Applied Soft Computing Journal 106 (2021) 107317

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

A self-adaptive multi-objective dynamic differential evolution algorithm and its application in chemical engineering

Xiaodong Zhang [a], Lu Jin [a], Chengtian Cui [b], Jinsheng Sun [a,*]

[a] School of Chemical Engineering and Technology, Tianjin University, Tianjin, 300072, PR China
[b] Institute of Intelligent Manufacturing, Nanjing Tech University, Nanjing, 211816, Jiangsu, PR China

ARTICLE INFO

Article history:
Received 21 September 2020
Received in revised form 11 February 2021
Accepted 10 March 2021
Available online 20 March 2021

Keywords:
Multi-objective optimization
Dynamic differential evolution
Parameter adaptation
Optimal control problem
Chemical and biochemical processes

ABSTRACT

This paper proposes a new multi-objective dynamic differential evolution algorithm with parameter self-adaptive strategies, named SA-MODDE. All components of the algorithm are synergically designed to reach its full potential, containing parental selection, mutation strategy, parameter setting, survival selection, constraint handling, and termination criteria. The improvement measures emphasize exploiting Pareto dominance information more efficiently. Particularly, parameter adaptation schemes are introduced based on both prior knowledges of current individual and feedback information on previous promising solutions, and their effectiveness is validated by comparison with three fixed-parameter combinations. Extensive numerical tests are conducted on multiple test suites with five state-of-the-art peer competitors. The statistical results demonstrated that the SA-MODDE exhibits good proximity and diversity in dealing with benchmark functions with various characteristics. Three industrial (bio)chemical processes, including two optimal control and one reformulated constrained tri-objective, are investigated to show the feasibility and robustness of the SA-MODDE.

1. Introduction

Engineering problems always require the simultaneous optimization of several competing objectives of interests. So far, multi-objective optimization (MOO) has been an active research field in process systems engineering [1,2]. Particularly, various multi-objective evolutionary algorithms (MOEAs), such as NSGA-II, GDE3, and MOPSO, have been widely used to solve both academic and industrial MOO problems [3–5]. Usually, MOEAs have two main advantages: (1) As many diverse non-dominated solutions as possible can be found in a single run; (2) Various types of MOO problems can be handled without assumptions on objective functions and their mathematical characteristics [6].

The algorithm structure and search operator jointly affect the performance of MOEAs. The algorithm structure can be classified into two main categories: Pareto-based [3,7] and decomposition-based [8,9]. The former provides detailed Pareto dominance information of the population to facilitate individual comparison. The latter decomposes MOO problems into a set of scalar aggregation subproblems, each of which is optimized using the current information from neighboring subproblems. The two methods have their own advantages on different types of problems and are considered to be evenly matched [10]. In terms of search operator, differential evolution (DE) is simple to implement with only a few control parameters, i.e., scale factor (F) and crossover rate (CR). Except for multi-objective DE (MODE) algorithms, many classic MOEAs also replaced the original evolutionary operators with DE and their performance was significantly enhanced, such as NSGA-II-DE, SPEA2-DE, IBEA-DE, and MOEA/D-DE [11,12]. Through updates by dynamic population rather than generation to generation, Qing [13] presented the dynamic DE (DDE) operator, superior in efficiency, robustness, and storage requirements to the conventional DE. That is, each new individual that performs better than or similar to the corresponding old counterpart will immediately participate in the current population to provide information for subsequent evolution. This makes DDE more responsive to changes in population status. Despite of researches on multi-objective DDE (MODDE) algorithms [14,15], it is still very inadequate compared to MODE. Herein, we propose several improvement measures on the MODDE under Pareto-based structure. For convenience, the background of MOO problems and DDE operators are given in **Supplementary Materials**.

Maintaining a good balance between the proximity and diversity of obtained non-dominated solutions is not a trivial task for MOEAs [16]. Most existing MODEs conquer this challenge by improving a particular component. Lin et al. [17] embedded three mutation schemes with different features and corresponding adaptive scheme selection strategy in their algorithm, which is better than variants with a single mutation scheme.

* Corresponding author.
E-mail address: jssun2006@vip.163.com (J. Sun).

https://doi.org/10.1016/j.asoc.2021.107317
1568-4946/

Nomenclature

Abbreviations

A	A kind of catalyst
CE	Crowding entropy
CSTC	Chi-squared-test based termination criterion
CVP	Control vector parameterization
DDE	Dynamic differential evolution
DE	differential evolution
DTLZ	Deb–Thiele–Laumanns–Zitzler
GRA	Gray relational analysis
HSS	Hybrid selection strategy
IQR	Interquartile range
MFE	Maximum number of function evaluation
MODDE	Multi-objective DDE
MODE	Multi-objective DE
MOEAs	Multi-objective evolutionary algorithms
MOO	Multi-objective optimization
PM	Performance metric
SA-MODDE	Self-adaptive MODDE
SSDTC	Steady-state detection termination criterion
ZDT	Zitzler–Deb–Thiele

Symbols

χ^2-test	Chi-squared-test
$\overline{\mathrm{x}}$	Median
GD_m	Modified generational distance
P^0	Initialized population
$P_0P_1P_2$	Points
P^1	Sorted population
PF^*	True Pareto front
X_p	Parents vector
e_m	The mth extreme solutions
δ_{PM}	User-defined tolerance value
δ_r	Standard deviation of rankings
μ_r	Average ranking
CR	Crossover rate
D	Decision variables
E	Distribution entropy
F	Scale factor
IGD	Inverted Generational Distance
M	Number of objective functions
NP	Population size
PF	Obtained Pareto front
PS	Pareto optimal set
R	Rank set
$Ratio\_CR$	Score level of X_i
S	Substances
SP	Spread
SS	Solution set
U	Trial vector
V	Mutation vector
X	Target vector
d	Euclidian distance
f	Objective function
n	Number of solutions or problems
$nadir\_point$	Point consisting of maximum
γ	Number of generations
$\lambda + \mu$	Original+offspring population

Subscripts

b	Base vector
$best$	Best value
i, q	Individual or solution index
$init$	Initial value
j	Objective function index
m	Modify
p	Parents
r	Ranking
s	Starting vector
t	Terminal vector

Superscripts

max	maximum value
min	minimum value

Xu et al. [18] partitioned the entire population into several subpopulations and constructed a hybrid selection strategy (HSS), and each subpopulation was assigned a survival selection scenario. Scalability experiment studies indicated that HSS was able to deal with high-dimension MOO problems. Lin et al. [19] adjusted the values of F and CR according to the success rate of offspring in each generation, and confirmed the validity of their work by comparing with two fixed parameter settings. In our work, considering the synergistic effect, several components are carefully devised based on the original MODDE framework. They contain parental selection, mutation strategy, parameter setting, survival selection, constraint handling, and termination criteria. Notably, through simultaneously exploiting prior and posteriori information during the evolutionary process, the new parameter self-adaptive (SA) strategies are devised to customize F and CR for each individual, respectively. Therefore, the proposed algorithm is named as SA-MODDE. Moreover, the external elitist archive typically used for diversity preservation is not required here.

An exhaustive overview of the application of MOEAs to process systems engineering was given by Rangaiah and Petric. [6], covering the areas of petrochemicals, biofuels, environment, and thermodynamics. In addition, Babu and Gujarathi [20] addressed a three-stage supply chain problem involving a network of suppliers, plants and customer areas, and considered three cases of objective functions using MODE. For the industrial manufacturing process, two styrene reactor configurations [21] considering productivity, selectivity and yield and the oxidation of pure terephthalic acid [22] were optimized using an improved MODE, respectively. The results show that MODE can cover a wider range and a better spread compared to NSGA-II. Additionally, two MODE algorithms were implemented in the problem of maximizing ethylene and propylene yield in naphtha crack unit [23]. In particular, dynamic optimization problems, i.e., problems modeled by a set of non-linear differential and algebraic equations (DAE), are quite challenging. Gujarathi et al. [24] demonstrated the usability of MODE in solid state fermentation processes, with the model formulated as DAE. Also, control vector parameterization (CVP) has been shown to be an effective and efficient method to solve such DAE problems by discretizing the control variables [25]. This work combines CVP to optimize chemical and biochemical processes related to either space or time. Furthermore, a constrained complex bi-objective problem including only

steady-state variables is reformulated as a tri-objective problem to develop more realistic solutions.

