Iranian J. Math. Chem. 10 (2) July (2019) 103 − 126

Iranian Journal of Mathematical Chemistry

Journal homepage: ijmc.kashanu.ac.ir

# *Solving Multi-Objective Optimal Control Problems of Chemical Processes Using Hybrid Evolutionary Algorithm*

**GHOLAM HOSEIN ASKARIROBATI[1], AKBAR HASHEMI−BORZABADI[2,•] AND AGHILEH HEYDARI[3]**

[1]Department of Mathematics, Payame Noor University, P. O. Box 19395−3697, Tehran, Iran
[2]Department of Mathematics and Computer Science, Damghan University, Damghan, Iran
[3]Department of Mathematics, Payame Noor University, Tehran, Iran

**ARTICLE INFO**

**Article History:**

Received 26 June 2018
Accepted 19 September 2018
Published online 30 July 2019
Academic Editor: Saeed Masoum

**Keywords:**

Multi-objective optimal control
Pareto optimal frontier
Non-dominated sorting
Invasive weed optimization
Fed Batch Reactor

**ABSTRACT**

This paper applies an evolutionary optimization scheme, inspired by Multi-objective Invasive Weed Optimization (MOIWO) and Non-dominated Sorting (NS) strategies, to find approximate solutions for multi-objective optimal control problems (MOCPs). The desired control function may be subjected to severe changes over a period of time. In response to deficiency, the process of dispersal has been modified in the MOIWO. This modification will increase the explorative power of the weeds and reduces the search space gradually during the iteration process. The performance of the proposed algorithmis compared with conventional Non-dominated Sorting Genetic Algorithm (NSGA-II) and Non-dominated Sorting Invasive Weed Optimization (NSIWO) algorithm.



## 1. INTRODUCTION

In many real-world optimal control problems, there are several (possibly conflicting) objectives that need to be optimized simultaneously. Such control problems arise in many applications such as designing optimal reactor feed ingrates in (bio)chemical engineering, optimal power management of fuel cells in

•Corresponding Author (Email address: a.hashemi.bor@gmail.com)
DOI: 10.22052/ijmc.2018.137247.1370

electrical engineering, optimal robot paths in mechanical engineering and optimal rocket trajectories in aerospace engineering [17].

In multiobjective problems (MOPs) there are no exists one single optimal solution that is best with respect to all objective. So we have a set of solutions, in which an objective can not be improved without worsening at least one of the other objectives. This solution set is said to be the Pareto optimal (or non-dominated) solution set and its image in the objective space is usually called the Pareto optimal frontier.

There are two different strategies for generating a set of Pareto optimal solutions representing the entire Pareto optimal frontier: One-at-a-time strategy, and Simultaneous strategy. In the former, a multi-objective optimizer may be applied one at a time with the goal of finding one single Pareto optimal solution. Most classical generating multi-objective optimization methods use such an iterative scalarization scheme, such as weighted sum (WS) method [23].

The WS method is a commonly used scalarization technique which consists of assigning each objective function a weight coefficient and then optimizing the function obtained by summing up all the objective functions scaled by their weight coefficients so that only one solution can be obtained.The trade off surface is then determined by repeating this process with different settings of the weights. The drawback of this approach is that the solution of the problem is dependent on the choice of the relative weights assigned to different objectives [33].

Moreover, the main criticism of this method is that although there are results for convergence, an equal distribution of weights does not guarantee an even distribution of the points along the Pareto frontier. Also this method has disadvantage of missing the non-convex portions of a Pareto frontier [6]. In response to deficiencies in the WS method, several algorithms such as Normal Boundary Intersection (NBI) [6] and Normalized Normal Constraint (NNC) [22] have been proposed that use scalarization schemes providing an uniform distribution among solutions along the Pareto frontier.

Recently, Logist et al. [17−20] successfully combined these methods with direct optimal control methods such as orthogonal collocation [1,2], single and multiple shooting [3,15] to efficiently solve MOCPs.

In the simultaneous approach, several Pareto-optimal solutions are found in a single simulation run, thereby there is no need to perform a series of separate runs as in the case of classical optimization techniques. These algorithms are usually population based and exchange important information among population members.

In past two decades several nature-inspired meta-heuristics were introduced to find approximate solutions of multi-objective problems, such as the Multi-

Objective Genetic Algorithm (MOGA) [10], Vector Evaluated Genetic Algorithm (VEGA) [34], Niched Pareto Genetic Algorithm (NPGA) [9], Strength Pareto evolutionary algorithms (SPEA) [37], SPEA2 [38], Pareto Archive Evolution Strategy (PAES) [12], Multi-Objective Particle Swarm Optimization (MOPSO) [5], Multi-Objective Invasive Weed Optimization (MOIWO) [14], Non-Dominated Sorting Genetic Algorithm (NSGA) [35] and NSGA-II [8].

### 1.1. HISTORICAL DEVELOPEMENTS – A SHORT OVERVIEW

Evolutionary strategies are potentially able to cover the entire Pareto set and have been also successfully used to solve MOOCPs. For instance, sharker and modak used evolutionary strategy to solve two optimal control problems related to fed-batch bioreactors under NSGA-II framework [33]. Zhang et. al. introduce an iterative multi-objective particle swarm optimization based control vector parameterization is proposed for the state constrained chemical and biochemical engineering problems [40]. Sun et. al. proposed a hybrid improved genetic algorithm (HIGA) for solving dynamic optimization problems of chemical processes [36]. Patel and Padhiyar introduce a modified genetic algorithm using Box Complex method and used it to solve optimal control problems [28].

Evolutionary optimization algorithms are better able to be converted to a global solution than the traditional methods in complex optimization problems [27]. Some of their advantages are: (I) the objective functions gradient is not required; (II) they are not sensitive to initial guess of solution and (III) they usually do not get stuck in to a local optimum [25].

The purpose of this study is to find approximate solutions of MOCPs by using an evolutionary optimization strategy. This strategy is an improved version of Non-dominated Sorting Invasive Weed Optimization (INSIWO) [24]. The most important motivation of the proposed strategy is to improve the process of dispersal in order to increase the explorative power of the weeds which causes further promote the convergence rate and diversity solution along Pareto frontier.

The remainder of this study is organized as follows. In Section 2, mathematical formulations of general multi-objective optimal control problem are briefly introduced. The heuristic approaches for solving multi-objective optimization problems are described in Section 3. Section 4 describes the proposed algorithm. Lastly in section 5, the ability of proposed strategy is demonstrated with three practical multi-objective optimal control problems.

