Computers and Chemical Engineering 173 (2023) 108225

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Surrogate-based optimization of capture chromatography platforms for the improvement of computational efficiency

Juan J. Romero [a], Eleanor W. Jenkins [b], Scott M. Husson [a,*]

[a] Department of Chemical and Biomolecular Engineering, Clemson University, Clemson, SC 29634, USA
[b] School of Mathematical and Statistical Sciences, Clemson University, Clemson, SC 29634, USA

ARTICLE INFO

Keywords:
Monoclonal antibodies
Mixed-integer optimization
Multi-objective optimization

ABSTRACT

In this work, we discuss the use of surrogate functions and a new optimization framework to create an efficient and robust computational framework for process design. Our model process is the capture chromatography unit operation for monoclonal antibody purification, an important step in biopharmaceutical manufacturing. Simulating this unit operation involves solving a system of non-linear partial differential equations, which can have high computational cost. We implemented surrogate functions to reduce the computational time and make the framework more attractive for industrial applications. This strategy yielded accurate results with a 93% decrease in processing time. Additionally, we developed a new optimization framework to reduce the number of simulations needed to generate a solution to the optimization problem. We demonstrate the performance of our new framework, which uses MATLAB built-in tools, by comparing its performance against individual optimization algorithms for problems with integer, continuous, and mixed-integer variables.

1. Introduction

Capture (a.k.a. bind-and-elute) chromatography is an essential operation in the purification process for monoclonal antibodies (mAb) and other biologics (von Lieres and Andersson, 2010). mAbs have gained importance for treating a number of cancers, Covid-19, heart conditions, and immunological disorders (Hernandez et al., 2018; Taylor et al., 2021). The selection of the most suitable manufacturing process based on techno-economic performance is key to increasing patient access to these therapeutics (Jones and Gerogiorgis, 2022). In chromatography process design for mAb purification, it is common to compare alternatives with fixed process parameters using process simulation software (Bansode et al., 2022; Ding et al., 2022; Yang et al., 2020). However, when parameters are treated as variables, the simulation and optimization require dynamic models and more complex computational frameworks (Behere and Yoon, 2020; Shekhawat and Rathore, 2019). For the evaluation of continuous process alternatives, the simulation involves solving a system of partial differential equations using, e.g., MATLAB or FORTRAN ODE solvers (Baur et al., 2016; Shi et al., 2020). This simulation strategy, in combination with tools like the genetic algorithm, can be used to optimize process variables like flow rate (Gomis-Fons et al., 2021). Moreover, artificial intelligence techniques like reinforcement learning can be used with the specialized software CADET to optimize continuous process variables in chromatography simulations (Nikita et al., 2021). Despite the successful results attained through these methods, their implementation requires software development, which poses an entry barrier to the utilization of these tools in the industry. There is a need for an easy-to-implement framework that can handle complex simulations and optimizations while using widely available software and basic knowledge by the user.

Recently, we developed a computational framework to simulate and optimize the capture chromatography process for mAb purification (Romero et al., 2022). The framework combines commercial software with complementary features to perform sensitivity analyses and optimization on multiple performance indicators from mixed-integer process variables. For the process optimization, we successfully employed the multi-objective genetic algorithm (*gamultiobj*) from the MATLAB optimization toolbox. Nevertheless, these genetic algorithms extend computing time and do not guarantee an optimal solution. With this framework, the optimization of a dual-objective problem, like minimizing process time and cost of goods with chromatography media volume as the variable, required up to two days to generate the set of optimal solutions on a laptop computer (a machine with 8GB of memory running 64-bit Windows 10 Education 21H2 on a Lenovo Ideapad

[*] Corresponding author.
E-mail address: shusson@clemson.edu (S.M. Husson).

https://doi.org/10.1016/j.compchemeng.2023.108225
Received 19 August 2022; Received in revised form 9 March 2023; Accepted 11 March 2023
Available online 16 March 2023

330S-15IKB with a Intel(R) Core(TM) i5–8250 U CPU @ 1.60 GHz). The long computing times resulting from the complexity of these simulations detract from the ultimate goal for developing the framework, which is for industrial scientists to use it to compare process alternatives including continuous purification platforms. This limitation motivates the development of new strategies to solve optimization problems rapidly, without sacrificing accuracy and customizability, or increasing system requirements.