Overall, the main contributions of the proposed SA-MODDE can be summarized as follows:

(1) Each component has been well-designed to reach its full potential. Pareto information is fully utilized in the parental selection, parameter setting and survival selection, even in constraint handling. Also, self-adaptive strategies of F and CR are presented based on both prior knowledges of current individual and feedback information of previous promising solutions. The effectiveness of the strategy is evaluated by comparing it with three fixed parameter combinations. Thus, only generic control parameters such as the population size (NP) and termination criterion are required. Besides, a performance-based termination criterion is incorporated into the SA-MODDE as an alternative to an arbitrary specified maximum number of generation or function evaluation.
(2) The performance is firstly investigated through the widely-used ZDT, DTLZ test suites and six constrained test instances [26–28]. Credible and thorough comparisons are conducted with 5 state-of-the-art peer competitor algorithms on two performance metrics (PMs). Then, two optimal control problems and a constrained steady-state (bio) chemical problem are employed to test the ability to track complex practical problems. The results show that SA-MODDE performs well on various types of MOO problems in terms of proximity and diversity and provides informative trade-off solutions for decision-makers.

The remainder of the paper is organized as follows. Section 2 expounds on each component of the proposed SA-MODDE in detail. Section 3 carries out extensive numerical experiments using two PMs. In Section 4, the use of the proposed SA-MODDE on three chemical and biochemical processes are discussed. Finally, conclusions are drawn in Section 5.

2. The proposed SA-MODDE algorithm

Algorithm S2 gives the main procedure of the proposed SA-MODDE algorithm. Firstly, the Pareto information of the initial population is provided by fast non-dominated sorting and crowding entropy assignment before the evolutionary iteration precedes. Then, during the mutation operation, the parents, i.e., the vectors participating in the mutation operator, are ranked and selected according to their dominance relationship. Besides, "rand/1" and "rand/2" are selected as mutation strategies with equal probability. Next, the binomial crossover operation is executed and the trial individual is generated. The values of F and CR of each individual are allocated by proposed parameter adaptation schemes. In the selection operation, which is crucial to maintain population diversity, substitution and truncation mechanisms are applied in combination. Finally, a simple dominance filtering operation is performed and the final solution set is obtained. The following subsections elaborate on each component of SA-MODDE.

2.1. Fast non-dominated sorting and crowding entropy assignment

Fast non-dominated sorting approach was proposed by Deb et al. [3], with two entities need to be calculated for each solution: (1) the number of solutions that dominate the ith solution, denoted as n_i; and (2) the set of solutions that dominated by the ith solution, named SS_i. Firstly, the solutions with $n_i = 0$ forms the first non-dominated front. Then, for the ith solution of the first front, each member q in SS_i is visited and the corresponding n_q value is reduced by one. Solution q will be collected in the second non-dominated front when $n_q = 0$. Next, for each member of the second front, the above procedure is repeated to make up the third front. The process loop continues until all fronts are identified.

The order of solutions in the same non-dominated front is usually determined through crowding degree assignment. The crowding distance is the universal crowding degree estimation measure [3]. It calculates the sum of the average distances between two solutions adjacent to the ith solution along with each objective, reflecting the solution density around the specific solution. On its basis, Wang et al. [14] introduced the crowding entropy (CE) concept by considering the distribution of the solution. The solutions in the same non-dominated front are firstly sorted in ascending or descending order by the function values of any objective. Then, the boundary solutions, namely, solutions with the largest or the smallest function values in any dimension, are set to an infinite CE value. Next, the calculation formulas of ith intermediate solution are defined as follows.

$$CE_i = \sum_{j=1}^{M} (c_{ij} \cdot E_{ij}) / (f_j^{max} - f_j^{min}) \tag{1}$$

$$E_{ij} = -\left[pl_{ij} \log_2(pl_{ij}) + pu_{ij} \log_2(pu_{ij}) \right] \tag{2}$$

$$pl_{ij} = \frac{dl_{ij}}{c_{ij}} \tag{3}$$

$$pu_{ij} = \frac{du_{ij}}{c_{ij}} \tag{4}$$

$$c_{ij} = dl_{ij} + du_{ij} \tag{5}$$

where E_{ij} is the distribution entropy of the ith solution along with the jth objective function, and the parameters f_j^{max} and f_j^{min} are the maximum and minimum values of the jth objective function; M is the number of objective functions; dl_{ij} and du_{ij} are the Euclidean distances of the ith solution to its lower and upper adjacent solution along the jth objective function, respectively. If a point locates in the middle of its two neighbors, then $pl_{ij} = pu_{ij} = 0.5$ and $E_{ij} = 1$, and it is regarded as the best distribution point.

In **Algorithm S2**, the initial population P^0 is first arranged in ascending order of the non-dominated front, and individuals in the same front are then sorted in descending order according to the CE values to obtain the new population P^1.

2.2. Parental selection and mutation strategies

The individuals acting as parents, denoted as X_p, are usually randomly selected from the whole population. This way is conducive to global search but affects the convergence rate. Abbass et al. [29] selected parents only from among non-dominated solutions based on the recognition that good parents always generate good offspring. However, this approach may in turn undermine global search capability. Chen et al. [30] calculated selection probability for each individual and found that the individual with high selection probability was more likely to be a parent. Although the method accelerated convergence while keeping global exploratory, it required a high computational complexity. Our work puts forward a simpler strategy: Several distinct individuals are randomly picked from the entire population to form parents, and then their own ranking is allocated according to their index in P^1. And the smaller the index value, the higher the ranking. The parent with the highest ranking is chosen as the base vector X_{pb}, and the parent with the lowest ranking is chosen as the terminal vector X_{pt}. Taking a two-dimensional problem as an example, the generation processes of mutation vector V_i with and

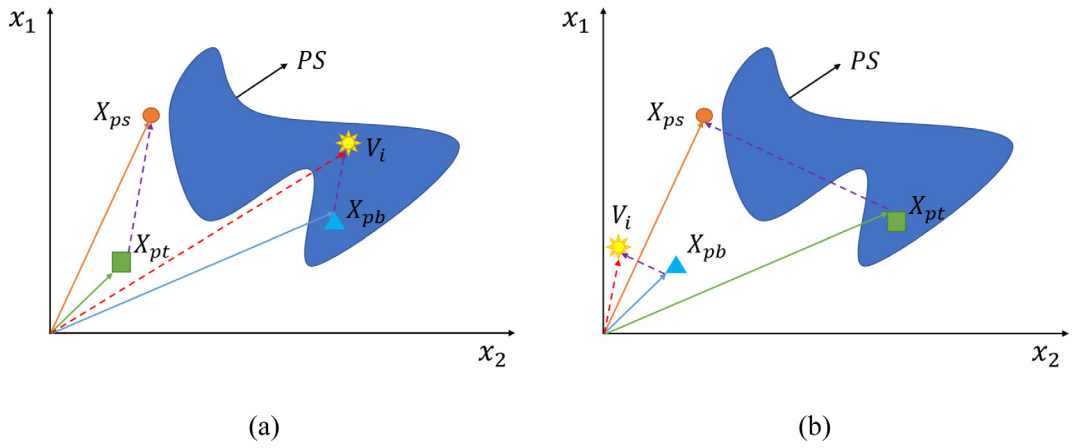


Fig. 1. The generation process of mutation vector V_i: (a) with parents ranking; (b) without parents ranking.

without parents ranking are illustrated in Fig. 1. As can be seen, under the guidance of the good base vector and the favorable direction of the difference vector $(X_{ps} - X_{pt})$, there are more opportunities to produce fine offspring. Switching X_{pb} and X_{pt} while holding individuals positions constant results in V_i is far from the Pareto optimal set (PS). Thus, the proposed parent ranking strategy not only facilitates the propagation of good information to the offspring without extra computational cost but also ensures powerful global exploration capabilities due to that there is no over-mine around specific vectors.

Multi mutation strategies have been recommended to solve MOO problems since no single strategy outperforms all others in the evolutionary process [18]. Several DE mutation strategies exhibit their own unique search characteristics and have different performance on diverse problems. Of these, "rand/1" and "rand/2" are the two most frequently used strategies, as appeared in **Algorithm S2**. Compared to "rand/1", "rand/2" adds a difference vector. This results in a better perturbation and provides more search directions, but may degrade search efficiency. In this work, the two strategies complement each other to exert different advantages and are employed by each individual with equal probability.

