Global Analysis and Discrete Mathematics
Volume 4, Issue 1, pp. 41–60
ISSN: 2476-5341

Pareto-optimal Solutions for multiobjective optimal control problems using hybrid IWO/PSO algorithm

Gholam Hosein Askarirobati* ·
Akbar Hashemi Borzabadi ·
Aghileh Heydari

Received: 31 March 2019 / Accepted: 3 October 2019

Abstract Heuristic optimization provides a robust and efficient approach for extracting approximate solutions of multiobjective problems because of their capability to evolve a set of non-dominated solutions distributed along the Pareto frontier. The convergence rate and suitable diversity of solutions are of great importance for multi-objective evolutionary algorithms. The focus of this paper is on a hybrid method combining two heuristic optimization techniques, Invasive Weed Optimization (IWO) and Particle Swarm Optimization (PSO), to find approximate solutions for multiobjective optimal control problems (MOCPs). In the proposed method, the process of dispersal has been modified in the MOIWO. This modification will increase the exploration power of the weeds and reduces the search space gradually during the iteration process. Thus, the convergence rate and diversity of solutions along the Pareto frontier have been promote. Finally, the ability of the proposed algorithm is evaluated and compared with conventional NSGA-II and NSIWO algorithms using three practical MOCPs. The results show that the proposed algorithm has better performance than others in terms of computing time, convergence and diversity.

* Corresponding author
Gh. H. Askarirobati
Department of Mathematics, Payame Noor University, P.O.Box 19395-3697, Tehran, Iran.
E-mail: askari@phd.pnu.ac.ir

A. H. Borzabadi
Department of Applied Mathematics, University of Science and Technology of Mazandaran, Behshahr, Iran.
E-mail: borzabadi@du.ac.ir

A. Heydari
Department of Mathematics, Payame Noor University, Tehran, Iran.
E-mail: a_heidari@pnu.ac.ir

Keywords Multiobjective optimal control · Pareto optimal frontier · Invasive weed optimization · Particle Swarm Optimization

1 Introduction

Many of engineering problems belong to multiobjective optimal control problems (MOPcs), which have multiple conflicting objectives to be optimized simultaneously to achieve a tradeoff, such as design of optimal reactor feeding rates in (bio)chemical engineering, optimal power management of fuel cells in electrical engineering, optimal robot paths in mechanical engineering and optimal rocket trajectories in aerospace engineering [14].

In single-objective optimization, the determination of the optimum among a set of given solutions is clear. However, in the absence of preference information, in multi-objective optimization there does not exist a unique or straightforward way to determine if a solution is better than other [16]. The notion of optimality most commonly adopted is the one called Pareto optimality [25] which leads to trade-offs among the objectives. Thus, by using this relation, it is not possible to obtain a single solution, but instead, we produce a set of them called the Pareto optimal set (the set of Pareto optimal solutions in the objective space is called Pareto front).

There are two different strategies for generating a set of solutions representing the entire Pareto-optimal frontier: One-at-a-time strategy, and Simultaneous strategy. In the former method, a multi-objective optimizer may be applied one at a time for finding one single Pareto-optimal solution. Most classical generating multi-objective optimization methods use such an iterative scalarization scheme of standard procedures, such as weighted-sum, ε-constraint, NC and NBI methods [5,17].

Recently, these methods have been successfully combined with direct optimal control techniques for construction the Pareto frontier of the non-convex MOCPs. For example, Logist et al. has been proposed an application of NBI and NNC for the multiple objective optimal control of (bio) chemical processes [13], and in [14] several scalarization skims for multi-objective optimization, such as WS, NNC, and NBI have been integrated with fast deterministic direct optimal control methods.

In the simultaneous approach, multiple Pareto-optimal solutions are found in a single simulation run, thereby not requiring multiple applications of an optimizer. In past two decades several nature-inspired meta-heuristics could show good performance detect approximate solutions of multi-objective problems, such as the Multi-Objective Genetic Algorithm (MOGA) [9], Niched Pareto Genetic Algorithm (NPGA) [8], Vector Evaluated Genetic Algorithm (VEGA) [28], Strength Pareto evolutionary algorithms (SPEA) [35], SPEA2 [36], Multi-

Objective Invasive Weed Optimization (MOIWO) [12], Multi-Objective Particle Swarm Optimization (MOPSO) [4], Non-Dominated Sorting Genetic Algorithm (NSGA) [27] and NSGA-II [7].

These algorithms have several advantages. Some of their advantages are: (I) the objective functions gradientis not required; (II) thet are not sensitive to initial guess of solution and (III) they usually do not get stuck in to a local optimum [19].

Evolutionary algorithms are potentially able to find the entire Pareto solutions set of multiobjective problems and have been also successfully used to solve MOCPs. For example, Zhang et. al. introduce an iterative multi-objective particle swarm optimization based control vector parameterization for the state constrained chemical and biochemical engineering problems [33]. sharker and modak used NSGA-II algorithm to solve two optimal control problems related to fed-batch bioreactors [26]. Patel and Padhiyar proposed a modified genetic algorithm using Box Complex method and used it to solve optimal control problems [22]. Sun et. al. introduce a hybrid improved genetic algorithm (HIGA) for solving dynamic optimization problems [30]. Borzabadi et. al. used NSGA-II and MOPSO algorithms to solve two multiobjective optimal control problems [2].

