Korean J. Chem. Eng., 38(1), 8-21 (2021)
DOI: 10.1007/s11814-020-0642-y

pISSN: 0256-1115
eISSN: 1975-7220

Hybridized multi-objective optimization approach (HMODE) for lysine fed-batch fermentation process

Zainab Al Ani, Ashish Madhukar Gujarathi†, and Gholamreza Vakili-Nezhaad

Department of Petroleum & Chemical Engineering, College of Engineering, Sultan Qaboos University, P. O. Box 33, Al-Khod, P.C. 123, Muscat, Sultanate of Oman

(*Received 2 February 2020 • Revised 17 July 2020 • Accepted 19 July 2020*)

Abstract−A new hybrid multi-objective differential evolution (MODE) algorithm is proposed that combines the MODE algorithm for the global space search with a dynamical local search (DLS) method for the local space search. HMODE-DLS algorithm was validated using the tri-objective DTLZ7 test problem and the results were compared with MODE algorithm with respect to four performance metrics. In addition to HMODE-DLS, another three algorithms were used to solve two multi-objective optimization cases in an industrial lysine bioreactor at different feeding conditions. Case 1 considers maximizing lysine's productivity and yield. While case 2 studies the maximization of productivity along with minimization of total operating time. In all cases, theoretical and industrial, HMODE-DLS showed a better performance with a better quality Pareto set of solutions. The Pareto front of case 1 found by HMODE-DLS was compared with a recent study trade-off, and the current non-dominated solutions values were found to be improved. This indicates that the lysine production process is enhanced. For case 2, the switching time from fed-batch to batch was found to be the key decision variable. Generally, these findings indicate the effectiveness of HMODE-DLS for the studied cases and its potential in solving real world complex problems.

Keywords: Lysine, Multi-objective Optimization, Hybrid Algorithms, Fed-batch Bioreactor, Evolutionary Algorithms

INTRODUCTION

Lysine, one of the most important amino acids, plays a role in the human nutrition system, but the human body cannot produce it. Because of its essentiality, huge amounts of lysine are manufactured every year to be used in the production of cosmetics, medicines and polymers [1,2]. Lysine is also used as an additive to animal feed. Several processes are used to produce lysine such as fermentation, enzymatic conversion and other chemical processes. However, fermentation is the most economic one, but with a drawback of low extraction yields [2-4]. The fermentation process is the most economically feasible one because of the low pressure and temperature operating conditions as well as the low cost of the required carbon source [3,5,6]. Sugars like sucrose and glucose are the main feeding raw materials used for lysine production by fermentation [7]. Batch and fed batch reactors are used to carry out this biochemical reaction in repeated cycles [2].

Multi-objective evolutionary algorithms (MOEA's) have been successfully used to solve several complex single and multi-objective optimization (MOO) problems in various fields like food processing [8-11], chemical industries [12-15] and biochemical processes [16-19]. Some of the biochemical processes that have been studied for MOO are ethanol [19-21], lactic acid [22], and lysine [23-25].

In MOO studies related to lysine production, the considered objectives were the simultaneous maximization of productivity and yield. A fed-batch bioreactor for lysine fermentation with singular feeding was optimized by Sarkar and Modak [23]. Non-dominated sorting genetic algorithm, NSGA-II, was used to carry out the MOO study and the algorithm was able to obtain the desired Pareto front [23]. For the same fed-batch bioreactor, two MOO cases were studied using different yield objective functions and constraints. This study combined NBI and NNC with different control methods. Singular feeding was considered and the results were as accurate as those found by Sarkar and Modak with wider Pareto range of solutions [26]. A single multi-objective optimization work was done on a lysine production process by Taras and Woinaroschy [27], where five objectives (minimizing each of the production cost per unit, total capital cost, environmental index, and biomass concentration and maximize the concentration of lysine in the product), were combined in a single objective function. In addition, a trade-off between production cost per unit and lysine concentration was obtained. The model used to conduct this work is the one proposed by Heinzle et al. [27,28]. Al-Siyabi et al. [25] studied the same process with two MOEA's, which are multi-objective differential evolution algorithm-III, MODE III, and harmonic MODE, considering the same objectives. The study covered constant, changing and singular feeding profiles. Harmonic MODE was found to be giving better results with good diversity compared to MODE-III algorithm [25].

In the current work, MODE algorithm was combined with a local search method to create HMODE-DLS algorithm. This technique was implemented to improve the performance of MODE algorithm in terms of convergence, diversity and time taken to converge. To evaluate the proposed algorithm, it was tested with the tri-objective DTLZ7 test problem and its performance was compared with MODE algorithm. Comparison bases were four performance met-

†To whom correspondence should be addressed.
E-mail: ashishgujrathi@gmail.com, ashishg@squ.edu.om

rics (convergence (Conv), spread (SP), generation distance (GD) and spacing (SPC)). Furthermore, HMODE-DLS along with other three well-established algorithms (NSGA-II, MODE, and multi-objective particle swarm algorithm (MOPSO)) were used to solve two MOO cases in the lysine biochemical process with constant and changing feeding conditions and the performance of the considered algorithms was evaluated and compared. The considered objectives in this study were the maximization of yield and productivity in case 1, and the maximization of productivity and minimization of process time in case 2. In addition, the results of case 1 of the current work were compared with the previously published work with MODE-based algorithms [25].

1. Hybridization of Multi-Objective Differential Evolution (MODE)

MOEA's have their own limitations that hinder them from achieving the global non-dominated optimum group of desired solutions, Pareto front, in reliable time span [25]. Thus, though the popular MOEA's can ensure the convergence to the Pareto front, there exists a scope in improving the performance of these algorithms in terms of time taken for solutions to converge to the Pareto front. To overcome this disadvantage, MOEA's can be combined with other local search methods (hybridization), which helps in accelerating the searching process of the algorithm. Several attempts were made to hybridize MOEA's such as NSGA-II with sequential quadratic programming [29], improved NSGA-II [30], hybridization-encouraged mechanism-based NSGA-II [31], Hybrid-MODE [32], Improved-MODE [33]. Focusing on the proposed MODE-hybridization methods, Hybrid-MODE showed some deficiency regarding the diversity of solutions [32]. Improved-MODE is hybridized with taboo search method, which is much more complex compared to the DLS technique [33].

In addition to the hybridization techniques, MODE algorithm has been improved by implementing several strategies, which helped in attaining better performance in solving complex engineering MOO problems. Some of these improved MODE algorithm versions are Harmonic MODE [34], MODE-II and MODE-III [35], E-MODE [36], and other popular versions of MODE algorithms [37-41]. MODE-II and MODE-III were found to achieve better results compared to MODE algorithm, but with the drawback of additional computing time and MODE-II was found to face a problem of deviating from the general simple approach of EA [35]. When E-MODE algorithm was tested for solving a real MOO problem (purified terephthalic acid oxidation process), in some cases the algorithm did not give Pareto fronts as wide as those found by MODE algorithm [36]. Regarding Harmonic MODE, although it showed better performance compared to some other tested algorithms, since it is totally based on evolutionary computation, it is not as fast as other hybrid MODE algorithms [34]. While, MODE with ranking-based mutation operator (MODE-RMO) was tested with non-constrained chemical engineering problems only [40]. On the other hand, summation based MODE (SMODE) proved its effectiveness in solving power system MOO problems only [41]. To overcome the drawbacks found in the discussed algorithms, new hybridization technique is presented and investigated in this work. This hybridization attempt aims to increase the speed of convergence and the quality of the solutions in any MOO problem (constrained and non-constrained).

Simple MODE algorithm randomly generates an initial population with the desired size, which is then sent to the MODE generation loop where new strings are formed from the already existing population using the specified recombination operators (crossover (CR) and frequency factor (ω)). ω can be randomly generated based on the following [42]:

$$\omega = \omega_l + \text{rand} \times (\omega_u - \omega_l) \tag{1}$$

where the lower limit of ω is 0.8 (ω_l) and the upper limit is 1.2 (ω_u) and rand is a random number between 0 and 1. Generating the new strings can be done using two strategies:

$$\text{str1: } x_{MODE\_new,i} = x_{c,i} + \omega \times (x_{a,i} - x_{b,i}) \tag{2}$$

$$\text{str2: } x_{MODE\_new,i} = x_{e,i} + \omega \times (x_{a,i} - x_{b,i}) + \omega \times (x_{c,i} - x_{d,i}) \tag{3}$$

where $x_{a,i}$, $x_{b,i}$, $x_{c,i}$, $x_{d,i}$ and $x_{e,i}$ are points from the current generation. In str1, three parents are used to generate the new child (x_{MODE_new}), while four are used to achieve the same target in str2 [43].

After the objective functions of the newly developed points are evaluated, they are combined with the parents' points and the non-dominating solutions are selected to continue the searching mechanism in MODE algorithm. To exit the generation loop, a termination criterion must be reached, such as the maximum number of generations [44]. This searching technique reduces the number of non-dominated solutions with each generation as the new point replaces at least three parent points, and the algorithm may have the risk of stopping without converging and with no sufficient number of solutions. A more detailed description for MODE algorithm can be found in Babu and Anbarasu [44].

The existing hybridization work done with MODE algorithm indicates that there is a potential for another hybridization technique proposal, as the algorithm has the capability of combining it with several efficient, yet simple, local search methods. Dynamical local search (DLS) method can be implemented in MODE algorithm (HMODE-DLS), where the population is divided into two parts. The biggest population part is handled by the main MODE algorithm reproduction operation as described previously using Eq. (2) or (3), while the smaller one is sent for the local search method, DLS in this case [45,46], for reproduction according to the following:

for q=1: Q
for i=1: n
$x_{DLS\_new}(q, i) = \text{rand} \times ((1 - D_l) \times x_{o,i} + D_l \times x_{m,i})$ (4)
end
end

where n is the number of decision variables, x_{DLS_new} is the newly produced population point from DLS, while x_o and x_m are random points that are selected from the generation and D_l is a local dynamic scaling factor that can vary from 0 to 1. Q is the number of population that is sent for DLS. With the implementation of this technique, the searching mechanism will be faster and the algorithm will generate Pareto fronts with a greater number of points compared to MODE algorithm alone.

