Analysis and Multi-objective Optimization of a Kind of Teaching Manipulator

Zhun Fan[a,b,c,d], Yugen You[a,b], Xinye Cai[f,g,*], Haodong Zheng[a,b], Guijie Zhu[a,b], Wenji Li[a,b], Akhil Garg[e], Kalyanmoy Deb[h], Erik Goodman[h]

[a]*Department of Electronic and Information Engineering, Shantou University, 515063, Guangdong, China*
[b]*Key Lab of Digital Signal and Image Processing of Guangdong Province, Shantou University, 515063, Guangdong, China*
[c]*Key Laboratory of Intelligent Manufacturing Technology (Shantou University), Ministry of Education, 515063, Guangdong, China*
[d]*State Key Lab of Digital Manufacturing Equipment & Technology, Huazhong University of Science and Technology, 43003, Wuhan, China*
[e]*Department of Mechanical Engineering, Shantou University, Guangdong, 515063, China*
[f]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Jiangsu, 210016, China*
[g]*Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, 210023, China*
[h]*BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, Michigan, USA*

Abstract

Designing and setting manipulator trajectories in a programming system can be a tedious and time-consuming task for manufacturers. In this paper, one kind of six degree-of-freedom (DOF) teaching manipulator is designed and developed for conveniently setting and recording trajectories for industrial robots. A constrained multi-objective optimization problem is formulated to optimize the design of the teaching manipulator. Two performance indexes, i.e. the magnitude of the peak operating force and difference between the maximum and minimum magnitude of operating forces are adopted as the objectives. Two PPS-based (push and pull search) algorithms, including PPS-MOEA/D and PPS-M2M, are suggested to solve the formulated CMOP. Several state-of-the-art CMOEAs, including MOEA/D-ACDP, MOEA/D-CDP, NSGA-II-CDP and CM2M, are also tested. The experimental results indicate that PPS-MOEA/D has the best performance among the six compared algorithms, and the PPS-based methods as a group outperform their counterparts without adopting the PPS framework, which demonstrates the superiority of the PPS framework for solving real-world optimization problems.

Keywords: Teaching manipulator design, Robot modeling, Constrained multi-objective optimization

1. Introduction

Robots are a kind of complex and tightly integrated mechantronic systems with multiple subsystems. It usually takes a long time for engineers to handle interactions between different subsystems to find a feasible and mature design, which involves a number of repetitive and routing tasks. The application of design automation on robots is a promising field, which involves systematic and iterative modeling and optimization efforts. For example, Lipson and Pollack [1] proposed a study about automatic design and manufacture of robots, which demonstrates the potential of robot design automation. To reduce non-creative and routine processes of design, Tarkian et al. [2, 3] proposed a kind of high level CAD templates for robot design, which provides a high fidelity CAD model to help engineers make more comprehensive choices in the early stage of design.

Optimization is an essential step in robot design automation. When optimizing a robot, a set of factors should be considered simultaneously. For example, the compromise between the low cost and the high performance [4], the high efficiency and the light weight [5], and the high dexterity and high stiffness [6]. Furthermore, the design of robots often involves a number of constraints, such as the limitations of the subsystems [7, 8], the constraints to preserve the geometry properties[9, 10, 11], the performance requirements [12], etc. A robot design optimization can be formulated as a constrained multi-objective optimization problem (CMOP). Without lose of generality, a CMOP can be defined as follows [13]:

$$\begin{cases} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T \\ \text{subject to} & g_i(\mathbf{x}) \geq 0, i = 1, \ldots, q \\ & h_j(\mathbf{x}) = 0, j = 1, \ldots, p \\ & \mathbf{x} \in \mathbb{R}^n \end{cases} \quad (1)$$

where $F(x) = (f_1(x), f_2(x), \cdots, f_m(x))^T$ is an objective vector with m dimensions. $g_i(x) \geq 0$ is an inequality constraint, and q is the number of inequality constraints. $h_j(x) = 0$ is an equality constraint, and p represents the number of equality constraints. $x \in \mathbb{R}^n$ is a decision vector with n dimensions. The equality constraints are usually converted into inequality constraints by introducing an extremely small positive number σ as follows:

$$h_j(x)' \equiv \sigma - |h_j(x)| \geq 0 \quad (2)$$

Constrained multi-objective evolutionary algorithms (CMOEAs) are widely used to solve CMOPs in recent years due to the following reasons.