Due to outstanding abilities of evolutionary algorithms in finding Pareto solutions of multi-objective problems, in this paper we propose a new approach based on evolutionary algorithms, named multiobjective hybrid IWO/PSO algorithm which is inspired from IWO and PSO, to find a Pareto optimal pair of state and control for multiobjective optimal control problems. In proposed approach, 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. Also, this improvement leads to the minimization of the distance of the Pareto front and the maximization the diversity of the solutions (uniform distribution).

The rest of the paper is organized as follows. In Section 2, mathematical formulations of general multi-objective optimal control problem are briefly introduced. In section 3, two heuristic approaches for solving multiobjective optimization problems are described. The details of multi objective hybrid IWO/PSO method with performance metrics are discussed in Section 4. In Section 5, the ability of this new method is demonstrated with various practical optimal control problems. Lastly, section 6 outlines the conclusion.

2 Multipleobjective Optimal Control Problem

A general multiobjective optimal control problem consists of optimizing a vector of functions is defined as below:

$$Opt\,(J(x,u) = (J_1(x,u), J_2(x,u), ..., J_m(x,u))) \tag{1}$$

subject to:

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

$$g(x,u,t) \geq 0, \tag{3}$$

$$\psi(x_0, x_f, t_0, t_f) \geq 0, \tag{4}$$

$$t \in [t_0, t_f] \ . \tag{5}$$

where J_i are functions of the state variable $x : [t_0, t_f] \to R^n$, control variable $u \in L^\infty$ and time t. For each individual cost function let's here consider the following typical optimal control problem (known as Bolza's problem):

$$J_i = \varphi(x_f, t_f) + \int_{t_0}^{t_f} L_i(x,u,t)dt,$$

The functions x belong to the Sobolev space $W^{1,\infty}$ while the objective functions are $J_i : R^{n+2} \times R^p \times [t_0, t_f] \to R$. The objective vector is subject to a set of dynamic constraints with $f : R^n \times R^p \times [t_0, t_f] \to R^n$, algebraic constraints $g : R^n \times R^p \times [t_0, t_f] \to R^s$, and boundary conditions $R^{2n+2} \to R^q$. The admissible set $\mathcal{P} \subset R^n \times R^p \times [t_0, t_f]$ is defined to be set of all feasible pair state and control (x,u) that satisfy in Eq.(2-5)

In MOCPs usually objectives in conflict with each other, so it is difficult to have an admissible pair (x^*, u^*) 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 multiobjective optimization problems. For this reason a classification of the solutions is introduced in terms of Pareto optimality, according to the following definitions [36]. In terms of minimization of objective functions:

Definition 1 (non-dominated and Pareto optimal solutions)

1. Pareto dominance: A admissible pair (x^*, u^*) is said dominate another admissible pair (x,u) (denote this relationship $(x^*, u^*) \succ (x,u)$) if the pair (x^*, u^*) is no worse than pair (x,u) in all objectives and the pair (x^*, u^*) is strictly better than (x,u) in at least one objective [36]. If there are no solutions which dominate (x^*, u^*), then it is non-dominated.
2. Pareto optimal: An admissible pair (x^*, u^*) is Pareto optimal solution of the MOCP if and only if there is no other admissible pair (x,u) which dominates (x^*, u^*).
3. A set of non-dominated admissible pair $\{(x^*, u^*) \in \mathcal{P} | \neg\exists (x,u) \in \mathcal{P} : (x,u) \succ (x^*, u^*)\}$ is said to be a Pareto set. The set of vectors in the objective space that are image of a Pareto set, i.e. $\{J(x^*, u^*) \in \mathcal{P} | \neg\exists (x,u) \in \mathcal{P} : (x,u) \succ (x^*, u^*)\}$.

3 Evolutionary Algorithms

Evolutionary algorithms include various methods, which are also called evolutionary computation methods. Evolutionary algorithms (EA) have been recognized to be particularly suitable to solve multi-objective optimization problems because they deal simultaneously with a set of possible solutions which allows an entire set of Pareto-optimal solutions to be evolved in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of traditional mathematical programming techniques. Moreover, EAs are less susceptible to the shape or continuity of the Pareto front [3].

3.1 Invasive weed optimization

Invasive weed optimization (IWO) was developed by Mehrabian and Lucas in 2006 [18]. 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 [34].

In the basic IWO, weeds represent the feasible solutions of problems and population is the set of all weeds. A population of initial solutions is being dispread over the D dimensional search space with random positions. These weeds will eventually grow up and execute steps of the algorithm as described below.