## 2. MULTI−OBJECTIVE OPTIMAL CONTROL PROBLEM

Mathematically, a general multi-objective optimal control problem contains a number of objectives to be minimized and (optional) constraints to be satisfied. In this case, a multi-objective optimal control problem consists of optimizing a vector of functions:

$$\mathrm{Opt}\big(\mathrm{J}(\mathrm{x},\mathrm{u})\big)=\big(\mathrm{J}_1(\mathrm{x},\mathrm{u}),\mathrm{J}_2(\mathrm{x},\mathrm{u}),\ldots,\mathrm{J}_m(\mathrm{x},\mathrm{u})\big) \quad (1)$$

$$\dot{\mathrm{x}}=\mathrm{f}(\mathrm{x}(\mathrm{t}),\mathrm{u}(\mathrm{t}),\mathrm{t}), \quad (2)$$

$$\mathrm{b}\big(\mathrm{x}(0)\big)=0, \quad (3)$$

$$\mathrm{b}_\mathrm{f}\big(\mathrm{x}(\mathrm{t}_\mathrm{f})\big)=0, \quad (4)$$

$$\mathrm{c}_\mathrm{p}(\mathrm{x}(\mathrm{t}),\mathrm{u}(\mathrm{t}),\mathrm{t})\leq 0, \quad (5)$$

$$\mathrm{c}_\mathrm{f}(\mathrm{x}(\mathrm{t}_\mathrm{f}),\mathrm{u}(\mathrm{t}_\mathrm{f}),\mathrm{t}_\mathrm{f})\leq 0, \quad (6)$$

where $\mathrm{J}_\mathrm{i}=\varphi(\mathrm{x}(\mathrm{t}_\mathrm{f}),\mathrm{t}_\mathrm{f})+\int_{\mathrm{t}_0}^{\mathrm{t}_\mathrm{f}}\mathrm{L}_\mathrm{i}\,(\mathrm{x}(\mathrm{t}),\mathrm{u}(\mathrm{t}),\mathrm{t})\mathrm{dt}$, $x \in R^n$ denotes the state variables and $u \in R^m$ is the control in a given time interval $[\mathrm{t}_0,\mathrm{t}_\mathrm{f}]$. The final time $t_f$ may be known or variable (free end time control problem). The functions $f: R^n \times R^m \times R \to R^n$, $\varphi: R^n \times R \to R$ and $L: R^n \times R^m \times R \to R^n$ are assumed to be sufficiently smooth on appropriate open sets. The initial and terminal boundary conditions are given by the vectors $b$ and $b_f$, respectively. The vectors $c_p$ and $c_f$ indicate path and final inequality constraints on the states and controls, respectively. The admissible set $\Omega \subseteq R^n \times R^m \times R$ is defined to be set of all feasible pair state and control $(\mathrm{x},\mathrm{u})$ that satisfy in Eq.(2−6).

In MOCPs are usually objectives in conflict with each other thus the concept of optimum is not well defined in this context, so it is difficult to have an admissible pair $\big(\mathrm{x}^*,\mathrm{u}^*\big)$ that optimizes all the objectives simultaneously. Therefore, the concept of Pareto optimality is used. The concept of optimality in single objective is not directly applicable in multi-objective optimization problems. For this reason a classification of the solutions is introduced in terms of Pareto optimality, according to the following definitions [40]. In terms of minimization of objective functions we have the following definitions:

**Definition 1.** An pair $\big(\mathrm{x}^*,\mathrm{u}^*\big)\in\Omega$ is Pareto optimal solution of the MOCP if and only if there is no other pair $(\bar{\mathrm{x}},\bar{\mathrm{u}})\in\Omega$, such that $J_i(\bar{\mathrm{x}},\bar{\mathrm{u}}) \leq J_i\big(\mathrm{x}^*,\mathrm{u}^*\big)$ for $i$ = 1, 2,...,$m$ and $\mathrm{J}_\mathrm{i}(\bar{\mathrm{x}},\bar{\mathrm{u}})<\mathrm{J}_\mathrm{i}\big(\mathrm{x}^*,\mathrm{u}^*\big)$ for at least one objective function $i$.

**Definition 2.** (Pareto dominance): A pair $\big(\mathrm{x}^*,\mathrm{u}^*\big) \in \Omega$ is said dominate another pair $(\bar{\mathrm{x}},\bar{\mathrm{u}})\in\Omega$, denoted by $(\bar{\mathrm{x}},\bar{\mathrm{u}})\prec\big(\mathrm{x}^*,\mathrm{u}^*\big)$, if the pair $\big(\mathrm{x}^*,\mathrm{u}^*\big)$ is no worse than pair $(\bar{\mathrm{x}},\bar{\mathrm{u}})$

in all objectives and the pair $(x^*,u^*)$ is strictly better than $(\bar{x},\bar{u})$ in at least one objective. If there are no solutions which dominate $(x^*,u^*)$, then it is non-dominated.

**Definition 3.** (Pareto set): A set of non-dominated pairs $(x^*,u^*) \in \Omega$ such that $\{\nexists(\bar{x},\bar{u})\in\Omega \mid (\bar{x},\bar{u})\succ(x^*,u^*)\}$ is said to be a Pareto set. Also, the set of vectors in the objective space that are image of a Pareto set is said to be a Pareto frontier.

## 3. EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EAs) are recognized to be suitable to solve multi objective optimization problems, since they deal simultaneously with a set of possible solutions, allowing an entire set of Pareto optimal solutions to be evolved in a single run of the algorithm.Moreover, EAs are less sensitive to the shape or continuity of the Pareto frontier [4].

### 3.1. NON-DOMINATED SORTING GENETIC ALGORITHM II (NSGA-II)

The Non-Dominated Sorting Genetic Algorithm (NSGA) was introduced by Srinivas and Deb [35]. The NSGA has been successfully applied to solve multi-objective optimization problems. This algorithm identifies non-dominated solutions in the population, at each generation, to form non-dominated fronts. Then the usual selection, crossover, and mutation operators are performed. The main criticisms of this algorithm have been high computational complexity of non-dominated sorting, lack of elitism and choosing the optimal parameter value for sharing parameter$\sigma$-share. Recently, Deb et al [8] proposed a modified version of NSGA, which they called NSGA-II, which is faster and more reliable than its predecessor and has a better sorting algorithm, incorporates elitism and no sharing parameter needs to be chosen a priori [30].

In this algorithm, the population is initialized as random, and the number of population is *N*. Next the population is sorted based on non-domination into each front. The first front is completely non-dominant set in the current population and the second front is dominated by the individuals in the first front only and the front goes so on. Each individual in the first front is given a value of 1 and individuals in second are assigned fitness value as 2 and so on. Next, the archive is created based on the order of ranking fronts, i.e. the best rank is selected first. If the number of individuals in the archive is smaller than the population size, the next front will be taken into account and so on. This procedure is called fast non-dominated sorting [24].

If adding a front increases the number of individuals in the archive to exceed the initial population size, a truncation operator is applied to the front based on the crowding distance (CD). For a member of non-dominated set, CD is calculated by finding distance between two nearest solutions on either side of the member along each of the objectives. These distances are normalized by dividing them by the difference between maximum and minimum values of corresponding objectives. For those members in the non-dominated set, which have maximum or minimum value for any objective (boundary solution), CD is assigned to have an infinite value. Let $J_k^{[i]}$ represent the fitness value of the individual $i$ in the sequence. Then, crowdedness of the individual $i$ in dimension $k$ in that rank can be expressed as follows:

$$\mathrm{CD}_{\mathrm{k}}^{[\mathrm{i}]}=\frac{\mathrm{J}_{\mathrm{k}}^{[\mathrm{i+1}]}-\mathrm{J}_{\mathrm{k}}^{[\mathrm{i-1}]}}{\mathrm{J}_{\mathrm{k}}^{\max}-\mathrm{J}_{\mathrm{k}}^{\min}}, \tag{7}$$

where $J_k^{max}$ and $J_k^{min}$ represent the maximum and minimum values in objective $k$, respectively. Let say individual $i$ in a Pareto rank has $m$ values for $m$ objectives according to (7). So, one can simply summarizes the distances to represent the overall crowdedness, crowding distance, of this individual as

$$\mathrm{CD}^{[\mathrm{i}]}=\sum_{\mathrm{k}=1}^{\mathrm{m}}\mathrm{CD}_{\mathrm{k}}^{[\mathrm{i}]}, \tag{8}$$

where $CD_k^{[i]}$ is calculated by (7) and $CD^{[i]}$ is the crowding distance of individual $i$. Finally, the members of the non-dominated set are sorted in monotonically decreasing order according to *CD*s and a desired number of members having the largest *CD* values are selected [13].