2.3. Survival selection

After generating the ith trial vector U_i, survival selection is executed to update the population. The survival schemes can be classified into three categories as summarized by Cheng et al. [31]. The first category conducts one-to-one selection. The replacement is performed when one solution dominates the other. When two solutions are non-dominated, one solution is randomly chosen or the one with less crowding degree is preferred. The second category implements a $(\lambda+\mu)$ design. In other words, the original population and the offspring population are merged, and then the combined population is truncated into the desired size. The third category considers the above two schemes synthetically, and its performance is better due to the hybridization effect. That is, the one-to-one strategy will be applied when two solutions are comparable, or the $(\lambda + \mu)$ procedure will be executed on the contrary.

The hybrid selection scheme is adopted in SA-MODDE, as shown in **Algorithm S3**, which includes three steps:

1. If U_i dominates target vector X_i, U_i will replace X_i. Then, the new population is reordered after U_i joins. At this time, it is unnecessary to recalculate the Pareto dominance relationships among all individuals, thanks to the original population that was sorted well in advance. In the worst case, U_i will be compared with all other individuals to obtain new non-dominated front classification, and then crowding entropy assignment will be recalculated only for the fronts where individuals' situations have changed.
2. If X_i dominates U_i, U_i is discarded. The original population has no change.
3. Otherwise, U_i is added to the population. The united population has $(NP + 1)$ individuals and is reordered after U_i joins, which is similar to step 1. Then, it is truncated at the last individual to keep size to be NP.

It is also noted that the F_i and CR_i values are collected into the set F_pool and CR_pool when X_i cannot dominate U_i, respectively, as listed in **line 4 and 13 in Algorithm S3**. The medians of the two sets are considered to be the best control parameter of the current evolution process, named F_{best} and CR_{best}, which can provide feedback information for parameter adaptation.

2.4. Parameter adaptation

The performances of DE-based algorithms are sensitive to the setting of F and CR. F changes the search step size and CR determines how much information U_i will inherit from X_i. Therefore, they affect the convergence speed and population diversity significantly [19]. In SA-MODDE, the parameters of each individual are dynamically adjusted by taking into account both prior and posteriori information during the evolutionary process. The parameter adaptation schemes are given in **Algorithm S4** and include the following three aspects.

1. Data normalization. Each objective function value in the obtained solution set is mapped to the range of $0 \sim 1$.
2. The self-adaptive strategy of F. F controls the scaling degree of the disturbance of the difference vector $(X_{ps} - X_{pt})$ to the base vector X_{pb}. When X_{ps} and X_{pt} are close in the search space, the generated difference vector value is small, and F should take a larger value at this time, otherwise the disturbance is too small to play the role of mutation, as the generation process of V_{i1} shown in Fig. 2(a). On the contrary, when X_{ps} and X_{pt} are farther apart, the generated difference vector value is large, and F should take a small value to limit the amount of disturbance, otherwise, the mutation vector may cross the boundaries of the feasible region, as the generation process of V_{i2} reflected in Fig. 2(a). The concept has been successfully applied in a single-objective DE algorithm by evaluating the difference between fitness values of different individuals to roughly reflect their closeness in the search space [32]. Despite

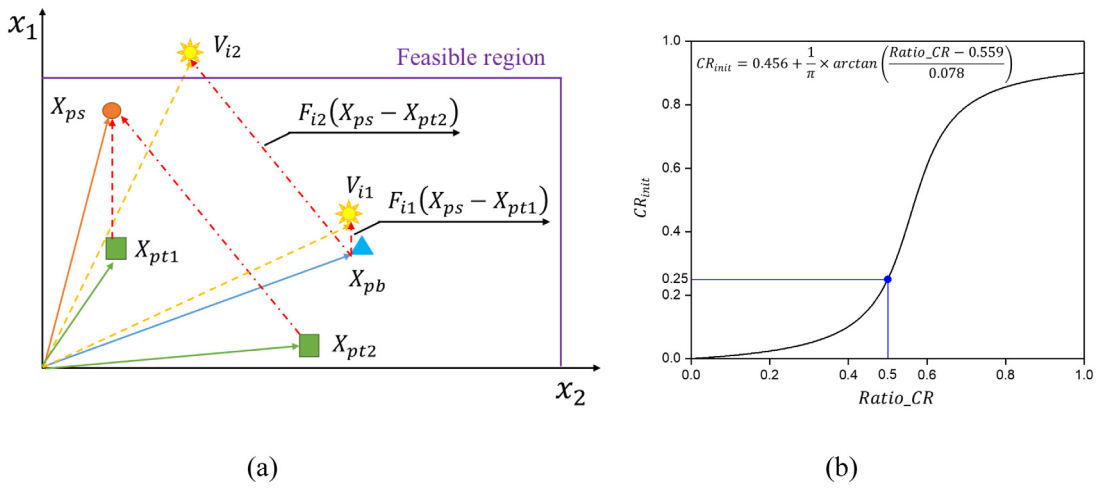


Fig. 2. Self-adaptive mechanism based on prior knowledge: (a) F; (b) CR.

that this mapping method is not always accurate, its computational cost is lower, especially for high-dimensional problems. Therefore, the setting of F first gets inspiration from the proximity of $f(X_{ps})$ and $f(X_{pt})$ in the solution space. The distance reference point is composed of the normalized maximum value on each objective function, called *nadir_point*. Then, the initial value of F_i is obtained through a linear relationship, denoted as F_{init}. Next, F_{init} approaches the most suitable control parameter F_{best} in the current population by a certain step and the final value of F_i is obtained. Furthermore, F_{best} is also used directly as F_i with the same probability. The upper and lower limit of F_i are 0.05 and 0.55, respectively.

3. The self-adaptive strategy of CR. The better X_i performs, the more its structure should be inherited, i.e., the smaller CR_i should be. The performance score of each X_i is obtained by summing its corresponding normalized objective function values. For minimization optimization problems, the X_i with the smallest score is considered to perform best. The score level of the ith individual in the score set is calculated in **line 20**, denoted as *Ratio_CR*. As shown in Fig. 2(b), CR_{init} is assigned within (0, 0.9) according to *Ratio_CR*. When *Ratio_CR* is below the average, the value of CR_{init} is small so that more individual information can be retained. In turn, when *Ratio_CR* is above the average, the value of CR_{init} increases rapidly, accelerating the elimination of poor individual structure. Then, CR_{init} will either learn experience from CR_{best} in a certain step, or be directly replaced by CR_{best}.

In addition, considering that F_{best} and CR_{best} have different ranges along with search stages, *F_pool* and *CR_pool* are emptied after each iteration, as shown in **line 5** of **Algorithm S2**. Thus, more appropriate and accurate values of the best control parameters can be obtained.

2.5. Constraint handling

Many real-world application problems are constrained besides bounds on decision variables. Three main categories of constraint handling approaches have been summarized [33]: (a) Penalty function method. This approach can be conveniently used in the formulations of various problems by adding penalty terms into the objective functions [34]. Thus, it is the most popular technique for constraint handling. However, its difficulty lies in choosing the suitable penalty factor. (b) Separation of objectives and constraints. A representative method is the feasibility approach proposed by Deb et al. [3]. It selects a feasible solution a prior over an infeasible solution based on the extent of constraint violation. (c) Repair algorithms. This approach converts the infeasible solution into a feasible or less feasible solution [35]. Recently, Chih [36,37] proposed two self-adaptive check and repair operators motivated by the fact that alternative pseudo-utility ratios must vary the approach directions in repairing infeasible solutions. They had been demonstrated to perform well on the multidimensional knapsack problem.

The feasibility approach is adopted in this work, as it can be seamlessly integrated with the proposed algorithm. The entire population is divided into a feasible group and an infeasible group based on whether or not individuals violate the constraints. The former is sorted according to the principle of non-dominance and crowding entropy, while the latter is first sorted according to the fewer number of constraint violations and then fewer total constraints violation. Four possible scenarios with corresponding handling methods are as follow.

1. If U_i and X_i are both feasible, the survival selection scheme introduced in Section 2.3 are applicable. The difference is that when they are non-dominated, U_i is accepted in the feasible group and the last individual of the infeasible group is discarded.
2. If U_i is feasible while X_i is infeasible, then U_i is added to the suitable position in the feasible group and X_i is abandoned from the infeasible group.
3. If U_i is infeasible while X_i is feasible, then U_i is refused.
4. If U_i and X_i are both infeasible, then U_i enters the infeasible group and the last individual of the reordered infeasible group is discarded.

