Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-7 (2016) 097–102

Multi-Objective Optimal Control Study of Fed-Batch Bio-Reactor ⋆

Narendra Patel * **Nitin Padhiyar** **

* *Chemical Engineering Department, Vishwakarma Government Engineering College, Chandkheda, Ahmedabad-382424, Gujarat, India. (e-mail: narendra@iitgn.ac.in)*

** *Department of Chemical Engineering, Indian Institute of Technology Gandhinagar, Ahmedabad-382424, Gujarat, India. (e-mail: nitin@iitgn.ac.in)*

Abstract: Evolutionary Algorithms have been successfully applied for offline optimal control problems of fed-batch bio-reactors. In such problems, productivity-yield maximization is carried out by optimizing the transient feed recipe. However, this is usually done for a fixed fed-batch time. The optimum batch time can be computed by solving single objective optimal control problems multiple times with different fed-batch times. Since this approach is quite computationally expensive, we in this work formulate a multi-objective optimization(MOO) problem to find the minimum fed-batch time along with maximizing productivity-yield. Such an MOO approach will result in saving significant computational effort. A single parameter based fast mesh sorting with multi objective differential evolution is used in this work for solving MOO problems. We have considered a case study of optimal control of fed-batch reactor for secreted protein production with volume constraint in this work.

Keywords: Attainable Region, Multi-objective Optimization, mesh sort, fed-batch reactor, optimal control

1. INTRODUCTION

Dynamic optimization (DO) is a tool for obtaining optimal operating conditions that maximizes productivity in a fed-batch bio-reactor, thus reducing experimental costs (Pham, 1998). DO has been extensively applied to fed-batch bio-reactor operations for optimal control problems. DO computes the optimum feed recipe such that nutrients are maintained into the bio-reactor to grow or synthesize the desired metabolite. DO guarantees both an optimal cell growth and a metabolite bio-synthesis, avoiding under and overfeeding of the substrate. Researchers have used methods like two-point collocation, Iterative Dynamic Programming (IDP) (Luus, 1994), Relaxed reduced space SQP strategy, IDP with absolute error penalty functions along with evolutionary approaches like Genetic Algorithm (GA), Iterative Ant-Colony Algorithm (Zhang et al., 2005), Particle Swam Optimization(PSO), Hybrid improved GA (Sun et al., 2013), other meta heuristics and hybrid methods like Box-Complex GA (Patel and Padhiyar, 2015b) for obtaining optimum feed trajectories to achieve the defined objective. Yield of the product is another important parameter apart from the productivity in bio-reactors. Both, the yield and the productivity are conflicting and hence form a MOO problem. The solution of this MOO problem can be obtained in the form of a pareto front. Very few articles are dedicated for the multi-objective optimal control (Maiti et al., 2011), (Sarkar and Modak, 2005), and (Patel and Padhiyar, 2015a).

Usually optimal control problems for fed-batch reactors are solved by maximizing productivity and/or yield for fixed fed-batch times. The optimum fed-batch time can be computed by solving numerous single objective optimal control problems at different fed-batch time (Luus (1994) and Lopez et al. (2010)). However, such an approach is computationally very expensive. This issue can efficiently be addressed by formulating and solving a multi-objective optimization (MOO) problem.

We in this work solve four multi-objective optimal control problems of fed-batch bio-reactor for a popular case study of secreted protein production by a yeast strain (Park and Fred Ramirez, 1988). The four MOO problems considered in this work are: (1) maximizing the productivity and yield, (2) maximizing the productivity and minimizing fed-batch time, (3) maximizing the yield and minimizing fed-batch time, and (4) maximizing the productivity and yield while minimizing fed-batch time. The multi objective optimal control problems have been solved using control vector parameterizaiton (CVP) with multi-objective differential evolution (MODE) algorithm. Further, the MODE is used along with recently proposed mesh sort algorithm (Patel and Padhiyar, 2015a) in this work.

The MODE algorithm with mesh-sort is presented in section 2. The multi-objective optimal control problems for fed-batch bio-reactor have been discussed in section 3. The fed-batch process model for secreted protein production by a yeast strain is presented in section 4. Results for this case study are discussed in Section 5. Finally, the concluding remarks are made in Section 6.

⋆ IIT Gandhinagar, Ahmedabad, Gujarat, India

10.1016/j.ifacol.2016.07.223

2. MULTI-OBJECTIVE DIFFERENTIAL EVOLUTION (MODE) WITH MESH SORT

Multi-objective optimization (MOO) problems with conflicting objectives will have a set of solutions, which are called pareto optimal solutions. Evolutionary algorithms have gained significant attention for solving MOO problems in past two decades. Non-dominated sorting, rank based sorting and evolution with decomposition are currently evolving approaches for MOO. The dominance based ranking of populations requires multiple comparisons of members for sorting and hence are computationally expensive. There are some computationally more efficient non-dominated sorting algorithms (Zhang et al., 2015). Recently, a novel mesh based sorting for genetic algorithm (Patel and Padhiyar, 2015a) has been presented, which is found to be computationally more efficient than non-dominated sorting.