---
*Corresponding author
Email address: xinye@nuaa.edu.cn (Xinye Cai)

1. The first reason is that CMOEAs are population-based algorithms which can obtain a set of optimal solutions in a single run.

2. CMOEAs are gradient-free methods, and can solve CMOPs with non-differentiable or discrete objectives and constraints.

3. CMOEAs can solve CMOPs with mixed variables, i.e., both continuous and discrete variables. As CMOEAs are heuristic algorithms, they can jump out of the local optimal and achieve the global optimal.

Compared with unconstrained multi-objective evolutionary algorithms (MOEAs), CMOEAs need to maintain not only the balance between the convergence and diversity of the working population, but also the balance between minimizing the objectives and satisfying the constraints. The existing CMOEAs can be classified into two different types according to their selection mechanisms. The first type is the dominance-based CMOEA, such as NSGA-II-CDP [14], which uses non-dominated sorting method to select solutions into the next generation. The other type is the decomposition-based CMOEA. MOEA/D-CDP [15] is a representative example of this type, which decomposes a CMOP into a number of constrained single-objective optimization problems(CSOPs) and each CSOP is solved in a collaborative way.

Recently, Fan et al [16] proposed a push and pull search (PPS) framework for solving CMOPs. PPS framework has two different search stages — the push and pull search. In the push stage, a CMOP is optimized without considering any constraints, so that the working population can get across infeasible regions efficiently. The constrained landscape of the CMOP is also probed and estimated, which can help to set parameters of constraint-handling mechanisms in the pull search stage. Infeasible individuals in the working population are pulled to the feasible and non-dominated regions in the pull stage.

In this paper, we formulate a teaching manipulator design optimization as a CMOP. Two PPS-based algorithms, including PPS-MOEA/D and PPS-M2M (embedding multi-objective to multi-objective decomposition method in the PPS framework), are suggested to solve the formulated CMOP. The suggested PPS-MOEA/D [16] and PPS-M2M are compared with four state-of-the-art CMOEAs, including MOEA/D-ACDP [17], MOEA/D-CDP [15], NSGA-II-CDP [14] and CM2M [18]. The experimental results indicate that PPS-MOEA/D has the best performance among the six compared algorithms, and the PPS-based methods are better than their counterparts without adopting PPS framework. The main contributions of the paper are as follows:

1. A model of a teaching manipulator is built up, and design optimization of the teaching manipulator is formulated as a CMOP.

2. A PPS-based algorithm called PPS-M2M is proposed. The experimental results demonstrate that the PPS-based methods outperform their counterparts without adopting PPS framework, which demonstrate the superiority of the PPS framework for solving real-world optimization problems.

3. Among the six compared CMOEAs, PPS-MOEA/D performs the best on the teaching manipulator design optimization problem.

The rest of this paper is organized as follows. In Section 2, the modeling of the teaching manipulator is detailed, with its optimization formulated as a CMOP. In Section 3, a modified PPS-based algorithm called PPS-M2M is proposed. Section 4 designs experiments to compare PPS-MOEA/D and PPS-M2M with four other CMOEAs, including MOEA/D-ACDP, MOEA/D-CDP, NSGA-II-CDP and CM2M. Finally, conclusions and some future research directions are summarized in Section 5.

2. Problem formulation

2.1. Design of the teaching manipulator

Designing and setting manipulator trajectories in a programming system is still a tedious and time-consuming task, especially when the trajectories need to be redesigned and reset frequently to adapt to different applications in a flexible production line. To simplify the process, a teaching manipulator is designed and developed in this paper. As shown in Fig.1, a human operator can hold the end-effector of a teaching manipulator and conveniently drive it along a proper trajectory. The rotation of each joint is recorded as a motion sequence while the end-effector moves along the trajectory. By given the same motion sequence, one or more manipulators can repeat the trajectory of the teaching manipulator in the production line. In the teaching process, the feeling of the motion of the human operator might be severely influenced if the design of the teaching manipulator is problematic. A large resistance force or variation of the resistance force from the end effector can make the operator fail to maneuver the manipulator properly as wanted. Furthermore, the teaching manipulator should be capable of maintaining static balance of the joints when stopping anytime in any position. However, it is not easy for a human designer to propose an optimal design of a teaching manipulator considering all aforementioned aspects simultaneously. Therefore, we conduct a robot design optimization process to help the designer to achieve the optimal design of the robot. In this section, we build up a model of the teaching manipulator and formulate its optimization problem as a CMOP.