When MODE and DLS finish generating the new population points separately, the objective functions are evaluated for each new vector point. After that, all points are combined for the non-dominating sorting purpose and the Pareto front is formed, which will then go through the same procedure until meeting the termination criteria. The advantage of MODE hybridization with DLS (HMODE-DLS) attempt is to obtain a better quality and quantity Pareto set of solutions compared to those obtained using the previously reported MODE hybridization methods and strategies with less computation time. DLS will handle the search for the nearby optimal solutions and MODE algorithm will be responsible for searching the global optimum solutions. The advantages and capabilities of both searching mechanisms will enhance the overall performance as they increase the speed of convergence and the quality of the Pareto set of solutions.

For solving the industrial lysine MOO problem, NSGA-II and MOPSO are used in addition to MODE and HMODE-DLS algorithms. In NSGA-II, the working mechanism is simply initiated by generating an initial population with the decided size that is then sent to the generation loop. In the generation loop, mutation factor (M_f) and CR operations take place to form the new generation. The new and the old generations are then combined, objective functions are evaluated and the final generation points are decided after selection and non-dominated sorting. Similar to MODE, the searching ends when reaching the maximum number of generations or any other criteria selected by the decision maker [47]. In MOPSO, in addition to the initial generation, a speed for each particle is also initialized and the particle values are stored as personal best. Then, domination is determined, a grid is created and a grid index is found. After that, in the generation loop, a leader is selected from the external repository, the position and speed of the particle is updated and objective functions are evaluated. M_f is then used to calculate new solutions, domination is determined and personal best is updated. The non-dominated particles are added to the repository and a new repository is created after non-dominated sorting. Furthermore, the grid and grid index are updated and w is modified. And this generation loop keeps working until reaching the criteria needed to be achieved [48,49].

KINETIC MODEL FOR INDUSTRIAL LYSINE PRODUCTION USING FED-BATCH FERMENTER

In this work, the model presented by Ohno et al. [50] was used. In this model, it is assumed that there is a perfect mixing inside the reactor with no degradation [50]. The model is represented in the following equations and the schematic for fed-batch bioreactor is shown in Fig. 1.

$$\frac{dX}{dt} = \mu X \quad (5)$$

$$\frac{dS}{dt} = \sigma X + uC_{S,F} \quad (6)$$

$$\frac{dP}{dt} = \pi X \quad (7)$$

$$\frac{dV}{dt} = \mu \quad (8)$$

where X, S, P and V are the biomass mass, substrate mass, product (lysine) all in g, and reactor volume in L. The initial conditions of X(0), S(0), P(0) and V(0) are 0.1 g, 14 g, 0 g, and 5 L respectively. $C_{S,F}$ is the concentration of S in the feed (2.8 g/L), while u is the volume flow rate of feed in L/h. μ is the growth rate in 1/h, σ is the rate of S consumed in $\frac{g}{gh}$ and π is the rate of product formation in $\frac{g}{gh}$ and they are calculated by:

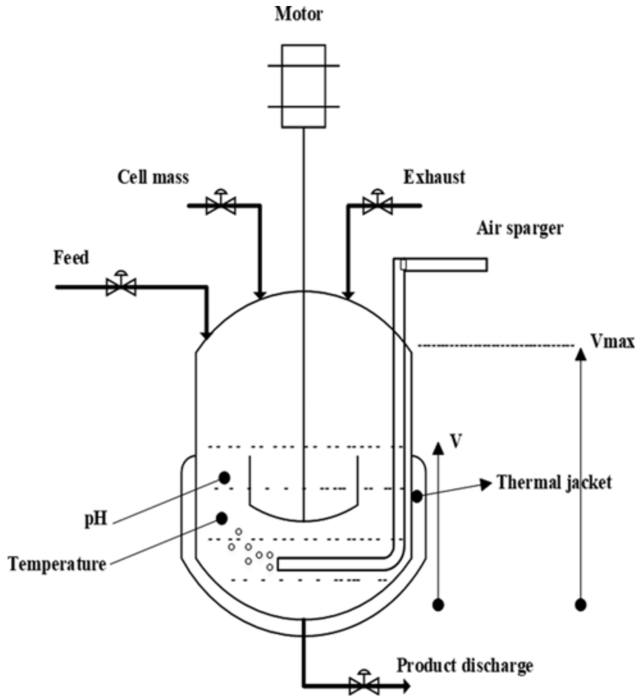

Fig. 1. Schematic of fed-batch lysine fermenter.

$$\mu = \mu_P \frac{S}{V} \quad (9)$$

$$\sigma = \frac{\mu}{\sigma_P} \quad (10)$$

$$\pi = -\pi_{P1}\mu^2 + \pi_{P2}\mu \quad (11)$$

where μ_P=0.125 g/L, σ_P=0.135 $\frac{g_S}{g_X h}$, π_{P1}=384 $\frac{g_P}{g_X h}$, and π_{P2}=134 $\frac{g_P}{g_X}$.

To solve lysine's kinetic mathematical model, ODE15s is used in MATLAB 2018b. Two MOO case studies were considered in this work. Four decision variables were selected in this study, which are two switching times for batch and fed-batch reactors (t_{s1}, t_{s2}), maximum operating time (t_f) and the initial volume of the reactor ($V(t_0)$).

PROBLEM FORMULATIONS FOR CASE STUDIES ON MULTI-OBJECTIVE OPTIMIZATION

Tri-objective DTLZ7 test function was considered to test the performance of HMODE-DLS algorithm with MODE algorithm. In addition, for further evaluation, both algorithms along with other widely used algorithms were used to solve lysine production MOO case study. HP Z6 G4 workstation with 2.19 GHz, total of 24 cores (2 x Intel Xeon 5120 14C CPU) and 48 GB RAM was used to perform the work. In the following, the methodology and algorithm's setting are elaborated.

1. Case Study 1: DTLZ 7 Test Problem

DTLZ 7 [51] test problem was solved with MODE and HMODE-DLS and results were compared. The comparison is based on calculated performance metrics. Thirty independent runs were conducted with MATLAB R2018b (for each algorithm) and the mean and standard deviation (std) of all performance metrics were calculated. Table 2 summarizes the objectives and decision variables of DTLZ7 test problem. The problem is solved for three objective functions (ob), which has four disconnected Pareto front regions with 22 decision variables.

2. Case Study 2: Industrial Lysine Fed-batch Fermenter Based Problem

In this work, two cases were considered for MOO:

- Case 1: Maximization of yield and productivity
- Case 2: Maximization of productivity and minimization of total operating time (t_f)

The objectives, decision variables and constraints of lysine fermentation study are summed up in Table 2. The ranges of these decision variables are specified according to literature [26,52]. For each case, the feeding flow rate (F) was changed from 0.6 to 2 g/h (with an increment of 0.2 g/h) to study its effect on the MOO results.

Three popular and widely used algorithms were considered to solve the MOO cases in this study for comparison. NSGA-II, MOPSO and MODE in addition to the proposed HMODE-DLS were used. The parameters used with NSGA-II (CR and mutation factor (M_f)), MOPSO (inertia weight (w), inertia damping rate (w_{Damp}), personal learning coefficient (c1) and global learning coefficient (c2), beta and gamma, and M_f), MODE and HMODE-DLS (CR and ω) are summarized in Table 1. These parameters were selected based on previous studies as they were found to give consistent results [25,53,54]. The parameters used in the algorithms' settings were fixed to the same and/or equivalent values for all the used algorithms for rational assessment with the same exact mathematical model.

For all algorithms used for solving the industrial lysine problem, runs were carried out with 250 population points and 300 genera-

Table 1. Parameters of the selected algorithms in lysine fermentation study

Algorithm	Parameters						
	M_f	CR	ω	w	w_{Damp}	c1 and c2	Beta and gamma
MODE	-	0.8	Eq. (1)	-	-	-	-
HMODE-DLS	-	0.8	Eq. (1)	-	-	-	-
NSGA-II	0.05	0.8	-	-	-	-	-
MOPSO	0.05	-	-	0.5	0.99	1	2

Table 2. MOO case studies, decision variables and constraints

Objective functions	Constraints	Decision variables
	Case Study 1	
Min $F1=x_1$ Min $F2=x_2$ .. .. Min $F_{ob-1}=x_{ob-1}$ Min $F_{ob}=(1+g(x_{ob}))h(F1, F2, \ldots, F_{ob-1}, g)$ where $g(x_3)=1+\frac{g}{\lvert x_{ob}\rvert}\Sigma_{x_i \in x_{ob}} x_i$ $h(F1, F2, \ldots, F_{ob-1}, g)=ob-\Sigma_{i=1}^{ob-1}\left[\frac{Fi}{1+g}(1+\sin(3\pi Fi))\right]$, where ob is the number of objectives	none	$0\leq x_i \leq 1$ for i=1, 2, 3, ..., n where n=22
	Case Study 2, Case 1	
Max Productivity $=\frac{P(t_f)}{t_f}$ Max Yield $=\frac{P(t_f)}{C_{S,F}(V(t_f)-V(0))}$	$0\leq t_{s1}$, $h\leq t_{s2}$ $t_{s1}\leq t_{s2}$, $h\leq t_f$ $20\leq t_f$, $h\leq 40$ $5\leq V(t_0)$ $L\leq 20$	$0\leq t_{s1}$, $h\leq t_{s2}$ $t_{s1}\leq t_{s2}$, $h\leq t_f$ $20\leq t_f$, $h\leq 40$ $5\leq V(t_0)$ $L\leq 20$
	Case Study 2, Case 2	
Max Productivity $=\frac{P(t_f)}{t_f}$ Min t_f	same as Case 1	same as Case 1

tions. 75% of the population was done with MODE algorithm, while the remaining 25% was sent to DLS in HMODE-DLS algorithm.

In Table 2, $P(t_f)$ is the product mass at the end of the processing time (t_f), $C_{S,F}$ is the concentration of the substrate in the feed, $V(t_f)$ and $V(0)$ are the volume for the reactor at the beginning and the end of the process respectively.