In this study, we investigated two strategies to reduce the optimization computing time: decreasing the simulation calculation time and reducing the number of simulations needed in the optimization algorithm. To decrease the simulation computing time, we replaced part of the simulation that involved complex calculations with a surrogate function (Kim and Boukouvala, 2020). The surrogate function was validated by comparing results against those of the original simulation. To reduce the number of simulations in the optimization loop, we proposed a new optimization framework that combines objective scalarization, variable discretization and optimization algorithms available as functions in MATLAB. We compared the newly developed framework against the direct implementation of unmodified MATLAB optimization algorithms for problems with integer, continuous, and mixed integer variables. The performance of the methods was assessed by comparing the number of function evaluations needed for the optimization. Finally, we generalized the observations of function evaluation performance to provide guidelines for the selection of the optimization algorithm based on the characteristics of the problem to be solved.

2. Methods

2.1. Application problem

In capture chromatography, fluid containing the mAb product passes through a chromatography column packed with media; the product adsorbs to the media while impurities pass through. Then, the product is desorbed from the media with an elution buffer. The adsorption process is represented with a breakthrough curve, which plots the concentration of mAb in the effluent from the chromatography column over time. This breakthrough curve is used to calculate the yield of the operation (i.e., the quotient of the total mass of mAb recovered as product and the total mass of mAb loaded on the column).

This breakthrough curve can be simulated by solving the mass-transport governing equation used to model the system (a non-linear partial differential equation). The selection of the model, the fitting parameters, and the model validation are discussed in Romero et al. (2022). A numerical approximate solution was generated by discretizing the governing equations in space using streamline-upwinded Galerkin finite elements and backward Euler discretization in time (Wilson et al., 2020). The resulting system was solved using FreeFEM finite element software (Hecht, 2012). Once yield was obtained in this dynamic simulation, a steady state simulation of the whole process implemented in SuperPro Designer (Intelligen Inc., 2020) was used to calculate the performance indicators that ultimately constitute the objective functions. The dynamic simulation in FreeFEM is the most computationally demanding calculation of the simulation yet only provides one parameter value (yield) for the evaluation of the objective functions. A general representation of the framework structure is presented in Fig. S1.

2.2. Surrogate function

The simulated breakthrough curve is affected by multiple parameters including the type of media, its volume, residence time (RT, i.e., the quotient of column volume and volumetric flow rate), void volume of the system, feed concentration, and load volume. Depending on the optimization problem, some of these parameters are fixed or can be combined. In our case study, we wanted to simulate large-scale production where it is common to use a single media type, with a fixed RT and feed concentration (Grilo et al., 2017; Hummel et al., 2019; Pollock et al., 2017). In this configuration, we obtain the same breakthrough curve for systems with the same load volume per unit of membrane volume, making the simulation outcome (yield) a function only of the relative load (quotient of load volume and membrane volume).

We take advantage of this behavior to construct a surrogate function to estimate the process yield as a function of the relative load. For this purpose, we built a library of yield values by evaluating different load volumes for a 1 L membrane chromatography module in the dynamic simulation. Then, we used MATLAB's implementation of shape-preserving cubic spline interpolation (The MathWorks Inc., 2022a) to obtain yield as a function of load volume and membrane volume. The surrogate function simulation strategy was validated by comparing its results against the finite element method simulation for a set of random points. The problem used for the validation employs a fixed load volume of 200 L and varies membrane volume. We took a set of 20 membrane volumes from a uniform distribution and calculated the root-mean-square error (RMSE) for yield as an indicator of the accuracy of the surrogate function. The magnitude of the error between the FreeFEM simulation and the surrogate function can be controlled through the point density of the library. To meet the desired RMSE of less than 10^{-3}, the point density was set to one point every 1 L load/L membrane, resulting in a library of 50 points for the selected load volume interval.

2.3. Problem formulation

The performance indicators obtained in the process simulation represent different benefits to the user. Depending on the user preferences the indicators are set as objective functions. To expand our alternatives in MATLAB beyond the *gamultiobj* algorithm, we decided to create a single objective as the sum of weighted objectives. The chosen objectives, cost of goods (COG [USD/g]) and process time (Pt [h]), were normalized by their minimum values (minCOG and minPt) and used to create the optimization objective function shown in Eqs. (1)–(3). The parameter Wcog represents the weight given to COG, which ranges from 0 to 1. Since we have two objectives, the weight for Pt is 1-Wcog. This parameter can be selected by the user or set as an array of values to evaluate tradeoffs between objectives.

