Optimal fuzzy inverse dynamics control of a parallelogram mechanism based on a new multi-objective PSO

A. Farokhi¹ and M.J. Mahmoodabadi¹*

Abstract: This work presents a multi-objective optimization method based on high exploration particle swarm optimization, called MOHEPSO, for optimization problems with multiple objectives. In order to convert the single-objective (HEPSO) algorithm to the multi-objective one, its fundamentals should be changed. The leaders’ selection in the proposed algorithm is based on the neighborhood radius concept for the global best position and the Sigma method for the personal best position. Also, a fuzzy elimination technique is used for pruning the archive. The numerical results of the MOHEPSO algorithm on mathematical test functions are compared with those of other multi-objective optimization algorithms for the performance evaluation of the algorithm. Finally, the proposed algorithm is implemented to find the optimum values of controller coefficients for a parallelogram five-bar linkage mechanism. The introduced control strategy is designed based on the inverse dynamics concepts, improved by fuzzy systems and optimized by regarding two objective functions. The simulation results are presented to demonstrate the efficiency and accuracy of this approach.

Keywords: fuzzy control; inverse dynamics; optimal control; multi-objective optimization; particle swarm optimization; parallelogram five-bar mechanism

ABOUT THE AUTHORS
A. Farokhi was born in Aleshtar, Iran in 1992. He received his BS degree in Mechanical Engineering from Islamic Azad University of Khorraramabad in 2015 and his MS degree in Mechanical Engineering from the Sirjan University of Technology, Iran, in 2017. His research interests include fuzzy control, optimal and nonlinear control, and robotics.

M.J. Mahmoodabadi received his BS and MS degrees in Mechanical Engineering from Shahid Bahonar University of Kerman, Iran in 2005 and 2007, respectively. He received his Ph.D. degree in Mechanical Engineering from the University of Guilan, Rasht, Iran in 2012. He worked for 2 years in the Iranian textile industries. During his research, he was a scholar visitor with Robotics and Mechatronics Group, University of Twente, Enschede, the Netherlands for 6 months. Now, he is an Assistant Professor of Mechanical Engineering at the Sirjan University of Technology, Sirjan, Iran. His research interests include optimization algorithms, non-linear and robust control, and computational methods.

PUBLIC INTEREST STATEMENT
The Particle Swarm Optimization (PSO) algorithm, originally was introduced by Kennedy (an American social psychologist) and Eberhart (an American electrical engineer), and is one of the modern heuristic algorithms. It was developed through simulation of simplified social systems as a stylized representation of the movement of organisms in a bird flock or fish school, and is robust in solving nonlinear optimization problems. In comparison with other evolutionary methods, the PSO technique can generate a high-quality solution with shorter calculation time and a more stable convergence characteristic. On the other hand, the relative simplicity of PSO, and the fact that it is a population-based technique have made it a natural candidate to be extended for multi-objective optimization. Multi-objective optimization which is also called multi-criteria optimization or vector optimization has been defined as finding a vector of decision variables satisfying constraints to give acceptable values to all objective functions.
1. Introduction
In engineering, the concept of optimization is noticed when complex problems are going to be solved. Such complexity may be associated with the kind of problem which an operator wants to solve (i.e. to be nonlinear or not) or the kind of solution which a person wishes to address (i.e. whether it is an exact or an approximate solution). Furthermore, the optimization problems fall into two categories: the single-objective and the multi-objective. The latter has been gaining increasing attention in the stochastic optimization community. A great variety of multi-objective optimizers have been developed, and their performance has been tested on many problems with different characteristics. In fact, the goal is to find a series of non-dominated solutions that represent a trade-off among the objectives, because unlike single-objective optimization problems, there is no single solution available for these problems (Donoso & Fabregat, 2016; Hughes, 2008).

In the practical multi-objective problems, the design criteria called objective functions may conflict with each other so that improving one of them will deteriorate another. When solving multi-objective problems, due to the presence of two or more conflicting objectives, it is often impossible to obtain a solution vector that simultaneously optimizes all of the objective functions and satisfies all of the constrains. In such cases, the concept of Pareto-optimal solutions or equivalently Pareto front arises. A state of solutions is named Pareto front if and only if there is no alternative state that would make an objective function better off without making anyone worse off (non-dominated solutions) (Cheng, Chen, Wai, & Wang, 2014; Liu, Chen, Deb, & Goodman, 2016; Meza, Espitia, Montenegro, Giménez, & González-Crespo, 2017; Razmi, Jafarian, & Amin, 2016).

Among different multi-objective optimization algorithms, multi-objective evolutionary algorithms (MOEAs), which make use of the strategy of the population evolution, are some effective methods for solving multi-objective problems. In recent years, several approaches such as multi-objective particle swarm optimization algorithms (MOPSO) (Mahmoodabadi, Bagheri, Mostaghim, & Bisheban, 2011), vortex multi-objective particle swarm optimization (MOVPSO) (Meza et al., 2017), multi-objective artificial bee colony algorithms (MOABC) (Atashkari, NarimanZadeh, Ghavimi, Mahmoodabadi, & Aghaieeneshad, 2011) and etc. have been proposed for the performance improvement of MOEAs. Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart, is one of the modern heuristic algorithms and was inspired by natural flocking and swarming behavior of birds and fish (Bratton & Kennedy, 2007; Valle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley, 2008). Further, it was developed through simulation of simplified social systems, and has shown robust performance, in solving nonlinear optimization problems. This method is able to quickly produce a proper solution which also has better convergence properties in comparison with the other evolutionary approaches (Ding, 2017).

Nonetheless, solving multi-objective problems by PSO is involved with two matters. The first one is to sustain an archive in order to make a trade-off between convergence and diversity. The non-dominated solutions are sustained by an external elitist archive which also can update itself so as to preserve the diversity. There are several methods used to update the archive. For example, many multi-objective algorithms adopt the crowding distance (Al Moubayed, Petrovski, & McCall, 2014) to prune the archive; the clustering mechanism for maintaining an archive is also applied to the multi-objective algorithms (Abbasiyan, Nezamabadi-pour, & Amoozegar, 2015; Padhye, Branke, & Mostaghim, 2009) to keep the size of external archive constant. The other matter in MOPSOs is updating the global best (gbest) and the personal best (pbest) positions for each particle, since there is no absolute best solution, but rather a set of non-dominated solutions. In recent years, several methods such as the Ranking methods implemented to find global best positions (Wang & Yang, 2009); the decomposition approach employed to select global and personal best positions for each particle (Gong, Cai, Chen, & Ma, 2014), etc. are proposed to determine the gbest and pbest.