Parents are selected from the population by using binary tournament selection based on the rank and crowding distance. The individual with lesser rank is selected. In case of solutions that have the same rank, an individual with the greatest crowding distance is chosen. The selected individuals generates offspring from crossover and mutation operators. The population with the current population and current offspring is sorted again based on non-domination and only the best $N$ individuals are selected and others are deleted. The selection is based on rank and on crowding distance on the last front. Then the new population will be selected as parents at the next round [30]. This cycle iterates until stop conditions are satisfied. After optimization process is completed, the Pareto frontier is produced by non-dominated solutions [11].

### 3.2. INVASIVE WEED OPTIMIZATION (IWO)

Invasive weed optimization was developed by Mehrabian and Lucas in 2006 [21]. The IWO algorithm is an adaptive algorithm based on the metaphor of natural

biological evolution of weed colonizing in opportunity spaces for function optimization. The algorithm is simple but has shown to be effective in converging to optimal solutions employing basic properties, e.g. seeding, growth and competition, in a weed colony [39].

The IWO algorithm works with an initial population of $N$ weeds dispersed on d-dimensional problem space with random positions. These weeds will eventually grow up and execute four steps: initialization, reproduction, spatial distribution and competitive exclusion [14].

In this algorithm, first a finite number of weeds (initial population) $W=\{w_1,w_2,...,w_n\}$, where each corresponding to a solution of the problem, are generated randomly in over feasible search space.Then the reproduction process is done, where each member of the population of weeds will produce seeds based on its fitness, the colony's fitness and the highest fitness, to simulate the natural survival of the fitness process. The number of seeds for each member begins with the value of $S_{min}$ for the worst member and increases linearly to $S_{max}$ for the best member. The number of seeds that weed $i$ can produce is calculated by Equation 9:

$$seed(i) = \frac{(Fit(i)-Fit_{min})(S_{max}-S_{min})}{Fit_{max}-Fit_{min}} \quad (9)$$

where, $Fit(i)$ is the fitness of the $i$-th plant, $S_{max}$ is the maximum number of seeds, $S_{min}$ is the minimum number of seeds, $Fit_{max}$ and $Fit_{min}$ are the maximum fitness and the minimum fitness in the colony, respectively.

In the next step (spatial distribution), the generated seeds are being randomly distributed over the entire d-dimensional search space by normally distributed random numbers with zero mean but varying variance. This step ensures that seeds will be randomly distributed at the neighborhood of parent weed. Here the standard deviation of the random function is given as follow:

$$\sigma_{it} = \left(\frac{it_{max}-i}{it_{max}}\right)^n (\sigma_{initial} - \sigma_{final}) + \sigma_{final} \quad (10)$$

where $\sigma_{it}$ is the standard deviation in the current iteration, $it_{max}$ is the maximum number of iterations, $\sigma_{initial}$ and $\sigma_{final}$ are the maximum and minimum standard deviation and $n$ is the nonlinear modulation index. This means that the standard deviation of the random function will be reduced from initial value $\sigma_{initial}$ to a final value $\sigma_{final}$ in every iteration of the algorithm.

Finally in the fourth step, a competitive exclusion is conducted in the algorithm. In fact, following a number of iterations the population size reaches its maximum. At this point an elimination mechanism is needed. To this end, the seeds and their parents are ranked together and those with better fitness survive and become reproductive.

### 3.3. MULTI− OBJECTIVE INVASIVE WEED OPTIMIZATION

In order to adopt IWO algorithm for solving multi-objectiveoptimization problems, Nikoofard et. al. use a non-dominating sorting technique to find the strength of the weeds or sort the weeds [24]. They called it the Non-dominated Sorting Invasive Weed Optimization (NSIWO). The sorting of weeds in MOIWO algorithm is similar to the NSGA-II.

In this algorithm, the initial population of weeds is generated randomly in a small region of the search space. In each generation, candidate parents are selected from the current population (archive) by a binary tournament selection. Next, a process of seed reproduction and seed dispersal are applied to generate the offspring population. The offspring solution set is added to the previous population, and the fronts are derived through fast nondominated sorting algorithm. The crowding distance is also assigned for each individualin the population. Then, the weakness (opposite of fitness) of each individual w is calculated according to the following formula:

$$\text{Weakness}\,(w) = rank(w) + 1/CD(w) + 2$$

where, rank(w) is the front number and CD(w) is the crowding distance for *w*. In this equation, the fitness (opposite of weakness) is proportionate to the crowding distance, but it is disproportionate to the rank. Thus, the individuals in the lower fronts and withbetter density have higher fitness. Finally,the individuals with lower fitness are eliminated from the combined population explained above, and a new population is formedfor the next iteration [24].

## 4. PROPOSED IMPROVED NSIWO ALGORITHM

In this section, we propose an improved version of NSIWO algorithm, which we called INSIWO.This algorithm is in the large part similar to NSIWO except in the process of dispersal.The whole procedure for INSIWO algorithm is given as follows.

First discretize the control space and then the state space. Choose an equidistance partition of the time interval $[t_0,t_f]$. with a step size $h = (t_f - t_0)/h$, and equidistant nodes on the set of control values corresponds to i-th component of the control vector function as $(u_{i0},u_{i1},...,u_{in})$.

At the primitive step, the algorithm generates a population of weeds in the entire search space. So, each weed is represented by a smooth solution curves,$u(t)$, that represent the candidate solutions for the optimal control signals. For each solution vector, $u(t)$, solve the given system of differential equations that represent the dynamic system to be controlled, for $x(t)$ numerically using any

numerical solver of high accuracy (*RK4*, for instance) using the given initial conditions on the state variables.

Based on the generated initialization population, where each individual of the population is a vector in $R^{2n+2}$ as $(x_i,u_i)=(x_{i0},x_{i1},...,x_{in},u_{i0},u_{i1},...,u_{in})$, find the value of the objective function $J_i(x,u);i=1,2,...,m$ to be minimized for each individual in the population using any numerical integration formula of high accuracy. Also, the fitness is calculated for allindividuals in the population in this step. Next, a rank is assigned based on non-dominated sorting for evaluation of the quality of the given solutions.

In the next step, a binary tournament selection is used to select the candidate parents from the current solution curves, $u(t)$, and correspondingly a higher fitness for individuals. then, the offspring population is generated by a process of seed reproduction, Equation 9, and dispersal.

Here, we improve the process of dispersal in order to increase the explorative power of the weeds and reduce the search space gradually during the iteration process so as to further promote the convergence rate and diversity of solutions. When a weed is near to true Pareto optimal frontier, we reduce the standard deviation $\sigma$ for it in the current population, so that the seeds will be dispersed over a small neighborhood of parent weed. Thus in this process, we alter the standard deviation for each weed based on its fitness value instead of using a fixed $\sigma$ for all weeds in each iteration. The process of varing the standard deviation $\sigma_i$ of the *i*-th weed is explain as follow

$$\sigma_i = (1 - e^{-\Delta_i})\left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \tag{11}$$

where

$$\Delta_i = min_{k=1}^{|p^*|}\left(\sum_{m=1}^{M}\left(J_m^{(i)} - J_m^{*(k)}\right)^2\right)^{\frac{1}{2}}$$

and $J_m^{(i)}$ represent the fitness value of the individual i in the sequence and also $J_m^{*(k)}$ represent the fitness value of the *i*-th individual in the Pareto optimal set, so when $\Delta_i \to 0$ then $\sigma_i \to \sigma_{final}$. This means that the i-th weed lies close to the Pareto frontier.