$$\min_{x} f(x) = Wcog \times \frac{COG(x) - minCog}{minCOG} + (1 - Wcog) \times \frac{Pt(x) - minPt}{minPt} \tag{1}$$

$$St.\ LL \leq x \leq UL \tag{2}$$

$$x = (V_{media}, V_{load}) \tag{3}$$

In this problem, the variables are bounded by a lower and upper limit (LL and UL) defined by the feasible range of operation (in our case [4.8–32] L and [50–200] L for V_{media} and V_{load}). Both variables can be treated as integer or continuous depending on the process specifications set by the user. If the chromatography module is custom-made, its volume (V_{media}) can be any value. However, it is more likely to come in a standard size with a discrete volume (multiples of 1.6 L in our case). Similarly, load volume (V_{load}) can be a continuous variable if there is a continuous supply of feed material. Conversely, if there is a fixed batch volume that needs to be processed, V_{load} would be a discrete value (increments of 50 L in our case). In this case study, we explored three types of problems: both variables are integer, both are continuous, and a mixed-integer problem where V_{media} is an integer variable and V_{load} is continuous.

2.4. Optimization algorithms

We want to select the optimization algorithm that yields the Pareto optimal front for a problem in the least amount of time. To this end, we solve our optimization problems with the suitable alternatives and compare their performance in terms of number of function evaluations. The multi-objective genetic algorithms directly provide a Pareto front, while the methods using the weighted single objective need to be run several times with different weights. In those cases, we build a Pareto front by optimizing the function f(x) for 11 evenly spaced values of Wcog from 0 to 1. The weighted sum method is suitable for our application, which is known to be a convex problem based on the previous results and the results from the genetic algorithms.

In the integer-variable problem, the search space is finite, so we have the alternative of evaluating each point and finding the global minimum instead of using an optimization algorithm. This search space evaluation (SSE) strategy is the base case against which all optimization algorithms are compared. The SSE also is used to set the minimal values for COG and Pt (minCOG and minPt) used to normalize the objective function (Eq. (1)).

The use of a single objective function gives us access to a more extensive selection of optimization algorithms to solve our problem in MATLAB, including the single objective genetic algorithm (*ga*) and the constrained nonlinear optimization function *fmincon* that uses an interior-point algorithm. These methods offer different advantages and drawbacks. *fmincon* usually requires fewer function evaluations than *ga* but it is limited to problems with continuous variables and is susceptible to terminations once a local minimum has been found (The MathWorks Inc., 2022b). Moreover, *fmincon* requires the selection of a starting point, which we set as the midpoint between the bounds of the search space.

The *ga* requires a high number of function evaluations but allows mixed-integer variables. This method is an evolutionary algorithm that works based on principles of natural selection. It selects random points in the search space to form an initial population. It evaluates these points and selects the ones displaying the lowest values for the objective function to be the parents of the next generation. The algorithms also introduce mutations to evaluate more conditions in the search space. After several generations the population evolves, and the Pareto optimal solutions can be found (The MathWorks Inc., 2022c). Increasing the size of the population and the number of generations makes it more likely for the algorithm to find optimal solutions. Nevertheless, these values must be limited because each member of the population from every generation represents a function evaluation. Therefore, the function *ga* requires specification of two parameters: maximum number of generations (MaxGen) and population size (PopSize).

To find the *ga* parameters that yield the optimal solution with the lowest number of function evaluations, we performed a sensitivity analysis by varying MaxGen and PopSize for a fixed Wcog of 0.5. We repeated this analysis ten times and compared the average minimum value of the objective function against the global minimum found with the SSE. The same procedure was applied to find the *ga* parameters for the continuous-variable and mixed-integer problem. This time *fmincon* was used to find the global minimum. For *gamultiobj*, the different combinations of MaxGen and PopSize were compared against the results of *fmincon* (Fig. S3), and the 10 × 100 (MaxGen = 10 x PopSize = 100) configuration was selected.

In addition to these off-the-shelf optimization tools, we developed a new optimization framework that formulates and solves two problems, first using integer variables, and then using continuous variables. The method starts by discretizing the continuous variables given a step size defined by the user. Once all variables are converted to integers, we perform a SSE; i.e., we evaluate every combination of discrete points and select the conditions that yield the minimum value for the objective function. Then, the results are used to formulate a new problem where the integer variables are kept fixed and the results corresponding to the continuous variables are used as the starting point for the optimization algorithm. The new problem featuring solely continuous variables is solved using *fmincon*. This new optimization framework can be used for mixed-integer or continuous-variable problems. This method requires the selection of the discretized values for the continuous variables, which for simplicity were set to be the values in the integer-variable problem. A representation of the information flow and method steps are provided in Fig. S2.