In order to have an effective multi-objective algorithm with the good population diversity, an external archive strategy is often used. In fact, the external elitist archive is implemented to store the non-dominated solutions obtained by the algorithm and these solutions are filtered by a certain
quality measure, such as fuzzy elimination technique, and etc. In this work, a new multi-objective particle swarm optimization algorithm based on the high exploration particle swarm optimization (HEPSO) which is one of the evolutionary methods introduced by one of the authors (Mahmoodabadi, Salahshoor Mottaghi et al., 2014) is proposed. The HEPSO seems particularly suitable for multi-objective optimization, mainly because of convergence speed, global optimality, solution accuracy, and algorithm reliability, in comparison with well-known and recent evolutionary algorithms for single-objective optimization. It is in fact the combination of PSO with two different operators, so as to increase the exploration capability of the PSO algorithm. The first operator was inspired by the multi-crossover mechanism of the genetic algorithm (Chang, 2007), and the other uses the bee colony mechanism to update the position of the particles (Akay & Karaboga, 2012).

The introduced MOHEPSO algorithm is tested on the mathematical benchmark functions and challenged by optimally design of fuzzy inverse dynamics controller for a parallelogram mechanism. The objective functions which often conflict with each other are appropriately defined as weighting normalized summation of angle errors and weighting normalized summation of control efforts. The constant parameters of the introduced controller are regarded as the design variables used for the optimization process. The obtained Pareto front is shown and three optimum design points are selected for computer simulation and verifying the effectiveness and robustness of the proposed strategy.

The inverse dynamics controller has a long history in the control engineering and is accepted in a lot of real applications due to the efficient performance. Peng, Lin, and Su (2009) applied a computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques. Zeinali and Notash (2010) introduced fuzzy logic-based inverse dynamic modelling for robot manipulators. Wang (2011) implemented adaptive inverse dynamics for free-floating space manipulators. Chen, Mei, Ma, Lin, and Gao (2014) proposed robust adaptive inverse dynamics control for uncertain robot manipulator. Giusti, Malzahn, Tsagarakis, and Althoff (2017) presented a combined inverse-dynamics/passivity-based controller for robots with elastic joints.

The rest of this paper is organized as follows: Section 2 gives a brief review on the multi-objective optimization. High exploration particle swarm optimization is presented in Section 3. In Section 4, the proposed algorithm is introduced. Experimental results and comparison studies to verify the capability of proposed algorithm are shown in Section 5. The description of a parallelogram controllable mechanism is presented in Section 6. Its fuzzy inverse dynamics controller is given in Section 7. The optimal fuzzy inverse dynamics controller of the parallelogram robot is simulated in Section 8. Finally, Section 9 concludes the paper.

2. Multi-objective optimization

From a classical standpoint, optimization of a single function simply entails determining a set of stationary points, identifying a local maximum or minimum, and possibly finding the global optimum. In contrast, the process of determining a solution for a multi-objective optimization problem is slightly more complex and less definite than a single-objective problem.

Multi-objective optimization problems (MOPs) are problems involving more than one objective to be optimized. In these problems, there are several objectives or cost functions (a vector of objectives) which are to be optimized (minimized or maximized) simultaneously. Unlike single objective optimization algorithms, the performance of multi-objective optimization solutions cannot be improved without sacrificing the performance of at least another one. Therefore, there is not a single optimal solution as the best solution with respect to all the objective functions (Kukkonen & Coello, 2017). Instead, there is a set of optimal solutions, known as Pareto-optimal solutions or Pareto front (Khishtandar & Zandieh, 2017) for such problems. The Pareto-optimal set is defined based on Pareto dominance. Each new addition to this set is compared with all the objective function values of potential solution points in order to determine if the new point is dominated. If it is non-dominated,
then it is kept in the set of potential solution points. In this section, the main concepts related to the multi-objective optimization problems are described.

A general multi-objective optimization problem is defined as follows:

\[
\text{minimize } f(x) = (f_1(x), f_2(x), \ldots, f_N(x)) \text{ subject to }
\]

\[
g_i(x) \geq 0; \ i = 1 \text{ to } k
\]

\[
h_j(x) = 0; \ j = 1 \text{ to } p
\]

where \(N\) is the number of objective functions involving \(N \geq 2\), \(k\) is the number of inequality constraints, and \(p\) is the number of equality constraints. \(f(x)\) is a \(N\)-dimensional vector of objective functions.

**Definition 1** (dominance): A decision vector \(x_1\) dominates another vector \(x_2\) (denotes as \(x_1 < x_2\)) if:

\[
f_i(x_1) \leq f_i(x_2) \ \forall i = 1, \ldots, N.
\]

**Definition 2** (Pareto-optimal): A vector of decision variables \(x^* \in s \subset \mathbb{R}^n\) (s is the feasible region) is Pareto-optimal if it is non-dominated with respect to \(s\).

**Definition 3** (Pareto-optimal set (PS)): The set of Pareto-optimal solutions (PS) is defined as:

\[
PS = \{x \in s | x \text{ is Pareto optimal}\}.
\]

**Definition 4** (Pareto front): The Pareto-optimal front (PF) is defined as: \(PF = \{S(x) | x \in PS\}\).

### 3. High exploration 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 was first introduced by Kennedy and Eberhart to propose the PSO method. The original version of PSO suffers from trapping in local minima and premature convergence. Several approaches, such as FIPSO (Wang, Wang, Yan, & Shen, 2017), APSO (Zhan, Zhang, Li, & Chung, 2009), ACOR–PSO (Huang, Huang, Chang, Yeh, & Tsai, 2013), HEPSO (Mahmoodabadi, Salahshoor Mottaghi et al., 2014) and etc., have been thus far proposed to modify the performance of PSO. The HEPSO algorithm has shown more successes in comparison with well-known and recent evolutionary algorithms for single-objective optimization.

In the PSO algorithm, each candidate solution to the problem is considered a particle and the set of solutions is called “the swarm”. Moreover, every candidate solution is associated with a velocity, and the position of each particle in population is changed according to its own experience and the other particles experience (velocity) in the flock. HEPSO is a combination of PSO with two different operators so as to increase the exploration capability of the PSO algorithm. The first one is based on the multi-crossover mechanism of the genetic algorithm (Chang, 2007), and the second one performs similarly to the bee colony mechanism (Akay & Karaboga, 2012). These formulae are used to improve the converging process and to escape from local minima in the original version of the PSO algorithm.

In PSO, the position of each particle is changed according to its own experience and that of its neighbors. Let \(\vec{x}_i(t)\) denote the position of particle \(i\) at time step \(t\). The position of particle \(i\) is then changed by adding a velocity \(\vec{v}_i(t)\) to its current position, i.e.:

\[
\vec{x}_i(t + 1) = \vec{x}_i(t) + \vec{v}_i(t + 1)
\]
and the velocity vector changes in the following way:

$$\vec{v}_i(t + 1) = w\vec{v}_i(t) + C_1\vec{r}_1(\vec{x}_{g\text{best}} - \vec{x}_i(t)) + C_2\vec{r}_2(\vec{x}_{p\text{best}} - \vec{x}_i(t))$$

(5)

where $\vec{r}_1$ and $\vec{r}_2$ are vectors which contain the random elements from the interval [0,1]; $C_1$ is the cognitive learning factor and represents the attraction of a particle toward its own success, and $C_2$ is the social learning factor and represents the attraction of a particle toward the success of the entire swarm; and finally, $w$ is the inertia weight which is used to balance local and global searches.