The proposed teaching manipulator has six DOFs, as illustrated in Fig.2. The link lengths of the teaching manipulator are fixed, which are equal to those of the manipulator in the production line, so that the motion data of the teaching manipulator can be directly used as the motion sequences of manipulators in the production line. There are no actuators in the teaching manipulator because it is actuated by a human operator. Six encoders are mounted in the corresponding joints to record the rotation of each joint. To reduce the load of the human operator, specific structures are designed to keep the joints in balance in static conditions, including two counterweights at Joint 3 and 5, a pneumatic balancer for Joint 2, and three frictional mechanisms inside Joint 1, 2 and 3. The mechanism of the pneu-
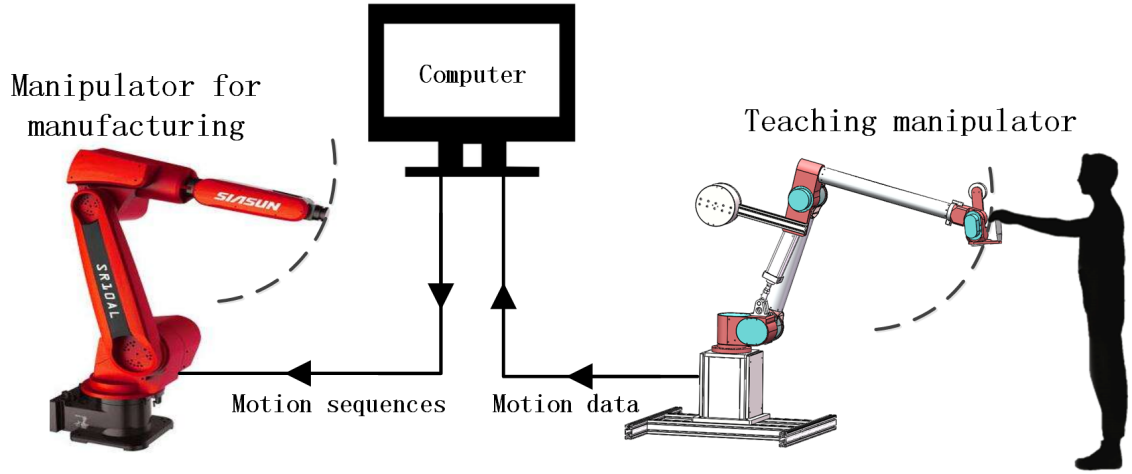
Figure 1: The human operator drives the teaching manipulator and demonstrates the trajectory. The computer system collects the motion data from the teaching manipulator and the motion sequences are transmitted as the command for the manipulator in the production line.
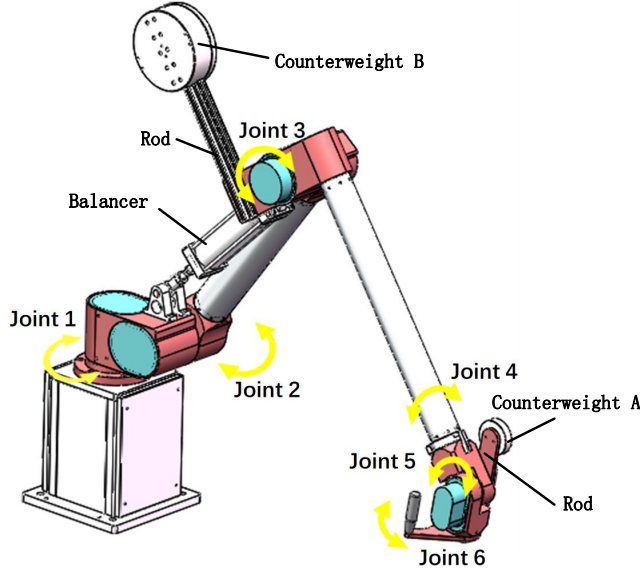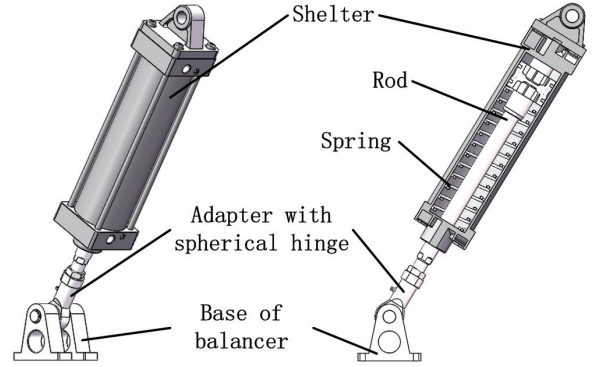

