Development of a computational tool for the solution of optimal control problems with metaheuristic techniques

Eduardo Herrera-Álvarez a, Oscar Daniel Lara-Montaño b, Mayra Margarita May-Vázquez c, Fernando Israel Gómez-Castro a,\*, Claudia Gutiérrez-Antonio b, Manuel Toledano-Ayala b

a Departamento de Ingeniería Química, División de Ciencias Naturales y Exactas, Campus Guanajuato, Universidad de Guanajuato, Noria Alta S/N, Guanajuato, Guanajuato, 36050, México. [fgomez@ugto.mx](mailto:fgomez@ugto.mx)

b Facultad de Ingeniería, Universidad Autónoma de Querétaro, Cerro de las Campanas S/N, Querétaro, Querétaro, 76010, México.

c Unidad Académica de Educación Virtual, Dirección General de Desarrollo Académico, Universidad Autónoma de Yucatán, Benito Juárez 421 Cd. Industrial, Mérida, Yucatán, 97288, México.

Abstract

This study explores the solution of a dynamic optimization problem with a focus on a Continuous Stirred-Tank Reactor (CSTR) known for its output multiplicities. The dynamic optimization problem is discretized into a non-linear programming (NLP) problem. The effectiveness of three metaheuristic algorithms—Differential Evolution (DE), Grey Wolf Optimizer (GWO), and Cuckoo Search (CS)— in solving this NLP problem is evaluated, considering both solution quality and computational efficiency. The findings provide insights into applying metaheuristic methods in dynamic optimization, highlighting the importance of discretization in transforming dynamic problems into manageable NLP tasks.

**Keywords**: dynamic optimization, optimal control, metaheuristic algorithms, continuous stirred-tank reactor

* 1. Introduction

Dynamic optimization is a key procedure widely used in various engineering and scientific fields to optimize the behavior of systems that predominantly depend on time or space. Typically, the goal is to optimize these systems based on a performance index, which is either maximized or minimized to achieve optimal outcomes. In the realm of chemical engineering, dynamic systems play a crucial role. Examples of such systems in this field include the determination of kinetic constants from time-series data, the control of batch and semi-batch chemical reactors, the start-up and shut-down processes of continuous systems, and the switching between different steady-state operating points. Differential-algebraic mathematical models are employed to represent and analyze these complex systems accurately. These models provide a robust framework for understanding and optimizing the dynamic behavior of various chemical engineering processes, mostly focusing on the analysis of the state variables, implying optimal control problems.

Traditional approaches to solve dynamic optimization problems, such as the calculus of variations, the Pontryagin's Maximum Principle, and the dynamic programming, each have their own drawbacks (Diwekar, 2008). The calculus of variations becomes complex when dealing with intricate constraints. Pontryagin's Maximum Principle often struggles with non-linear systems. On the other hand, although versatile and robust, dynamic programming becomes less effective and more computationally demanding when the dynamic optimization problem increases in dimensions; furthermore, these classic techniques necessitate a comprehensive system model, which may not be practical for extensive or complex problems, particularly those with uncertain variables. These challenges underscore the importance of choosing the most suitable method for each unique dynamic optimization scenario. Another alternative to address dynamic optimization problems is discretizing and solving the problem numerically as a non-linear programming (NLP) problem. An example of this approach has been reported for a fermentation system by Sridhar and Lopez Saucedo (2015). However, to solve relatively complex models by deterministic approaches, it is mandatory to determine adequate values for the variables at the initial time (May-Vázquez et al., 2022). Also, determining proper initial guesses for the trajectory of the variables could be difficult.

This research examines the use of various metaheuristic optimization algorithms in a case study of a Continuous Stirred-Tank Reactor (CSTR) exhibiting output multiplicities. The case study, initially reported by Hicks & Ray (1971) and later modified by Flores-Tlacuahuac et al. (2008), serves as the basis for this exploration. The study begins by discretizing the dynamic optimization problem, converting it into a Nonlinear Programming (NLP) problem. Then, the NLP problem is solved using metaheuristic algorithms. The use of metaheuristic optimization algorithms can be a suitable strategy to solve optimal control problems, mainly due to the non-linear behavior that the model can present, and the capacity of the metaheuristics to perform a global search and identify the feasible sub-space that contains the best solution. Three distinct optimization algorithms—differential evolution (DE), grey wolf optimizer (GWO), and cuckoo search (CS)—are then applied to solve the resulting NLP. The effectiveness of these algorithms is compared based on the number of iterations and the variety of candidate solutions generated. A dynamic optimization solution tool is also provided, enabling users to apply these methodologies to different case studies.