3. Results and discussion

3.1. Surrogate function

Fig. 1 shows the dynamic simulation results for the finite element method and the surrogate function. In the figure we see an overlap between the values obtained with the two simulation strategies. The error of the surrogate function was quantified through the yield RMSE (8.7 × 10^{-4}). This value met the desired level of accuracy for our case: an error lower than 10^{-3} which is the uncertainty of the original data. In addition to the calculation of the residuals, the validation points were used to measure the computing time. The use of the surrogate function reduced the average computing time from 177 s to 13 s per function evaluation. In the context of an optimization using a genetic algorithm with a 10 × 100 (10 generations with population of 100) configuration, it translates to a reduction in computing time from 49.2 h to 3.6 h.

In our application, we used breakthrough data obtained through simulations to build the library used by the surrogate functions. However, the library values can be obtained directly from experimental breakthrough curves. This way, this framework can be implemented without requiring a complex dynamic simulation. Moreover, the user can employ experimental data specific to their system, bypassing the task of finding a suitable model and model parameters.

3.2. Parameters selection for the optimization methods

Table 1 presents the percent deviation from the minimum value of f (x) for different combinations of *ga* parameters. As a result of this analysis, we selected the 3 × 25 configuration, as it yields the lowest number of function evaluations from all alternatives with our acceptable level of convergence (≤1% of deviation from the minimum value of the objective function).

3.3. Optimization algorithms performance

Once the optimization parameters were set, the Pareto fronts were calculated for all three types of problems (Figs. 2–4). In Fig. 2 for the integer-variable problem, we see how the *ga* captures the shape of the Pareto front but yields suboptimal results when Wcog approaches 1 (i.e., when COG reaches its minimum value). In contrast, for the continuous-variable problem (Fig. 3), we see how the *ga* does not provide optimal points as Wcog approaches 0 (i.e., when Pt reaches its minimum value). This behavior shows that the 3 × 25 configuration may not be suitable for all weights. The other three methods yielded optimal points, with

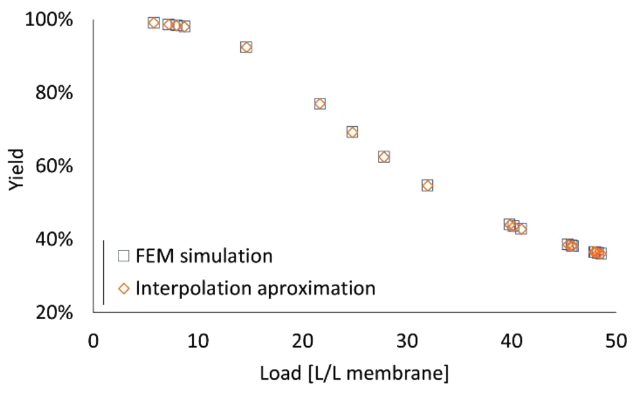

Fig. 1. Dynamic simulation results obtained with FreeFEM and surrogate function for a set of 20 random points.

Table 1
Deviation from the minimum value of the objective function for different *ga* parameter configurations.

Integer	MaxGen			
PopSize	1	2	3	5
5	30%	20%	5.3%	9.4%
10	8.1%	5.5%	3.6%	4.6%
25	1.9%	1.6%	0.0%	0.0%
50	1.4%	0.0%	0.0%	0.0%

Continuous	MaxGen			
PopSize	1	2	3	5
5	26%	9.9%	17%	9.8%
10	8.4%	6.7%	4.3%	6.7%
25	4.7%	3.6%	1.4%	1.2%
50	2.3%	2.0%	1.3%	0.9%

Mixed-Integer	MaxGen			
PopSize	1	2	3	5
5	14%	10%	5.7%	7.9%
10	5.6%	2.4%	1.7%	3.0%
25	1.6%	1.1%	0.3%	0.2%
50	1.0%	0.7%	0.1%	0.0%

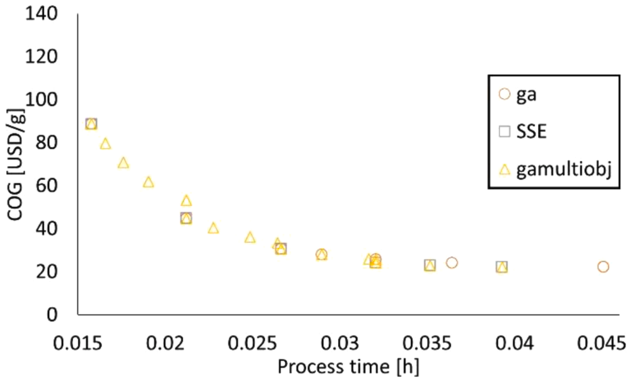

Fig. 2. Pareto fronts for integer-variable problem.