Reproduction: Each member of the population of plants 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 higher the weed's fitness, the more seeds it produces. The formula of weeds producing seeds is

$$seed(i) = \frac{(Fit(i) - Fit_{\min})\,(S_{\max} - S_{\min})}{Fit_{\max} - Fit_{\min}} \tag{6}$$

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

Spatial dispersal: The generated seeds are randomly distributed over the D dimensional search space by normally distributed random numbers with a mean equal to zero, but with a varying variance. This ensures that seeds will be randomly distributed at the neighborhood of parent weed. However, standard deviation σ of the random function will be reduced from a previously defined initial value $\sigma_{initial}$ to a final value σ_{final} in every generation. In simulations, a nonlinear alteration has shown satisfactory performance, given as follows

$$\sigma_{it} = \left(\frac{it_{\max} - it}{it_{\max}}\right)^n (\sigma_{initial} - \sigma_{final}) + \sigma_{final} \tag{7}$$

where σ_{it} is the standard deviation in the current iteration, $it_{\max}$ is the maximum number of iterations and n is the nonlinear modulation index.
Competitive exclusion: After passing some iteration, the number of weeds in a colony will reach its maximum $pop_{\max}$ by fast reproduction. At this time, each weed is allowed to produce seeds. The produced seeds are then allowed to spread over the search area. When all seeds have found their position in the search area, they are ranked together with their parents. Next, weeds with lower fitness are eliminated to reach the maximum allowable population in a colony. In this way, weeds and seeds are ranked together and the ones with better fitness survive and are allowed to replicate. The population control mechanism is also applied to their offspring to the end of a given run, realizing competitive exclusion [32] .

3.2 Particle Swarm Optimization

In nature, birds seek food by considering their personal experience and the knowledge of the other birds in the flock. This idea motivated Kennedy and Eberhart [10] to propose the Particle swarm optimization (PSO) method. This method has been successfully applied in many engineering and management optimization problems because of its simple principle and easy implementation.

In standard PSO, the velocity and position of particle i in the search space are calculated based on the following equation:

$$v_i(t+1) = wv_i(t) + c_1 r_1 \left(x_{i,best}(t) - x_i(t)\right) + c_2 r_2 \left(x_{g,best}(t) - x_i(t)\right) \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

where, $v_i(t+1)$ is the velocity of particle i in generation $t+1$, $x_i(t+1)$ is the position of particle i in generation $t+1$, $x_{i,best}$ is the current optimal position of particle i, and $x_{g,best}$ is current global optimal position. w is an inertia weight which determines the influence of velocity memory and is employed on the favor of global or local search [29], r_1 and r_2 are two random numbers with uniform distribution on the interval [0, 1], and c_1 and c_2 are acceleration factors, which represent the weights of each particle being pushed towards the statistical $x_{i,best}$ and $x_{g,best}$ position, respectively.

3.3 The Multiobjective Hybrid IWO/PSO Algorithm

In this section, we mix the two algorithms and present a hybrid algorithm and discuss the infrastructure and rationale of the hybrid algorithm. From the two previous sections it can be concluded that IWO and PSO have two different approaches for optimization. IWO offers good exploration and diversity, while PSO is an algorithm with fairly deliberate and to the point movements in each iteration and also, it allows individuals to benefit from their previous experiences. The whole procedure for multiobjective hybrid IWO/PSO 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 $h = \frac{t_f - t_0}{n}$, 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 $u(t)$, 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, find the value of the objective function $J_i(x, u);\ i = 1, ..., m$ to be minimized for each individual in the population using any numerical integration formula of high accuracy. Also, the fitness is calculated for all individuals in the population in this step. 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. Next, the archive is created based on the order of ranking fronts. 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 [20].

In the next step, a binary tournament selection is used to select the candidate parents from the current solution. Then, the maximum and minimum weakness (is calculated according to the formula (13)) in w is found. The member having minimum weakness value has got its position x_{gbest} and the i-th member has its corresponding position $x_{pbest,i}$. Number of seeds of w is computed corresponding to its weakness. Each member generates seeds depending on their corresponding weakness value. The number of seeds varies from maximum seed S_{max} for the minimum weakness member to the minimum seed S_{min} for the maximum weakness member.
For each seed w, the velocity and position is calculated according to (8) and (9) respectively. Next, randomly distribute generated seeds over the search space with process of 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 σ 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 σ for all weeds in each iteration. The process of varying the standard deviation σ_i of the i-th weed is explain as follow

$$\sigma_i = \sigma_{final} + (1 - e^{-\Delta_i})(\sigma_{initial} - \sigma_{final}) \tag{10}$$

where $\Delta_i = \min_{k=1}^{|p*|}(\sum_{m=1}^{M}(J_m^{(i)} - J_m^{*(k)})^2)^{1/2}$ 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 truncation operator is applied to the front based on the crowding distance (CD) and weakness.
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 [11]. 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

$$CD_k^{[i]} = \frac{J_k^{[i+1]} - J_k^{[i-1]}}{J_k^{\max} - J_k^{\min}}, \tag{11}$$

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

$$CD^{[i]} = \sum_{k=1}^{m} CD_k^{[i]}, \tag{12}$$

where $CD_k^{[i]}$ is calculated by (7) and $CD^{[i]}$ is the crowding distance of individual i.
The weakness (opposite of fitness) of each individual w is calculated according to the following formula:

$$Weakness(w) = rank(w) + 1/CD(w) + 2 \tag{13}$$

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 [20]. Thus, the individuals in the lower fronts and with better density have higher fitness.