RESULTS AND DISCUSSION

1. Case Study 1: DTLZ 7 Test Problem

For solving DTLZ7 test problem, after detailed investigation from earlier studies on MODE algorithm, it was found that CR value of 0.7 and a random ω between 0.2 to 1.2 provide consistent results [32,55]. The population size was set to 100 and the maximum number of generations was fixed to 800. These settings were also used with HMODE-DLS algorithm with random D_l. Several performance metrics were used to validate the new and improved algorithm. In this work, four metrics were selected: convergence (Conv), which displays how close is the attained Pareto front compared to the actual one; generational distance (GD), which indicates to what level the algorithm has converged compared to the true Pareto front; spacing (SPC), that measures how each minimum distance between solutions deviates from the average of all

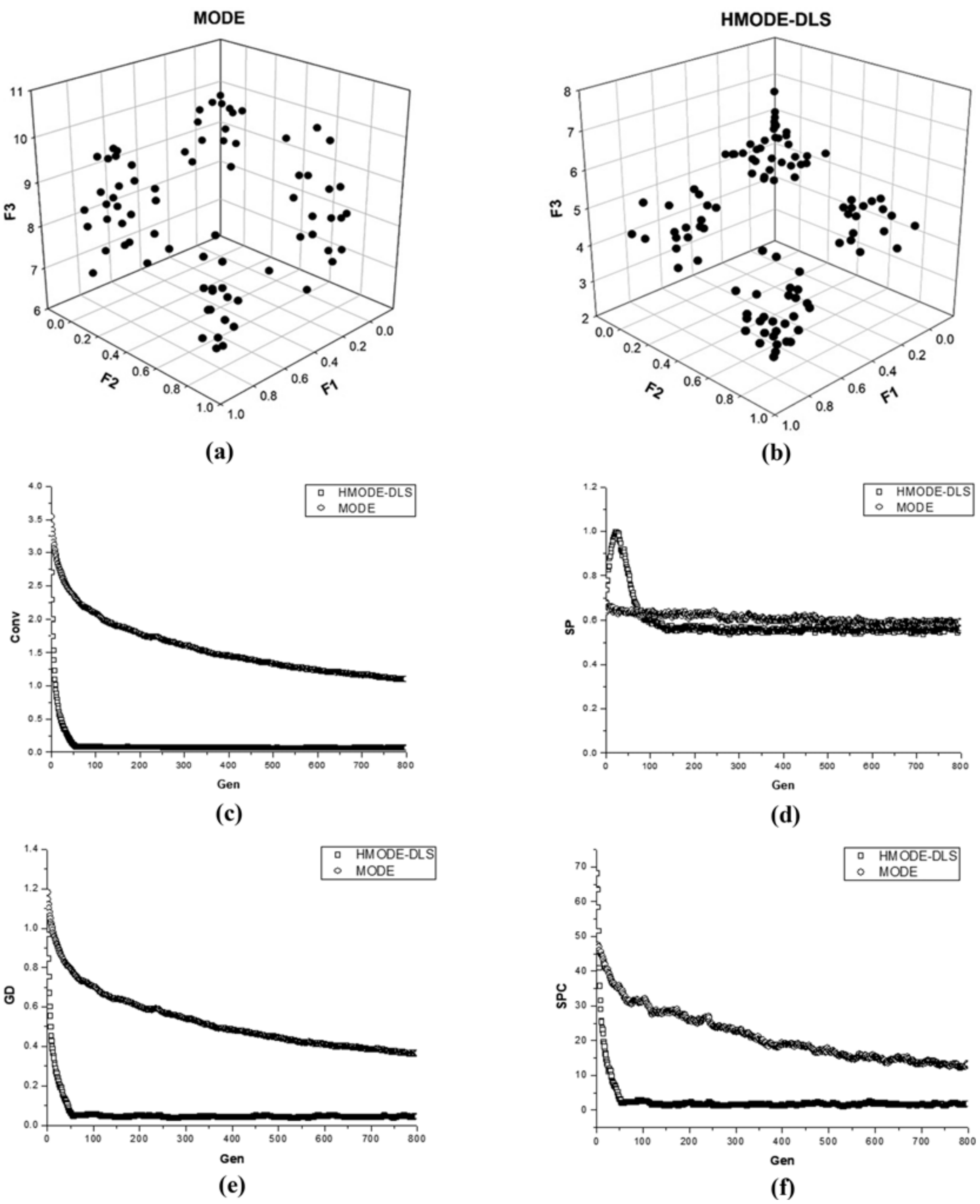

Fig. 2. DTLZ7 Pareto front with (a) MODE, (b) HMODE-DLS algorithms and their performance metrics (c)-(f).

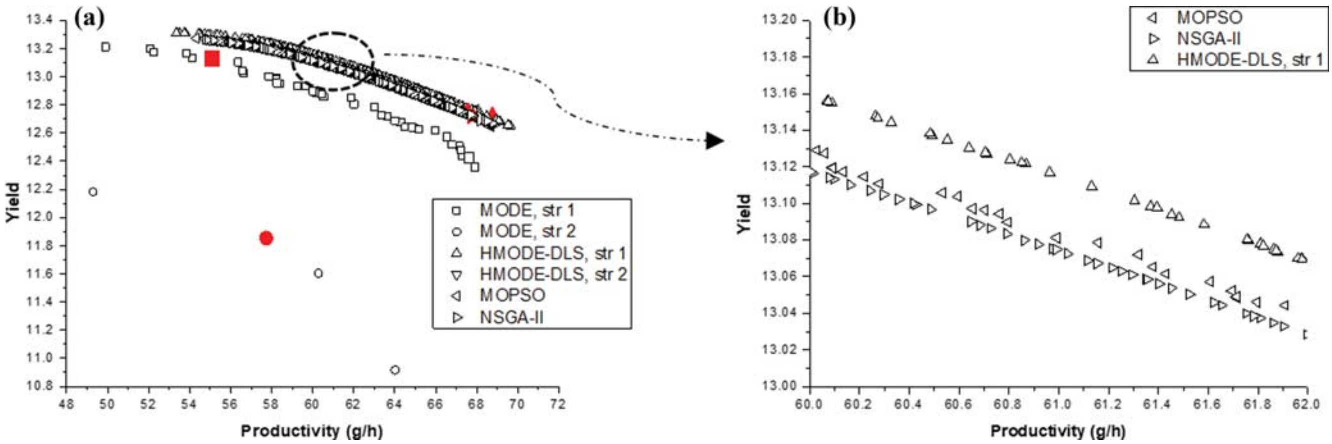

Fig. 3. Case 1: simultaneous maximization of yield and productivity (a) trade-off with different algorithms with best solutions in red and (b) with the best performing algorithms with a magnified area at constant F (2 g/h).

minimum distances, and spread (SP) [56,57].

Fig. 2 represents the achieved Pareto front using MODE (Fig. 2(a)) and HMODE-DLS algorithm (Fig. 2(b)). It is quite clear that the non-dominated solutions obtained using HMODE-DLS algorithm are relatively better in terms of both quality and quantity. HMODE-DLS was able to converge with 100 Pareto solutions, while those of MODE algorithm were 77 points only. Moreover, MODE algorithm Pareto solutions are more scattered and off of the range of the true Pareto front of DTLZ7 test problem compared to HMODE-DLS algorithm. These observations are clearer in the performance metrics (Fig. 2(c)-(f)).

Fig. 2(c)-(f) illustrates the performance metrics of MODE and HMODE-DLS algorithms with respect to the number of generations. As can be seen in Fig. 2(c), HMODE-DLS algorithm converged earlier than MODE algorithm. The latter was not able to converge even after 800 generations ($Conv_{@800}=1.09\pm0.18$). On the other hand, at the 150th generation, HMODE-DLS algorithm Conv metric value was 0.07 ± 0.02. In the same manner, HMODE-DLS algorithm achieved better SP metric results compared to MODE algorithm as shown in Fig. 2(d). A better performance with HMODE-DLS algorithm was also noticed from GD metric results. By analyzing Fig. 2(e), at the 150th generation, GD metric value obtained by MODE was 0.64 ± 0.04, while with HMODE-DLS the value was 0.04 ± 0.02, which is more than twenty times less than that of MODE. SPC metric value, at generation 150, was also found to be better with HMODE-DLS algorithm (1.34 ± 1.30) in comparison to MODE algorithm (28.10 ± 5.83) as shown in Fig. 2(f). From the calculated std results, the advantage of using HMODE-DLS is manifest and justified. This performance improvement in HMODE-DLS algorithm is attributed to the high capability of DLS method in handling the local search in the neighborhood effectively. DLS with MODE enhanced the convergence time and quality, diversity and the spread of the Pareto front.

2. Case Study 2: Industrial Lysine Fed-batch Fermenter based Problem

Using three of the widely used established algorithms with the developed HMODE-DLS algorithm, a comparison was made for solving the lysine industrial problem MOO cases in terms of spread, speed and distribution of the converged non-dominated solutions.

2-1. Case Study 1: Simultaneous Maximization of Yield and Productivity

- Constant feeding

The Pareto front of this case at a F=2 g/h is plotted in Fig. 3 for the considered algorithms. Fig. 3(a) clearly shows the deficiency of MODE algorithm in converging to the global Pareto front with both strategies (str1 and str2). MODE exhibits the worst performance, quality and quantity wise, compared to the other used algorithms. So, for a clearer presentation and easier evaluation, algorithms with competing performances are plotted separately in Fig. 3(b) for a magnified section from Fig. 3(a). Since both strategies (str1 and str2) gave consistent results with HMODE-DLS, the plot includes str1 results only. The superior performance of HMODE-DLS is seen with its more diversified and better converged solutions. It is observed, from Fig. 3(b), that the performance of HMODE-DLS is slightly better than MOPSO and NSGA-II. For example, at a productivity value of 60.8 g/h, the corresponding yield was 13.12 with HMODE-DLS algorithm, while it was 13.08 and 13.09 with MOPSO and NSGA-II, respectively. In addition to HMODE-DLS' better performance, it outshines in the time domain and the total number of final solutions as well. It converged with 250 solutions at the 47th generation. Which is not the case with MOPSO and NSGA-II algorithms, as they did not converge to the same Pareto front and only 137 Pareto solutions were attained with MOPSO algorithm. On the other hand, NSGA-II was able to converge to 250 number of solutions by generation 69. Net flow method was used to rank the solutions in the resulted Pareto fronts and they are shown in with Fig. 3(a) red symbols [58]. The best solution obtained with HMODE-DLS was 69.63 g/h for productivity and 12.65 for yield. While for MOPSO it was 68.53 g/h and 12.66 for productivity and yield, respectively. NSGA-II best solution is very close to that of MOPSO with 68.78 g/h and 12.66 for productivity and yield, respectively. While it is clear that MODE algorithm best solution with str2 (57.73 g/h, 11.85) is worse than the ones discussed. On the other hand, MODE with str1 best solution was 56.36 g/h with yield of 13.10. So, HMODE-DLS best solution was