2.6. Computational complexity

The proposed SA-MODDE algorithm is developed by integrating the above ingredients. The complexity of fast non-dominated sorting is $O(M \cdot NP^2)$, while the crowding entropy assignment needs $O(M \cdot NP \cdot \log_2 NP)$. In the survival selection step, the complexity is $O(M \cdot NP)$ when conducting dominant relationship comparison between U_i and X_i, and under the worst scene, reordering population requires $O(M \cdot NP^2)$. In the parameter adaptation step, the main computational complexity lies in data normalization and distance calculation of two solutions, which are $O(NP^2)$ and $O(M^2)$, respectively. Besides, parental selection and constraint handling do not introduce additional complexity. Therefore, the overall complexity of SA-MODDE is $O(M \cdot NP^2)$ at each iteration, which is the same as NSGA-II.

3. Numerical experiments

In the section, the performance of SA-MODDE is assessed through numerical experimental studies. Firstly, test problems and performance metrics are stated. Secondly, the proposed algorithm is compared with 5 state-of-the-art MOEAs. Thirdly, the effectiveness of parameter adaptation strategies is verified by comparing it with three fixed parameter settings. Finally, a termination criterion based on evolution performance is applied. See **Supplementary Materials** for relevant tables and figures.

3.1. Benchmark problems and performance metrics

12 unconstrained test instances, including 5 bi-objective MOPs from the ZDT test suite and 7 tri-objective MOPs from the DTLZ test suite, and 6 constrained benchmark functions are covered in experimental studies. They have a variety of characteristics, such as multi-local optimal fronts, non-uniform, and discontinuity, making the numerical testing comprehensive and reliable. The parentheses in the first column of **Table S1** list the number of the objective function (M) and decision variables (D) for each unconstrained test instance, and the parentheses in the first column of **Table S3** list M, D, and the number of constraints for each constrained problem.

Two quantitative assessment metrics, Inverted Generational Distance (IGD) and Spread (SP), are employed to reflect the performance of different algorithms. Assume that PF is a set of achieved non-dominated solutions and PF^* is a set of uniformly distributed solutions along the true Pareto front. IGD metric mirrors both the convergence and diversity of PF and a lower value mean a better approximation [12]. SP metric measures the solution distribution of PF and a lower value means a more even distribution [3]. Each objective in PF and PF^* is firstly normalized and the calculation formulas are defined as follows.

$$IGD = \frac{\sqrt{\sum_{i \in PF^*} d^2(i, PF)}}{|PF^*|} \tag{6}$$

where $d(i, PF)$ is the minimum Euclidian distance between ith solution in PF^* and solutions in PF and $|PF^*|$ is the number of solutions in PF^*.

$$SP = \frac{\sum_{m=1}^{M} d(e_m, PF) + \sum_{i=1}^{|PF|} |d_i - \bar{d}|}{\sum_{m=1}^{M} d(e_m, PF) + |PF| \cdot \bar{d}} \tag{7}$$

where $(e_1, e_2, \ldots, e_M)$ are m extreme solutions of PF^*; $d(e_m, PF)$ is the Euclidian distance between the extreme solution of mth objective in PF^* and its nearest solution in PF; d_i is the Euclidian distance between ith solution in PF and its nearest solution in the same set; $\bar{d}$ is the average of d_i; and $|PF|$ is the number of solutions in PF.

3.2. Peer algorithms and experimental settings

Four of the recent MOEAs with different structural features, including two DE variants, are chosen as peer competitors on unconstrained test problems. They outperformed several classical algorithms, such as GDE3, MOEA/D-DE, and MOPSO. A representative algorithm, NSGA-II, is chosen as a rival on constrained problems. A brief introduction of these five algorithms is given below.

1. MODE-RMO [30]: Multi-objective differential algorithm evolution with ranking-based mutation operator. The base and terminal vector in the mutation operator had a large probability to be selected from better vectors instead of random assignment.
2. MODE-PMSMO [38]: Multi-objective differential evolution with performance-metric-based self-adaptive mutation operator. The highlight of the algorithm was the automatic selection of a suitable mutation operator in different evolution stages.
3. NS-GWO [39]: Non-Dominated Sorting Gray Wolf Optimizer algorithm. The algorithm was proposed based on the natural hunting process of gray wolves, including searching, encircling, and attacking, and a leader selection mechanism was established.
4. INM-TLBO [10]: Multi-objective Individualized-Instruction Teaching Learning Based Optimization Algorithm. The algorithm assigned a specific teacher or interactive object for each individual and emphasized the guiding role of the non-dominated solution.
5. NSGA-II [3]: Non-dominated Sorting Genetic Algorithm II. The algorithm proposed a series of innovative concepts along with low computational complexity and has been successfully applied in many real-world constrained optimization problems [40].

Each algorithm performs 25,000 function evaluations and 30 independent runs for each test problem. NP is set to 100 for algorithms except for INM-TLBO, which is set to 20 due to its optimization mechanism. Other control parameters adopt recommended values in their original publications and all algorithms are implemented in the MATLAB platform. Additionally, the Wilcoxon rank sum test at a 0.05 significance level is applied to determine whether the experimental results of different algorithms are statistically significantly different. In **Table S1-6**, the symbols "+ ", "−", and "≈" denote that other algorithms perform better than, worse than, and similar to SA-MODDE, respectively, and the best results are bolded. At last, the average ranking method is used to give a comprehensive ranking of each algorithm, and the calculation formulas are as follows [41].

$$\mu_r = \frac{\sum_{i=1}^{n} R_i}{n} \tag{8}$$

$$\delta_r = \sqrt{\frac{\sum_{i=1}^{n} (R_i - \mu_r)^2}{n}} \tag{9}$$

where $R = \{R_1, R_2, \ldots, R_n\}$ is a rank set of one algorithm and n is the number of test problems. μ_r is the average ranking and δ_r is the standard deviation of the rankings.

3.3. Experimental results and discussions

The Median ($\bar{x}$) and interquartile range (IQR) of IGD metric on ZDT and DTLZ test suites of all algorithms are recorded in **Table S1**. SA-MODDE wins the first place in most problems, except for losing to INM-TLBO on ZDT6 and DTLZ7. The results of ZDT6, DTLZ1, and DTLZ7 are no statistically different from those of NS-GWO, MODE-PMSMO and MODE-RMO, and MODE-PMSMO, respectively. Moreover, the results of DTLZ3, DTLZ5, and DTLZ6 are several orders of magnitude lower than those of other competitors. In particular, regarding DTLZ3, which has many local optima, only the results of SA-MODDE are acceptable, while the data median and dispersion of other algorithms are not satisfactory, especially NS-GWO. To visually display the evolutionary behaviors of different algorithms, the curses of the median IGD value of 30 runs versus the number of function evaluations on each test instance are illustrated in **Fig. S1 (a–l)**. It can be observed that (1) INM-TLBO exhibits convergence speed advantage on the ZDT test suite, while SA-MODDE has fast declines on the DTLZ test suite except for DTLZ6 and DTLZ7. (2) SA-MODDE can quickly reach a stable lowest platform area on most problems

within 25,000 function evaluations. (3) The DE-based algorithms offer advantages in descending speed and solution quality on DTLZ1, DTLZ4, and DTLZ5, while exposing the weaknesses on these two aspects on ZDT6 and DTLZ7.

The statistical numerical results of *SP* metric on ZDT and DTLZ test suites are presented in **Table S2**. All competitors are inferior to SA-MODDE in the final average ranking. INM-TLBO provides the best results on ZDT6, DTLZ2, and DTLZ7, and obtains the runner-up again, while other competitors only have better or similar performance on DTLZ7. **Fig. S2 (a–l)** depicts the evolutionary curses of the median *SP* value of 30 runs versus the number of function evaluations. Compared with *IGD* curves, the *SP* curves oscillate significantly, especially the tri-objective problems. This is due to the characteristic that *SP* value is affected by the distribution uniformity of the currently obtained solutions. Besides, the *SP* curves of SA-MODDE drop rapidly within a certain abscissa interval on most problems, and the decline is larger than other algorithms, such as the interval [3,000–7,000] of ZDT1. In order to intuitively reflect the approximation and distribution of the solution set, the typical fronts obtained by each algorithm on ZDT6 and DTLZ3 after 25,000 function evaluations are exemplified in **Fig. S3 (a–b)**.