In HEPSO, the first added operator uses the global best position of the population ($\vec{x}_{g\text{best}}$) as the premier parent, and the personal best position ($\vec{x}_{p\text{best}}$) as the second parent. It will randomly generate the new velocity for the selected particle $\vec{x}_i(t)$ as below:

$$\vec{v}_i(t + 1) = \vec{r}(C_2\vec{x}_{g\text{best}}(t) - \vec{x}_{p\text{best}}(t) - \vec{x}_i(t))$$

(6)

where $\vec{x}_i(t)$ denotes the position of particle $i$ at iteration (time step) $t$, $\vec{r} \in [0,1]$ is a random vector which contains values, $C_2$ is the social learning factor, $\vec{x}_{g\text{best}}$ is the global best position, and $\vec{x}_{p\text{best}}$ is the personal best position.

Furthermore, the second operator is inspired by the foraging behavior of honey bees (Karaboga, 2005), and the food source obtaining operator of Artificial Bee Colony (ABC) algorithm is used for the selected particles in the HEPSO method. The position of the randomly selected particle $\vec{x}_i(t)$ will change in the dimensions as below:

This is an important issue, because, the archive has to be updated

$$\vec{x}_d^j(t + 1) = \vec{x}_d^j(t) + (2r - 1)(\vec{x}_d^j(t) - \vec{x}_d^j(t))$$

(7)

where $d$ is a random integer in the range [1, dimension], $\vec{r} \in [0,1]$ is a random value, and $j$ is also a random integer in the range [1, number of particles]. After calculation of Equation (7), the superior value between $\vec{x}_i(t + 1)$ and $\vec{x}_i(t)$ should be selected. The flowchart of this algorithm is shown in Figure 1.

4. Multi-objective high exploration particle swarm optimization

The proposed MOHEPSO algorithm is designed, as briefly described with regard to such items as diversity and convergence here. Initially, a random population is generated and in each iteration, the learning factors ($C_1$ and $C_2$) and the inertia weight ($w$) would be allocated. Also, after calculation of the fitness values of all particles, the non-dominated solutions are determined and the archive is formed.

Now, it is possible to identify the $\vec{x}_{g\text{best}}$ and $\vec{x}_{p\text{best}}$ from the Pareto front based on the considered strategies mentioned in the following sections. The random values $\rho_1$ and $\rho_2 \in [0,1]$ are allocated to every particle. If, for a particle, $\rho_1$ is not greater than the standard deviation of the fitness values; or if $\rho_2$ is not smaller than $\frac{\rho_1}{\text{max iteration}}$, the operator in Equation (7) will generate a new particle. Note that, $\rho_1$ denotes the probability of the bee colony, $t$ refers to the present iteration, and max iteration denotes the biggest possible iteration number. For the particles that are not selected for the bee colony operation, another random number $\theta \in [0,1]$ should be assigned. If a particle has $\theta < P_c$, then the multi-crossover operator would generate a velocity for it Equation (6). Here, $P_c$ is the multi-crossover probability. Furthermore, the particles that are not chosen for these two operations will be enhanced by PSO, and this cycle should be repeated until the user-defined stopping criterion is satisfied.

In the following subsections, the strategies to restrain the archive size and select $\vec{x}_{p\text{best}}$ and $\vec{x}_{g\text{best}}$ are discussed.
4.1. Archive size and fuzzy elimination

In most of multi-objective optimization methods, the archive must contain a set of specified criteria solutions, while maintaining a good spread of solutions in the obtained Pareto front. However, if all of non-dominated solutions are maintained in the archive, then the size of the archive grows very quickly.

This is an important issue, because, the archive has to be updated in each iteration and with a growing archive size, the update may become computationally demanding. Herein, a fuzzy elimination technique (Mahmoodabadi, Taherkhorsandi et al., 2014) is used to prune the archive. It involves a fuzzy parameter $\varepsilon_{Fuzzy}$ which sets the extent of fuzzy desired in a problem, In this approach all the particles in the archive have a neighborhood radius equal to $\varepsilon_{Fuzzy}$ and if the Euclidean distances of particles (in the objective function space) from a certain particle are fewer than $\varepsilon_{Fuzzy}$, then they will be simply eliminated from the objective function space. Figure 2 illustrates this technique as an example in a bi-objective space. The technique is formulated as:
while \( t \) refers to the present iteration; \( \text{max iteration} \) denotes the biggest possible iteration number; \( E_{\text{constant}} \) represents a positive constant value; and \( E_{\text{variable}} \) denotes a fuzzy variable that its rules and membership functions are brought in Table 1 and Figure 3 respectively. The \( E_{\text{variable}} \), as the inference outcome of the consequent variables, can be evaluated by applying any of the simplified inference, the product–sum-gravity, and the min–max gravity approaches (Rao & Kumar, 2017).

The input of the membership function ought to be evaluated as follows:

\[
T = \left( \frac{1}{T} \right) - \text{fix}\left( \frac{1}{T} \right)
\]

where \( \text{fix}\left( \frac{1}{T} \right) \) is the rounding function of \( \frac{1}{T} \) to the closest integer toward zero.

In fact, Equation (9) normalizes the input of the membership functions for the fuzzy variables. Equation (8), however, makes the archive retain more non-dominated solutions at first iterations due to low elimination radius, and consequently increases the algorithm convergence. An increase in the iteration number causes a rise in the elimination radius and omits more close solutions. Therefore, the algorithm spread becomes broader and the non-dominated solutions are more uniformly diversified.

### 4.2. The strategy for the global best position (gbest)

To update the position of particles in MOPs, one should employ the whole swarm’s best particle (\( \vec{x}_{\text{gbest}} \)) as a leader. Every particle can only has a single leader, which must be suitably selected so as

| Table 1. The fuzzy rules for the fuzzy variable |
|-----------------------------------------------|
| \( S \)          | 0.0 |
| \( M \)          | 0.5 |
| \( B \)          | 1.0 |
to raise the diversity and the convergence of the solutions. To this end, the selection of \( \vec{x}_{\text{gbest}} \) is done through assigning a neighborhood radius \( R_{\text{neighborhood}} \) to all of the non-dominated solutions with regard to density measures (Mahmoodabadi et al., 2011). Accordingly, when the Euclidean distance between two non-dominated solutions is smaller than this radius, they are said to be the neighbors of each other. Hence, after counting the neighbors of every single non-dominated solution at every iteration, the one with fewer neighbors is considered to be the leader.