The document's structure is organized in the following manner: Section 2 outlines the methodology used in developing the tool. Section 3 presents and discusses the results of this specific case study, including an analysis of the tool's scope and limitations. The document concludes with Section 4, which summarizes the findings and conclusions of the work.

* 1. Methodology
     1. Case study

The case study under consideration involves a CSTR exhibiting output multiplicities, as detailed in the work of Flores-Tlacuahuac et al. (2008). This model operates within a highly non-linear region. For enhanced analysis and to facilitate the emergence of multiple steady states, the model is formulated in a dimensionless form, as shown in equations 1 and 2.

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |

In this model, denotes the dimensionless concentration, expressed as , while is the dimensionless temperature, represented as . The term corresponds to the dimensionless feed temperature, defined as , and represents the cooling flow rate. Table 1 shows the values employed in the parameters used by the model. The measured variable is , and the manipulated variable is .

Table 1. Parameters required by the model.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter |  |  |  |  |  |  |  |  |
| Value | 20 min | 100 K-L/mol | 7.6 mol/L | 1.95x10-4 l/L | 300 K | 300 l/min | 290 K | 5 |

Table 2 presents four distinct nominal steady states applicable to our case study. These states represent different operational conditions, and the system can transition between these steady states as required.

Table 2. Nominal steady states for the case study.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
|  | 0.0944 | 0.1367 | 0.1926 | 0.2632 |
|  | 0.7766 | 0.7293 | 0.6881 | 0.6529 |
|  | 340 | 390 | 430 | 455 |

* + 1. Optimal control problem

The optimal control problem is shown in equations 3 to 6.

|  |  |
| --- | --- |
|  | (3) |
|  | (4) |
|  | (5) |
|  | (6) |

In this optimal control problem, the objective is to minimize , representing the squared deviation of the state vector from the target over time to . The system's evolution is governed by , with as the state variables, as the control variables, and defining the initial state. In this case study, such relationships are given by equations (1) and (2). Constraints and limit the state and control variables within feasible ranges.

This dynamic optimization problem is solved using a Python-based tool. The process involves discretizing the time variable, which is independent, into equal intervals. To minimize the integral, its value is computed numerically using the 1/3 Simpson's method. The optimization of the integral is carried out using metaheuristic algorithms from the Mealpy library. The code for solving this dynamic optimization problem, after converting it into a Nonlinear Programming (NLP) problem, can be accessed at https://github.com/DanlaraIQ/DynamicOpt.

The explicit discretized objective function is shown in equations 7-9. is the discretized time, 20 intervals are used for its calculation. , and are weights for each squared term.

|  |  |
| --- | --- |
|  | (7) |
|  | (8) |
|  | (9) |

The parameters of the metaheuristic optimization algorithms are given in Table 3. 50 candidate solutions and 500 iterations are employed. The lower and upper limits used for the control variable are 200 and 550, respectively.

Table 3. Parameters used in the metaheuristic algorithms.

|  |  |  |
| --- | --- | --- |
| CS | DE | GWO |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Results

Table 4 displays the results for the objective function's value and the CPU time taken for each optimization algorithm across various steady-state transitions. This table highlights the best performances regarding the objective function's quality and CPU time efficiency in bold. These results show that the CS algorithm is not the best choice for solving this optimization problem. This is not only due to its longer computational time for completing 500 iterations, which might be attributed to the inherent structure of the code but also because of its inability to effectively minimize the value of the objective function through the optimal selection of the vector of values.

Table 4. Comparative results for the objective function value/time (seconds) in each transition between steady states.