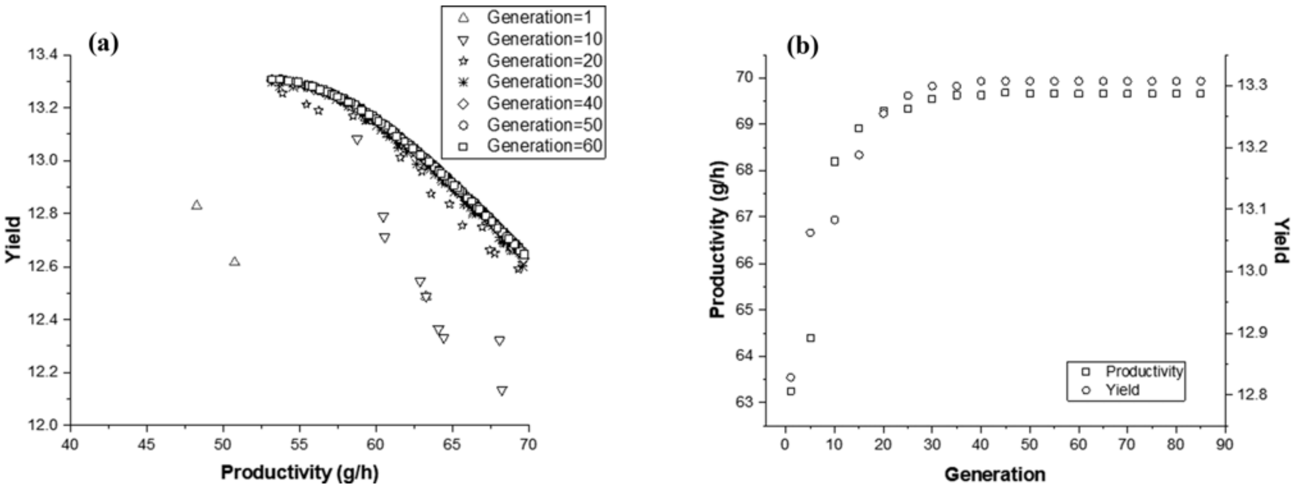

Fig. 4. The efficacy of HMODE-DLS in case 1 (a) Pareto front improvement and (b) improvement of best relative solution with iterations.

relatively better than other algorithms' best solutions.

To show the fast convergence of HMODE-DLS algorithm, Pareto fronts of several generations (iteration) are plotted in Fig. 4(a). The high convergence speed can be clearly observed, especially between generation 1 and 20. By the 20th generation, the Pareto front is already close to that of generation 50 (where the Pareto front has already fully converged). The Pareto front of generation 60 is overlapping that of the 50th generation, which ensures that 50 generations are more than enough for this problem to achieve the desired Pareto set of solutions. In Fig. 4(b), the improvement of the best solution (with respect to each of the objectives individually) with generations is represented. It also exhibits the previous finding, that the problem can be solved in less than 50 generations. Therefore, if productivity is more preferable over the yield, the maximum value that can be achieved is 69.7 g/h, which corresponds to 12.65 for yield. On the other hand, if yield is the more preferable

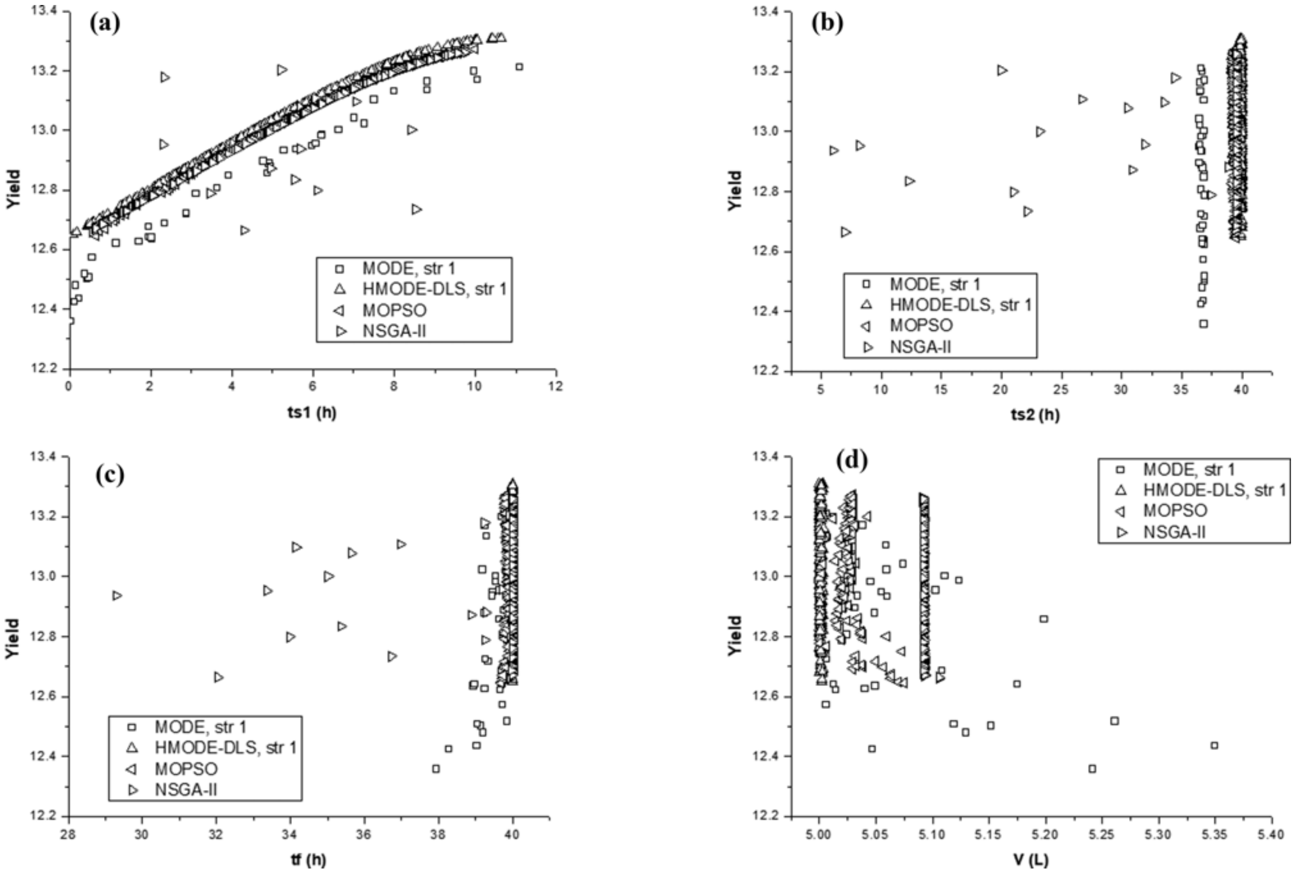

Fig. 5. Case 1: simultaneous maximization of yield and productivity decision variables' trend with different algorithms at constant F (2 g/h).

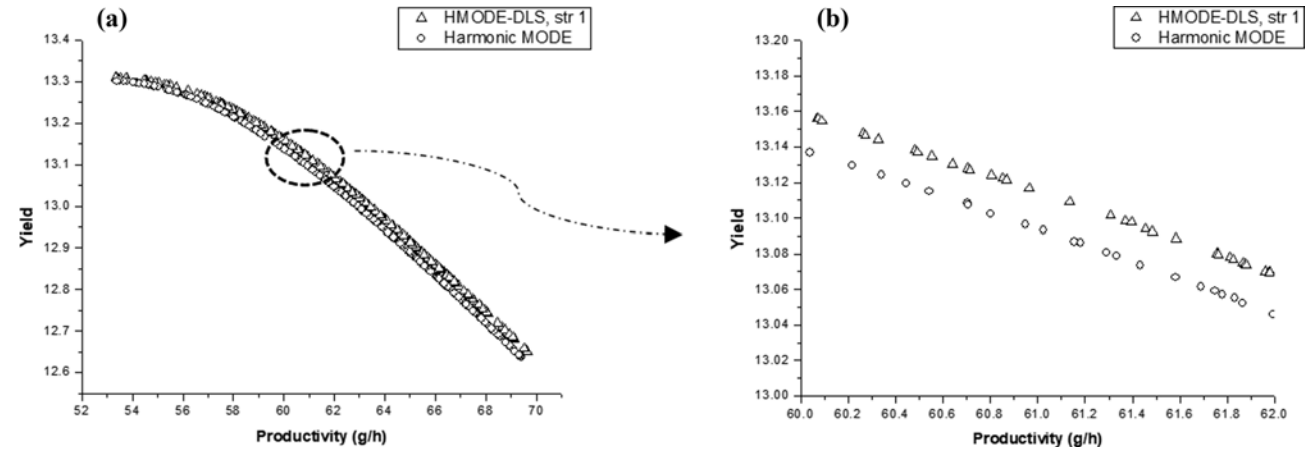

Fig. 6. Case 1: simultaneous maximization of yield and productivity trade-off with HMODE-DLS and Harmonic MODE [25] at constant F (2 g/h).

objective, the maximum value that can be obtained is 13.31 with productivity of 52.87 g/h.