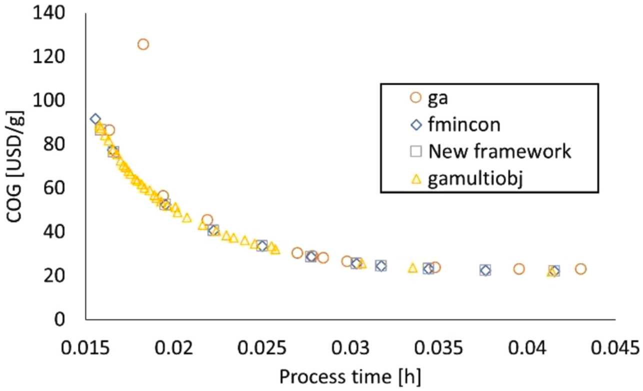

Fig. 3. Pareto fronts for continuous-variable problem.

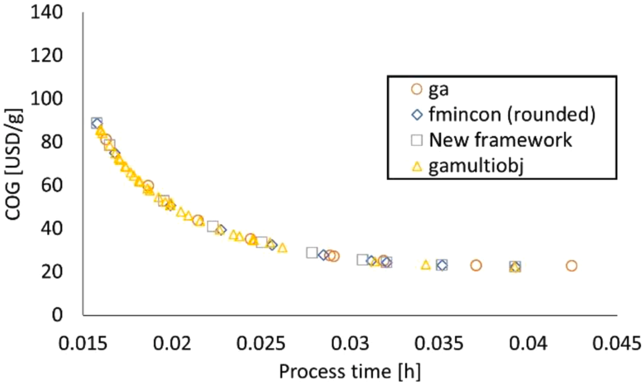

Fig. 4. Pareto fronts for mixed-integer problem.

fmincon and the new optimization framework yielding the same results for almost all weights. This result is consistent with the fact that both methods utilize *fmincon* but with different initial values.

In Fig. 4 for the mixed-integer problem, all methods formed overlapping Pareto fronts. This finding shows that the strategy used with *fmincon* of solving the problem as if all variables were continuous and then rounding the results can yield accurate results for our study. We caution that this strategy may not be suitable for other cases where the rounding of multiple variables can yield suboptimal results. In addition, although the *ga* yields suboptimal points, deviations from optimality are minor compared with the continuous-variable problem.

Table 2 summarizes the performance in terms of number of function evaluations for all problems and methods. Using this information with the Pareto fronts, we can select the best optimization method for each problem in our study. For integer variables, we get the best accuracy and efficiency with the SSE. The lower number of function evaluations for SSE over the genetic algorithms is due to the relatively small search space of our problem, which makes the evaluation of every condition computationally efficient. With continuous variables, the best option is *gamultiobj* since it yields all optimal points in the Pareto front and displays the lowest number of function evaluations among the four algorithms. For the same reasons, the new framework is the best option for the mixed integer-variable problem. In this case, there is a large reduction in number of function evaluations because the optimization step using *fmincon* is only applied to the continuous variable (V_{load}).

3.4. General selection criteria

Since the performance of the algorithms depends on the characteristics of the optimization problem, we developed a flowchart (Fig. 5) to aid the selection of the best optimization method for any given case. The flowchart includes mathematical relationships to determine the method that yields the lowest number of function evaluations. It can be applied to integer, continuous, or mixed integer problems with n variables having l_i levels each and m objectives with W_j weightings. It uses a representative number of function evaluations (FE) for each optimization algorithm (FEGA for *ga*, FEMOGA for *gamultiobj*, FEFmin for *fmincon*, FENF for the new framework), which is estimated by solving the problem with sample weights.

The selection algorithm was applied to the continuous-variable problem of our case study as proof of concept. In our problem, the significant reduction in the number of function evaluations using the new framework favors this method over *fmincon* in decision point 4. Then, the lower number of function evaluations favors the *ga* over the new

Table 2
Number of function evaluations for each algorithm and problem.

Algorithm	Integer	Continuous	Mixed-integer
SSE	72	N/A	N/A
ga	1111	1100	1100
fmincon	N/A	1283	1283*
gamultiobj	716	1000	1000
New framework	N/A	1122	264

* Rounding the results for the integer variable.