4.3. The strategy for the personal best position (pbest)

By selecting a proper technique to find \( \vec{x}_{\text{pbest}} \), the diversity within the swarm is maintained. In the proposed algorithm, each particle of population selects a personal best position from the archive during each iteration via the Sigma method (Mostaghim & Teich, 2003). These selections are distinct from each other; in the first step, the value \( \sigma_i \) is calculated for particle \( i \) in the archive; and in the second step, \( \sigma_j \) for particle \( j \) of the population is assigned. Then, the distance between \( \sigma_i \) and \( \sigma_j \) is evaluated for all the particles of both the population and the archive. Finally, the particle \( k \) of the archive is selected as the personal best position of the particle \( j \) of the population if the distance between their sigmas is the minimum. In other words, each particle with a closer sigma value (in the case of two dimensional objective space, closer means having less difference of sigma values) to the sigma value of an archive member must select that archive member as its best local guide (\( \vec{x}_{\text{pbest}} \)).

For a two-objective space, the sigma parameter is defined as follows:

\[
\sigma = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^2}
\] (10)

Figure 3. The membership functions for the fuzzy variable \( E_{\text{variable}} \).

Figure 4. The determination of the personal best position of the particles using the Sigma method.
where \( f_1 \) and \( f_2 \) are the first and second objective functions. Figure 4 shows the strategy for selecting the personal best position among the archive members for each particle, based on the Sigma method, in a bi-objective space.

5. Numerical experiments on multi-objective high exploration particle swarm optimization

This section contains the computational results obtained by the MOHEPSO algorithm using mathematical test functions compared to the other multi-objective methods.

5.1. Test functions

The test functions for multi-objective optimization should impose sufficient difficulty to challenge the algorithm in searching for the Pareto-optimal solutions. Here, SCH, FON, ZDT1, and ZDT6 benchmark problems, all of them have two objective functions, are utilized to examine the performance of the proposed method (Mahmoodabadi, Taherkhorsandi et al., 2014). Table 2 shows the formulae of the functions along with the number of dimensions, the admissible bound of their variables, the Pareto-optimal solutions, and the nature of the Pareto-optimal front for each problem.

5.2. Comparison metrics

In order to provide a quantitative assessment of the performance of multi-objective optimization on the test functions, two goals are often taken into consideration: a good convergence toward the Pareto-optimal set; and the maintenance of diversity in obtained solutions existing in the Pareto-optimal set. Since both of them are conflicting in nature, comparing two sets of trade-off solutions require different performance measures. Herein, we consider five performance metrics for evaluating each of the above two goals in the set of Pareto-optimal solutions.

(1) The first is the diversity metric (\( \Delta \)). This measures the extent of spread achieve among the non-dominated solutions (Mahmoodabadi, Taherkhorsandi et al., 2014). It is given as:

\[
\Delta = \frac{d_f + d_l + \sum_{i=1}^{n-1} |d_i - \bar{d}|}{d_f + d_l + (n-1)d} \quad \text{and} \quad d = \frac{\sum_{i=1}^{n-1} d_i}{n-1}
\]

Table 2. Multi-objective optimization test functions

| Name | Dimension | Formula | Range | True pareto optimal solutions | Comment |
|------|-----------|---------|-------|------------------------------|---------|
| SCH  | 1         | \( f_1(x) = x^2 \) \( f_2(x) = (x - 2)^2 \) | \([-10^5, 10^5]\) \( x \in [0, 2]\) | Convex | |
| FON  | 3         | \( f_1(x) = 1 - \exp \left(-\sum_{i=1}^{n} \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) \) \( f_2(x) = 1 - \exp \left(-\sum_{i=1}^{n} \left(x_i + \frac{1}{\sqrt{n}}\right)^2\right) \) | \([-4,4]\) \( x_i = x_j = x_k \) | Nonconvex | |
| ZDT1 | 30        | \( f_1(x) = x_1 \) \( f_2(x) = g(x) \times h(x) \) | \([0,1]\) \( x_j = 0 \) | Nonconvex | |
|      |           | \( g(x) = 1 + 9 \sum_{i=1}^{n-1} \frac{x_i}{n-1}, h(x) = 1 - \sqrt{\frac{\sum_{i=1}^{n} \frac{x_i}{n}}{n-1}} \) | | | |
| ZDT6 | 10        | \( f_1(x) = 1 - e^{-4x_1} \times \sin^7(6x_1) \) \( f_2(x) = g(x) \times h(x) \) | \([0,1]\) \( x_i = 0 \) | Nonconvex, nonuniformly and spaced | |
|      |           | \( g(x) = 1 + 8 \left( \sum_{i=1}^{n} \frac{x_i}{n}\right)^{0.25} \) \( h(x) = 1 - \left( \frac{10}{n+9}\right)^{0.2} \) | | | |

Table 2. Multi-objective optimization test functions
where the parameters $d_e$ and $d_l$ are the Euclidean distances between the extreme solutions and the boundary solutions of the non-dominated set; $d_f$ is the Euclidean distance between consecutive solutions in the obtained non-dominated set; $\bar{d}$ denotes the average of these distances; and finally, $n$ is the number of members in the set of non-dominated solutions. A good distribution would make all distances $d_i$ equal to $\bar{d}$, and would make: $d_f = d_l = 0$. Thus, for the most extensive spread set of non-dominated solutions, $\Delta = 0$.

(2) The metric of generational distance (GD) (Azzouz, Bechikh, & Said, 2017) gives a good indication of the distance between the true Pareto-optimal front and non-dominated solution members as follows:

$$GD = \sqrt{\frac{\sum_{i=1}^{n} d_i^2}{n}} \quad (12)$$

where $n$ is the number of members in the set of non-dominated solutions archive and $d_i$ is the minimum Euclidean distance between the member $i$ in the non-dominated set and the Pareto-optimal front. If all members of non-dominated set are in the true Pareto-optimal front, then $GD = 0$.

(3) The metric of the maximum spread (MS), calculates the normalized Euclidean distance of boundary solutions in the objective space. It is formulated for a two-objective problem as:

$$MS = \sqrt{\frac{1}{2} \sum_{m=1}^{2} \left( \frac{\max_{i=1}^{n} f_{m}^{i} - \min_{i=1}^{n} f_{m}^{i}}{F_{m}^{\max} - F_{m}^{\min}} \right)^2} \quad (13)$$

where $n$ is the numbers in the discovered Pareto front; $f_{m}^{i}$ is the $m$th objective of member $i$; and $F_{m}^{\max}$ and $F_{m}^{\min}$ are respectively the maximum and minimum of the $m$th objective function in the Pareto-optimal front. The greater the maximum spread is, the more area of Pareto-optimal front is covered by the non-dominated solutions. If the above metric equals 1, a widely spread set of solution is obtained (Khishtandar & Zandieh, 2017).