The corresponding decision variables' results, for the four used algorithms are displayed in Fig. 5, where they are plotted with respect to the yield. Since MODE algorithm showed better results with str1 and HMODE-DLS algorithm performance is equally good with both strategies, the discussion will include results with str1 only. It is seen from Fig. 5(a) that the yield has a proportional relationship with decision variable t_{s1} in general, which means that higher batch time resulted in a higher yield value. Although the trend is similar with all algorithms, the yield value for a corresponding t_{s1} value is different and MODE algorithm values are a bit behind the other algorithms. HMODE-DLS converged to the highest yield values. For example, at t_{s1} of 4 h, the yield values are 12.84, 12.94, 12.95 and 12.96 for MODE, MOPSO, NSGA-II and HMODE-DLS, respectively. The difference in the objective function results is not similar due to the effect of other decision variables' results. The influence of t_{s2} decision variable is elucidated in Fig. 5(b). HMODE-DLS algorithm converged to higher values compared to other algorithms as all of its t_{s2} values are between 39.76-39.99 h. The values of the range's limits are smaller with MOPSO (39.13-39.87 h), NSGA-II (37.37-39.47 h) with some scattered points of smaller values seen from 6-34 h and MODE algorithm's results ranging from 36.41-36.91 h. Likewise, there are some differences between the algorithms in t_f decision variable results as shown in Fig. 5(c), where HMODE-DLS algorithm points are between 39.99-40 h. While the range limits values are smaller with other algorithms (39.69-39.85 h for MOPSO, 29.27-39.98 h for NSGA-II and 37.93-39.99 h for MODE). Results of t_{s2} and t_f decision variables indicate that higher time is preferable for higher yield results. Fig. 5(d) shows the impact of the reactor's initial volume (V) on the MOO case results. It is very clear that small V values are required for maximizing both of the objectives as all of the converged values are between 5 and 5.35 L. HMODE-DLS obtained the lowest V values followed by MOPSO and NSGA-II, while MODE algorithm's results are scattered along the range. These results are good evidence for the benefit of using DLS method with MODE algorithm.

The Pareto front from a previous study [25], where Harmonic MODE was found to accomplish the best results at this operating F (2 g/h), is plotted with that of HMODE-DLS for comparison in Fig. 6(a)-(b). The advantage of HMODE-DLS over Harmonic MODE is fairly recognizable, especially from the magnified section shown in Fig. 6(b). Maximum productivity obtained with HMODE-DLS is 69.63 g/h corresponding to a yield value of 12.65, while with Harmonic MODE, the equivalent point is 69.40 g/h for productivity and 12.64 for yield. From Fig. 6(b), when the productivity is approximately 61.87 g/h, a yield value of 13.05 is obtained with Harmonic MODE, while 13.07 is obtained with HMODE-DLS. In addition to HMODE-DLS better achieved Pareto, the algorithm converged at an earlier generation compared to Harmonic MODE. Although the same parameters were used with both of the algorithms, Harmonic MODE failed to converge to the non-dominated solutions obtained by HMODE-DLS.

- Effect of feed changing

Due to the impact of F on the optimization results, this factor is studied and results are displayed in Fig. 7. In general, with all algorithms, as F increases, productivity improves but yield decreases. A glance at Fig. 7(a)-(d) reveals the advantage of using HMODE-DLS (Fig. 7(b)). MODE algorithm, presented in Fig. 7(a), is the worst performing one, as its covered plotting area is not comparable to other algorithms, especially when it comes to the productivity objective, where the maximum productivity value is 67.90 g/h when F is 2 g/h. In addition, all algorithms, except HMODE-DLS, experienced a disconnection in one of the Pareto fronts. This indicates the effect of implementing DLS in the MODE algorithm, as it kept good diversity and distribution of solutions along the Pareto front range. At all F conditions, HMODE-DLS was able to achieve the Pareto front at an earlier generation with a better number of solutions. To illustrate, when F is 1.2 g/h, 250 converged Pareto solutions were obtained with HMODE-DLS at the 50th generation. On the contrary, MOPSO and NSGA-II algorithms were able to achieve 250 non-dominated solutions, but at generation number 274 and 95, respectively. This confirms the fast and efficient performance of HMODE-DLS algorithm with this industrial problem.

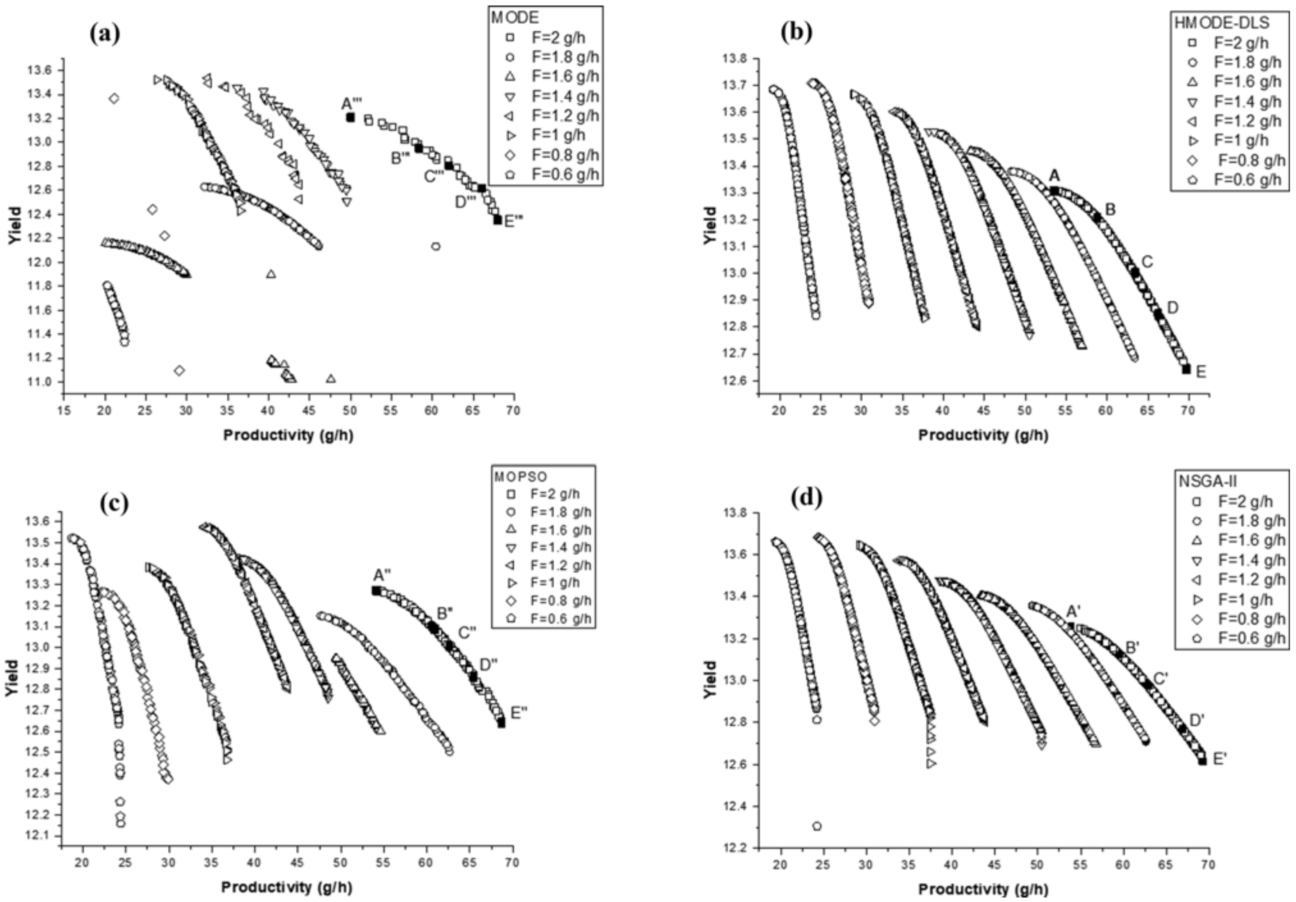

Fig. 7. Case 1: simultaneous maximization of yield and productivity at changing F with different algorithms.

The profiles of the biomass, substrate and product with time are elucidated in Fig. 8(a)-(c) for extreme chromosomes (Pareto solutions) obtained with different algorithms at F=2 g/h. The decision variables corresponding to these selected chromosomes were used to run a lysine kinetic model and obtain the profiles seen in Fig. 8(a)-(c). Additional mid non-dominated chromosomes and which includes the corresponding decision variables' values, objective functions' results and the final amount of biomass, substrate and product, are listed in Table 3. These chromosomes are A, B, C, D and E for HMODE-DLS, A′, B′, C′, D′ and E′ for NSGA-II and A″, B″, C″, D″, E″ for MOPSO and A‴, B‴, C‴, D‴ and E‴ for MODE. The location of these chromosomes in the Pareto front can be found in Fig. 7(a)-(d). A, B, C, D and E for HMODE-DLS are shown in Fig. 7(b) while those of MODE, MOPSO and NSGA-II are shown in Fig. 7(a), (c) and (d) respectively. Note that in moving from chromosome A to E, the productivity increases for HMODE-DLS algorithm (from 52.874 to 69.698 g/h, respectively), while yield decreases (from 13.310 to 12.647 respectively). This behavior is seen with all of the tested algorithms. The impact of t_{s1} on the results is obvious as it decreases with moving from chromosome A to E in all algorithms. For example, at A, t_{s1} is 10.891 h and at E, it is 0.036 h, which means that less batch time is needed and preferable to achieve maximum possible productivity (see Fig. 7(b) and Table 3). For t_{s2} decision variable, there is no specific trend with all algorithms. For instance, from A to E with HMODE-DLS algorithm, all t_{s2} values are 39.967 h or higher, which is higher than any other t_{s2} value obtained with the other three algorithms. The lowest t_{s2} values are attained with MODE algorithm, as discussed previously. In general, t_f seems to slightly decrease when targeting the maximization of productivity. For instance, with HMODE-DLS algorithm, where productivity maximization is favored as moving from chromosome A to E, t_f shows a trivial reduction of about 0.001 h. This trend generally applies for other algorithms. For V decision variable, it can be said that slightly higher V values are needed for maximizing productivity in HMODE-DLS and NSGA-II algorithms, as the value changed from 5.001 to 5.003 L (chromosomes A to E, respectively) and from 5.091 to 5.106 L for chromosomes A′ to E′, respectively. On the other hand, no specific trend is seen with the other two algorithms. So, the key decision variable which caused the conflicting nature between the objectives is t_{s1}, as its highest achieved values are preferred for higher yield, while lower values are needed to maximize productivity. When only the maximization of yield is desired (A, A′, A″ and A‴), lower biomass growth is needed as seen in Fig. 8(a) compared with when only the productivity maximization is desirable (E, E′, E″ and E‴). From the same graph, Fig. 8(a), it can be noticed that the profile of A is the lowest amongst all, and E is the highest, which supports Table 3 discussed values. Fig. 8(b) represents the profile of substrate with time and in that, it can be clearly seen that earlier t_{s1} values are needed when the max-