Figure 2: A 6-DOF teaching manipulator

Figure 3: The mechanism for the pneumatic balancer

Figure 4: The joint mechanism for Joint 1

matic balancer and the frictional mechanisms inside Joint 1 are as shown in Fig. 3 and 4, respectively.

The forward kinematics of the teaching manipulator is formulated based on the Denavit-Hartenberg (D-H) convention [19]. The coordinate frame $o_0x_0y_0z_0$ is the reference inertial coordinate. The coordinate frames $o_ix_iy_iz_i$ ($i = 1, 2, \cdots, 6$) are assigned based on the design of the manipulator coordinate system, which is shown in Fig.5. The D-H parameters are listed in Table 1. It is notable that we assume that the end-effector and the Joint 6 share the same coordinate frame.

In this study, the method of Lagrange is utilized to establish the dynamic model of the teaching manipulator. The Lagrange equation is given as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_{frc}(\dot{q}) = J^T f_{opr}(t) + \tau_b \quad (3)$$

where $\ddot{q} \in \mathbb{R}^6$ is the generalized angular acceleration, $M(q) \in$
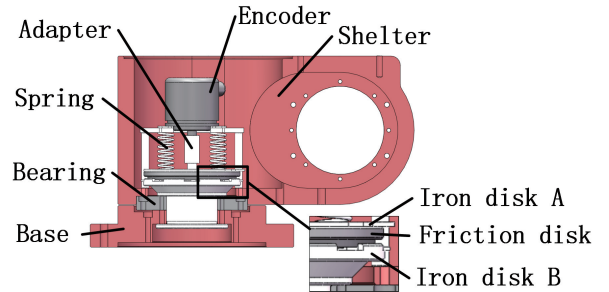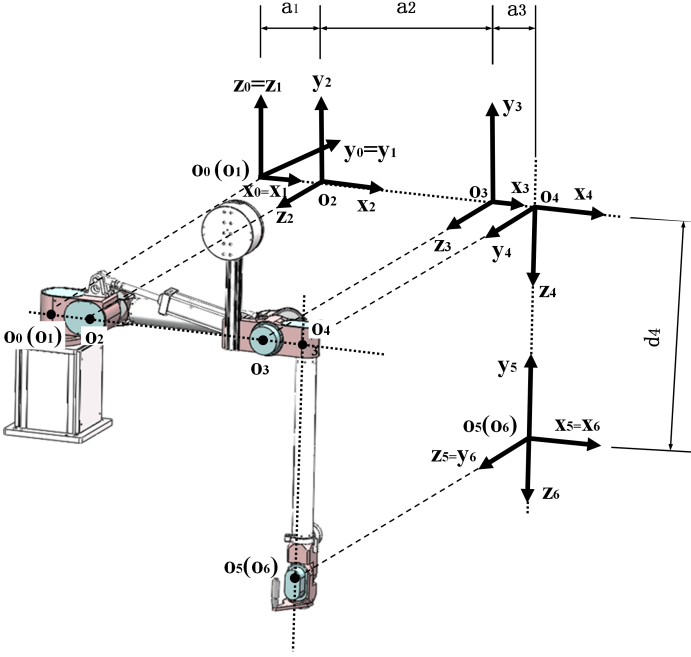
3

Figure 5: Manipulator coordinate system

Table 1: D-H Parameters of the teaching manipulator.