Table S3 shows the comparison results of NSGA-II and SA-MODDE in six constrained test problems. The first five are bi-objective problems with 2 or 6 constraints, and the last one is a five-objective problem with 7 constraints. In terms of statistical numerical results, SA-MODDE outperforms NSGA-II except for the *IGD* performance of OSY. **Fig. S1 (m–r)** and **Fig. S2 (m–r)** show the evolutionary behaviors corresponding to *IGD* and *SP*, respectively. SA-MODDE performs well on the first three problems, while the evolving speed on the fourth problem is slow along with a slight advantage in solution quality. For the fifth problem, which has five separated regions, SA-MODDE is prone to stagnation. For the sixth problem with 5 objectives, *IGD* curve of NSGA-II increases first and then stays steady. To explain the phenomena, **Fig. S3 (c–d)** plot the projections of the typical fronts realized by NSGA-II after 100 function evaluations, NSGA-II after 25,000 function evaluations, and SA-MODDE after 25,000 function evaluations on the space of objective 1 (f_1) and objective 2 (f_2), as well as objective 3 (f_3) and objective 4 (f_4), respectively. In the early stage of evolution, the solutions of NSGA-II in the feasible region (coved by blue dots) are relatively uniform distributed, while in the later stage of evolution, all solutions are concentrated on a certain boundary. This leads to an increase in the *IGD* curve instead of a decrease, indicating that NSGA-II deteriorates population diversity during the evolution process. On the contrary, SA-MODDE evenly covers the entire projection region including boundary points, which proves its ability to simultaneously maintain good proximity and diversity when dealing with constrained complex problems.

3.4. Effectiveness testing of parameter adaptation

To verify the effectiveness of proposed parameter adaptation strategies in SA-MODDE, three fixed control parameters are adopted for performance comparison, i.e., (1) Case1: $F = 0.3$, $CR = 0.1$; (2) Case2: $F = 0.5$, $CR = 0.5$; (3) Case3: $F = 0.7$, $CR = 0.9$, and other settings remain the same.

Table S4 gives the compared results in terms of *IGD*. Case 1 has 6 problems worse than, 12 problems similar to, and no problems better than SA-MODDE, respectively. Case 2 has a slight advantage over ZDT3 and CONSTR, is on the downside on 3 problems, and obtains comparable results on 13 problems. Case 3 scores higher on SRN and CONSTR, ties on 3 problems and loses on 13 problems. Finally, SA-MODDE prevails in the final average ranking. Notably, SA-MODDE obtains a *IGD* result with an accuracy level of $10^{-4}/(10^{-6})$ and $10^{-4}/(10^{-5})$ for ZDT4 and DTLZ3 respectively, while the best accuracy level of the results for three cases are $10^{-3}/(10^{-2})$ and 10^{-3} / (10^{-4}). These two problems include many local Pareto fronts, and are suitable for testing the ability of algorithms to handle multimodality problems. Conversely, on the test problems that SA-MODDE lost, its results are in the same order of magnitude as those obtained by the first-place. These demonstrate the critical role of parameter adaptation. In addition, the performance of three cases and SA-MODDE on DTLZ6 and BNH are statistically equivalent, indicating that these two problems are not sensitive to control parameters.

Table S5 gives the compared results in terms of *SP*. Among the 18 test problems, the performances of case 1 are weaker than SA-MODDE in 7 problems, and are equivalent in the others. Case 2 outperforms SA-MODDE on ZDT series (except ZDT4) and CONSTR, and achieves the same ranking on 9 problems and poor ranking on 4 problems. Case 3 yields better results on ZDT1 and 4 constrained test problems, comparable results on 7 problems, and worse results on 6 problems. Finally, despite the μ_r value of case 2 is the same as SA-MODDE, the latter is ranked first by virtue of the lower δ_r value.

There is no doubt that a certain of test problems perform well under some specific parameter combinations. However, SA-MODDE enables the selection of control parameters less dependent on the type of optimization problems, and consistently yields reliable and high-quality results in terms of convergence and diversity.

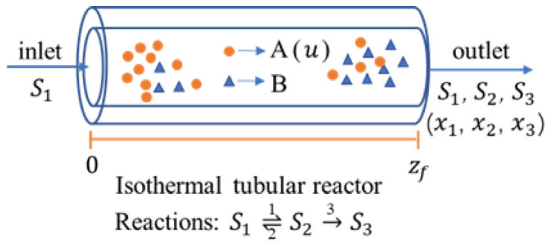
3.5. Performance-based termination criterion

As shown in **Fig. S1** and **S2**, before the evolution processes reach the maximum number of function evaluation (MFE), *IGD* values of SA-MODDE no longer decrease significantly on most problems, and *SP* values fluctuate in a narrow range. This suggests that the current stopping condition overuse computational resources. Therefore, it is necessary to adopt a more reliable and efficient termination criterion. Recently, performance-based termination criteria have been growing studied, which can stop search progress in time by monitoring the improvement of PMs. Sharma and Rangaiah [42] screened out two suitable PMs, namely, modified generational distance (GD_m) and modified spread (SP_m), and proposed the chi-squared-test based termination criterion (CSTC). Wong et al. [43] successfully applied a steady-state detection termination criterion (SSDTC) in heat exchangers design problems. Rangaiah et al. [44] evaluated the performance of these two criteria through several chemical processes and found that CSTC was more reliable and time-saving over SSDTC. Concerning the outstanding performance of CSTC, it is adopted in the work, with the difference that modified *IGD* and *SP*, i.e., IGD_m and SP_m, are chosen as PMs. Firstly, PMs are calculated using the non-dominated solution set in the previous and current generations. Then, a statistical verification of PM variations over the latest γ generations is required. The related calculation formulas are as follows.

$$IGD_m = \frac{\sqrt{\sum_{i\in N_{curr}} d^2_{i,IGD_m}}}{N_{curr}} \tag{10}$$

where d_{i,IGD_m} is the minimum Euclidian distance between ith solution in the current generation and solutions in the previous generation, and N_{curr} is the number of non-dominated solutions in the current generation.

$$SP_m = \frac{\sum_{i=1}^{N_{curr}} |d_{i,SP_m} - \bar{d}_m|}{N_{curr} \cdot \bar{d}_m} \tag{11}$$



$$Max\ f_1 = 1 - x_1(z_f) - x_2(z_f)$$

$$Min\ f_2 = \int_0^{z_f} u(z)dz$$

Subject to:

$$\frac{dx_1}{dz} = u(z)[k_2 x_2(z) - k_1 x_1(z)]$$

$$\frac{dx_2}{dz} = -u(z)[k_2 x_2(z) - k_1 x_1(z)] - [1 - u(z)]k_3 x_2(z)$$

$$0 \le u(z) \le 1$$

$$x(0) = [1\ 0]^T,\ k = [1\ 10\ 1]^T$$

$$0 \le z \le z_f, z_f = 1$$

Fig. 3. The schematic diagram and model equation of catalysts mixing process. Where x_1 and x_2 (state variables) are the concentrations of S_1 and S_2, respectively. u (control variable) is the fraction of catalyst A, and z denotes the spatial coordinate. k_1, k_2, k_3 are the kinetic factors of reactions 1–3. The rates of reaction 1 and 2 are affected by catalyst A, while reaction 3 does not occur unless catalyst B is present.

where d_{i,SP_m} is the Euclidian distance between ith solution in the current generation and its nearest solution in the same set; $\overline{d}_m$ is the average of d_{i,SP_m}.

$$Chi(PM) = \frac{Variance\,[PM_1, PM_2, \ldots, PM_\lambda]\,(\gamma - 1)}{\delta_{PM}^2} \tag{12}$$

$$P(PM) = \chi^2\,[Chi(PM), (\lambda - 1)] \tag{13}$$

where δ_{PM} is the user-defined tolerance value for standard deviation of PM, and $P(PM)$ is the probability that χ^2-test supports the hypothesis that the variance of PM is lower than δ_{PM}^2. A value of 10 is used for γ. The iterative search will stop if each $P(PM)$ exceeds 99% simultaneously at a certain generation. Moreover, the stopping condition based on MFE is also used, which acts as a guarantee to avoid indefinite looping.