(4) The spacing metric (SP) aims at assessing the spread distribution of vectors throughout the set of non-dominated solutions. It is calculated with a relative distance measure between consecutive solutions in the obtained non-dominated set (Mirjalili, Saremi, Mirjalili, & Coelho, 2016):

$$SP = \frac{1}{\sum_{i=1}^{n-1} d_i} \sqrt{\frac{\sum_{i=1}^{n} (d_i - \bar{d})^2}{n}}, \quad d = \frac{\sum_{i=1}^{n-1} d_i}{n} \quad (14)$$

where $n$ is the number of members in the set of non-dominated solutions and $d_i$ is the Euclidean distance between the member $i$ and its nearest neighbor in the non-dominated set. The desired value for this metric is zero, which means that the elements of the set of non-dominated solutions are equidistantly spaced.

(5) The metric of convergence ($\Upsilon$) (Pal & Bandyopadhyay, 2016) measures the proximity of the generated set of Pareto-optimal solutions to the true Pareto front. First, a set of uniformly spaced solutions of the true Pareto-optimal front in the objective space is defined; and then, for each solution obtained by the algorithm, its minimum Euclidean distance ($d$) from the chosen solutions on the Pareto front is calculated. Also, $n$ is the number of members in the set of non-dominated solutions. The average of these distances is computed, and the lower the value of $\Upsilon$, the closer the generated Pareto front to the true Pareto front.

$$\Upsilon = \frac{1}{n} \sum_{i=1}^{n} d_i \quad (15)$$
5.3. Numerical results and discussion
The MOHEPSO algorithm is tested with a population of size 150. The maximum iteration, $E_{\text{constant}}$ and $R_{\text{neighborhood}}$ are respectively set at 250, 200 and 0.1. Moreover, $C_1$ is linearly decreased from $C_{1i} = 2.5$ to $C_{1f} = 0.5$, while $C_2$ is linearly increased from $C_{2i} = 0.5$ to $C_{2f} = 2.5$ over iteration. Furthermore, the bee colony probability is set at $P_B = 0.02$, and the multi-crossover probability is set for SCH and FUN at $P_c = 0.95$ and for ZDT1 and ZDT6 at $P_c = 0.1$.

The non-dominated solutions obtained, in one arbitrary run, by the new considered multi-objective algorithm for the test functions given in Table 2 are shown in Figure 5. As these figures show, good diversity and convergence have been achieved by the proposed algorithm.

In order to evaluate the performance of the proposed algorithm on the test functions via the metrics, three well-known versions of multi-objective optimization algorithms are used for comparison, as detailed in Table 3.

Each algorithm is implemented in 20 independent runs, for which the average values of the performance metrics corresponding to each test function are brought in Table 3.

In other words, this table includes the average values of the metrics obtained from the four algorithms (MOHEPSO, MOPSO Ingenious, MOEA-D, and NSGA-II) for the test functions given in Table 2. As it is specified from the table, the proposed algorithm delivers a better performance in all of the functions for the GD, MS and $\Upsilon$ metrics except ZDT1. The MOPSO Ingenious in turn has desirable $\Delta$ and SP values in all the test functions except ZDT6. The MOEA-D, however, performs poorly in MS and $\Upsilon$ metrics along with the SCH and FUN test functions. In addition, the MOHEPSO has desirable $\Delta$
values in ZDT6; and MOPSO Ingenious has a better performance in Y metric for the ZDT1 test function (Table 4).

6. Dynamical equation of a parallelogram five-bar linkage mechanism

The mechanism in Figure 6 is called a five-bar linkage, where having all revolute joints. Clearly there are only four bars in the figure, but in the theory of mechanisms, it is a convention to count the ground as an additional linkage, which explains the terminology. In Figure 6, it is assumed that the lengths of links 1 and 3 are the same, and that the two lengths marked $l_2$ are also the same. In this way, the closed path in the figure is in fact a parallelogram, which greatly simplifies the computations. Notice, however, that the quantities $l_c$, and $l_c$ need not to be equal. In other words, even though links 1 and 3 have the same length, they need not to have the same mass distribution (Spong & Vidyasagar, 2008).

It is clear from the figure that, even though there are four links being moved, there are in fact only two degrees of freedom, identified as $q_1$ and $q_2$. At first, the coordinates of the centers of mass of the various links are written as a function of the generalized coordinates. This gives

$$
\begin{align*}
  x_{c1} &= l_1 \cos q_1 \\
  y_{c1} &= l_1 \sin q_1 \\
  x_{c2} &= l_2 \cos q_2 \\
  y_{c2} &= l_2 \sin q_2 \\
  x_{c3} &= l_2 \cos q_2 + l_3 \cos q_1 \\
  y_{c3} &= l_2 \sin q_2 + l_3 \sin q_1 \\
  x_{c4} &= l_1 \cos q_1 + l_4 \cos(q_2 - \pi) \\
  y_{c4} &= l_1 \sin q_1 + l_4 \sin(q_2 - \pi)
\end{align*}
$$

(16)

Next, with the aid of these expressions, the velocities of the various centers of mass are written as a function of $\dot{q}_1$ and $\dot{q}_2$. For convenience, the third row of each of the following Jacobian matrices is dropped as it is always zero. The result is:

$$
\begin{align*}
  v_{c1} &= \begin{bmatrix} -l_1 \sin q_1 & 0 \\
  l_1 \cos q_1 & 0 \end{bmatrix} \dot{q} \\
  v_{c2} &= \begin{bmatrix} 0 & -l_2 \sin q_2 \\
  0 & l_2 \cos q_2 \end{bmatrix} \dot{q} \\
  v_{c3} &= \begin{bmatrix} -l_3 \sin q_1 & -l_2 \sin q_2 \\
  l_3 \cos q_1 & l_2 \cos q_2 \end{bmatrix} \dot{q} \\
  v_{c4} &= \begin{bmatrix} -l_1 \sin q_1 & -l_2 \sin q_2 \\
  l_1 \cos q_1 & l_2 \cos q_2 \end{bmatrix} \dot{q}
\end{align*}
$$

(17)
The velocity Jacobians $J_{vc}^i$, $i = 1, \ldots, 4$ are defined as the four matrices appearing in the last equations. Next, it is clear that the angular velocities of the four links are simply given by

$$\omega_1 = \omega_3 = q_1 k, \quad \omega_2 = \omega_4 = q_2 k$$

Thus the inertia matrix is given by

$$D(q) = \sum_{i=1}^{4} m_i J_{vc}^i J_{vc}^i + \begin{bmatrix} I_1 + I_3 & 0 \\ 0 & I_2 + I_4 \end{bmatrix}$$

If Equation (17) is substituted into Equation (19) and the standard trigonometric identities are used then
d_{11}(q) = m_1 l_c^2 + m_3 l_c^2 + m_4 l_1^2 + I_1 + I_3
\quad d_{12}(q) = d_{21}(q) = (m_3 l_c^2 - m_4 l_1 l_{c4}) \cos(q_2 - q_1)
\quad d_{22}(q) = m_2 l_c^2 + m_3 l_2^2 + m_4 l_{c4}^2 + I_2 + I_4

\text{IF}
\begin{align*}
m_3 l_c l_{c3} &= m_4 l_1 l_{c4} 
\end{align*}

then the inertia matrix is diagonal and constant, and as a consequence the dynamical equations will contain neither Coriolis nor centrifugal terms.