We in this work use a modified mesh based sorting mechanism for better computational efficiency. Instead of classifying the population members in various ranked pareto fronts, in the mesh-sort approach, the population is divided into an m-dimensional mesh, where m is the number of objectives. Here, the location of each population member in the mesh determines the quality of the population member. A fitness value for each population member is assigned depending upon the location of it in the m-dimensional mesh. Further, various fitness criteria for good pareto solutions such as uniformity and the coverage of total pareto span are also accounted for in the fitness value of every population member. This single fitness value of each member is used in survival selection step in the MOO algorithm. Though, this sorting is applicable to any population based evolutionary algorithm, we apply it with multi-objective differential evolution (MODE).

2.1 Multi-Objective Differential Evolution with Mesh Sort

DE is a population based stochastic optimization technique, which can provide potentially a global optimum solution. At every generation of DE, mutation and crossover steps are performed to add diversity in the population members and survival selection is applied for eliminating the inferior ones. A step wise implementation of the proposed mesh sort based MODE for minimization of m objectives has been summarized as follows,

(1) *Initialize population of size Np.*
(2) *Calculate all the m objective function values for each population member.*
(3) *Carry out DE crossover and mutation operations to produce Np children.*
(4) *Calculate all the m objective function values for each member of the child population.*
(5) *Elitism selection: Sort the $2Np$ members (Np parents and Np children) using mesh sort and select the best Np members.*
(6) *This completes one generation. Stop if convergence criteria has been met; else go to step (3).*

Crossover in DE for each element $\overline{y}_k$ in $\overline{y} = (\overline{y}_1, \overline{y}_2, ..., \overline{y}_n)^T$ is done as,

$$\overline{y}_k = \begin{cases} x_k^{r_1} + F \times (x_k^{r_2} - x_k^{r_3}) & \text{with probability}\, CR \\ x_k^{r_1} & \text{with probability}\, 1 - CR \end{cases} \quad (1)$$

where, CR and F are two control parameter for DE crossover and r_1, r_2 and r_3 are randomly selected different parents. The mutation operation for DE to generate $y = (y_1, y_2, ..., y_n)^T$ from $\overline{y}$ is done as,

$$y_k = \begin{cases} \overline{y}_k + \sigma_k \times (b_k - a_k) & \text{with probability}\, p_m \\ \overline{y}_k & \text{with probability}\, 1 - p_m \end{cases} \quad (2)$$

$$\text{with, } \sigma_k = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}} - 1 & \text{if}\, rand < 0.5 \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases} \quad (3)$$

where, $rand \in [0, 1]$ and a_k and b_k are lower and upper bounds of the k-th variable. Distribution index η and mutation probability p_m are two control parameters for mutation.

Different components of mesh-weight and stepwise algorithm for mesh-weight assignment to the population members are discussed in the next subsections.

2.2 Mesh weight components

In the survival stage at every generation in population based evolutionary methods best Np members are selected out of $2Np$ members. Sorting for this purpose is proposed to be carried out using mesh concept in this work. This sorting is also useful in the selection for reproduction. As Np members are to be selected from $2Np$, it is a natural choice to divide every objective space in Np sections. Different weights are assigned to each population member to account for convergence and uniform distribution of pareto optimal solutions.

For two objectives ($m = 2$) and a population size of ten ($Np = 10$), a mesh of size 10×10 is created as shown in Fig. (1). Here, cell dimensions for the ith objective are calculated by maximum ($f_{i,max}$) and minimum ($f_{i,min}$) value of f_i as follows,

$$\Delta f_i = \frac{(f_{i,max} - f_{i,min})}{Np} \quad (4)$$

Note that this cell size is dynamically changing generation wise depending upon the maximum and minimum values of ith objective function. A rank in descending order from 10 (best) to 1 (worst) to all the grids along each objective dimension is assigned as shown in Fig. (1). To direct the evolution of the population toward the solution and obtain the uniformity in the pareto solutions, different weights are assigned to each population member based on its location in the mesh. The proposed mesh sorting algorithm uses five components contributing to total weight for each member: (1) Macro mesh weight (2) Micro mesh weight (3) Best weight (4) Strip weight and (5) Neighbour weight. The population is sorted based on the sum total of all these five components. The detail of all the five weight components is discussed below,

- **Macro mesh weight (mw):** Macro weight is assigned to each member depending upon its location in mesh. It is the sum total of all the ranks along m dimensions. Macro weight value for each cell is shown in the corresponding cell in Fig. (1). The values range from 2 to 20 as we move from top right corner (worst cell) to left bottom corner (best cell). Macro mesh weight values are identical for all the off diagonal cells as shown in Fig. (1).