The offspring solution set is added to the previous population, and the fronts are derived through fast non-dominated sorting algorithm for this combined population. If adding a front increases the number of individuals in the archive to exceed the initial population size, a trunction operator is applied to the front based on the crowding distance (CD), Equation 8.

The weakness of each weed in the current population is calculate that it sorts the weeds in the some front according to their crowding distances. Finally, the individuals with lower fitness are eliminated from the combined population

explained above, and a new population is generated for the next iteration. The Pseudo-code for NSIWO algorithm is summarized as follows.

### 4.1. PSEUDO–CODE FOR INSIWO ALGORITHM

**Initialization step: Discretization**

First discretize the control space and then the state space. Choose an equidistance partition of the time interval $[t_0,t_f]$ as h with $h = (t_f - t_0)/h$, and equidistant nodes on the set of control values corresponds to $i$-th component of the control vector function as $(u_{i0},u_{i1},...,u_{in})$.

**Main Steps:**

1. Randomly generate the original population of $N$ individuals, from the time-control space and numerical solving system of differential equations i.e., random (2n + 2) tuples as $(x_i,u_i)=(x_{i0},x_{i1},...,x_{in},u_{i0},u_{i1},...,u_{in})$ where each individual of the population is a weed ($w$).
2. Evaluate each weed in $W$
3. For each weed $w \in W$
3.1. Assign the rank based on fast non-dominated sorting
3.2. Assign the crowding distance
3.3. Compute the weakness of each weed according to its rankand crowding distance
4. Set the maximum number of generation MaxIt as the terminal condition of the algorithm
5. For *iter = 1* to *MaxIt*
5.1. Use the binary tournament selection to obtain a selected parent population ($W$)
5.2. Find the maximum and minimum weakness in $W$
5.3. For each weed $w \in W$
5.3.1. Compute the number of seeds of w, corresponding to its weakness
5.3.2. Randomly distribute the generated seeds over the search space by Equation 11 around the parent plant ($w$)
5.3.3. Evaluate each weed in the population
5.3.4. Add the generated seeds to the previous solution archive $W$
5.4. For each weed $w \in W$
5.4.1. Assign the rank based on the fast non-dominated sorting
5.4.2. Assign the crowding distance
5.4.3. Compute the weakness of each weed according to its rank and crowding distance

5.5. If $|W| = N > pop_{max}$
5.5.1. Sort the population W in descending order of their fitness
5.5.2. Truncate the population of weeds with smaller fitness until $N = pop_{max}$.

## 5. PERFORMANCE METRICS

Performance metrics are important performance assessment measure, which also allow us to compare algorithms. Deb [7] classified them in three categories, metrics evaluating closeness to the pareto optimal frontier, metrics evaluating diversity amongst non-dominated solutions andmetrics evaluating closeness and diversity. We in this work choose Generational Distance (GD) metric ($\sigma$) to represent convergence to true Pareto frontier and metric $\Delta$ to represent diversity among the non-dominated solutions. They are defined as the follows, respectively:

$$\gamma = \frac{\left(\sum_{i=1}^{|Q|} d_i^p\right)^{1/p}}{|Q|} \tag{12}$$

where $Q$ represents solution set having $|Q|$ members. weuse $p=2$ and$d_i$is minimum distance between the memberin solution set and nearest member is true Pareto set, which is defined as

$$\mathrm{d_i}=\min \sqrt{\sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\mathrm{J_m^{(i)}}-\mathrm{J_m^{*(k)}}\right)^2} \tag{13}$$

Here $M$ represents number of objectives, $i$ and $k$ represent member index in solution set and true pareto set respectively.

$$\Delta = \frac{d_f+d_l+\sum_{i=1}^{|Q|-1}|d_i-\bar{d}|}{d_f+d_l+(|Q|-1)\bar{d}} \tag{14}$$

where $\bar{d}$ is the average of all distances $d_i$, and $d_f$ and $d_l$ are the Euclidean distance between the extreme solutions in true Pareto optimal frontier and the boundary solutions of the obtained non-dominated set. The smaller their values are, the better performance the algorithm shows.

## 6. NUMERICAL RESULTS

In this section, using practical examples the improved non-dominated solutioninvasive weed optimization (INSIWO) method compared with NSIWO and NSGA II methods. Since the selected problems does not have known true Pareto frontier, an expected Pareto frontier is generated by running EA for large number of generations. INSIWO and NSGA-II are run for 1000 generations with each 500 population size. The final populations of both runs are combined and the

obtained Pareto frontier of the combined population is considered as the expected Pareto frontier for optimal control problems.

The algorithms are implemented in MATLAB R2010a on Intel(R) Core(TM) i5, 2.5 GHz, 4 GB RAM, Windows 7 computer configuration. The parameters used for three algorithms are specified in Table 1.

| Parameter | NSGA II | NSIWO | INSIWO |
|---|---|---|---|
| N | 100 | 100 | 100 |
| Max It | 150 | 150 | 150 |
| $P_c$(Crossover Probability) | 0.8 | --- | --- |
| $\mu$(mutation rate) | 0.3 | --- | --- |
| $S_{max}$ | --- | 3 | 3 |
| $S_{min}$ | --- | 1 | 1 |
| $\sigma_{initial}$ | --- | 0.1 | 0.1 |
| $\sigma_{final}$ | --- | 0.01 | 0.01 |
| n | --- | 3 | 3 |

**Table 1.** The parameters of NSGA II, NSIWO and INSIWO for problems.

### 6.1. FED-BATCH BIOREACTOR

The first MOCP is based on the fed-batch lysine fermentation process investigated by Ohno et al. ]26]. The aim is to determine an optimal feeding profile and batch length with respect to conflicting yield and productivity objectives. Assuming perfect mixing and no product degradation yields the following model equations

$$\frac{dx_1}{dt}=\mu x_1,\ \frac{dx_2}{dt}=-\sigma x_1+uC_{S,F},\ \frac{dx_3}{dt}=\pi x_1,\ \frac{dx_4}{dt}=u.$$

t [h]: as the independent variable.

**States variables:**

$x_1$ [g] : the biomass.

$x_2$[g] : the substrate.

$x_3$ [g] : the product (lysine).

$x_4$[L] : the fermenter volume.

**Control variables:**

u [L/h] : the volumetric rate of the feed stream, which contains a limiting substrate concentration $C_{S,F}$ of 2.8 g/L.

**Parameters:**

$\mu$[1/h] : the specific rates for growth, $\mu=0.125C_{S,F}$.

$\sigma$[g/gh] : the substrate consumption, $\sigma=\mu$ /0.135.

$\pi$[g/gh] : the production,$\pi = -138\mu^2 + 134\mu$.

With the substrate concentration. The final time is fixed at $h$ and the initial conditions are specified as:

$$[x_1(0),x_2(0),x_3(0),x_4(0)]=[0.1,14,0,5].$$

The goal is to derive a feeding strategy and batch duration which maximize the productivity, i.e., the ratio between the product formed and the process duration:

$$J_p = \frac{x_3(t_f)}{t_f}.$$

While maximizing the yield, i.e., the mass of product is added over the mass of substrate during the operation:

$$J_y=\frac{x_3(t_f)}{\left(x_4(t_f)-x_4(0)\right)C_{S,F}}.$$

To cast these maximization problems into a minimization framework, the objective functions are defined as the negative productivity and yield $J_1 = -J_p$ and $J_2 = -J_y$. Several constraints are imposed on the volume, the feed rate, the operation time and the amount to be added [16].