Turning now to the potential energy:
\begin{align*}
P &= g \sum_{i=1}^{n} y_{ci} = g \sin(q_1(m_1 l_c + m_3 l_c + m_4 l_1)) = g \sin(q_2(m_2 l_c + m_3 l_2 - m_4 l_{c4}))
\end{align*}

Hence
\begin{align*}
\varphi_1 &= g \cos(q_1(m_1 l_c + m_3 l_c + m_4 l_1)) \\
\varphi_2 &= g \cos(q_2(m_2 l_c + m_3 l_2 - m_4 l_{c4}))
\end{align*}

Notice that \( \varphi_1 \) depends only on \( q_1 \) but not on \( q_2 \), and similarly that \( \varphi_2 \) depends only on \( q_2 \) but not on \( q_1 \). Hence, if Equation (21) is satisfied, then the rather complex looking manipulator in Figure 6 is described by the decoupled set of equations as follows.
\begin{align*}
d_{11} \dot{q}_1 + \varphi_1(q_1) &= u_1 \\
d_{22} \dot{q}_2 + \varphi_2(q_2) &= u_2
\end{align*}

where \( u_1 \) and \( u_2 \) are the control inputs. Equation (24) can be rewritten as follows.
\begin{align*}
\ddot{q}_1 &= -\frac{\varphi_1(q_1)}{d_{11}} + \frac{u_1}{d_{11}} \\
\ddot{q}_2 &= -\frac{\varphi_2(q_2)}{d_{22}} + \frac{u_2}{d_{22}}
\end{align*}

The state-space formulation of the system can be described as Equation (26).
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\frac{\varphi_1(q_1)}{d_{11}} + \frac{u_1}{d_{11}} \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -\frac{\varphi_2(q_2)}{d_{22}} + \frac{u_2}{d_{22}}
\end{align*}

7. Fuzzy inverse dynamics controller for the mechanism

The inverse dynamic model of an arbitrary manipulator can be expressed as:
\begin{align*}
u &= M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q)
\end{align*}

where \( u = [u_1, u_2, \ldots, u_n]^T \) is the vector of input generalized forces, and \( n \) denotes the number of generalized coordinates of the manipulator. Where inertia matrix \( M(q) \) is an \( n \times n \) symmetric positive definite matrix, \( C(q, \dot{q}) \) is an \( n \times n \) matrix of centripetal and Coriolis terms and \( g(q) \) is an \( n \times 1 \) vector of gravitational terms; \( q = [q_1, q_2, \ldots, q_n]^T \), \( \dot{q} = [\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n]^T \) and \( \ddot{q} = [\ddot{q}_1, \ddot{q}_2, \ldots, \ddot{q}_n]^T \) are the displacement, velocity and acceleration vectors of joints respectively. In Equation (27), if
\[
\dot{q} = \dot{a}_q, a_q = -K_p q - K_d \dot{q} + r \quad \text{and} \quad r = \dot{q}^d + K_d \dot{q}^d + K_p q^d
\]

then

\[
\ddot{q} = K_p \left( q^d - q \right) + K_d \left( \dot{q}^d - \dot{q} \right) + \ddot{q}^d
\]

Finally, the tracking error \( e(t) = q^d - q \) satisfies

\[
\dot{e}(t) + K_d \dot{e}(t) + K_p e(t) = 0 \tag{28}
\]

where \( K_p \) and \( K_d \) are diagonal matrices with diagonal elements consisting of position and velocity gains, respectively. The following choice for the gain matrices \( K_p \) and \( K_d \) causes an exponentially stability of the system.

\[
K_p = diag \{ y_{1s}^2, \ldots, y_{ns}^2 \} \quad \text{and} \quad K_d = diag \{ 2y_1, \ldots, 2y_n \} \tag{29}
\]

The fuzzy inverse dynamics controller with input of the membership function \( e(t) = q^d - q \) and output \( F \) for the robot manipulator could be calculated using Equations (30) and (31) for each link as the following equations.

\[
F_1 = \frac{V B \times R_1 + P O \times R_2 + Z B \times R_3}{V B + P O + Z B} \tag{30}
\]

\[
F_2 = \frac{H S \times K_x + H M \times K_y + H B \times K_b}{H S + H M + H B} \tag{31}
\]

where

\[
y_1 = F_1 \Delta M_1 + M_1
\]

\[
y_2 = F_2 \Delta M_2 + M_2 \tag{32}
\]

Then, in Equation (29):

\[
K_p = diag \{ y_{1s}^2, y_{2s}^2 \}
\]

\[
K_d = diag \{ 2y_1, 2y_2 \} \tag{33}
\]

The membership functions are shown in Figures 7 and 8, and their rules are mentioned in Tables 5 and 6.

8. Optimal fuzzy inverse dynamics controller of the parallelogram mechanism

In this section, the proposed MOHEPSO algorithm introduced in Section 4 with a population of size, maximum iteration, \( E_{constant} \) and \( R_{neighborhood} \) 250, 300, 200 and 0.2 is used to determine the appropriate parameters of the fuzzy inverse dynamics controller for the five-bar linkage mechanism. In this paper, weighting normalized summation of angle errors and weighting normalized summation of control effort of the both links are regarded as the objective functions and calculated using Equation (34).

Weighting normalized summation of angle errors

\[
W_1 \times \int |q_1| \, dt + W_2 \times \int |q_2| \, dt
\]

Weighting normalized summation of control effort

\[
W_1 \times \int |u_1| \, dt + W_2 \times \int |u_2| \, dt \tag{34}
\]
where $W_1$ and $W_2$ are the weighting coefficients. $q_1$ and $q_2$ are displacement of joints. $u_1$ and $u_2$ are the input generalized forces.