Finally, the individuals with lower fitness are eliminated from the combined population explained above, and a new population is formed for the next iteration.
The Pseudo-code for IWO/PSO algorithm is summarized as follows.

3.4 Pseudo-code for MO hybrid IWO/PSO algorithm

initialization step: Discretization
First discretize the control space and then the state space. Choose an equidistance partition of the time interval $[0, t_f]$ as h with $h = \frac{t_f - t_0}{n}$, 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. Generate Random the population of N individuals, from the time-control space and numerical solving system of differential equations i.e., random (2n + 2) tuples as $(u_{i0}, u_{i1}, ..., u_{in}, x_{i0}, x_{i1}, ..., x_{in})$ where each individual of the population is a weed (w).
2. Initialize the velocity of each weed to zero.
3. Calculate fitness for each weed in W
4. For each weed $w \in W$
4.1. Assign the rank based on fast non-dominated sorting
4.2. Assign the crowding distance
4.3. Compute the weakness of each weed according to its rank and crowding distance
5. For iter=1 to maximum number of generation (iter max):
5.1. Use the binary tournament selection to obtain a selected parent population (w)
5.2. Find the maximum and minimum weakness in W. The member having minimum weakness value has got its position x_{gbest} and the i-th member has its corresponding position $x_{pbest,i}$.
5.3. For each weed $w \in W$, compute the number of seeds of w, corresponding to its weakness
5.4. For each seed w
5.4.1. Calculate the velocity according to (8)
5.4.2. update the position according to (9)
5.5. Randomly distribute the generated seeds over the search space by Eq.(10) around the parent plant (w)
5.6 Add the generated seeds to the previous solution archive W
5.7. If $(|W| = N > P_{max}$
5.7.1. Sort the population on W according to the nondominated sorting, and assign rank and crowding distance to each individual.
5.7.2. Truncate the population of weeds with smaller fitness until $N = pop_{max}$.
5.8. Next iter

4 Performance metrics

To evaluate the performances of the multi objective evolutionary algorithms, many performance metrics have been proposed in the literature [6]. We in this work choose Generational Distance metric (γ) to represent convergence to true Pareto frontier and metric Δ to represent diversity among the non-dominated solutions. They are defined as the follows, respectively:

$$\gamma = \frac{(\sum_{i=1}^{|Q|} d_i^p)^{1/p}}{|Q|} \tag{14}$$

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

$$d_i = \min \sqrt{\sum_{m=1}^{M} (f_m^{(i)} - f_m^{*(k)})^2} \tag{15}$$

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

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{|Q|-1}(|d_i - \overline{d}|)}{d_f + d_l + (|Q|-1)\overline{d}} \tag{16}$$

where $\overline{d}$ 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.

5 Numerical Results

In this section, using practical examples the MO hybrid IWO/PSO 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. NSIWO 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. We use the piecewise constant functions for solving multiobjective optimal control problem using control vector parameterization (CVP) approach. The simulation studies are carried out in MATLAB environment, Windows 7 based computer with 3.4 GHz i7 CPU and 4 GB RAM specifications. Table. 1 shows the control parameter values for NSGA-II, NSIWO and MO hybrid IWO/PSO algorithms.

Table 1 The parameters of NSGA II, NSIWO and MO hybrid IWO/PSO for problems

Parameter	NSGA II	NSIWO	MO hybrid IWO/PSO
N	100	100	100
It_{Max}	150	150	150
P_c (Crossover Probability)	0.8	–	–
μ (mutation rate)	0.3	–	–
S_{max}	–	3	3
S_{min}	–	1	1
$\sigma_{initial}$	–	0.1	0.1
σ_{final}	–	0.01	0.01
n	–	3	3
c_1	–	–	2
c_2	–	–	2
w_{max}	–	–	0.9
w_{min}	–	–	0.4

5.1 Home Heating System

The first MOCP concerns the home heating system involve a heat pump coupled to floor heating system with two conflicting energy and thermal comfort objectives. A dynamic model is as follows [1]:

$$\frac{dx_1}{dt} = \frac{-\kappa_{WR}}{\rho_W C_{PW} V_H} x_1 + \frac{\kappa_{WR}}{\rho_W C_{PW} V_H} x_2 + \frac{u}{\rho_W C_{PW} V_H},$$
$$\frac{dx_2}{dt} = \frac{-\kappa_{WR}}{\kappa_G \tau_G} x_1 - \frac{K_{WR} + K_G}{K_G \tau_G} x_2 + \frac{d}{\tau_G}.$$

where the time t [s] is the independent variable. The states $x_1[C]$ and $x_2[C]$, denote the temperature of the water returning from the heating and the room temperature, respectively. The control $u[W]$ is the heat supplied from the heat pump to the floor, while the outside temperature $d[C]$ induces a disturbance. Here, $\rho_W[kg/m3]$, $c_{pW}[J/kgK]$ and $V_H[m^3]$ are the density, the specific heat and the volume of the water in the floor heating system. $\kappa_{WR}[W/K]$ and κ_G $[W/K]$ are the thermal conductivities between respectively, the water and the room, and the room and the environment, while $\tau_G[s]$ indicates the thermal time constant of the room.
Initially, the water and room temperature are both at $19.5C$. The objective is to minimize the energy input over an entire day

$$J_1 = \int_0^{t_f=24h} \frac{u(t)}{P_{\max}} dt,$$

while maximizing the thermal comfort, i.e., remaining as close as possible to a desired value $x_{2,ref} = 20^oC$

$$J_2 = \int_0^{t_f=24h} (x_2 - x_{2,ref})^2 dt,$$

Table 2 Values of the parameters

Parameter	*Value*	*Unit*
V_H	1.28	m^3
κ_{WR}	1160	$[W/K]$
κ_G	260	$[W/K]$
τ_G	240	s