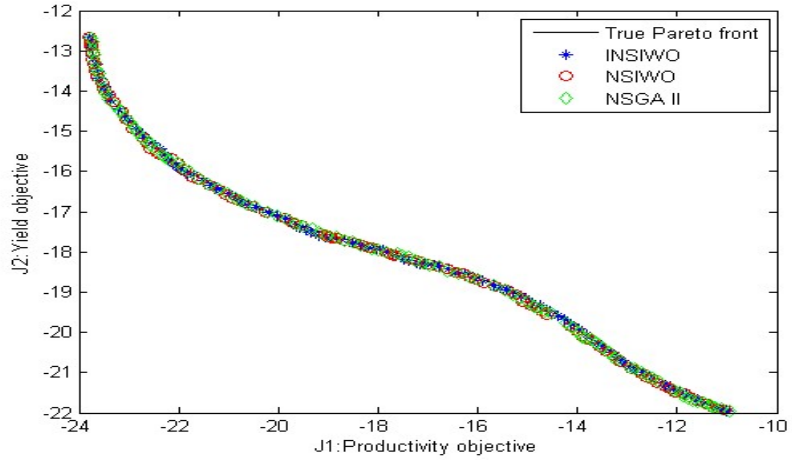


**Figure 1.** Pareto frontier at the end of 150 generations for FED-BATCH bioreactor.

The Figure 1 depicts the Pareto frontier obtained by three strategies. It is visually difficult to observe convergence from in those Pareto frontier plots. The$\gamma$ metric average and $\Delta$diversity values at the end of 150 generations are shown at Table 2. It is clearly reflects that the INSIWO have better average CPU time, convergence metric and diversity values for 150 generations consistently.

The Figure 2 shows the results obtained from the implementation of the algorithms by considering the $\gamma = 0.04$ as the terminal condition of the algorithm. The convergence ($\gamma = 0.04$) is achieved at 59 s and 44 s using NSGA II, NSIWO

respectively, while it is 42 s using the INSIWO. Also, the convergence ($\gamma = 0.04$) for INSIWO is achieved in less generation than the NSIWO and NSGA II.

| Parameters | NSGA II | NSIWO | INSIWO |
|---|---|---|---|
| CPU Time | 56 | 43 | 42 |
| Convergence Rate $\gamma$ | 0.061 | 0.053 | 0.048 |
| Diversity $\Delta$ | 0.643 | 0.511 | 0.492 |

**Table 2.** Average CPU time, $\gamma$and $\Delta$measures at the end of 150 generations using different algorithms.

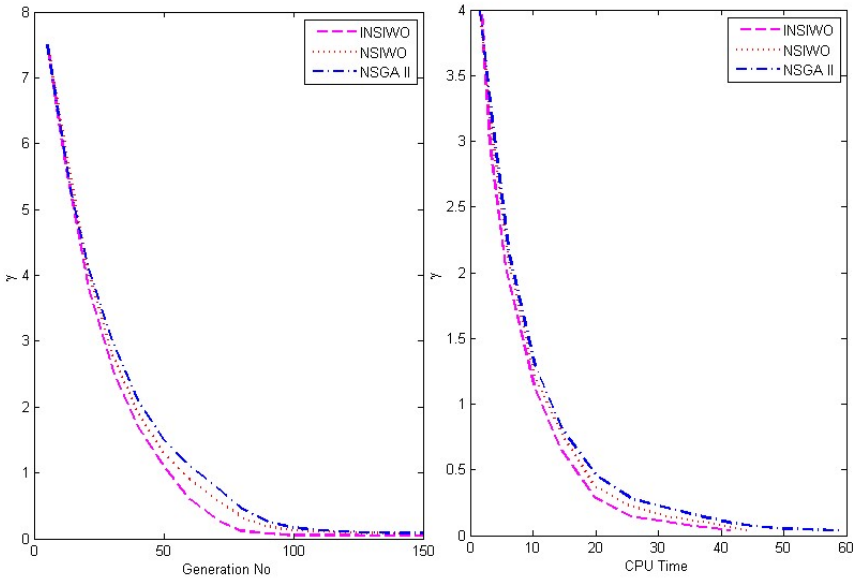


**Figure 2.** Generation wise convergence (left) and convergence as a function CPU time (right).

The optimal feeding profiles is shown in Figure 3. When only productivity is considered the initial max part is provide in order to stimulate biomass growth,and hence lysine production. However, when concentrate on yield optimisation the max piece shortens and the singular profile reduces in height but lasts longer.

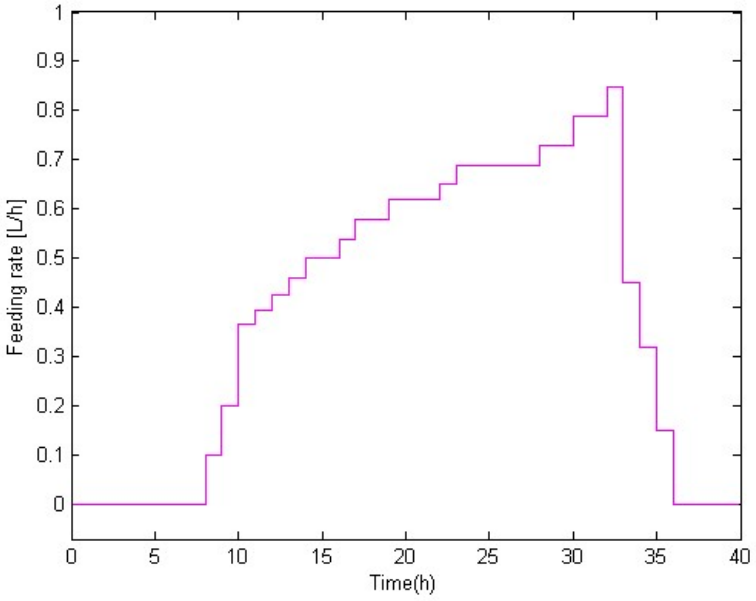


**Figure 3.** Optimal control profile for FED-BATCH bioreactor.

### 6.2. FED BATCH REACTOR

The second MOCP considers a fed batch reactor for induced foreign protein production by recombinant bacteria [32]. The objective is to maximize the profitability of the process using the nutrient and the inducer feeding rates as the control variables. The dynamic model is

$$\frac{dx_1}{dt}=u_1+u_2,\quad \frac{dx_2}{dt}=\mu x_2-(u_1+u_2)\frac{x_2}{x_1},\quad \frac{dx_3}{dt}=C_{S,in}\frac{u_1}{x_1}-(u_1+u_2)\frac{x_3}{x_1}-\mu\frac{x_2}{0.51},$$

$$\frac{dx_4}{dt}=\pi x_2\frac{u_1}{x_1}-(u_1+u_2)\frac{x_4}{x_1},\quad \frac{dx_5}{dt}=C_{i,in}\frac{u_2}{x_1}-(u_1+u_2)\frac{x_5}{x_1},\quad \frac{dx_6}{dt}=-\frac{0.09x_5}{0.034+x_5}x_6,$$

$$\frac{dx_7}{dt}=\frac{0.09x_5}{0.034+x_5}(1-x_7),\quad \mu=\frac{x_3}{14.35+x_3+\frac{x_3^2}{111.5}}\left(x_6+\frac{0.22x_7}{0.22+x_5}\right),\quad \pi=\frac{0.233x_3}{14.35+x_3+\frac{x_3^2}{111.5}}\left(\frac{0.0005+x_3}{0.022+x_5}\right).$$

**State variables:**

$x_1$ [L] : the reactor volume.