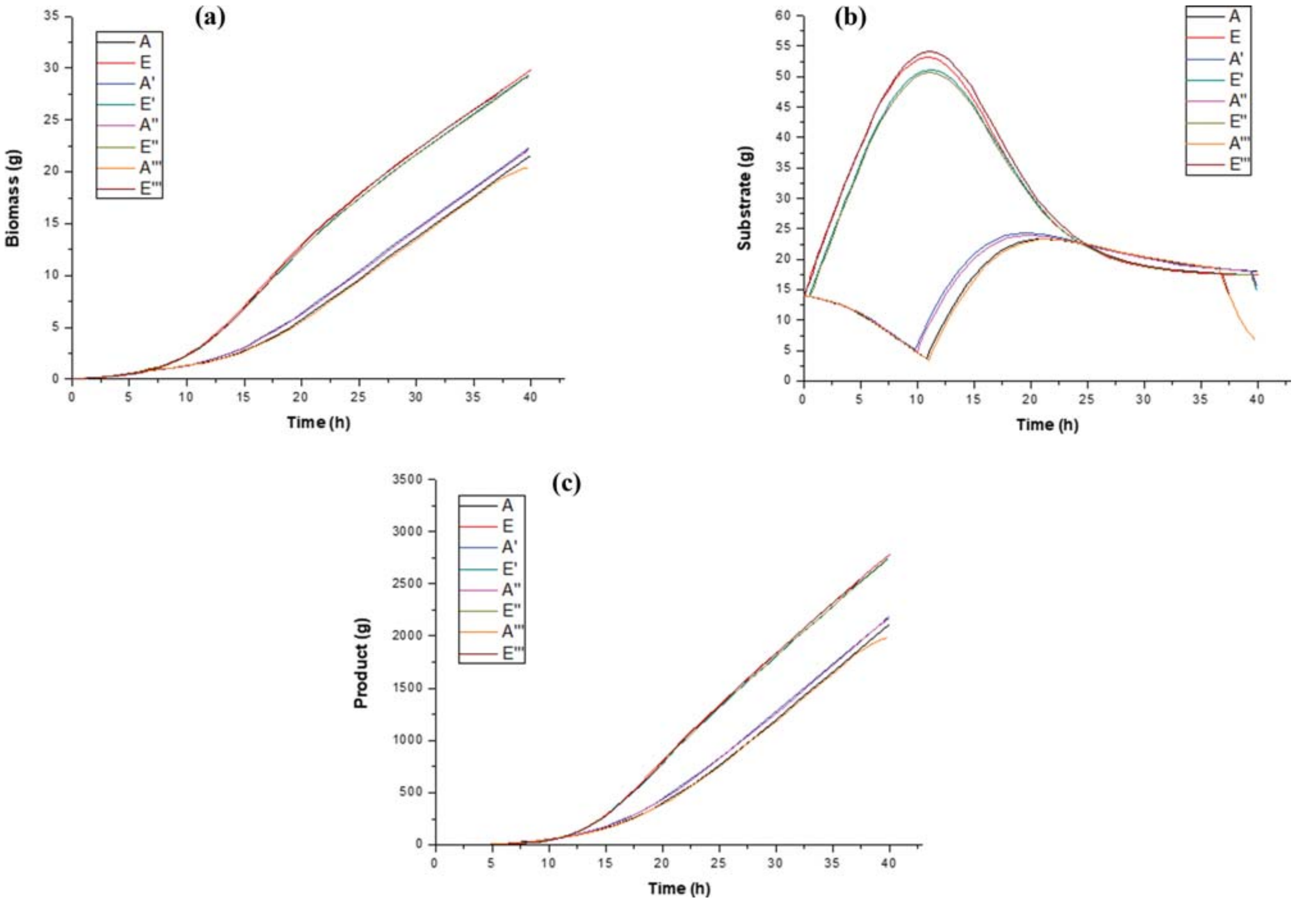

Fig. 8. Optimum profiles for selected chromosomes (A, E, A′, E′, A″, E″, A‴ and E‴) of Pareto fronts in Fig. 7 and Table 3 at F=2 g/h.

Table 3. Results of the used algorithms for selected data points (chromosomes) from Fig. 7 at F=2 g/h

Algorithm	Point	t_{s1} (h)	t_{s2} (h)	t_f (h)	V (L)	Productivity (g/h)	Yield
HMODE-DLS, str1	A	10.891	39.997	39.998	5.001	52.874	13.310
	B	7.247	39.967	39.998	5.001	59.232	13.189
	C	4.820	39.967	39.999	5.002	62.960	13.021
	D	2.582	39.992	39.998	5.003	66.193	12.849
	E	0.036	39.982	39.997	5.003	69.698	12.647
NSGA-II	A′	9.791	39.457	39.979	5.091	54.764	13.264
	B′	7.309	39.460	39.978	5.091	58.971	13.157
	C′	5.008	39.466	39.978	5.092	62.525	13.003
	D′	2.560	39.466	39.978	5.092	66.081	12.820
	E′	0.557	39.449	39.943	5.106	68.848	12.659
MOPSO	A″	9.987	39.336	39.828	5.029	54.311	13.273
	B″	7.077	39.313	39.848	5.028	59.272	13.151
	C″	5.221	39.309	39.862	5.029	62.143	13.025
	D″	3.558	39.377	39.887	5.026	64.626	12.904
	E″	0.630	39.460	39.694	5.075	68.657	12.647
MODE, str1	A‴	11.104	36.609	39.770	5.006	49.893	13.211
	B‴	6.230	36.767	39.908	5.123	58.231	12.987
	C‴	3.921	36.854	39.829	5.019	61.879	12.848
	D‴	1.703	36.719	39.257	5.040	65.149	12.626
	E‴	0.038	36.860	37.938	5.241	67.902	12.358

imization of productivity is given more priority (E, E′, E″ and E‴) over the yield. The profiles of HMODE-DLS and MODE algorithms are very close to each other, but since t_{s2} occurs earlier with MODE algorithm (A‴ in Fig. 8(b)), the substrate starts depleting at 36.609 h. But with HMODE-DLS (A in Fig. 8(b)) the starting depletion time is 39.997 h. Additionally, substrate formation and depletion is higher and more rapid with chromosomes E, E′, E″ and E‴, which helps in achieving more production rate. On the other hand, less substrate in the fed-batch operation works on keeping the yield values as high as possible (Fig. 8(b)). For the

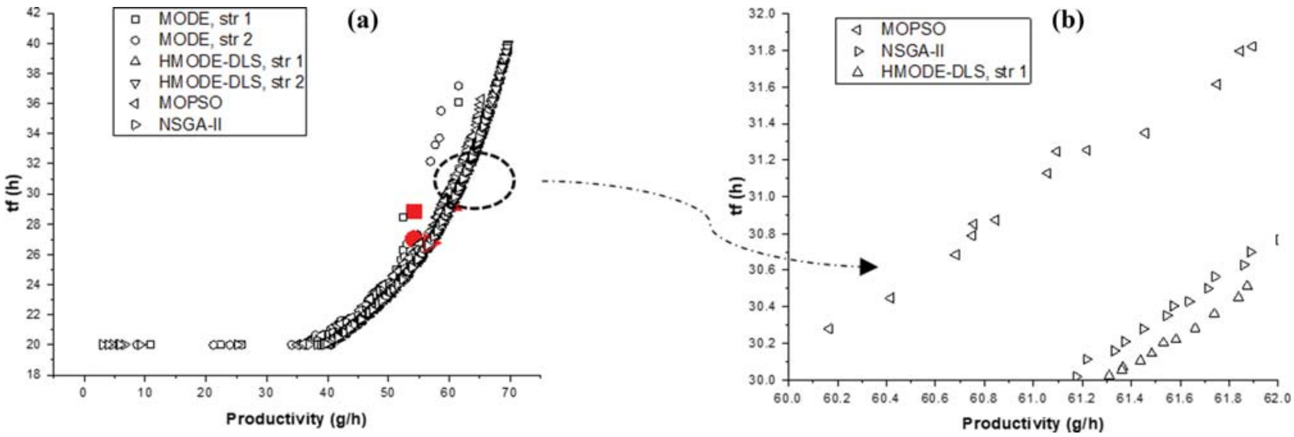

Fig. 9. **Case 2: simultaneous maximization of productivity and minimization of t_f (a) trade-off with different algorithms with best solutions in red and (b) with the best performing algorithms with a magnified area at constant F (2 g/h).**