The vector $[VB, PO, ZB, HS, HM, HB, M_1, \Delta M_1, M_2, \Delta M_2]$ is the chosen parameters of fuzzy inverse dynamics controller. VB, PO, ZB, HS, HM and HB are input of the membership function. $M_1$, $\Delta M_1$, $M_2$ and $\Delta M_2$ are positive coefficients of the fuzzy inverse dynamics controller. The weighting normalized summation of angles errors and weighting normalized summation of control effort are the functions of this vector’s components. This means that by selecting various values for the selective parameters, we can make changes in the weighting normalized summation of angles errors and weighting normalized summation of control effort. Indeed, this is an optimization problem with two objective functions and ten decision variables ($VB, PO, ZB, HS, HM, HB, M_1, \Delta M_1, M_2, \Delta M_2$). The design variables corresponding to the optimal design points A, B, and C are illustrated in Table 7. The parameters of the multi-objective algorithms are chosen in Section 4 in a similar way.

In Figure 9, points A and C stand for the best weighting normalized summation of angles errors and weighting normalized summation of control effort, respectively. This figure illustrates that all the optimum design points in the Pareto front are non-dominated and can be chosen by the designer based on the design criteria. In Figure 9, point B can be regarded as a trade-off optimum choice when the minimum values of both the weighting normalized summation of angles errors and weighting normalized summation of control effort are considered. The real tracking trajectories of the optimal design points A, B, and C are shown in Figure 10. The tracking errors of the optimal design points A, B, and C are shown in Figure 11. Moreover, Figure 12 illustrates the control effort of the parallelogram mechanism for the optimal design points A, B and C.
Table 7. Design variables and objective functions of optimum points A, B and C for fuzzy inverse dynamics controller of the parallelogram mechanism

| Optimum design point | A   | B   | C   |
|----------------------|-----|-----|-----|
| Weighting normalized summation of angle errors | 0.03| 0.27| 0.91|
| Weighting normalized summation of control effort | 0.96| 0.42| 0.04|
| VB                   | 0.88| 0.95| 1.07|
| PO                   | 1.40| 1.38| 1.23|
| ZB                   | 4.10| 4.10| 3.94|
| HS                   | 2.25| 2.16| 1.42|
| HM                   | 3.92| 3.20| 4.00|
| HB                   | 3.47| 4.51| 5.23|
| M₁                   | 5.58| 5.59| 5.52|
| ΔM₁                  | 1.38| 1.60| 1.38|
| M₂                   | 4.55| 1.67| 1.40|
| ΔM₂                  | 0.41| 0.30| 0.30|

Figure 9. The obtained Pareto fronts by using the proposed algorithm for the optimal fuzzy inverse dynamics controller of the parallelogram mechanism.

Figure 10. Simulations of the desired trajectory and the trajectory tracking of optimal design points A, B, and C shown in the Pareto front for (a) angle $\theta_1$ and (b) angle $\theta_2$. 
9. Conclusions

This paper has proposed a new and simple generic evolutionary multi-objective optimization algorithm based on HEPSO which is called MOHEPSO. Different properties of the proposed MOHEPSO method have made it robust and competitive among the other existing methods in the literature. In order to supply a proper infrastructure for comparison, a fuzzy elimination technique is utilized to prune the archive instead of previous techniques in the literature. Also, the Sigma method is used to find the personal best positions of particles. Moreover, a neighborhood relationship among all the non-dominated solutions based on the Euclidean distance is defined to find and update the global best positions. Several experiments have been conducted to compare the performance of the proposed method with other recent and well-known multi-objective evolutionary algorithms. As the experiments demonstrate, the obtained optimal Pareto fronts in the proposed algorithm (MOHEPSO) outperform three effective multi-objective optimization algorithms (i.e. MOEA/D, MOPSO Ingenious and NSGA-II) for different complex multi-objective test functions in terms of both solution diversity and convergence. Finally, the proposed algorithm is used for the optimum design of a fuzzy inverse dynamics controller for a parallelogram five-bar linkage mechanism. Two conflicting objective functions, the weighting normalized summation of angle errors and weighting normalized summation of control effort, are regarded to design the optimal controller. Three points of Pareto front, obtained from MOHEPSO, are selected to work out the ten parameters of the optimal fuzzy inverse dynamic control; hence, the designer has the ample opportunity to select the finest point based on the design criteria. The result elucidated that the first selected point had minimum normalized summation of angles errors and maximum normalized summation of control effort. However, the third point was the direct opposite of the first one. Indeed, it had maximum normalized summation of angles errors and minimum normalized summation of control effort. Thus, the designer can regard the second well-chosen point as the trade-off optimum choice.
Funding
The authors received no direct funding for this research.

Author details
A. Farokhi1
E-mail: aminfarokhi11@yahoo.com
M.J. Mahmoodabadi1
E-mail: Mahmoodabadi@sjiran.ac.ir
ORCID ID: http://orcid.org/0000-0002-4249-8623
1 Department of Mechanical Engineering, Sirjan University of Technology, Sirjan, Iran.

Citation information
Cite this article as: Optimal fuzzy inverse dynamics control of a parallelogram mechanism based on a new multi-objective PSO, A. Farokhi & M.J. Mahmoodabadi, Cogent Engineering (2018), 5: 1443675.