$x_2$ [g/L] : the cell density.

$x_3$ [g/L] : the nutrient concentration.

$x_4$ [g/L] : the foreign protein concentration.

$x_5$ [g/L] : the inducer concentration.

$x_6$ [-] : the inducer shock factor on cell growth rate.

$x_7$ [-]: the inducer recovery factor on cell growth rate.

**Control variables:**

$u_1$ [L/h] : the volumetric rates of the glucose.

$u_2$ [L/h] : the volumetric rates of the inducer.

The control variables are bounded between 0 and 1 (L/h). The concentrations of inducer and glucose in the feed streams are $C_{i,in}$= 4.0 g/L and $C_{S,in}$= 100.0 g/L, respectively. The initial conditions are $[1, 0.1, 40, 0, 0{,}1, 0]^T$ and the final time is fixed at $t_f = 10$h. As stated above, the objectives are maximizing the final amount of foreign protein, $J_1 = x_1(t_f)x_4(t_f)$ and minimizing the amount of inducer added, $J_2 = C_{i,in}\int_0^{t_f} u_2(t)dt$.

The Pareto-optimal frontier obtained at the end of maximum allowable generation of 150 is shown in Figure 4 consist of three regions. Region bottom left is shown a small increase in the volume of inducer added results in substantial increase in the amount of protein produced. Regiontop right demonstrate a large increase in the volume of inducer added results in insignificant increase in the amount of protein. The curve part of the Pareto frontier represents a transition area between two regions.

The Table 3 is shown that although the CPU time of the NSIWO is less than the INSIWO, the INSIWO have better convergence metric and diversity values when compared to NSIWO and NSGA II algorithms. Figure 5 shows the results obtained from the implementation of the algorithms by considering the $\gamma = 0.04$ as the terminal condition of the algorithm. It is obviously seen that the INSIWO algorithm needs less computational time and number of generations compared to two other algorithms.

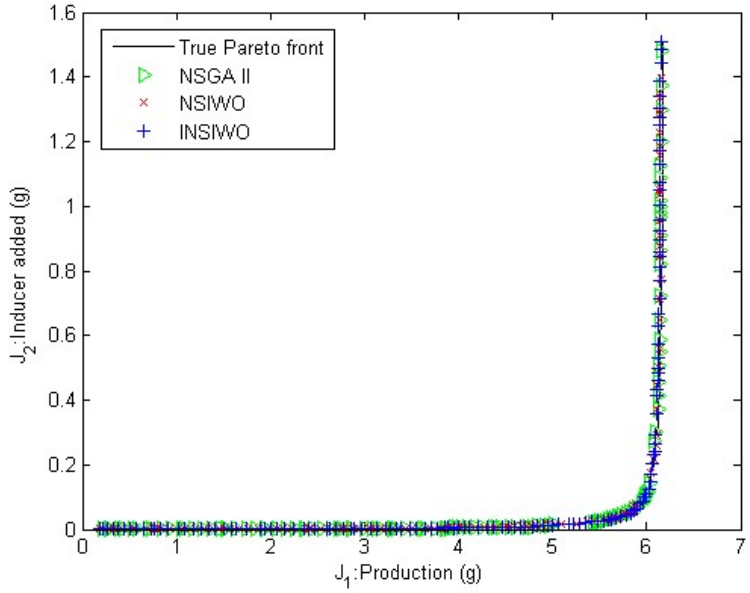


**Figure 4.** Pareto frontier at 150 generation for fed batch reactor.

| Parameters | NSGA II | NSIWO | INSIWO |
|---|---|---|---|
| CPU Time | 51 | 31 | 39 |
| Convergence Rate $\gamma$ | 0.069 | 0.057 | 0.046 |
| Diversity $\Delta$ | 0.537 | 0.421 | 0.406 |

**Table 3.** Average CPU time, $\gamma$and $\Delta$measures at the end of 150 generations using different algorithms

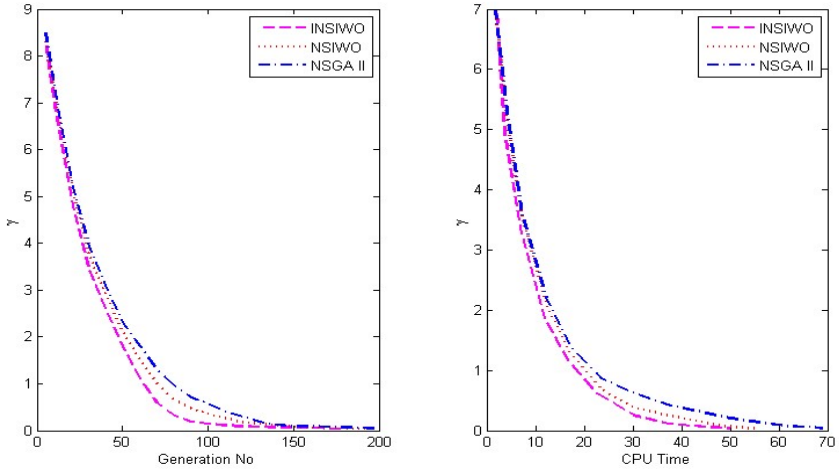


**Figure 5.** Generation wise convergence (left) and convergence as a function CPU time (right).

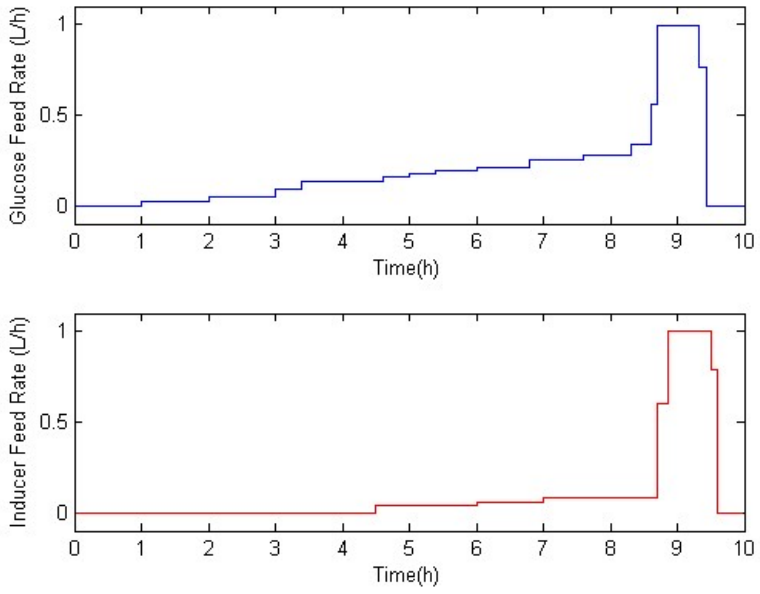


**Figure 6.** Optimal control profiles for glucose and inducer.

The corresponding optimal control trajectory profiles for the substrate $u_1$ and induser $u_2$ feed rate are shown in Figure 6. When we focus on the minimization of the inducer, the values forboth controls remain quite down. Also, when the maximization of the foreign protein production gradually increases, the inducer feed rates increase.