$Joint_i$	α_i	a_i	d_i	θ_i
1	$\pi/2$	0.160	0	q_1
2	0	0.790	0	q_2
3	$\pi/2$	0.155	0	q_3
4	$-\pi/2$	0	0.995	q_4
5	$\pi/2$	0	0	q_5
6	0	0	0	q_6

$\mathbb{R}^{6\times6}$ is an inertia matrix, which is symmetric and positive definite. $G(q) = [G_1, G_2, \cdots, G_6]^T$ is a six-dimension vector which represents the gravitational moments acting on joints. Moreover, the mass of counterweights and the length of the rods, which connect the counterweights and the corresponding joints, can change the distribution of mass and the inertia of the links, thus influencing the operating force. $C(q, \dot{q}) \in \mathbb{R}^{6\times6}$ is the centripetal and Coriolis matrix [20]. In this matrix, the elements only involving $\dot{q}$ are centripetal which are related to the centripetal force of each joint, while the rest elements involving q are Coriolis which are related to the Coriolis force of each joint. $J \in \mathbb{R}^{6\times6}$ is the Jacobian matrix of the teaching manipulator with respect to the reference inertial frame, while $f_{opr}(t) = [F_x, F_y, F_z, n_x, n_y, n_z]$ is the operating force and torque vector of the human operator, in which F_x, F_y, F_z are the components of the operating force, and n_x, n_y, n_z are the components of the operating torques. The magnitude of the operating torque is tiny, because the three joints of the spherical wrist are not equipped with friction disks and the counterweight A can help keep Joint 5 in balance. Thus, rather than the magnitude of the operating torque, the magnitude of the operating force should be a focus in design. $\tau_b \in \mathbb{R}^6$ is the moment vector of force act-
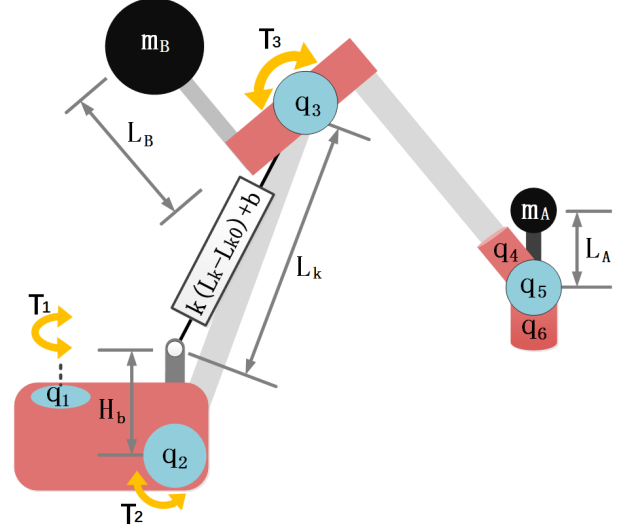

Figure 6: A sketch of the teaching manipulator

ing on joints by the balancer. $\tau_{frc}(\dot{q}) \in \mathbb{R}^6$ is a friction moment vector of the friction disks on joints. A more detailed analysis of the modeling is given in the supplemental material.

2.2. Design variables

Ten design variables are considered in the optimization problem, which is defined as:

$$x = [m_A, L_A, m_B, L_B, k, H_b, b, T_1, T_2, T_3]$$

where, m_A and m_B denote the masses of the counterweights of Joint 5 and Joint 3, while L_A and L_B denote the length of the corresponding rods, respectively. k is the stiffness coefficient of the spring inside the balancer, H_b is the vertical distance between the lower attachment point of the balancer and the rotational axis of Joint 2. b is the pulling force of the pneumatic cylinder. T_i ($i = 1, 2, 3$) is the kinetic friction moment of the ith friction disk. All the variables are shown in Fig.6.

2.3. Objective functions

To guarantee that the feeling of the motion of the human operator is not influenced too much during the operation of the manipulator, two objectives are considered: the magnitude of the peak operating force and the difference between the maximum and minimum magnitude of the operating force.