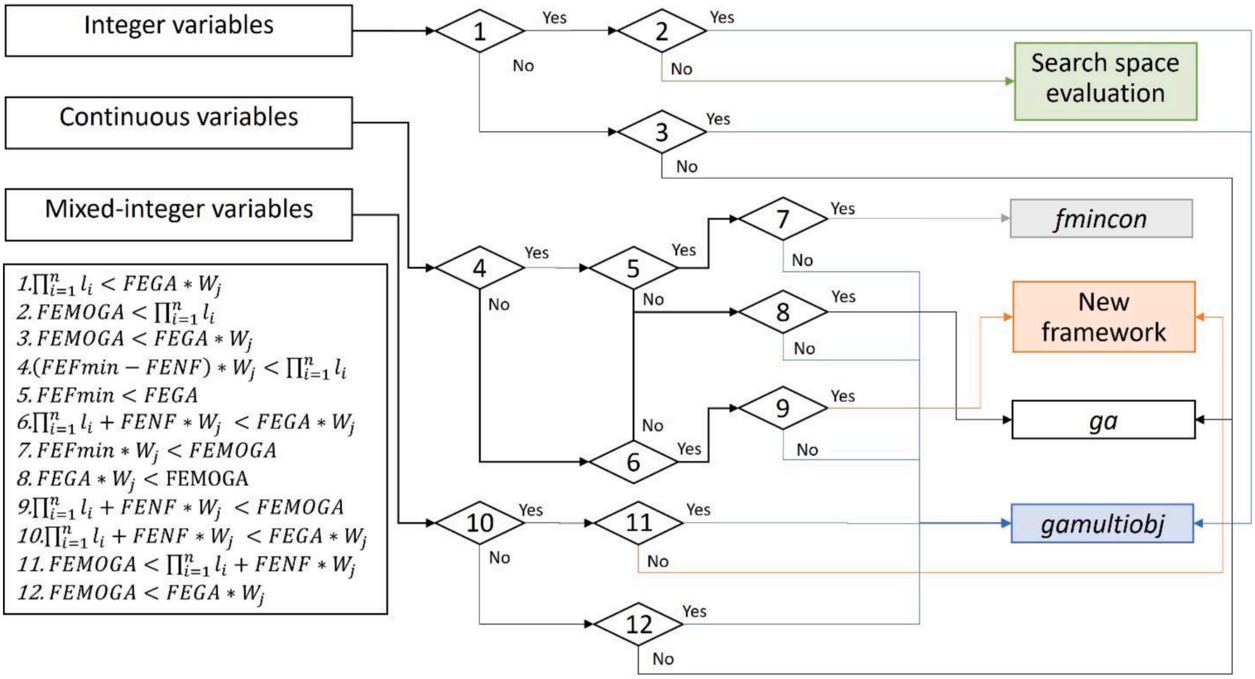
Integer variables
Continuous variables
Mixed-integer variables
1
2
3
4
5
6
7
8
9
10
11
12
Yes
No
Search space evaluation
fmincon
New framework
ga
gamultiobj
$1. \prod_{i=1}^{n} l_i < FEGA * W_j$
$2. FEMOGA < \prod_{i=1}^{n} l_i$
$3. FEMOGA < FEGA * W_j$
$4. (FEFmin - FENF) * W_j < \prod_{i=1}^{n} l_i$
$5. FEFmin < FEGA$
$6. \prod_{i=1}^{n} l_i + FENF * W_j < FEGA * W_j$
$7. FEFmin * W_j < FEMOGA$
$8. FEGA * W_j < \mathrm{FEMOGA}$
$9. \prod_{i=1}^{n} l_i + FENF * W_j < FEMOGA$
$10. \prod_{i=1}^{n} l_i + FENF * W_j < FEGA * W_j$
$11. FEMOGA < \prod_{i=1}^{n} l_i + FENF * W_j$
$12. FEMOGA < FEGA * W_j$

Fig. 5. Guideline for the selection of an optimization method.

framework in decision point 6. Finally, in decision point 8, the relatively large number of weightings favors *gamultiobj* over algorithms that use the sum of weighted objectives for single objective optimization. In those cases, the algorithms solve the optimization problem for every weight configuration, which increases the number of function evaluations.

In our case, the guidelines lead to the same algorithms that yielded the lowest number of function evaluations as in Table 2. However, the selection of the best method depends on the ability to obtain a representative number of function evaluation values for every optimization algorithm, which requires the solution of the problem at different conditions of weighting factors and algorithm parameters. Such a process can be challenging and time-consuming depending on the susceptibility of the algorithm to changes in those conditions. Once representative function evaluations are estimated, the guidelines can be used to select the algorithm that yields the lowest number of function evaluations for the problem.