The setting of δ_{IGD_m} and δ_{SP_m} needs to consider a balance between solution quality and computational expense. The small values will produce better results but will consume a lot of time, while large values will reduce computational costs but may cause premature stops. Here, $\delta_{IGD_m} = 0.0002$ and $\delta_{SP_m} = 0.05$ are recommended for bi-objective problems and these values are 0.0008 and 0.02 for problems with higher objective numbers. **Table S6** summarizes the comparison results applying CSTC and MFE on 18 test instances. As expected, the quality of the solutions obtained by the former is slightly inferior to the latter on most problems. Remarkably, there is no compromise for CSTC on the *IGD* values of 2 problems and the *SP* values of 6 problems. Besides, the median number of function evaluations required by CSTC are all located at the beginning of platform areas in **Fig. S1**. This indicates that adopted δ_{PM} values are suitable to stop the search timely without premature or overripe. In short, CSTC is well incorporated into the SA-MODDE framework, avoiding the waste of computing resources caused by arbitrarily specifying the maximum number of generation or function evaluation.

4. Chemical engineering processes optimization

Three process applications taken from literature, namely, catalysts mixing policy, Lee–Ramirez bioreactor, and alkylation process, are optimized by SA-MODDE to verify its performance in practical engineering problems. The first two problems are unconstrained optimal control problems, and the third is a constrained steady-state optimization problem. The number of independent runs is setting to 30 for all problems. Process model equations and optimization results are listed in **Supplementary Materials.**

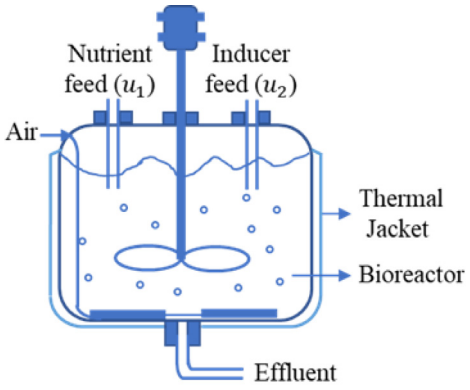
4.1. Catalysts mixing policy

The problem studies optimal mixing policies of two catalysts packed in a plug-flow reactor. These catalysts promote a series of reactions including one reversible and one irreversible, that is, $S_1 \Longleftrightarrow S_2 \rightarrow S_3$. The optimization objectives are maximizing the conversion of S_3 and minimizing the consumption of expensive catalyst A. The schematic diagram and model equation [45,46] are depicted in Fig. 3.

In order to solve and optimize the problem, the spatial interval is divided into 10 stages of equal length, and NP and MFE are set to 100 and 10,000. The typical PFs obtained by SA-MODDE under CSTC and MFE are plotted in **Fig. S4 (a)**. It can be observed that the non-dominated solutions under CSTC are close enough to that under MFE, except for an extreme solution P_0, while the median of the number of function evaluations under CSTC is only 4850. In addition, the conflicting relationship between the two objectives is clearly visible. In terms of maximum value of f_1, i.e., the abscissa value of P_0 under MFE or P_1 under CSTC, the median value of 30 runs is 0.047928 and 0.047837, respectively. Moreover, gray relational analysis (GRA), as a decision-making approach without objectives weight or other user inputs [47], is employed to select the appropriate solution. **Fig. S6 (c–d)** illustrates the control trajectories of boundary points (P_1 and P_3) and the recommended point P_2 under CSTC. The common thread is that catalyst A should be loaded more at the reactor inlet.

4.2. Lee–Ramirez bioreactor

Lee and Ramirez [48] presented a strong nonlinear optimal control problem for a fed-batch reactor using recombinant bacteria to induce foreign protein production, containing 7 state variables and 2 control variables. The objectives are maximizing



$$Max\ f_1 = x_1(t_f)x_4(t_f)$$

$$Min\ f_2 = \int_0^{t_f} u_2(t)dt$$

Subject to:

$$\frac{dx_1}{dt} = u_1 + u_2$$

$$\frac{dx_2}{dt} = \mu x_2 - \frac{u_1 + u_2}{x_1}x_2$$

$$\frac{dx_3}{dt} = C_{nf}\frac{u_1}{x_1} - \frac{u_1 + u_2}{x_1}x_3 - Y^{-1}\mu x_2$$

$$\frac{dx_4}{dt} = R_{fp}x_2 - \frac{u_1 + u_2}{x_1}x_4$$

$$\frac{dx_5}{dt} = C_{if}\frac{u_2}{x_1} - \frac{u_1 + u_2}{x_1}x_5$$

$$\frac{dx_6}{dt} = -k_1 x_6$$

$$\frac{dx_7}{dt} = k_2(1 - x_7)$$

$$\mu = \frac{x_3}{14.35 + x_3 + {x_3}^2/111.5}\left(x_6 + \frac{0.22x_7}{0.22 + x_5}\right)$$

$$R_{fp} = \frac{0.233x_3(0.0005 + x_3)}{(14.35 + x_3 + {x_3}^2/111.5)(0.22 + x_5)}$$

$$k_1 = k_2 = \frac{0.09x_5}{0.034 + x_5}$$

$$x(0) = [1\ \ 0.1\ \ 40\ \ 0\ \ 0\ \ 1\ \ 0]^{\mathrm{T}}$$

$$0 \le u_1, u_2 \le 1,\ t_0 = 0, t_f = 10$$

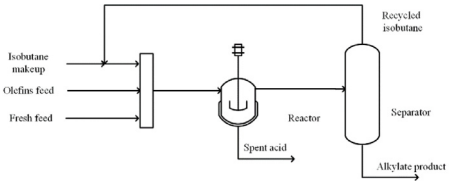
$$C_{nf} = 100.0, C_{if} = 4.0, Y = 0.51$$

Fig. 4. The schematic diagram and model equation of Lee–Ramirez bioreactor. Where $x_i(t)$ $(i = 1, 2, \ldots, 7)$ are state variables, indicating the reactor volume, cell density, nutrient concentration, foreign protein concentration, inducer concentration, the inducer shock and inducer recovery factors on cell growth rate, respectively. u_1 and u_2 are control variables, representing nutrient feeding rate and inducer feeding rate. C_{nf} and C_{if} are concentration of nutrient and inducer. μ is the specific growth rate, R_{fp} is the foreign protein production rate, and Y is the growth yield coefficient. k_1, k_2, k_3 are the shock and recovery parameters, respectively.

the yield of foreign protein and minimizing the consumption of inducer. The schematic diagram and model equation are shown in Fig. 4.

The time interval is partitioned into 10 stages of equal length, and *NP* of 100 and MFE of 30,000 are set. A visual comparison result for non-dominated solutions obtained by SA-MODDE under CSCT and MFE are depicted in **Fig. S5 (a)**. The results obtained by CSTC in a lower computation cost are in good agreement with those of MFE, except that some extreme solutions are not covered. When only f_1 is considered, the medians of maximum values achieved by MFE and CSTC are 6.1191(P_0) and 6.0859 (P_1), respectively. Also, there is a trade-off between two objectives and P_2 is suggested to decision-makers by GRA. The control trajectories of P_1, P_2, and P_3 are plotted in **Fig. S5 (b–d)**, indicating that the nutrient and inducer feeding rates at the onset of reactions should remain low.

In order to verify the feasibility and robustness of SA-MODDE, **Table S7** and **S8** present a comparison of the maximum of f_1 with earlier studies and four peer algorithms with respect to catalysts mixing and Lee–Ramirez bioreactor problems, respectively. In fact, since there are differences in terms of optimization type, the maximum number of function evaluation, even discretization level, earlier studies involving single-objective optimization are regarded as references. Four multi-objective peer algorithms, as well as the current work, are compared fairly under the same number of function evaluations and 30 independent runs. For catalyst mixing problem, the best conversion value of S_3 obtained by SA-MODDE is sufficiently close to the theoretical solution [49], and the Wilcoxon rank sum test shows that NS-GWO has a slight advantage in solution quality. For Lee–Ramirez bioreactor problem, INM-TLBO wins the first place and SA-MODDE second in terms of the yield of foreign protein. Remarkably, NS-GWO and INM-TLBO were the worst performers on the Lee–Ramirez bioreactor and catalyst mixing problems, respectively. Furthermore, as expected, the best values of multi-objective optimization



$$Max\ f_1 = 0.063x_4x_7 - 5.04x_1 - 0.035x_2 - 10x_3 - 3.36x_5\ [\$/day]$$

$$Max\ f_2 = x_7$$

$$Min\ f_3 = x_2\ [barrels/day]$$

Subject to:

$$0 \le [x_2 \equiv x_1x_8 - x_5] \le 16000$$

$$0 \le [x_3 \equiv 0.001x_4x_6x_9\ /\ (98 - x_5)] \le 120$$

$$0 \le [x_4 \equiv x_1(1.12 + 0.13167x_8 - 0.0066667{x_8}^2)] \le 5000$$

$$0 \le [x_5 \equiv 1.22x_4 - x_1] \le 2000$$

$$85 \le [x_6 \equiv 89 + (x_7 - 86.35 - 1.098x_8 + 0.038{x_8}^2)\ /\ 0.325] \le 93$$

$$1.2 \le [x_9 \equiv 35.82 - 0.222x_{10}] \le 4$$

$$145 \le [x_{10} \equiv -133 + 3x_5] \le 162$$

$$0 \le x_1 \le 2000$$

$$90 \le x_7 \le 95$$

$$3 \le x_8 \le 12$$

Fig. 5. The schematic diagram and model equation of the alkylation process. Where $x_i(t)$ $(i = 1, 2, \ldots, 10)$ are the olefin feed rate (barrels/day), isobutane recycle rate (barrels/day), acid addition rate ($10^3\times$ pounds/day), alkylate production rate (barrels/day), isobutane feed rate (barrels/day), spent acid strength (wt%), octane number, isobutane to olefins ratio, acid dilution factor, and F-4 performance number, respectively.

are inferior to that of single-objective optimization, due to the fact that the former requires more effort to explore the entire Pareto fronts, whereas the latter is exploited around the optimal solution.

4.3. Alkylation process

Light olefins react with isobutane under the catalysis of acid to produce alkylate products, before mixing with refinery products to increase the octane number. The process consists of the reactor and fractionator modules as well as recycle streams. Sharma and Rangaiah [42] discussed two different bi-objective problems for the process. Here, in order to test SA-MODDE's ability to deal with complex engineering problems, a tri-objective problem is considered for the first time, i.e., maximum profit, maximum octane number, and minimum isobutane recycling. Moreover, the results obtained by SA-MODDE are compared with that of NSGA-II. The schematic diagram and model equation are shown in Fig. 5, including 10 variables and 7 inequality constraints [42,50].

Fig. S6 (a–b) shows the front and side views of typical *PF*s achieved by SA-MODDE and NSGA-II, with *NP* of 100 and MFE of 50,000. Apparently, the solutions of NSGA-II account for only a fraction of that of SA-MODDE. This once again proves SA-MODDE's strengths in upholding proximity and diversity. From the projections on f_1-f_2 and f_1-f_3 planes (**Fig. S6 (c–d)**), the correlation between profit and isobutane recycling can be approximated by a linear relationship, while the correlation between profit and octane number is irregular. However, profit and octane number existed a strong linear correlation when isobutane recycling was not considered [42]. This shows that more accurate and realistic solutions can be obtained considering three objectives at the same time. **Fig. S7** illustrates the results of SA-MODDE under different termination criteria. The results of CSTC are close enough to that of MFE and consume less computational resources. In addition, when the non-dominated solutions obtained by SA-MODDE under MFE are regarded as the known Pareto-optimal fronts, median *IGD* and *SP* values of 30 runs for NSGA-II are 0.0250 and 0.6400, and these values are 0.0069 and 0.4019 for SA-MODDE under CSTC.

5. Conclusion

In this work, each component of SA-MODDE is designed in the following the ways: (1) Crowding entropy rather than crowding

distance is used to improve the uniformity and dispersion of non-dominated solutions. (2) Parents selection scheme based on the Pareto dominance relationship provides good guidance information for offspring generation. (3) Multi mutation strategies cooper with each other to balance search capability and diversity maintenance. (4) Parameter self-adaptive strategies of F and CR enable the algorithm less sensitive to the types of optimization problems and their effectiveness is experimentally quantified. (5) A hybrid survival selection is implemented to update the population, avoiding the need for an external elitist archive. (6) The feasibility approach for constraint handling is fine-tuned to accelerate the phase-out of infeasible individuals. (7) A performance-based termination criterion greatly improves computing efficiency with slightly compromise solution quality. Furthermore, low computational complexity enhances the availability and efficiency of the SA-MODDE algorithm.

18 benchmark functions including various types of MOPs, make the numerical experimental results more comprehensive and convincing. Judging from the overall statistical performance of 30 runs and the average evolutionary behavior of a single run, SA-MODDE exhibits stronger exploration and exploitation capabilities with respect to five powerful competitors. Additionally, the test results also revealed that no algorithms can completely conquer others due to the unique advantages of different frameworks on specific problems. Two optimal control problems and a constrained steady-state problem are studied to further examine the applicability of SA-MODDE in tracking chemical engineering problems. The results show that SA-MODDE is efficient to locate and obtain a set of diversified trade-off solutions for decision-makers to refer to.

CRediT authorship contribution statement

Xiaodong Zhang: Conceptualization, Methodology, Software, Validation, Visualization, Writing - original draft. **Lu Jin:** Validation, Data curation. **Chengtian Cui:** Writing - review & editing. **Jinsheng Sun:** Supervision, Reviewing and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2021.107317.

References

[1] I.T. Cameron, S. Engell, C. Georgakis, N. Asprion, D. Bonvin, F. Gao, B.R. Young, Education in process systems engineering: Why it matters more than ever and how it can be structured, Comput. Chem. Eng. 126 (2019) 102–112, http://dx.doi.org/10.1016/j.compchemeng.2019.03.036.

[2] S. Avraamidou, S.G. Baratsas, Y. Tian, E.N. Pistikopoulos, Circular economy-A challenge and an opportunity for process systems engineering, Comput. Chem. Eng. 133 (2020) 106629, http://dx.doi.org/10.1016/j.compchemeng.2019.106629.

[3] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comp. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[4] S. Kukkonen, J. Lampinen, GDE3: The third evolution step of generalized differential evolution, in: 2005 IEEE Congress on Evolutionary Computation, vol. 1, IEEE, 2005, pp. 443–450, http://dx.doi.org/10.1109/CeC.2005.1554717, September.

[5] C.C. Coello, M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation. CeC'02 (Cat. No. 02TH8600), vol. 2, IEEE, 2002, pp. 1051–1056, http://dx.doi.org/10.1109/CeC.2002.1004388, May.

[6] G.P. Rangaiah, A.B. Petriciolet, Multi-Objective Optimization in Chemical Engineering, John Wiley & Sons, Ltd., Publication, ISBN: 9781118341704, 2013.

[7] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, TIK-report, 103, 2001, http://dx.doi.org/10.3929/ethz-a-004284029.

[8] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comp. 11 (6) (2007) 712–731, http://dx.doi.org/10.1109/TEVC.2007.892759.

[9] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints, IEEE Trans. Evol. Comp. 18 (4) (2013) 577–601, http://dx.doi.org/10.1109/TEVC.2013.2281535.

[10] D. Yu, J. Hong, J. Zhang, Q. Niu, Multi-objective individualized-instruction teaching-learning-based optimization algorithm, Appl. Soft Comput. 62 (2018) 288–314, http://dx.doi.org/10.1016/j.asoc.2017.08.056.

[11] T. Tušar, B. Filipič, Differential evolution versus genetic algorithms in multiobjective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, Berlin, Heidelberg, ISBN: 9783540709275, 2007, pp. 257–271, March.

[12] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comp. 13 (2) (2008) 284–302, http://dx.doi.org/10.1109/TeVC.2008.925798.

[13] A. Qing, Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems, IEEE Trans. Geosci. Remote Sens. 44 (1) (2005) 116–125, http://dx.doi.org/10.1109/TGRS.2005.859347.

[14] Y.N. Wang, L.H. Wu, X.F. Yuan, Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure, Soft Comput. 14 (3) (2010) 193, http://dx.doi.org/10.1007/s00500-008-0394-9.

[15] L. Wu, Y. Wang, X. Yuan, Z. Chen, Multiobjective optimization of HeV fuel economy and emissions using the self-adaptive differential evolution algorithm, IEEE Trans. Veh. Technol. 60 (6) (2011) 2458–2470, http://dx.doi.org/10.1109/TVT.2011.2157186.

[16] P.A. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, IEEE Trans. Evol. Comp. 7 (2) (2003) 174–188, http://dx.doi.org/10.1109/TEVC.2003.810761.

[17] Q. Lin, Y. Ma, J. Chen, Q. Zhu, C.A.C. Coello, K.C. Wong, F. Chen, An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies, Inform. Sci. 430 (2018) 46–64, http://dx.doi.org/10.1016/j.ins.2017.11.030.