2.3.1. Peak operating force

In the study, we minimize the magnitude of the peak operating force to ensure that a human operator can easily operate the teaching manipulator without feeling much resistance. The magnitude of the operating force can be obtained through the following equation.

$$f_{mag}(x) = \sqrt{F_x^2 + F_y^2 + F_z^2} \tag{4}$$

Thus, the objective function can be written as follows:

$$f_1(x) = \max_t f_{mag}(x, t) \tag{5}$$

2.3.2. Difference between the maximum and minimum operating forces

We also minimize the difference between the maximum and minimum magnitude of the operating forces, because the feeling of motion of human operators may be influenced by a large variation of operating force during operation of the teaching manipulator. The objective function can be written as follows:

$$f_2(x) = \| \max_t f_{mag}(x, t) - \min_t f_{mag}(x, t) \| \tag{6}$$

2.4. Constraints

It is desired that the teaching manipulator is able to maintain the angular positions of the Joint 1,2 and 3 when it stops in any position of the trajectory. Thus, we have the following three constraints:

$$\max_t |G_2(q_t, m_A, L_A, m_B, L_B) - \tau_{b,2}(q_t, k, H_b, b)| \le T_2 \tag{7}$$

where $\tau_{b,2}$ is the second element of the moment vector of force τ_b, which represents the moment generated from the balancer acting on Joint 2.

$$\max_t |G_3(q_t, m_A, L_A, m_B, L_B)| \le T_3 \tag{8}$$

To keep the balance of Joint 5, the following constraint has to be satisfied.

$$\max_t |G_5(q_t, m_A, L_A)| \le \epsilon \tag{9}$$

Here, we give $\epsilon = 0.1 N \cdot m$. A more detailed analysis can be found in the supplemental material.

Table 2: The ranges of design variables.

Variables	Range	Units	Original design
m_A	[0, 20]	kg	1.5
L_A	[0, 0.2]	m	0.18
m_B	[0, 50]	kg	15
L_B	[0, 0.8]	m	0.46
k	[0, 50000]	N/m	3730
H_b	[0.11, 0.50]	m	0.12
b	[450, 4000]	N	2500
T_1	[0, 90]	$N \cdot m$	75.7
T_2	[0, 90]	$N \cdot m$	75.7
T_3	[0, 90]	$N \cdot m$	75.7

The ranges of the design variables are listed in Table 2, which are provided by a human engineer. Other relevant constant parameters are listed in Table 3, which are obtained from the virtual prototype in SolidWorks, where m_{Li} is the mass of Link i, ρ_A and ρ_B are the masses per unit length of the connecting rods of counterweight A and counterweight B, respectively.

Table 3: The values of the related parameters.

Parameter	value	Units
m_{L1}	13.0	kg
m_{L2}	11.7	kg
m_{L3}	5.3	kg
m_{L4}	4.5	kg
m_{L5}	2.0	kg
m_{L6}	1.5	kg
ρ_A	1.8	kg/m
ρ_B	3.7	kg/m

2.5. Trajectory design

To evaluate a design, a specific trajectory must be defined as an essential part of the application scenario. We choose a trajectory adopted by the work of [8] and [21], which is commonly used for performance evaluation of 6-DOF manipulators. The trajectory can be formulated in the base coordinate system by the following equations:

$$
\begin{aligned}
X_{ef} &= 1.2 - 0.3 \times (1 - \cos(\frac{2\pi}{20}t)) \\
Y_{ef} &= 0.8 \sin(\frac{2\pi}{20}t) \\
Z_{ef} &= 0.3 + 0.5 \cos(\frac{2\pi}{20}t)
\end{aligned}
\tag{10}
$$

The functions are all with unit of m. The Euler angle for the end-effector are given as $[0, \pi, \pi]$, which implies that the end-effector remains vertical and points at the ground during the prescribed motion. We equally divide the trajectory into $N = 2000$ segments with respect to time and give each segment $0.01s$. It is assumed that the end-effector has uniformly accelerated motion in each segment.

To figure out the operating force in the segmented positions of the aforementioned trajectory, we build up a simulation model of teaching manipulator using the Robotics toolbox for MATLAB [22]. The model of the teaching manipulator in the simulation environment, and the trajectory design of the end effector are shown in Fig.7.

3. PPS framework

This section gives a detailed description of the PPS framework. Besides, the details of two instantiation of PPS-based algorithms, PPS-MOEA/D [16] and PPS-M2M are provided.