4. Conclusions

This work introduces a new, efficient computational framework to evaluate process design alternatives. Surrogate representations and readily available software tools are used to make the framework robust, easy to implement, and computationally inexpensive.

In our simulation scenario, the surrogate function yielded results with the required level of accuracy for the application. The implementation of this function reduced computing time by 93%. We formulated a bi-objective optimization problem with different types of variables. In the integer variable problem, we observed that SSE outperforms genetic algorithms due to the relatively small size of the search space. For continuous-variable problems, the *gamultiobj* is recommended for Pareto front calculations. Alternately, if the decision maker has preferred values for the objective weight factors, the new optimization framework may be more efficient. Finally, we observed high efficiency with the new framework in mixed-integer problems.

The developed tools can be applied broadly to other chromatography problems. Since the surrogate function can use experimental or simulated data, the framework is suitable for situations where the available models cannot represent the system, experimental data can be easily obtained, or there are no resources for the simulation of dynamic systems. The new optimization framework also can be adapted to other process design problems. Nevertheless, the selection of this method over other optimization algorithms will depend on the characteristics of the problem. The proposed guidelines for comparing optimization methods can be used for this task. Despite the need to solve the optimization problem to obtain representative values for number of function evaluations, the guidelines are useful to explore the impact of different problem characteristics on the optimization performance. Overall, the implemented strategies proved to be effective in reducing computing time for our problem and can be extended to other process design applications requiring an efficient and easy to implement framework for simulation and optimization.

CRediT authorship contribution statement

Juan J. Romero: Conceptualization, Funding acquisition, Methodology, Project administration, Writing – review & editing. **Eleanor W. Jenkins:** Conceptualization, Funding acquisition, Investigation, Project administration, Supervision, Writing – original draft, Writing – review & editing. **Scott M. Husson:** Formal analysis, Funding acquisition, Investigation, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

Scott Husson has an ongoing financial interest in Purilogics and provides consulting services to the Company. Juan Romero and Eleanor Jenkins declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The source code for the simulation and the optimization framework and the data used to generate the surrogate function are available at https://github.com/juanjoromeroc/Supplementary-material.

Acknowledgements

This work was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number R15 GM131341 and the National Science Foundation under award DMS-2011902. S.M.H. acknowledges support from the William B. “Bill” Sturgis, ‘57 and Martha Elizabeth “Martha Beth” Blackmon Sturgis Distinguished Professorship in Chemical and Biomolecular Engineering.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2023.108225.

References

Bansode, V., Gupta, P., Kateja, N., Rathore, A.S., 2022. Contribution of protein A step towards cost of goods for continuous production of monoclonal antibody therapeutics. J. Chem. Technol. Biotechnol. 97, 2420–2433. https://doi.org/10.1002/jctb.6686.

Baur, D., Angarita, M., Müller-Späth, T., Steinebach, F., Morbidelli, M., 2016. Comparison of batch and continuous multi-column protein A capture processes by optimal design. Biotechnol. J. 11, 920–931. https://doi.org/10.1002/biot.201500481.

Behere, K., Yoon, S., 2020. Chromatography bioseparation technologies and in-silico modelings for continuous production of biotherapeutics. J. Chromatogr. A 1627, 461376. https://doi.org/10.1016/j.chroma.2020.461376.

Ding, C., Ardeshna, H., Gillespie, C., Ierapetritou, M., 2022. Process design of a fully integrated continuous biopharmaceutical process using economic and ecological impact assessment. Biotechnol. Bioeng. 119, 3567–3583. https://doi.org/10.1002/bit.28234.

Gomis-Fons, J., Yamanee-Nolin, M., Andersson, N., Nilsson, B., 2021. Optimal loading flow rate trajectory in monoclonal antibody capture chromatography. J. Chromatogr. A 1635, 461760. https://doi.org/10.1016/j.chroma.2020.461760.

Grilo, A.L., Mateus, M., Aires-Barros, M.R., Azevedo, A.M., 2017. Monoclonal antibodies production platforms: an opportunity study of a non-protein-A chromatographic platform based on process economics. Biotechnol. J. 12, 1–10. https://doi.org/10.1002/biot.201700260.

Hecht, F., 2012. New development in freefem++. J. Numer. Math. 20, 251–266. https://doi.org/10.1515/JNUM-2012-0013.