Despite, a disturbance is raised from the outside temperature:

$$d(t) = 2.5 + 7.5\sin(2\pi t/t_f - \pi/2),$$

However, the heating power of the heat pump is limited to:

$$0 \leq u(t) \leq P_{\max} = 15000,$$

The parameter values are given in the Table 2.
Figure 1 depicts the resulting trade-off between thermal comfort and energy

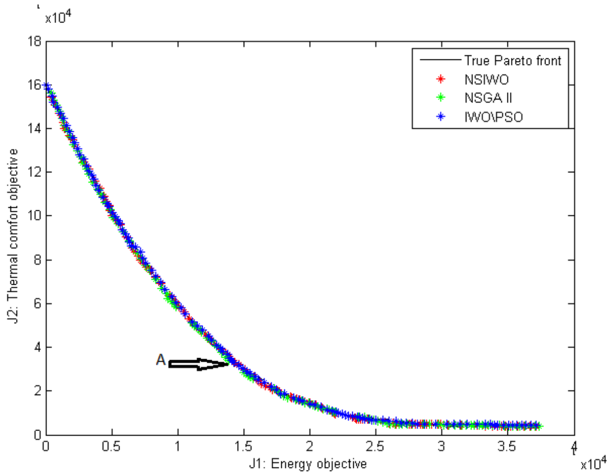

Fig. 1 Pareto frontier at the end of 150 generations

objectives at the end of 150 generations by all the three algorithms NSGA II, NSIWO and MO hybrid IWO/PSO. As the figure show, when the energy input increases from 0 to 3.7 the thermal comfort objective gradually decreases from 16 to 0.47 (i.e., improves the comfort). As the figure shows, there is no significant difference among the three strategies in respect to the construct of Pareto frontier. But Table 3 clearly illustrates that the measure values of γ and Δ reflect MO hybrid IWO/PSO has resulted in better convergence rate and diversity when compared to NSIWO and NSGA-II. It is noteworthy that all the algorithms are measured by 10 times so as to avoid the randomness in

Table 3 Average CPU time, γ and Δ measures at the end of 150 generations using different algorithms

	NSGA II	NSIWO	IWO/PSO
CPU time	50	38	22
convergence rate γ	0.050	0.046	0.038
DiversityΔ	0.443	0.334	0.321

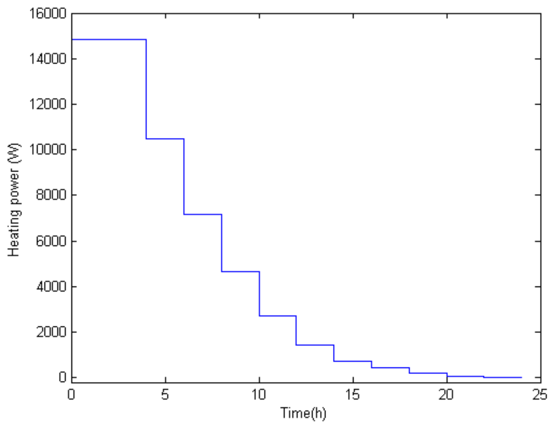

Fig. 2 Optimal control profile for home heating system

the experiment, and their averages are taken as results.

We show the control trajectory for point A on the Pareto front in Fig. 2 corresponding to the utopia point. The utopia point, A, is selected such that it has the minimum Euclidean distance from the reference point. The reference point is a point corresponding to minimum values of the thermal comfort and energy input. The profile consists of two parts. In the first part, the heating uses its full potential in order to counteract the cold of the night. In the second part, it decrease throughout the day since the environment temperature is higher.

5.2 Goddard's rocket problem

The Goddard's rocket problem is to find an optimal ascent trajectory from a flat celestial body with no atmosphere to a prescribed altitude. The control variable is the thrust angle and both gravity and thrust accelerations are constant. The final altitude is assigned and the final vertical component of the velocity has to be zero. A dynamic model is as follows [31]:

$$\begin{aligned}\dot{x} &= v_x \\ \dot{v_x} &= a\cos u \\ \dot{y} &= v_y \\ \dot{v_y} &= -g + a\sin u\end{aligned}$$

where g is the gravity acceleration, a the thrust acceleration, x and y are the components of the position vector, v_x and v_y the components of the velocity vector and u the control. The dynamics is integrated from time $t = 0$ to time $t = t_f$. The objective is to minimize the mission time $J_1 = t_f$, while maximizing the horizontal velocity $J_2 = v_x(t_f)$. Thus, the objective functions are defined as

