2021**, *67*, e17204. [CrossRef]
13. Roy, S.; Eng, P. Consider modular plant design. *Chem. Eng. Prog.* **2017**, *113*, 28–31.

14. Reisinger, D. Is Modular Technology the Next 'It' Trend? *Fortune*, 7 September 2015. Available online: https://fortune.com/2015/09/07/modular-technology-business/ (accessed on 7 August 2019).
15. Beckett, S. World's First Modular Phone Goes on Sale. *BBC News*, 16 December 2015. Available online: https://www.bbc.com/news/av/technology-34982724/fairphone-2-world-s-first-modular-phone-goes-on-sale (accessed on 7 August 2019).
16. Chernoff, A.B.; Borroni-Bird, C.E.; Shabana, M.D.; Vitale, R.L.; Johnson, T.J. Vehicle Body Interchangeability. U.S. Patent 202456, 22 May 2003.
17. Tian, Y.; Pappas, I.; Burnak, B.; Katz, J.; Pistikopoulos, E.N. A Systematic Framework for the synthesis of operable process intensification systems—Reactive separation systems. *Comput. Chem. Eng.* **2019**, *134*, 106675. [CrossRef]
18. Bishop, B.A.; Lima, F.V. Novel module-based membrane reactor design approach for improved operability performance. *Membranes* **2021**, *11*, 157. [CrossRef] [PubMed]
19. Radcliffe, A.J.; Singh, R.P.; Berchtold, K.A.; Lima, F.V. Modeling and optimization of high-performance polymer membrane reactor systems for water–gas shift reaction applications. *Processes* **2016**, *4*, 8. [CrossRef]
20. Tian, Y.; Pistikopoulos, S. Generalized modular representation framework for the synthesis of extractive separation systems. In Proceedings of the 9th International Conference on Foundations of Computer-Aided Process Design, Frisco, CO, USA, 14–18 July 2019; pp. 475–480. [CrossRef]
21. Tylko, M.; Barkmann, S.; Sand, G.; Schembecker, G.; Engell, S. Synthesis of reactive separation processes. In *Integrated Reaction and Separation Operations: Modelling and Experimental Validation*; Schmidt-Traub, H., Górak, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 7–94. [CrossRef]
22. Demirel, S.E.; Li, J.; Hasan, M.M.F. Systematic process intensification using building blocks. *Comput. Chem. Eng.* **2017**, *105*, 2–38. [CrossRef]
23. Demirel, S.E.; Li, J.; Hasan, M.F. Systematic process intensification. *Curr. Opin. Chem. Eng.* **2019**, *25*, 108–113. [CrossRef]
24. Bishop, B.A.; Lima, F.V. Modeling, simulation, and operability analysis of a nonisothermal, countercurrent, polymer membrane reactor. *Processes* **2020**, *8*, 78. [CrossRef]
25. AVEVA. SimCentral Simulation Platform: Process Simulation Reinvented. 2019. Available online: https://sw.aveva.com/hubfs/pdf/datasheet/Datasheet_SE-LIO-SimSci_SimCentral_02-17-updated.pdf (accessed on 8 August 2019).
26. Lima, F.V.; Jia, Z.; Ierapetritou, M.; Georgakis, C. Similarities and differences between the concepts of operability and flexibility: The steady-state case. *AIChE J.* **2010**, *56*, 702–716. [CrossRef]
27. Choi, Y.; Stenger, H.G. Water gas shift reaction kinetics and reactor modeling for fuel cell grade hydrogen. *J. Power Sources* **2003**, *124*, 432–439. [CrossRef]
28. Pascal, D. geneticalgorithm2: Supported Highly Optimized and Flexible Genetic Algorithm Package for Python. Available online: https://github.com/PasaOpasen/geneticalgorithm2 (accessed on 25 October 2021).