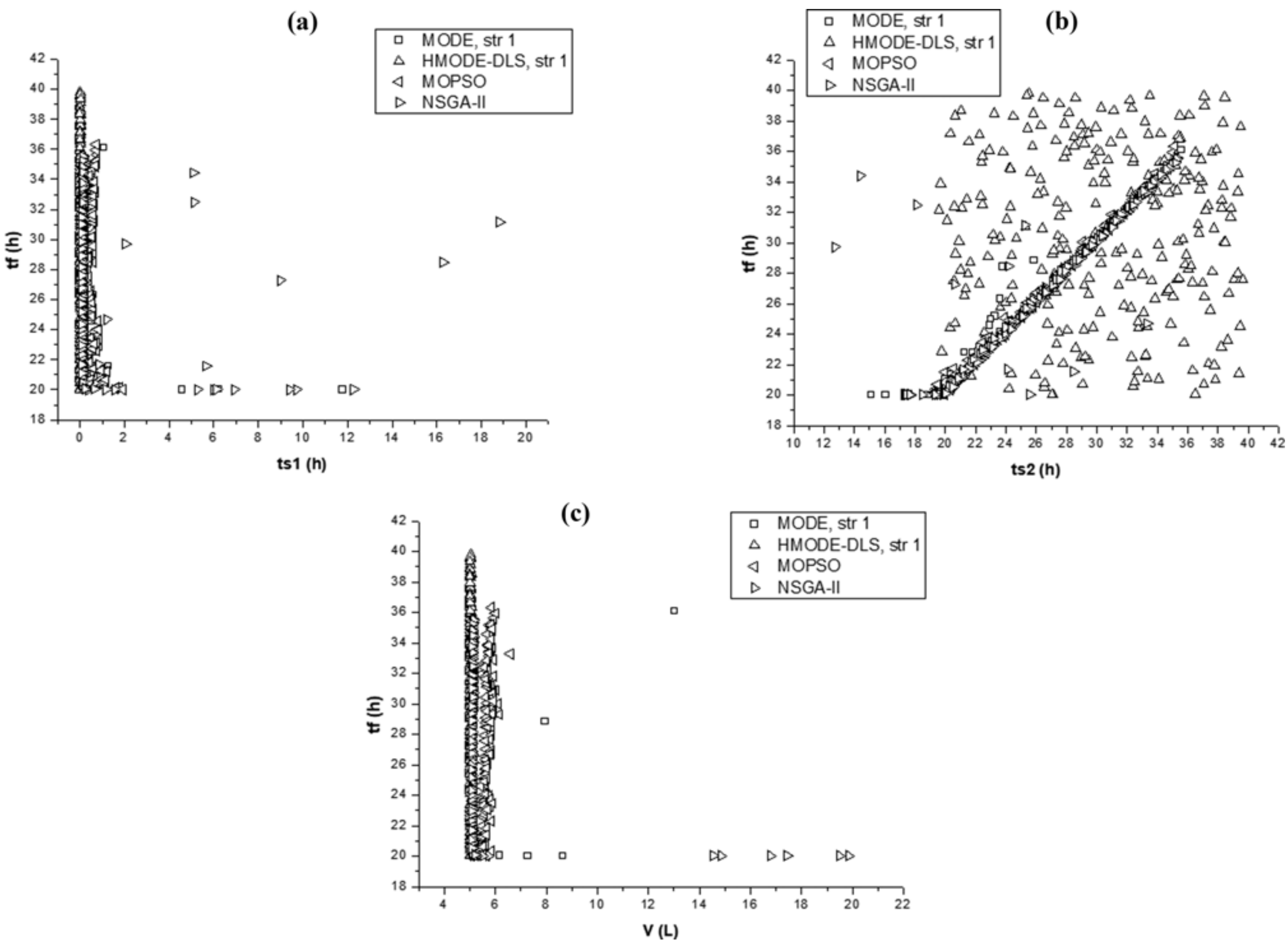

Fig. 10. **Case 2: simultaneous maximization of productivity and minimization of t_f decision variables trends at changing F with different algorithms.**

product formation with time, Fig. 8(c), less product is manufactured when higher yield is preferred (A, A′, A″ and A‴) and vice versa.

2-2. Case Study 2: Simultaneous Maximization of Productivity and Minimization of t_f

• Constant feeding

In accordance with case 1, a constant F of 2 g/h was used to carry out this part of the study and the Pareto fronts are shown in Fig. 9(a). The convergence deficiency is significant with MODE algorithm, while the other three algorithms exhibit a contending execution. But, the Pareto front of MOPSO and NSGA-II is disconnected as seen at productivity values between 2.5 and 34 g/h. To differentiate between the exact Pareto front of each algorithm (MOPSO, NSGA-II and HMODE-DLS), a small section is magnified and presented separately in Fig. 9(b). It is noticeable that MOPSO Pareto front is not as good as those of NSGA-II and HMODE-DLS algorithms. A minor, yet important, difference exists between the two latter algorithms, where HMODE-DLS proves to be the better one. To elaborate, a productivity of 61.44 g/h is approached in approximately 30.10 h with HMODE-DLS, while it takes 30.28 and 31.35 h with NSGA-II and MOPSO, respectively. In addition, the better spread of HMODE-DLS is distinctly salient. HMODE-DLS Pareto front was achieved in 70 generations only. The best solution for each algorithm was selected with net flow method and identified with red symbols in Fig. 9(a). With HMODE-DLS, the best solution is 61.20 g/h productivity with 29.88 h operation time. With NSGA-II, the best solution is 56.53 g/h and the total time is 26.60 h. While with MOPSO it is 56.51 g/h corresponding to 26.98 h operation time. The best solutions obtained by MODE algorithm are 54.20 g/h and 28.83 h, and 54.76 g/h and 27.33 h for str1 and str2, respectively.

The corresponding decision variable results for the discussed Pareto fronts are demonstrated in Fig. 10. To minimize t_f, minimum t_{s1} is required, which is seen in Fig. 10(a), where HMODE-DLS converged to the minimum needed time, while the other algorithms could not do as efficiently. With t_{s2} decision variable, HMODE-DLS results are scattered between 20 and 40 h, while other algorithms' results are showing a proportional trend over the same time range (Fig. 10(b)). In the same manner, HMODE-DLS algorithm converged to the minimum V possible, whereas the other algorithms did not as plotted in Fig. 10(c).

• Effect of feed changing

Contrary to case 1, increasing F values has a positive impact on the productivity objective, but a non-sensible influence is noticed on the t_f objective. In similar behavior with case 1, HMODE-DLS provided better quality Pareto fronts in terms of convergence and spread for all the studied F values shown in Fig. 11(b). Consequently, MODE is the least favored algorithm due to the low number of solutions and the lack of good convergence and spread compared to other algorithms (Fig. 11(a)).

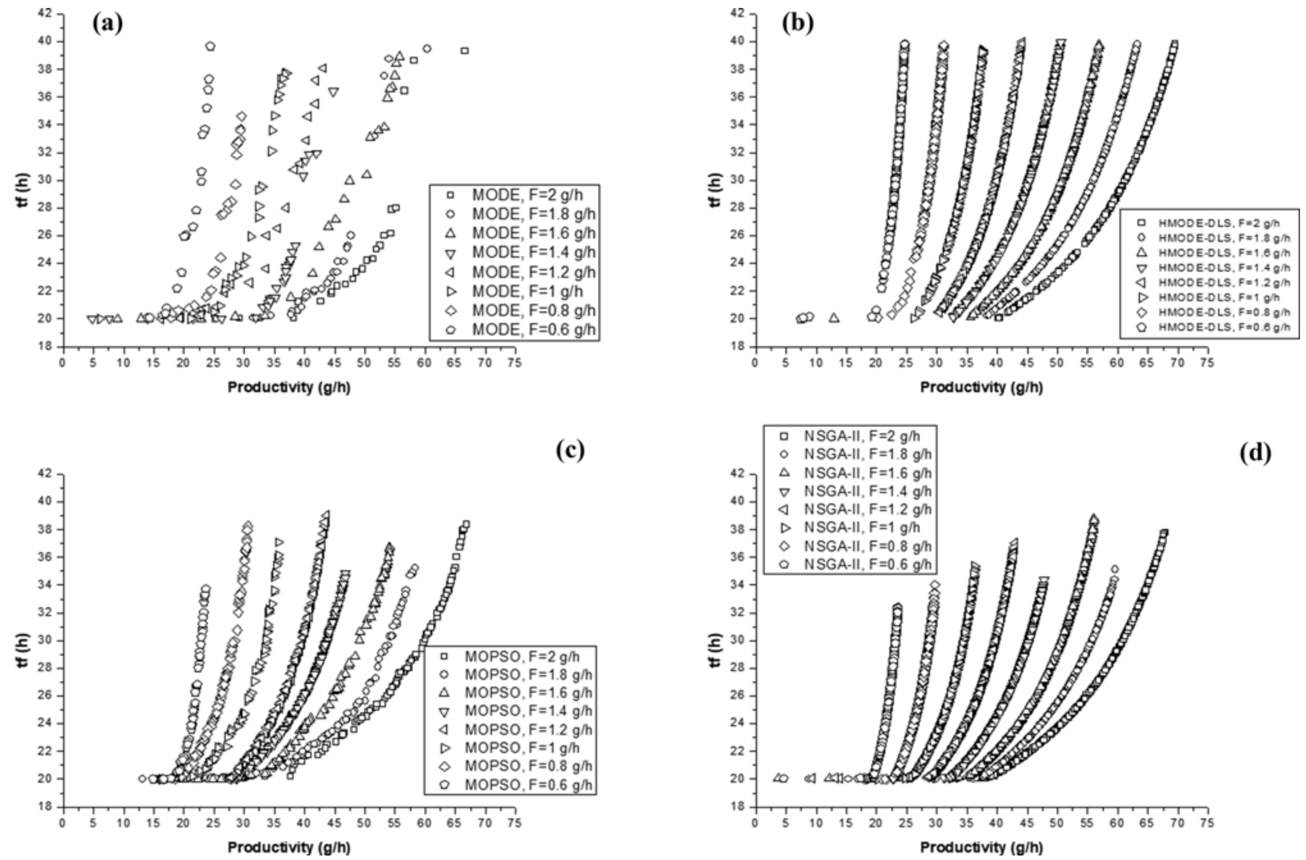

Fig. 11. Case 2: simultaneous maximization of productivity and minimization of t_f trade-off at changing feeding rate with different algorithms.

CONCLUSIONS

MOO of the tri-objective DTLZ7 test problem and two industrial cases of lysine biochemical process were carried out with the proposed algorithm, HMODE-DLS, NSGA-II, MODE and MOPSO algorithms. Two considered lysine cases were the maximization of productivity and yield (Case 1) and the maximization of productivity and the minimization of t_f (Case 2) at constant and changing feeding rate. From the obtained results, it can be concluded that HMODE-DLS has the ability of converging to the true Pareto front of DTLZ7 test problem, as well as achieving a better trade-off in the lysine industrial cases.

With DTLZ7 test problem study, four performance metrics were calculated: Conv, SP, GD and SPC to compare between MODE and HMODE-DLS algorithms. The results revealed the relatively faster and more accurate performance of HMODE-DLS compared to the other one, as all its performance metric's mean values are of better quality. 150 generations were found to be enough for the algorithm to converge, while MODE algorithm did not converge with 800 generations. This improvement in the performance is accredited to the involvement of DLS in the decision space searching mechanism with the evolutionary algorithm side by side.

In the industrial lysine production case study, the conflicting objectives (maximization of productivity and yield) in case 1 were considered with a constant feeding rate. Furthermore, the effect of changing the feed was also studied. In all feeding conditions, HMODE-DLS was able to attain Pareto fronts with wider spread and better convergence when compared to MODE, MOPSO and NSGA-II. In addition, the global Pareto front was reached in an earlier generation with a higher number of non-dominated solutions. With all algorithms, increasing the feeding rate resulted in higher productivity and yield values. MODE algorithm was the worst performing algorithm as it lacked in converging to the final number of non-dominated solutions. When the constant feeding flow rate study results were compared with another reported work, HMODE-DLS algorithm showed its dominance over the best stated algorithm in that study (Harmonic MODE).