$$min_{t_f,u}[J_1,\ J_2]^T = [t_f,\ J_2 = -v_x(t_f)]^T$$

The initial conditions are $[x(0)\ v_x(0)\ y(0)\ v_y(0)] = [0\ 0\ 0\ 0]$ and the terminal conditions are $[y(t_f)\quad v_y(t_f)] = [h\quad 0]$. The parameters g, a and h were respectively set to 1.6×10^{-3}, 4×10^{-3} and 10. The control angle was bounded between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, while total mission time was bounded between 100 and 250.
Figure 3 depicts the resulting trade-off between two objectives at the end of 150 generations by all the three algorithms. The metric average and diversity values at the end of 150 generations are shown at Table 5. It clearly illustrates that the measure values of γ and Δ reflect MO hybrid IWO/PSO has better performance when compared to NSIWO and NSGA-II.

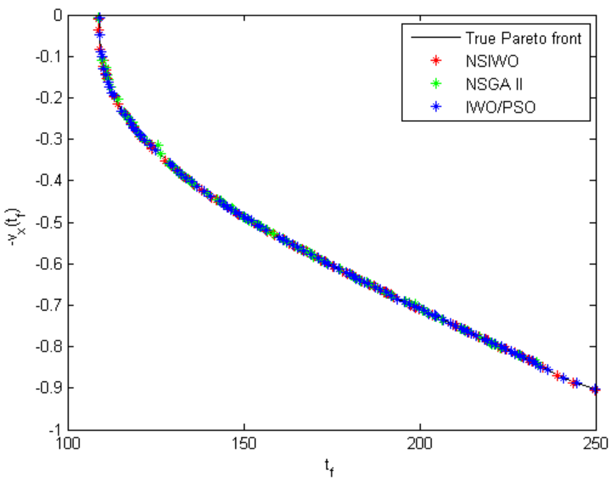

Fig. 3 Pareto frontier at the end of 150 generations

Table 4 Average CPU time, γ and Δ measures at the end of 150 generations using different algorithms

	NSGA II	NSIWO	IWO/PSO
CPU time	44	39	37
convergence rate γ	0.041	0.034	0.030
DiversityΔ	0.443	0.411	0.392

The time history of the control and trajectory for utopia point is plotted in figures 4 and 5.

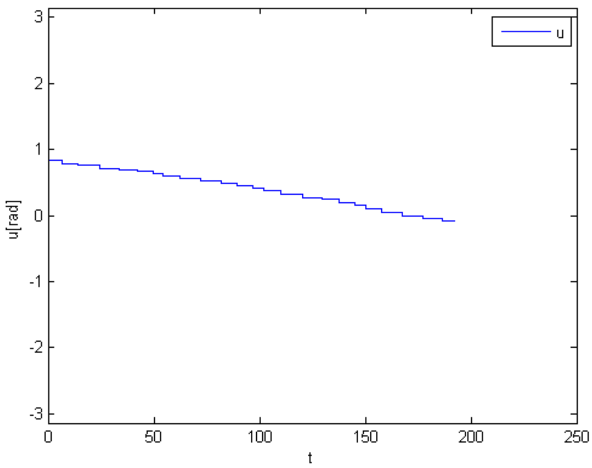

Fig. 4 Time history of the control corresponding to the selected point on the Pareto front

5.3 Fed batch bio-reactor with three objective

A model for the production of secreted protein in fed-batch bio-reactor was reported by Park and Fred Ramirez [24], which is also studied for three objective

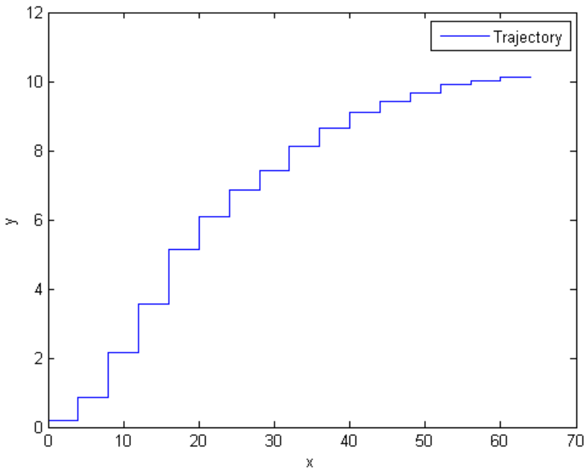

Fig. 5 Trajectory corresponding to the selected point on the Pareto front

dynamic optimization application [23].