PPS framework consists of two different stages: push search and pull search. In the stage of push search, the constraints are ignored and the population converges to the unconstrained PF, so that the population can cross the infeasible region without being blocked. Once the population in the push stage has not been changed in a number of generations, the state of search is switched to the pull stage. In the pull stage, the infeasible solutions are pulled to the feasible and non-dominated regions by using constraint-handling mechanisms. Two instantiation of
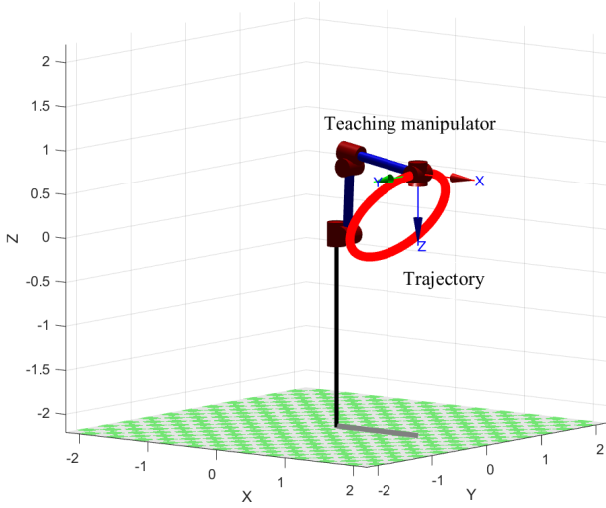
Figure 7: The simulation of the manipulator and the trajectory in MATLAB.

PPS-based algorithms are demonstrated in the following sub-sections.

3.1. PPS-MOEA/D

In PPS-MOEA/D, MOEA/D algorithm is integrated in the PPS framework. In the push search stage, MOEA/D is employed to search for non-dominated solutions disregarding any constraints. The Tchebycheff [23] decomposition method is adopted in this paper, with the definition of the decomposition function given as follows.

$$g^{te}(\mathbf{x}, \lambda^i, z^*) = \max_{j=1,...,m} \frac{1}{\lambda^i_j}(|f_j(\mathbf{x}) - z^*_j|) \qquad (11)$$

where λ^i is a weight vector, and $\sum_{j=1,...,m} \lambda^i_j = 1, \lambda^i_j \geq 0$. z^* is the ideal point, and $z^*_j = \min_{k=1,...,N} f_j(\mathbf{x}^k)$.

In the pull search stage, the infeasible solutions are pulled to the feasible and non-dominated regions by using an improved epsilon constraint handling technique [24, 25]. The ε is set as follows [25]:

$$\varepsilon(k) = \begin{cases} (1-\tau)\varepsilon(k-1), \text{if } rf_k < \alpha \\ \varepsilon(0)(1 - \frac{k}{T_c})^{cp}, \text{if } rf_k \geq \alpha \end{cases} \qquad (12)$$

where rf_k represents the proportion of feasible solutions in the k-th generation. α is is a threshold to control the searching preference between the feasible and infeasible regions. τ is a parameter to control the speed of reducing the relaxation of constraints in the case of $rf_k < \alpha$. cp is a parameter to control the speed of reducing relaxation of constraints in the case of $rf_k \geq \alpha$. The initial value $\varepsilon(0)$ is set to the maximum overall constraint violation of the working population at the end of the push search. $\varepsilon(k)$ is updated until the generation counter k reaches the control generation T_c. The pseudocode of PPS-MOEA/D can be found in [16].
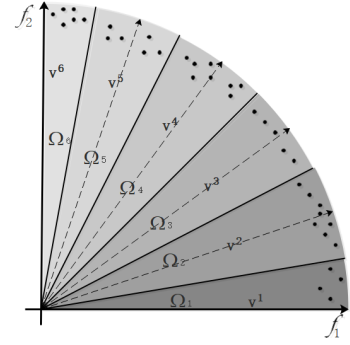

Figure 8: An illustration of the setting of $\mathbf{v}^j$ and Ω_k.

3.2. PPS-M2M

In this section, we introduce a new PPS-based algorithm – PPS-M2M, in which two basic components are included. The first component is the M2M decomposition method, and the second component is the PPS framework.