### 6.3. FED BATCH BIO-REACTOR WITH THREE OBJECTIVES

A model for the production of secreted protein in fed-batch bio-reactor was reported by Park and Fred Ramirez [31], which is also studied for three objective dynamic optimization application [29].

$$\frac{dx_1}{dt}x_5=\mu(x_2-x_1)x_5,\ \frac{dx_2}{dt}x_5=x_5x_3\frac{x_4\exp(-5x_4)}{0.1+x_4},\ \frac{dx_3}{dt}x_5=x_5x_3K,$$
$$\frac{dx_4}{dt}x_5=-7.3x_5x_3K+uS,\ \frac{dx_5}{dt}=u,\ \mu=\frac{4.75}{0.12+k},\ K=\frac{21.87x_4}{(0.4+x_4)(62.5+x_4)}.$$

**State variables:**

$x_1$ [g/L] : the secreted protein concentration.
$x_2$ [g/L] : the total protein concentration.
$x_3$ [-] : the culture cell density.
$x_4$ [g/L] : the substrate concentration.
$x_5$ [L] : the hold-up volume.
Control variable:
u [L/h]: the substrate volumetric feed rate.
Parameter:
S [g/L] : the substrate feed concentration.
The initial conditions aer$X(0) = [0, 0, 1, 5, 1]$. Substrate feed concentration is S= 20 with bounds onfeed rate as $0\leq u\leq 2.5$. Three objectives, namely the productivity, yield and fed-batchtime have been considered in this subsection. The productivity, $J_1$ is defined as the ratio of the end point product concentration, $x_1(t_f)$ and the total time of bio-reactor operation,

$$J_1 = \frac{P(t_f)}{t_f}$$

The yield, $J_2$ is defined as the ratio of the total amount of product formed, $x_1(t_f)x_4(t_f)$ and the amount of substrate added in the reactor,

$$J_2 = \frac{P(t_f)x_4(t_f)}{\int_0^{t_f} u(t)Sdt}.$$

Finally, the total fed-batch time, $t_f$ is the third objective, $J_3 = t_f$.

In this problem, the upper bound on the reactor volume is kept as 10 L while that on the volumetric flow rate of the substrate is 2.5 L/h. The fed-batch time, $t_f$ has lower and upper limit of 10 and 30 h, respectively.

The resulting 3-dimensional Pareto frontier has been shown in Figure 7. We use population size of 150 for this case to capture the Pareto frontier. The convergence metric $\gamma$and diversity$\Delta$ for all three algorithms are shown in Table 4, it clearly demonstrate that the INSIWO is more efficient than the other algorithms.

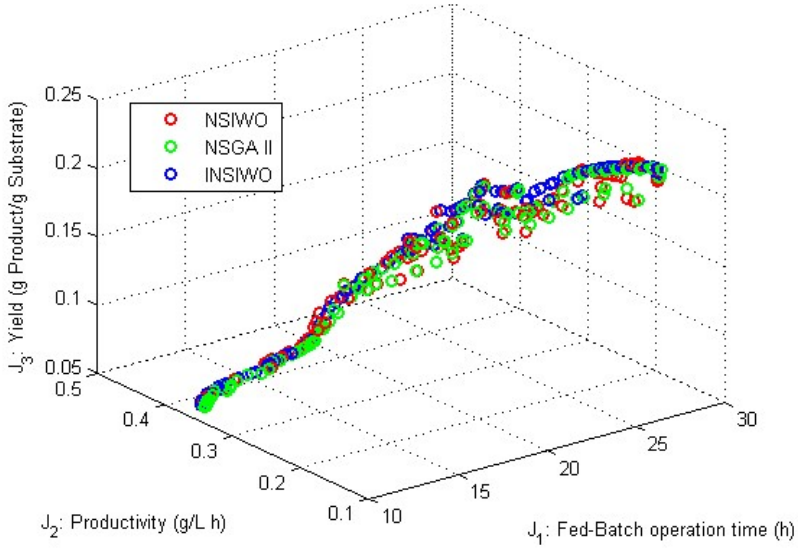


**Figure 7.** Pareto frontier in 3D objective space for productivity, yield and fed batch time.

| Parameters | NSGA II | NSIWO | INSIWO |
|---|---|---|---|
| CPU Time | 76 | 58 | 47 |
| Convergence Rate $\gamma$ | 0.098 | 0.087 | 0.082 |
| Diversity $\Delta$ | 0.846 | 0.754 | 0.689 |

**Table 4.** Average CPU time, $\gamma$ and $\Delta$ measures at the end of 150 generations using different algorithms.

The corresponding optimal control trajectory profile for the substrate volumetric feed rate is shown in Figure 8.

Table 5 shows the results obtained from the implementation of the algorithms by considering the $\gamma$ =0.04 as the terminal condition of the algorithm. It is clearly seen that the INSIWO algorithm needs less number of generations and also less computational time compared to NSIWO algorithm.

| Example | Generation No | | CPU Time | |
|---|---|---|---|---|
| | INSIWO | NSIWO | INSIWO | NSIWO |
| 6.1 | 165 | 174 | 42 | 44 |
| 6.2 | 180 | 196 | 50 | 55 |
| 6.3 | 204 | 219 | 57 | 72 |

**Table 5.** Average CPU time and generation number using different algorithms.

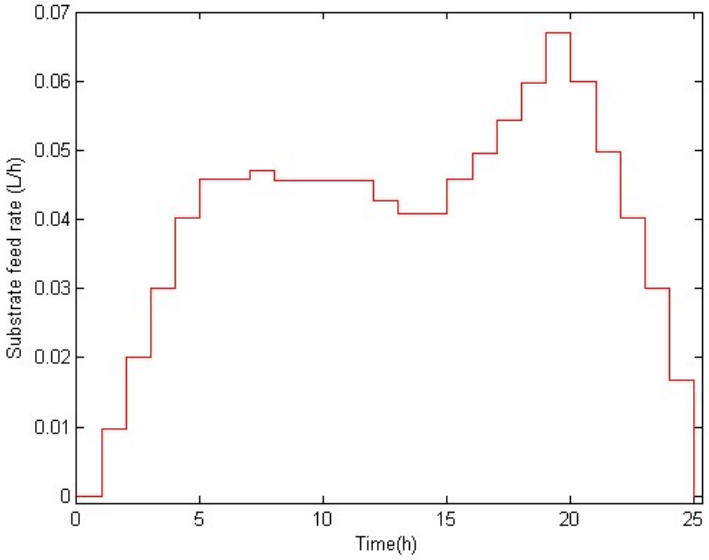


**Figure 8.** Optimal control profile for substrate volumetric feed rate.

## 6. CONCLUSION

In this paper, a new multi objective evolutionary algorithm is presented in which some parameters of MOIWO have modified in order to reduce some shortcomings and find the more suitable solutions for multi objective optimal control problems. In this algorithm a non-dominating sorting strategy and crowding distance method are employed to adopt IWO for solving multi-objective optimal control problems.

The efficiency of the proposed algorithm in finding the entire Pareto optimal frontier is illustrated by solving several engineering examples involving bi- and three-objective MOCPs related to fed-batch reactor and fed-batch bioreactor.

The numerical results show that the proposed algorithm at the end of 150 generations in Examples 6.1 and 6.3 has a better convergence rate and dispersal. It also has a better performance in terms of computational time.In 6.2 the proposed algorithm was more successful in terms of convergence rate and dispersal while the NSIWO method has performed better in terms of computational time.Once, the

convergence rate $\gamma$ =0.04 is used instead of 150 generations as the terminal condition of the algorithm. The results illustrated that the INSIWO algorithm converges to this value in less number of generations and less CPU time.