Hernandez, I., Bott, S.W., Patel, A.S., Wolf, C.G., Hospodar, A.R., Sampathkumar, S., Shrank, W.H., 2018. Pricing of monoclonal antibody therapies: higher if used for cancer? Am. J. Manag. Care 24, 109–112.

Hummel, J., Pagkaliwangan, M., Gjoka, X., Davidovits, T., Stock, R., Ransohoff, T., Gantier, R., Schofield, M., 2019. Modeling the downstream processing of monoclonal antibodies reveals cost advantages for continuous methods for a broad range of manufacturing scales. Biotechnol. J. 14 https://doi.org/10.1002/biot.201700665.

Intelligen, Inc., 2020. SuperPro Designer Product Features [WWW Document]. Intelligen, Inc. URL https://www.intelligen.com/products/superpro-product-features/ (accessed 1.11.22).

Jones, W., Gerogiorgis, D.I., 2022. Dynamic simulation, optimisation and economic analysis of fed-batch vs. perfusion bioreactors for advanced mAb manufacturing. Comput. Chem. Eng. 165 https://doi.org/10.1016/j.compchemeng.2022.107855.

Kim, S.H., Boukouvala, F., 2020. Surrogate-based optimization for mixed-integer nonlinear problems. Comput. Chem. Eng. 140, 106847 https://doi.org/10.1016/j.compchemeng.2020.106847.

Nikita, S., Tiwari, A., Sonawat, D., Kodamana, H., Rathore, A.S., 2021. Reinforcement learning based optimization of process chromatography for continuous processing of biopharmaceuticals. Chem. Eng. Sci. 230, 116171 https://doi.org/10.1016/j.ces.2020.116171.

Pollock, J., Coffman, J., Ho, S.V., Farid, S.S., 2017. Integrated continuous bioprocessing: economic, operational, and environmental feasibility for clinical and commercial antibody manufacture. Biotechnol. Prog. 33, 854–866. https://doi.org/10.1002/btpr.2492.

Romero, J.J., Jenkins, E.W., Osuofa, J., Husson, S.M., 2022. Computational framework for the techno-economic analysis of monoclonal antibody capture chromatography platforms. J. Chromatogr. A 1689, 463755. https://doi.org/10.1016/j.chroma.2022.463755.

Shekhawat, L.K., Rathore, A.S., 2019. Preparative Biochemistry and Biotechnology An overview of mechanistic modeling of liquid chromatography. Prep. Biochem. Biotechnol. 40, 623–638. https://doi.org/10.1080/10826068.2019.1615504.

Shi, C., Gao, Z.Y., Zhang, Q.L., Yao, S.J., Slater, N.K.H., Lin, D.Q., 2020. Model-based process development of continuous chromatography for antibody capture: a case study with twin-column system. J. Chromatogr. A 1619, 460936. https://doi.org/10.1016/j.chroma.2020.460936.

Taylor, P.C., Adams, A.C., Hufford, M.M., de la Torre, I., Winthrop, K., Gottlieb, R.L., 2021. Neutralizing monoclonal antibodies for treatment of COVID-19. Nat. Rev. Immunol. https://doi.org/10.1038/s41577-021-00542-x.

The MathWorks Inc., 2022. 1-D Data Interpolation (Table Lookup) - MATLAB Interp1 [WWW Document]. The MathWorks Inc. URL https://www.mathworks.com/help/matlab/ref/interp1.html?s_tid=doc_ta (accessed 6.22.22).

The MathWorks Inc., 2022. Global Optimization Toolbox - MATLAB [WWW Document]. The MathWorks Inc. URL https://www.mathworks.com/products/global-optimization.html (accessed 8.2.22).

The MathWorks Inc., 2022. What Is the Genetic Algorithm? - MATLAB & Simulink [WWW Document]. The MathWorks Inc. URL https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html (accessed 12.17.22).

von Lieres, E., Andersson, J., 2010. A fast and accurate solver for the general rate model of column liquid chromatography. Comput. Chem. Eng. 34, 1180–1191. https://doi.org/10.1016/j.compchemeng.2010.03.008.

Wilson, A.B., Jenkins, E.W., Wang, J., Husson, S.M., 2020. Numerical simulation of chemical separations using multimodal adsorption isotherms. Results Appl. Math. 7, 100122 https://doi.org/10.1016/j.rinam.2020.100122.

Yang, O., Qadan, M., Ierapetritou, M., 2020. Economic analysis of batch and continuous biopharmaceutical antibody production: a review. J. Pharm. Innov. 15, 182–200. https://doi.org/10.1007/s12247-018-09370-4.