References
Abbasian, M. A., Nematodabi-pour, H., & Amoozegar, M. (2015). A clustering based archive multi objective gravitational search algorithm. Fundamenta Informaticae, 138(4), 387–409. https://doi.org/10.3233/FI-2015-1218
Akay, B., & Karaboga, D. (2012). A modified Artificial Bee Colony algorithm for real-parameter optimization. Information Sciences, 192, 120–142. https://doi.org/10.1016/j.ins.2010.07.015
Al Mouhneyed, N., Petrovski, A., & McCall, J. (2014). DP-MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces. Evolutionary Computation, 22(1), 47–77. https://doi.org/10.1162/evco_a_00104
Atashkari, K., NarimanZadeh, N., Ghaivami, A. R., Mahmoodabadi, M. J., & AghaieNeshad, F. (2011). Multi-objective optimization of power and heating system based on artificial bee colony. In Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on IEEE (pp. 64–68).
Azzouz, R., Bechkih, S., & Said, L. B. (2017). Dynamic multi-objective optimization using evolutionary algorithms: A survey. In Recent advances in evolutionary multi-objective optimization (pp. 31–70). Springer International Publishing.
Bratton, D., & Kennedy, J. (2007). Defining a standard for particle swarm optimization. In Proceedings of the IEEE Symposium on Swarm Intelligence, pp. 120–345.
Chang, W. D. (2007). A multi-crossover genetic approach to multivariable PID controllers tuning. Expert Systems with Applications, 33, 620–626. https://doi.org/10.1016/j.eswa.2006.06.003
Chen, Y., Mei, G., Ma, G., Lin, S., & Gao, J. (2014). Robust adaptive inverse dynamics control for uncertain robot manipulator. International Journal of Innovative Computing, Information and Control, 10(2), 575–587.
Cheng, S., Chen, M. Y., Wei, R. J., & Wang, F. Z. (2014). Optimal placement of distributed generation units in distribution systems via an enhanced multi-objective particle swarm optimization algorithm. Journal of Zhejiang University Science Computer Electronic, 15(4), 300–311. https://doi.org/10.1631/jzus.C1300250
Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182–197. https://doi.org/10.1109/4235.996017
Ding, Z. (2017). Research of improved particle swarm optimization algorithm. AIP Conference Proceedings, 1839(1). AIP Publishing.
Donoso, Y., & Fabregat, R. (2016). Multi-objective optimization in computer networks using metaheuristics. Boca Raton: CRC Press.
Giusti, A., Malzahn, J., Tsagarakis, N., & Althoff, M. (2017). Combined inverse-dynamics/passivity-based control for robots with elastic joints. In Proceedings of the IEEE International Conference on Robotics and Automation. Singapore, (pp. 5281–5288).
Gong, M., Cai, Q., Chen, X., & Ma, L. (2014). Complex network clustering by multi-objective discrete particle swarm optimization based on decomposition. IEEE Transactions on Evolutionary Computation, 18(1), 82–97. https://doi.org/10.1109/TEVC.2013.2260862
Huang, C. L., Huang, W. C., Chang, H. Y., Yeh, Y. C., & Tsai, C. Y. (2013). Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering. Applied Soft Computing, 13(9), 3864–3872. https://doi.org/10.1016/j.asoc.2013.05.003
Hughes, E. J. (2008). Fitness assignment methods for many-objective problems. In J. Knowles, D. Corne, K. Deb, & D. R. Chair (eds.), Multi-objective problem solving from nature. Natural Computing Series. Berlin Heidelberg: Springer. https://doi.org/10.1007/978-3-540-72964-8
Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
Khisti and, S., & Zandle, M. (2017). Comparisons of some improving strategies on NSGA-II for multi-objective inventory system. Journal of Industrial and Production Engineering, 34(1), 61–69. https://doi.org/10.1080/21681015.2016.1210681
Kukkonen, S., & Coello, C. A. C. (2017). Generalized differential evolution for numerical and evolutionary optimization. In NEOS 2015, Springer International Publishing, pp. 253–279.
Liu, H. L., Chen, L., Deb, K., & Goodman, E. (2016). Investigating the effect of imbalance between convergence and diversity in evolutionary multi-objective algorithms. IEEE Transactions on Evolutionary Computation, 21(3), 408–425.
Mahmoodabadi, M. J., Bagheri, A., Mostaghim, S. A., & Bisheban, M. (2011). Simulation of stability using Java application for Pareto design of controllers based on a new multi-objective particle swarm optimization. Mathematical and Computer Modelling, 54(5), 1584–1607. https://doi.org/10.1016/j.mcm.2011.04.032
Mahmoodabadi, M. J., Salahshoor Mottaghi, Z., & Bagheri, A. (2014). High exploration particle swarm optimization. Information Sciences, 273, 101–111. https://doi.org/10.1016/j.ins.2014.02.150
Mahmoodabadi, M. J., Toherkhorsandi, M., & Bagheri, A. (2014). Optimal robust sliding mode tracking control of a biped robot based on ingenious multi-objective PSO. Neurocomputing, 124, 194–209. https://doi.org/10.1016/j.neucom.2013.07.009
Meza, J., Espitia, H., Montenegro, C., Giménez, E., & González-Crespo, R. (2017). MOVPSO: Vortex multi-objective particle swarm optimization. Applied Soft Computing, 52, 1042–1057. https://doi.org/10.1016/j.asoc.2016.09.026
Mirjalili, S., Soremi, S., Mirjalili, S. M., & Coelho, L. D. S. (2016). Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. Expert Systems with Applications, 47, 106–119. https://doi.org/10.1016/j.eswa.2015.10.039
Mostaghim, S., & Teich, J. (2003). Strategies for finding good local guides in multi-objective particle swarm optimization. In Proceedings of IEEE Swarm Intelligence Symposium (pp. 26–33).
Padhye, N., Branke, J., & Mostaghim, S. (2009). Empirical comparison of MOPSO methods – guide selection and diversity preservation. In Proceedings of the IEEE Congress of Evolutionary Computation (pp. 2516–2523).
Pal, M., & Bandyopadhyay, S. (2016). Reliability of convergence metric and hypervolume indicator for many-objective optimization. In Control, Instrumentation Energy & Communication (CIEC), 2nd International Conference on IEEE (pp. 511–515).

Peng, W., Lin, Z., & Su, J. (2009). Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques. IET Control Theory & Applications, 3(6), 701–711. https://doi.org/10.1049/iet-cta.2008.0259

Piltan, F., & Tayebi Haghighi, S. (2012). Design gradient descent optimal sliding mode control of continuum robots. IAES International Journal of Robotics and Automation (IAES-IJRA), 1(4), 175–189.

Rao, M. A., & Kumar, N. P. (2017). Speed control of parallel connected DSIM fed by six phase inverter with IFOC strategy using ANFIS. International Journal of Research and Engineering, 4(9), 244–250.

Razmi, J., Jafari, E., & Amin, S. H. (2016). An intuitionistic fuzzy goal programming approach for finding pareto optimal solutions to multi-objective programming problems. Expert Systems with Applications, 65, 181–193. https://doi.org/10.1016/j.eswa.2016.08.048

Spong, M. W., & Vidyasagar, M. (2008). Robot dynamics and control. Hoboken, NJ: John Wiley & Sons.

Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J., & Harley, R. G. (2008). Particle swarm optimization: Basic concepts, variants and applications in power systems. IEEE Transactions on Evolutionary Computing, 12(2), 171–195. https://doi.org/10.1109/TEVC.2007.896686

Wang, H. (2011). On adaptive inverse dynamics for free-floating space manipulators. Robotics and Autonomous Systems, 59(10), 782–788. https://doi.org/10.1016/j.robot.2011.05.013

Wang, Y., Wang, L., Yan, X., & Shen, P. (2017). Fuzzy immune particle swarm optimization algorithm and its application in scheduling of MVB periodic information. Journal of Intelligent & Fuzzy Systems, 32(6), 3797–3807. https://doi.org/10.3233/JIFS-152067

Wang, Y., & Yang, Y. (2009). Particle swarm optimization with preference order ranking for multi-objective optimization. Information Sciences, 179(12), 1944–1959. https://doi.org/10.1016/j.ins.2009.01.005

Zeinali, M., & Notash, L. (2010). Fuzzy logic-based inverse dynamic modeling of robot manipulators. Transactions of the Canadian Society for Mechanical Engineering, 34(1), 137–150.

Zhan, Z., Zhang, J., Li, Y., & Chung, H. S. (2009). Adaptive particle swarm optimization. IEEE Transactions on Systems Man and Cybernetics. Part B, 39(6), 1362–1381. https://doi.org/10.1109/TSMCB.2009.2015956

Zhang, Q., & Li, H. (2007). A multi-objective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 11(6), 712–731. https://doi.org/10.1109/TEVC.2007.892759