[18] B. Xu, X. Chen, X. Huang, L. Tao, A multistrategy-based multiobjective differential evolution for optimal control in chemical processes, Complexity 2018 (2018) http://dx.doi.org/10.1155/2018/2317860.

[19] Q. Lin, Q. Zhu, P. Huang, J. Chen, Z. Ming, J. Yu, A novel hybrid multi-objective immune algorithm with adaptive differential evolution, Comput. Oper. Res. 62 (2015) 95–111, http://dx.doi.org/10.1016/j.cor.2015.04.003.

[20] B.V. Babu, A.M. Gujarathi, Multi-objective differential evolution (MODE) for optimization of supply chain planning and management, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 2732–2739, http://dx.doi.org/10.1109/CEC.2007.4424816, September.

[21] A.M. Gujarathi, B.V. Babu, Multi-objective optimization of industrial styrene reactor: Adiabatic and pseudo-isothermal operation, Chem. Eng. Sci. 65 (6) (2010) 2009–2026, http://dx.doi.org/10.1016/j.ces.2009.11.041.

[22] A.M. Gujarathi, B.V. Babu, Improved multiobjective differential evolution (MODE) approach for purified terephthalic acid (PTA) oxidation process, Mater. Manuf. Process. 24 (3) (2009) 303–319, http://dx.doi.org/10.1080/10426910802679337.

[23] A.M. Gujarathi, A.H. Motagamwala, B.V. Babu, Multiobjective optimization of industrial naphtha cracker for production of ethylene and propylene, Mater. Manuf. Process. 28 (7) (2013) 803–810, http://dx.doi.org/10.1080/10426914.2012.746706.

[24] A.M. Gujarathi, A. Sadaphal, G.A. Bathe, Multi-objective optimization of solid state fermentation process, Mater. Manuf. Process. 30 (4) (2015) 511–519, http://dx.doi.org/10.1080/10426914.2014.984209.

[25] X. Chen, W. Du, H. Tianfield, R. Qi, W. He, F. Qian, Dynamic optimization of industrial processes with nonuniform discretization-based control vector parameterization, IEEE Trans. Autom. Sci. Eng. 11 (4) (2013) 1289–1299, http://dx.doi.org/10.1109/TASE.2013.2292582.

[26] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195, http://dx.doi.org/10.1162/106365600568202.

[27] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proceedings of the 2002 Congress on Evolutionary Computation. CeC'02 (Cat. No. 02TH8600), vol. 1, IEEE, 2002, pp. 825–830, http://dx.doi.org/10.1109/CeC.2002.1007032, May.

[28] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, 16, John Wiley & Sons, ISBN: 0-471-87339-X, 2001.

[29] H.A. Abbass, R. Sarker, C. Newton, PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems, in: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), vol. 2, IEEE, 2001, pp. 971–978, http://dx.doi.org/10.1109/CeC.2001.934295, May.

[30] X. Chen, W. Du, F. Qian, Multi-objective differential evolution with ranking-based mutation operator and its application in chemical process optimization, Chemometr. Intell. Lab. Syst. 136 (2014) 85–96, http://dx.doi.org/10.1016/j.chemolab.2014.05.007.

[31] J. Cheng, G.G. Yen, G. Zhang, A grid-based adaptive multi-objective differential evolution algorithm, Inform. Sci. 367 (2016) 890–908, http://dx.doi.org/10.1016/j.ins.2016.07.009.

[32] L. Wu, Y. Wang, S. Zhou, X. Yuan, Self-adapting control parameters modified differential evolution for trajectory planning of manipulators, J. Control Theory Appl. 5 (4) (2007) 365–373, http://dx.doi.org/10.1007/s11768-006-6178-9.

[33] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Engrg. 191 (11–12) (2002) 1245–1287, http://dx.doi.org/10.1016/S0045-7825(01)00323-1.

[34] M. Chih, C.J. Lin, M.S. Chern, T.Y. Ou, Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem, Appl. Math. Model. 38 (4) (2014) 1338–1350, http://dx.doi.org/10.1016/j.apm.2013.08.009.

[35] K. Harada, J. Sakuma, I. Ono, S. Kobayashi, Constraint-handling method for multi-objective function optimization: Pareto descent repair operator, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, Berlin, Heidelberg, 2007, pp. 156–170, http://dx.doi.org/10.1007/978-3-540-70928-2, March.

[36] M. Chih, Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem, Appl. Soft Comput. 26 (2015) 378–389, http://dx.doi.org/10.1016/j.asoc.2014.10.030.

[37] M. Chih, Three pseudo-utility ratio-inspired particle swarm optimization with local search for multidimensional knapsack problem, Swarm Evol. Comput. 39 (2018) 279–296, http://dx.doi.org/10.1016/j.swevo.2017.10.008.

[38] Q. Fan, W. Wang, X. Yan, Multi-objective differential evolution with performance-metric-based self-adaptive mutation operator for chemical and biochemical dynamic optimization problems, Appl. Soft Comput. 59 (2017) 33–44, http://dx.doi.org/10.1016/j.asoc.2017.05.044.

[39] P. Jangir, N. Jangir, A new Non-Dominated Sorting Grey Wolf Optimizer (NS-GWO) algorithm: Development and application to solve engineering designs and economic constrained emission dispatch problem with integration of wind power, Eng. Appl. Artif. Intell. 72 (2018) 449–467, http://dx.doi.org/10.1016/j.engappai.2018.04.018.

[40] G.P. Rangaiah, S. Sharma, B.K. Sreepathi, Multi-objective optimization for the design and operation of energy efficient chemical processes and power generation, Curr. Opin. Chem. Eng. 10 (2015) 49–62, http://dx.doi.org/10.1016/j.coche.2015.08.006.

[41] C.K. Chow, S.Y. Yuen, A multiobjective evolutionary algorithm that diversifies population by its density, IEEE Trans. Evol. Comp. 16 (2) (2011) 149–172, http://dx.doi.org/10.1109/TeVC.2010.2098411.

[42] S. Sharma, G.P. Rangaiah, An improved multi-objective differential evolution with a termination criterion for optimizing chemical processes, Comput. Chem. Eng. 56 (2013) 155–173, http://dx.doi.org/10.1016/j.compchemeng.2013.05.004.

[43] J.Y. Wong, S. Sharma, G.P. Rangaiah, Design of shell-and-tube heat exchangers for multiple objectives using elitist non-dominated sorting genetic algorithm with termination criteria, Appl. Therm. Eng. 93 (2016) 888–899, http://dx.doi.org/10.1016/j.applthermaleng.2015.10.055.

[44] G.P. Rangaiah, S. Sharma, H.W. Lin, Evaluation of two termination criteria in evolutionary algorithms for multi-objective optimization of complex chemical processes, Chem. Eng. Res. Des. 124 (2017) 58–65, http://dx.doi.org/10.1016/j.cherd.2017.05.030.

[45] D.J. Gunn, W.J. Thomas, Mass transport and chemical reaction in multifunctional catalyst systems, Chem. Eng. Sci. 20 (2) (1965) 89–100, http://dx.doi.org/10.1016/0009-2509(65)85002-3.

[46] P. Liu, G. Li, X. Liu, Z. Zhang, Novel non-uniform adaptive grid refinement control parameterization approach for biochemical processes optimization, Biochem. Eng. J. 111 (2016) 63–74, http://dx.doi.org/10.1016/j.bej.2016.03.006.

[47] J.D. Martinez-Morales, U. Pineda-Rico, E. Stevens-Navarro, Performance comparison between MADM algorithms for vertical handoff in 4G networks, in: 2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control, IEEE, 2010, pp. 309–314, http://dx.doi.org/10.1109/ICEEE.2010.5608646, September.

[48] J. Lee, W.F. Ramirez, Optimal fed-batch control of induced foreign protein production by recombinant bacteria, AIChE J. 40 (5) (1994) 899–907, http://dx.doi.org/10.1002/aic.690400516.

[49] G. Li, X. Liu, Comments on optimal use of mixed catalysts for two successive chemical reactions, J. Optim. Theory Appl. 165 (2) (2015) 678–692, http://dx.doi.org/10.1007/s10957-014-0641-4.

[50] G.P. Rangaiah, Multi-Objective Optimization: Techniques and Applications in Chemical Engineering, 1, World Scientific, ISBN: 9789812836519, 2009.