$$\begin{aligned}
\frac{dx_1}{dt}x_5 &= \mu(x_2 - x_1)x_5, \\
\frac{dx_2}{dt}x_5 &= x_3x_5\frac{x_4exp(-5x_4)}{0.1 + x_4}, \\
\frac{dx_3}{dt}x_5 &= x_3x_5K, \\
\frac{dx_4}{dt}x_5 &= -7.3x_3x_5K + uS, \\
\frac{dx_5}{dt} &= u \\
\mu &= \frac{4.75K}{0.12 + K}, \\
K &= \frac{21.87x_4}{(0.4 + x_4)(62.5 + x_4)}.
\end{aligned}$$

where the time $t[s]$ is the independent variable. The states variables are $x_1[g/L]$, 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 and $x_5[L]$ the hold-up volume. The control $u[L/h]$ is the substrate volumetric feed rate and the Parameter $S[g/L]$ is substrate feed concentration.
The initial conditions are $X(0) = [0\ 0\ 1\ 5\ 1]^T$. Substrate feed concentration is $S = 20$ with bounds on feed 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 concen-

tration, $x_1(t_f)$ 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 Fig. 6. We use population size of 150 for this case to capture the Pareto frontier. The convergence metric and diversity for all three algorithms are shown in Table 6, it clearly demonstrate that the MO hybrid IWO/PSO is more efficient than the other algorithms. Fig. 7 is shown the corresponding optimal control trajectory profile for the substrate volumetric feed rate.

Table 5 Average CPU time, γ and Δ measures at the end of 150 generations using different algorithms

	NSGA II	NSIWO	IWO/PSO
CPU time	76	58	47
convergence rate γ	0.098	0.087	0.082
DiversityΔ	0.846	0.754	0.689

Table 7 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 MO hybrid IWO/PSO algorithm needs less number of generations and also less computational time compared to NSGA-II and NSIWO algorithms.

6 Conclusion

In this paper, the IWO and PSO algorithms are combined in order to design a new hybrid method in which some parameters of MOIWO have modified in order to reduce some shortcomings and find the more suitable solutions for

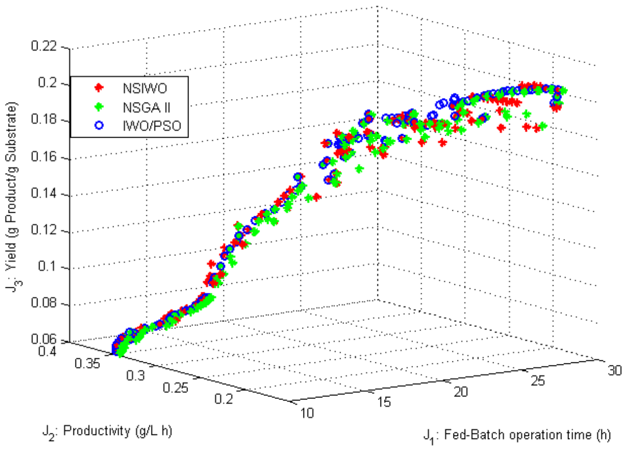

Fig. 6 Pareto frontier in 3D objective space for Productivity, yield and fedbatch time

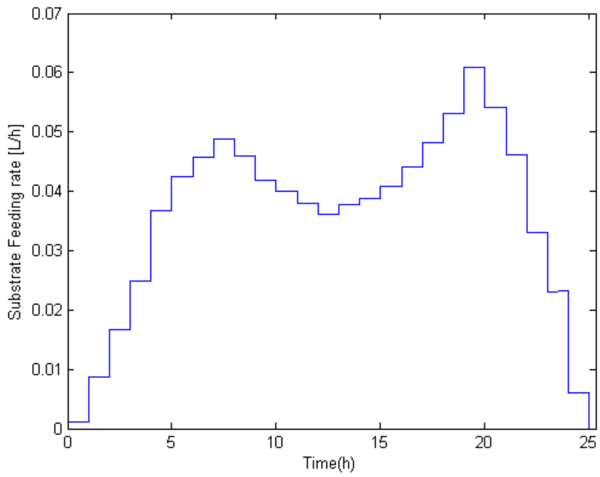

Fig. 7 Optimal control profile for substrate volumetric feed rate

Table 6 Average CPU time and generation number using different algorithms

Example	Generation No			CPU time		
	IWO/PSO	NSIWO	NSGA-II	IWO/PSO	NSIWO	NSGA-II
6.1	140	160	171	19	42	50
6.2	180	195	202	48	59	68
6.3	205	220	218	57	72	71

multi objective optimal control problems. In the proposed method, the process of dispersal has been modified. This modification will increase the explorative power of the weeds and reduces the search space gradually during the iteration process. Moreover, to improve the convergence of MO hybrid IWO/PSO, the crowding distance is used to calculate the fitness value of these solutions. 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.

The numerical results show that the proposed algorithm has a better convergence rate, dispersal and less 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 MO hybrid IWO/PSO algorithm converges to this value in less number of generations and less CPU time.