## REFERENCES


1. L. Biegler, Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation, *Comput. Chem. Eng.* **8** (1984) 243−248.
2. L. Biegler, An overview of simultaneous strategies for dynamic optimization, *Chem. Eng Process: Process Intensif*. **46** (11) (2007) 1043−1053.
3. H. Bock and K. Plitt, A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings of the 9th IFAC world congress, Budapest. Pergamon Press, 1984, pp. 243-247.
4. C. A. Coello, A comprehensive survey of evolutionary-based multi-objective optimization techniques, *Knowl. Inf. Syst*.**1** (3) (1999) 269−308.
5. C. A. Coello andM. S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, In: Proceeding of Congress on Evolutionary Computation (CEC2002), Honolulu, HI. **1** (2002) 1051−1056.
6. I. Das and J. E. Dennis, Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multi-criteria optimization problems, *SIAM. J.Optimiz.* **8** (3) (1998) 631−657.
7. K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*,Wiley 2001.
8. K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multi-objective geneticalgorithm: NSGA-II, *IEEE. Trans. Evolut. Comput*. **6** (2) (2002) 182−197.
9. T. Erickson, A. Mayer andJ. Horn, The niched Pareto genetic algorithm 2 applied to the design of ground water remediation systems, Evolutionary Multi-Criterion Optimization: First International Conference, EMO. (2001) 681−695.
10. C. M. Fonseca, *Multi-Objective Genetic Algorithms with Application to Control Engineering Problems*, Ph.D. Thesis, University of Sheffield. Sheffield, 1995.
11. S. M. K. Heris and H. Khaloozadeh, Open- and closed-loop multi-objective optimal strategies for HIV therapy using NSGA-II, *IEEE. Trans. Biomed. Eng.* **58** (6) (2011) 1678−1685.

12. J. Knowles andD. Corne, ThePareto archived evolution strategy: A new baseline algorithm for Pareto multi-objective optimization, Proceedings of the 1999 IEEE Congress on Evolutionary Computation 1999.
13. S. Kukkonen and K. Deb, Improved Pruning of Non-Dominated Solutions Based on Crowding Distance for Bi-Objective Optimization Problems, IEEE Congress on Evolutionary Computation, pp. 91−98, 2007.
14. D. Kundu, K. Suresh, S. Ghosh, S. Das and B. K. Panigrahi, Multi-objective optimization with artificial weed colonies, *J. Inf. Sci.* **181** (2011) 2441−2454.
15. D. Leineweber, I. Bauer, H. Bock and J. Schlder, An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization, Part I: Theoretical aspects, *Comput. Chem. Eng.* **27** (2003) 157−166.
16. F. Logist, P. M. Van Erdeghem and J. F. Van Impe, Efficient deterministic multiple objective optimal control of (bio)chemical processes, *Chem. Eng. Sci.* **64** (2009) 2527−2538.
17. F. Logist, B. Houska, M. Diehl and J. Van Impe, Fast Pareto set generation for nonlinear optimal control problems with multiple objectives, *Struct. Multidisc. Optim.* **42** (2010) 591−603.
18. F. Logist, B. Houska, M. Diehl and J. Van Impe, Robust multi-objective optimal control of uncertain (bio)chemical processes, *Chem. Eng. Sci.* **66** (2011) 4670−4682.
19. F. Logist, S. Sager, C. Kirchesand and J. Van Impe, Efficient multiple objective optimal control of dynamic systems with integer controls, *J. Process Control* **20** (2010) 810−822.
20. F. Logist, M. Vallerio, B. Houska, M. Diehl and J. Van Impe, Multi-objective optimal control of chemical processes using ACADO toolkit, *Comput. Chem. Eng.* **37** (2012) 191−199.
21. R. Mehrabian and C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, *Ecol. Inform.* **1** (4) (2006) 355−366.
22. A. Messac, A. Ismail-Yahaya and C. A. Mattson, The normalized normal constraint method for generating the Pareto frontier, *Struct. Multidisc. Optim.* **25** (2) (2003) 86−98.
23. K. Miettinen, *Nonlinear Multi-Objective Optimization*, Kluwer, Boston, 1999.
24. A. H. Nikoofard, H. Hajimirsadeghi, A. Rahimi-Kian and C. Lucas, Multi-objective invasive weed optimization: Application to analysis of Pareto

improvement models in electricity markets, *Appl. Soft. Comput.* **12** (2012) 100−112.

25. H. Modares and M. N. Sistani, Solving nonlinear optimal control problems using a hybrid IPSOSQP algorithm, *Eng. Appl. Artif. Intel.* **24** (2011) 476−484.
26. H. Ohno, E. Nakanishi and T. Takamatsu, Optimal control of a semi-batch fermentation, *Biotechnol. Bioeng.* **18** (1976) 847−864.
27. G. C. Onwubolu and B. V. Babu, *New Optimization Techniques in Engineering*, Springer Verlag, Heidelberg, Germany 2004.
28. N. Patel and N. Padhiyar, Modified genetic algorithm using box complex method: Application to optimal control problems, *J. Process Control* **26** (2015) 35−50.
29. N. Patel and N. Padhiyar, Multi-objective dynamic optimization study of fed-batch bio-reactor, *Chem. Eng. Res. Des.* **119** (2017) 160−170.
30. S. Panuganti, P. Roselyn John, D. Devraj and S. Sekhar Dash, Voltage stability constrained optimal power flow using NSGA-II, *Comput. Water Energy Envir. Eng.* **2** (2013) 1−8.
31. S. Park and W. Fred Ramirez, Optimal production of secreted protein in fed-batch reactors, *AIChE. J.* **34** (1988) 1550−1558.
32. D. Sarkar and J. Modak, Genetic algorithms with filters for optimal control problems infed-batch bioreactors, *Bioprocess Biosyst. Eng.* **26** (2004) 295−306.
33. D. Sarkar and J. M. Modak, Pareto-optimal solutions for multi-objective optimization offed-batch bioreactors using non-dominated sorting genetic algorithm, *Chem. Eng. Sci.* **60** (2) (2005) 481−492.
34. J. D. Schafier, Multiple objective optimization with vector evaluated genetic algorithms, Proceedings of the First International Conference of Genetic Algorithms, Pittsburgh, pp. 93−100, 1985.
35. N. Srinivas and K. Deb, Multi-objective function optimization using non-dominated sorting genetic algorithms, *Evolut.Comput*.**2** (3) (1995) 221−248.
36. F. Sun, W. Du, R. Qi, F. Qian and W. Zhong, A hybrid improved genetic algorithm and its application in dynamic optimization problems of chemical processes, *Chin. J. Chem. Eng.* **21** (2013) 144−154.
37. E. Zitzler and L. Thiele, Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Trans. Evolut. Comput.* **3** (1999) 257−271.

38. E. Zitzler, M. Laumanns and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, Zurich, Switzerland: Swiss Federal Institute Technology, 2001.
39. X. Zhang, J. Xu and G. Cui, Research on invasive weed optimization based on the cultural framework, 3rd International Conference on Bio-Inspired Computing: Theories and Applications, IEEE Conference Publications, pp. 129–134, 2008.
40. P. Zhang, H. Chen, X. Liu and Z. Zhang, An iterative multi-objective particle swarm optimization-based control vector parameterization for state constrained chemical and biochemical engineering problems, *Biochem. Eng. J.* **103** (2015) 138−151.