In case 2, where the objectives involved minimizing operation time and maximizing productivity, some findings were noticed regarding the advantage of using HMODE-DLS in comparison with MODE, MOPSO and NSGA-II algorithms. The best solution obtained with each algorithm was selected based on net flow method. According to the entire studied problems in this work, HMODE-DLS has great potential in being successfully used in solving theoretical and real world problems. It is able to find the desired Pareto front with the required number of solutions in much less computation time and number of function evaluations than other algorithms, included in the study, require. Therefore, solving other types of real world MOO problems will give a better indication for its performance.

NOMENCLATURE

Abbreviation/Symbol

c1 : personal learning coefficient
c2 : Global learning coefficient
Conv : convergence metric
CR : crossover
$C_{S,F}$: concentration of S in the feed [g/ L]
D_l : local scaling factor
DLS : dynamical local search
F : feeding flow rate [g/h]
M_f : mutation factor
GD : generational distance metric
i : number of decision variables
MODE : multi-objective differential evolution algorithm
MOEA's : multi-objective evolutionary algorithms
MOO : multi-objective optimization
n : number of decision variables
NSGA-II : non-dominated sorting genetic algorithm-II
ob : number of objective functions for DTLZ7 test problem
P : product mass [g]
Q : number of population handled by DLS
S : substrate mass [g]
SP : spread metric
SPC : spacing metric
str1 & str2 : strategies of MODE algorithm
t_f : maximum operating time [h]
t_{s1} : first switch time [h]
t_{s2} : second switch time [h]
u : volume flow rate of the feed [L/h]
V : reactor volume [L]
w inertia weight
w_{Damp} : inertia damping rate
X : biomass mass [g]
x_{DLS_new} : new population point generated with DLS
x_{MODE_new} : new population point generated with MODE algorithm
x_o, x_m, x_a, x_b, x_c, x_d & x_e : random points selected from the generation
π : rate of S consumed [g/gh]
σ : rate of P formation [g/gh]
μ : growth rate [1/h]
ω : frequency factor
ω_l : lower ω limit
ω_u : upper ω limit

REFERENCES

1. M. Singh, *Int. J. Res. Pharm. Sci. (Madurai, India)*, **2**, 637 (2016).
2. S. Anastassiadis, *Recent Pat. Biotechnol.*, **1**, 11 (2007).
3. R. Faurie, T. Scheper and J. Thommel, *Amino Acids*, **79**, 1 (2003).
4. D. Kumar and J. Gomes, *Biotechnol. Adv.*, **23**, 41 (2005).
5. M. Moosavi-Nasab, S. Ansari and Z. Montazer, *Iran Agric. Res.*, **25**, 99 (2008).
6. I. Ekwealor and J. Obeta, *Afr. J. Biotechnol.*, **4**, 633 (2005).
7. S. Mitsuhashi, *Curr. Opin. Biotechnol.*, **26**, 38 (2014).
8. T. Escobet, V. Puig, J. Quevedo, P. Palá-Schönwälder, J. Romera and W. Adelman, *Comput. Chem. Eng.*, **124**, 228 (2019).
9. D. Deka and D. Datta, *Knowl-Based Syst.*, **121**, 71 (2017).
10. J. O. H. Sendín, A. A. Alonso and J. R. Banga, *J. Food Eng.*, **98**, 317 (2010).
11. A. M. Enitan and J. Adeyemo, *Afr. J. Biotechnol.*, **10**, 16120 (2011).

12. G. P. Rangaiah, *Multi-objective optimization: Techniques and applications in chemical engineering*, World Scientific, Singapore (2009).
13. G. P. Rangaiah, *Multi-objective optimization: Techniques and applications in chemical engineering*, World Scientific, Singapore (2016).
14. T. Seifert, *A contribution to the design of flexible modular chemical plants*, Verlag Dr. Hut, Germany (2015).
15. S. S. Parhi, G. P. Rangaiah and A. K. Jana, *Appl. Therm. Eng.*, **150**, 1273 (2019).
16. S. Garcia and C. T. Trinh, *Processes*, **7**, 316 (2019).
17. A. D. Rodman, E. S. Fraga and D. Gerogiorgis, *Comput. Chem. Eng.*, **108**, 448 (2018).
18. Z. Guo and X. Yan, *Chemom. Intell. Lab. Syst.*, **177**, 8 (2018).
19. J. Q. Albarelli, S. Onorati, P. Caliandro, E. Peduzzi, M. A. A. Meireles, F. Marechal and A. V. Ensinas, *Energy*, **138**, 1281 (2017).
20. J. Shadbahr, Y. Zhang, F. Khan and K. Hawboldt, *Renew. Energy*, **125**, 100 (2018).
21. S. Sharma and G. Rangaiah, *Multi-objective optimization of a fermentation process integrated with cell recycling and inter-stage extraction*, Computer Aided Chemical Engineering, Elsevier (2012).
22. T. Y. Kim, J. M. Park, H. U. Kim, K. M. Cho and S. Y. Lee, *Metab. Eng.*, **28**, 63 (2015).
23. D. Sarkar and J. M. Modak, *Chem. Eng. Sci.*, **60**, 481 (2005).
24. A. M. Gujarathi, B. Al-Siyabi, N. Sivakumar and M. Mathew, *Proceedings of the 10th international conference on thermal engineering*, Muscat, Oman, 26-28 Febreuary (2017).
25. B. Al-Siyabi, A. M. Gujarathi and N. Sivakumar, *Mater. Manuf. Processes*, **32**, 1152 (2017).
26. F. Logist, P. Van Erdeghem and J. Van Impe, *Chem. Eng. Sci.*, **64**, 2527 (2009).
27. S. Taras and A. Woinaroschy, *Comput. Chem. Eng.*, **43**, 10 (2012).
28. E. Heinzle, A P. Biwer and C. L. Cooney, *Development of sustainable bioprocesses: Modeling and assessment*, John Wiley & Sons, England (2007).
29. X. Gao, B. Chen, X. He, T. Qiu, J. Li, C. Wang and L. Zhang, *Comput. Chem. Eng.*, **32**, 2801 (2008).
30. T. Liu, X. Gao and L. Wang, *J. Taiwan Inst. Chem. Eng.*, **57**, 42 (2015).
31. S. Yijie and S. Gongzhang, *Chin. J. Aeronaut.*, **21**, 540 (2008).
32. A. M. Gujarathi and B. Babu, *Ind. Eng. Chem. Res.*, **48**, 11115 (2009).
33. S. Sharma and G. P. Rangaiah, *Comput. Chem. Eng.*, **56**, 155 (2013).
34. V. L. Huang, P. N. Suganthan, A. K. Qin and S. Baskar, *Singapore: Nanyang Technological University, Technical Report* (2005).
35. B. Babu, A. M. Gujarathi, P. Katla and V. Laxmi, *Proceedings of the international conference on emerging mechanical technology: macro to nano*, Pilani, India, 16-18 Febreuary (2007).
36. A. M. Gujarathi and B. V. Babu, *Mater. Manuf. Processes*, **26**, 455 (2011).
37. B.-y. Qu and P.-N. Suganthan, *J. Zhejiang Uni. SCIE. C*, **11**, 538 (2010).
38. B. Qu and P. N. Suganthan, *Proceedings of the IEEE congress on evolutionary computation*, Barcelona, Spain, 18-23 July (2010).
39. B. Y. Qu and P. N. Suganthan, *Proceedings of the IEEE symposium on differential evolution*, Paris, France, 11-15 April (2011).
40. X. Chen, W. Du and F. Qian, *Chemom. Intell. Lab. Syst.*, **136**, 85 (2014).
41. B. Y. Qu, J. J. Liang, Y. S. Zhu, Z. Y. Wang and P. N. Suganthan, *Inf. Sci.*, **351**, 48 (2016).
42. Y.-N. Wang, L.-H. Wu and X.-F. Yuan, *Soft Comput.*, **14**, 193 (2010).
43. A. M. Gujarathi and B. Babu, *Proceedings of the 4th indian international conference on artificial intelligence*, India, December 16-18 (2009).
44. B. Babu and B. Anbarasu, *Proceedings of the international symposium 58th annual session of IIChE*, New Delhi, India, 14-17 December (2005).
45. A. Percus, G. Istrate and C. Moore, *Computational complexity and statistical physics*, Oxford University Press, United States of America (2006).
46. M. Zhang, H. Wang, Z. Cui and J. Chen, *Memet. Comput.*, **10**, 199 (2018).
47. K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, *IEEE Trans. Evol. Comput.*, **6**, 182 (2002).
48. A. Sundaram, *Particle swarm optimization with applications*, InTech, United Kingdom (2018).
49. C. A. C. Coello and M. S. Lechuga, *Proceedings of the 2002 congress on evolutionary computation*, Honolulu, HI, USA, 12-17 May (2002).
50. H. Ohno, E. Nakanishi and T. Takamatsu, *Biotechnol. Bioeng.*, **18**, 847 (1976).
51. K. Deb, L. Thiele, M. Laumanns and E. Zitzler, *Proceedings of the 2002 congress on evolutionary computation*, Honolulu, HI, USA, 12-17 May (2002).
52. F. Logist, B. Houska, M. Diehl and J. F. Van Impe, *Chem. Eng. Sci.*, **66**, 4670 (2011).
53. A. M. Gujarathi and B. V. Babu, *Chem. Eng. Sci.*, **65**, 2009 (2010).
54. A. M. Gujarathi and B. V. Babu, *Mater. Manuf. Processes*, **24**, 303 (2009).
55. A. M. Gujarathi and B. V. Babu, *Int. J. Comput. Int. Stud.*, **2**, 157 (2013).
56. E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications*, Citeseer, Switzerland (1999).
57. K. Deb, *Multi-objective optimization using evolutionary algorithms*, Wiley & Sons, England (2002).
58. E. S.-Q. Lee and G. Rangaiah, *Ind. Eng. Chem. Res.*, **48**, 7662 (2009).