|  |  |  |  |
| --- | --- | --- | --- |
| transition | CS | DE | GWO |
| A🡪B | 2,141.8058/13.58 | 2,051.2866/**12.05** | **2,040.9517**/12.54 |
| B🡪A | 2,151.7355/14.01 | 1,957.2324/**12.43** | **1,951.9298**/12.56 |
| A🡪C | 18,017.8751/14.43 | 15,337.9467/13.11 | **15,315.1204**/**12.83** |
| C🡪A | 13,331.4591/14.06 | 12,814.7678/**13.10** | **12,718.0579**/14.79 |
| A🡪D | 67,931.6923/15.07 | **58,127.4895**/**13.39** | 58,140.7994/14.52 |
| D🡪A | 44,602.0470/14.09 | **42,802.0694**/**12.66** | 42,811.31/13.43 |
| B🡪C | 4,707.0856/14.03 | 4,268.6486/13.38 | **4,257.5752**/**13.57** |
| C🡪B | 3,775.9691/13.70 | **3,506.7442**/13.30 | 3,507.2127/**12.84** |
| B🡪D | 36,388.9497/13.75 | 30,688.6344/**13.16** | **30,484.4022**/13.35 |
| D🡪B | 24,548.2092/13.39 | **22,476.6771**/**12.01** | 22,498.8302/13.08 |
| C🡪D | 9,542.4709/13.81 | 8,156.1741/**13.07** | **8,127.4643**/13.38 |
| D🡪C | 7,309.2156/13.80 | **6,144.5463**/**13.17** | 6,151.5801/18.59 |

The efficiency in execution time is generally comparable across the three optimization algorithms. However, GWO shows some variability in certain instances, notably in the transition from D to C. Regarding the best solution's quality, the DE and GWO algorithms demonstrate very similar performances. With comparison purposes, the problem has been discretized using the orthogonal collocation on finite elements approach and codified in the GAMS Studio environment. By solving for the first transition using the deterministic IPOPT solver, a value of 2,677.63 is obtained for the objective function. This value is comparable to those obtained with the metaheuristic methods, although slightly higher. This may reflect the existence of various local optima. On the other hand, the deterministic solution is obtained in 0.227 seconds with the deterministic approach, which is a considerably lower computing time.

Figure 1 presents the plotted trajectories of and during the transitions from state B to A and from D to C. It is observable that the GWO and DE algorithms produce comparable trajectories for the B to A transition, with minor differences observed in the transition from C to D. The plots visually demonstrate the CS algorithm's inability to identify satisfactory solutions within the given number of iterations for this optimization problem.

|  |  |  |
| --- | --- | --- |
| Transition from B to A. | | |
|  |  |  |
| Transition from C to D. | | |
|  |  |  |

Figure 1. The behavior of the transitions between steady-state B🡪A and C🡪D.

A notable challenge associated with the strategy involving metaheuristic optimization algorithms is the exponential increase in running time as the number of dimensions grows. Specifically, in this optimization problem, the dimensionality is directly tied to the discretization of the independent variable, which, in this context, is time. This aspect becomes particularly critical in scenarios with broader time frames. In such cases, a larger number of discrete time points is required to represent the system's trajectory accurately. Consequently, as the time frame expands, the increased need for finer discretization leads to a higher dimensional space, thereby significantly impacting the computational time.

* 1. Conclusion

The research indicates that DE and GWO outperform CS in addressing the discretized dynamic optimization problem, demonstrating superior solution quality and time efficiency performance. The strategy of discretizing the dynamic optimization into an NLP problem is critical for effectively applying these metaheuristic algorithms. The limitations of the CS algorithm are noted within the iteration constraints of the NLP context. The developed computational tool, capable of adapting these algorithms, proves versatile for various dynamic optimization scenarios. The study underscores the significance of a well-chosen strategy in discretized dynamic optimization and opens new avenues for future research in this field. It is also noted that an increase in problem complexity leads to an exponential growth in the computational time required for these metaheuristic algorithms.

References

U. Diwekar, 2008, Introduction to applied optimization, Springer Science & Business Media.

A. Flores-Tlacuahuac, S.T. Moreno, L. Biegler, 2008. Global Optimization of Highly Nonlinear Dynamic Systems, Industrial & Engineering Chemistry Research, 47, 2643–2655.

G.A. Hicks, W.H. Ray, 1971, Approximation methods for optimal control synthesis, The Canadian Journal of Chemical Engineering, 49, 522–528.

M.M. May-Vázquez, F.I. Gómez-Castro, E.S. Rawlings, V. Rico-Ramírez, M.A. Rodríguez-Ángeles, 2022, Optimal control of a rate-based modelled batch distillation column: Initialization strategy, Computers & Chemical Engineering, 162, 107811.

L.N. Sridhar, E.S. López Saucedo, 2015, Optimal control of Saccharomyces cerevisiae. Fermentation process, Chemical Engineering Communications, 203, 318-325