References

1. M. Bianchi, Adaptive modellbasierte prdiktive regelung einer kleinwrmepumpen anlage, PhD Thesis, ETH, Zurich (2006).
2. A.H. Borzabadi, M. Hasanabadi, and N. Sadjadi, Approximate Pareto optimal solutions of multi objective optimal control problems by evolutionary algorithms, Journal of Control and Optimization in Applied Mathematics, 1(1), 1–19 (2015).
3. C.A. Coello, A comprehensive survey of evolutionary-based multi-objective optimization techniques, Knowledge and Information Systems: An International Journal, 1(3), 269–308 (1999).
4. C.A. Coello and M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, In Proceeding Congress on Evolutionary Computation, Honolulu, HI., 1, 1051–056 (2002).
5. I. Das and J.E. Dennis, Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems, SIAM Journal on Optimization, 8(3), 631–657 (1998).
6. K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, Wiley, 16 (2001).
7. K. Deb, A. Pratap, and S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, 6(2), 182–197 (2002).
8. M. Erickson, A. Mayer, and J. Horn, The niched Pareto genetic algorithm 2 applied to the design of ground water remediation systems, In EMO, 681–695 (2001).
9. C.M. Fonseca, Multiobjective genetic algorithms with application to control engineering problems. Ph.D. Thesis, University of Sheffield, Sheffield, U.K. (1995).
10. J. Kennedy and R. Eberhart, Particle swarm optimization, in Proceedings of the IEEE International Conference on Neural Networks, 4, 1942–1948, (1995).
11. S. Kukkonen and K. Deb, Improved Pruning of Non-Dominated Solutions Based on Crowding Distance for Bi-Objective Optimization Problems, IEEE International Conference on Evolutionary Computation, 132, 91–98 (2007).
12. D. Kundu, K. Suresh, S. Ghosh, S. Das, and B.K. Panigrahi, Multi-objective optimization with artificial weed colonies, Journal of Information Science, 181, 2441–2454 (2011).
13. F. Logist and J. Van Impe, Multiple objective optimization of cyclic chemical systems with distributed parameters. Control. Appl. Optim., 7, 295–300 (2009).
14. F. Logist, B. Houska, M. Diehl, and J. Van Impe, Fast Pareto set generation for nonlinear optimal control problems with multiple objectives, Structural and Multidisciplinary Optimization, 42, 591–603 (2010).
15. F. Logist, P.M.M. Van Erdeghem, and J. Van Impe, Efficient deterministic multiple objective optimal control of (bio)chemical processes, Chem. Eng. Sci, 64, 2527–2538 (2009).
16. A.L. Jaimes, L.V. Santana Quintero, and C.C.A Coello, Ranking methods in many-objective evolutionary algorithms, Nature-Inspired Algorithms for Optimisation, 413–434 (2009).

17. A. Messac, A. Ismail-Yahaya, and C. A. Mattson, The normalized normal constraint method for generating the Pareto frontier, Structural and Multidisciplinary Optimization, 25(2), 86–98 (2003).
18. R. Mehrabian and C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecological Informatics, 1(4), 355–366 (2006).
19. H. Modares and M.B. Naghibi Sistani, Solving nonlinear optimal control problems using a hybrid IPSO-SQP algorithm, Engineering Applications of Artificial Intelligence, 24, 476–484 (2011).
20. A.H. Nikoofard, H. Hajimirsadeghi, A. Rahi-mi-Kian, and C. Lucas, Multiobjective invasive weed optimization: application to analysis of Pareto improvement models in electricity markets, Applied Soft Computing, 12, 100–112 (2012).
21. H. Ohno, E. Nakanishi,and T. Takamatsu, Optimal control of a semibatch fermentation, Biotechnology and Bioengineering, 18 , 847–864 (1976).
22. N. Patel and N. Padhiyar, Modified genetic algorithm using Box Complex method: application to optimal control problems, J. Process Control, 26, 35–50 (2015).
23. N. Patel and N. Padhiyar, Multi-objective dynamic optimization study offed-batch bioreactor, Chemical Engineering Research and Design, 119, 160–170 (2017).
24. S. Park and W. Fred Ramirez, Optimal production of secretedprotein in fed-batch reactors, AIChE J., 34, 1550–1558 (1988).
25. V. Pareto, Cours d'conomie Politique, F. Rouge, Lausanne, 97 (1896).
26. D. Sarkar and J.M. Modak, Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using nondominated sorting genetic algorithm, Chemical Engineering Science, 60(2), 481–492 (2005).
27. N. Srinivas and K. Deb, Multi-objective function optimization using nondominated sorting genetic algorithms, Evolutionary Computations, 2(3), 221–248 (1995).
28. J.D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, Proceedings of the First International Conference of Genetic Algorithms, Pittsburgh, 93–100 (1985).
29. Y. Shi and R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Congress of Evolutionary Computation, 69–73 (1998).
30. F. Sun, W. Du, R. Qi, F. Qian, and W. Zhong, A hybrid improved genetic algorithmand its application in dynamic optimization problems of chemical processes, Chin. J. Chem. Eng., 21, 144–154 (2013).
31. M. Vasile and L. Ricciardi, A Direct Memetic Approach to the Solution of Multi-Objective Optimal Control Problems, IEEE Symposium Series on Computational Intelligence (2016).
32. Y. Zhou, H. Chen, and G. Zhou, Invasive weed optimization algorithm for optimization no-idle flow shop scheduling problem. Neurocomputing, 137, 285–292 (2014).
33. Y. Zhang, D.W. Gong, and N. Gong, Multi-objective optimization problems using cooperative evolvement particle swarm optimizer, J. Comput. Theor. Nanosci., 10(3), 655–663 (2013).
34. 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, 129–134 (2008).
35. E. Zitzler and L. Thiele, Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation, 3, 257–271 (1999).
36. E. Zitzler, M. Laumanns, and L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Zurich, Switzerland: Swiss Federal Institute Technology (2001).