3.2.1. M2M

M2M is a kind of decomposition-based method for solving multi-objective optimization problems. Unlike MOEA/D, M2M decomposes a MOP into a set of simple MOPs. Each simple MOP is a subproblem which is defined as follows:

$$\begin{cases} \text{minimize} & F(x) = (f_1(x), ..., f_m(x)) \\ \text{subject to} & x \in \prod_{i=1}^n [a_i, b_i] \\ & F(x) \in \Omega_k \end{cases} \qquad (13)$$

where Ω_k is the k-th sub-region, and it is defined as follows:

$$\Omega_k = \{\mathbf{u} \in \mathbb{R}^m_+ | \langle \mathbf{u}, \mathbf{v}^k \rangle \leq \langle \mathbf{u}, \mathbf{v}^j \rangle \text{ for any } j = 1, ..., K\} \quad (14)$$

where $\langle \mathbf{u}, \mathbf{v}^j \rangle$ is the acute angle between $\mathbf{u}$ and $\mathbf{v}^j$. $\mathbf{v}^1, ..., \mathbf{v}^K$ are K unit vectors in the first octant of the objective space ($\mathbb{R}^m_+$). An illustration of the setting of $\mathbf{v}^j$ is shown in Fig. 8.

In M2M, each subpopulation (P_k) maintains S_k solutions. If the number of solutions in P_k is fewer than S_k, we select $S_k - |P_k|$ solutions randomly from the entire population and add them to the subpopulation. If the number of solutions in P_k is greater than S_k, then the best S_k solutions are selected according to non-dominated sorting.

3.2.2. Push and Pull Search

In the push search stage, a CMOP is divided into a set of subproblems. Each subproblem is solved by employing a subpopulation. The pseudocode of push search is given in Algorithm 1. In line 2, non-dominated sorting is carried out on the sup-population P_k. In lines 4-7, a number of solutions are selected into P'_k until the number of solutions in P'_k is greater than S_k. Lines 8-10 select $S_k - |P'_k|$ solutions into P'_k from F_i. In line 11, the sup-population P_k is updated by setting $P_k = P'_k$.

In the pull stage, we need to prevent the population from falling into local optima, and balance the search between feasible and infeasible regions. To achieve these goals, an improved epsilon constraint-handling mechanism [25] is used to

6

Algorithm 1: Push Subproblems

---

1 **Function** *PushSubproblems(P_k,S_k)*
2   F = nondominated-sort(P_k), $F = (F_1, F_2, ...)$;
3    $P'_k = \varnothing$ *and* $i = 1$;
4   **while** $|P'_k| + |F_i| \leq S_k$ **do**
5      $P'_k = P'_k \bigcup F_i$;
6      $i = i + 1$;
7   **end**
8   calculate crowding-distance in F_i;
9   sort solutions in F_i by crowding-distance in a
     descending order
10    $P'_k = P'_k \bigcup F_i[1 : (S_k - |P'_k|)]$;
11    $P_k = P'_k$;
12 **end**

---

deal with constraints, and an ε-dominance [26] is employed to deal with objectives. In the pull search stage, the selection using the constraint-handling mechanism is described as Algorithm 2.

Algorithm 2: Pull Subproblem

---

1 **Function** *result = PullSubproblems(P_k,gen,T_{max},ε_0)*
2   //T_{max}: the maximum generation.
3   //e: parameter setting in ε-domination;
4   *result* = *false*;
5   **if** $gen \leq 0.9T_{max}$ **then**
6   **for** $k \leftarrow 1$ **to** K **do**
7   An improved epsilon constraint-handling
        mechanism is used to search for
        non-dominated and feasible solutions in P_k;
8   **end**
9   **else**
10   An improved epsilon constraint-handling
       mechanism [25] and the ε-dominance [26] are
       used to search non-dominated and feasible
       solutions in P_k;
11   **end**
12   **return** *result*;
13 **end**

3.2.3. An instantiation of PPS-M2M

Combining M2M decomposition method with PPS framework, we can obtain a new algorithm, called PPS-M2M.

The pseudo-code of PPS-M2M is introduced in Algorithm 3. The algorithm runs repeatedly from line 4 to 40 until the termination condition is met. Lines 5-13 is used to generate new solutions for each sub-population. A number of new solutions are generated at lines 6-10. At lines 11-12, The solution set Q is allocated to each sub-population according to Eq. (14). The max rate of change between the ideal and nadir points during the last l generations r_k is calculated at line 14-16. Parameter $\varepsilon(k)$ is updated at lines 18-27. The updating process for each sub-population is described in lines 28-38. If the size of sub-population P_k is less than S, then $S - |P_k|$ individual solutions

---

Algorithm 3: PPS-M2M

Input:
   K: the number of the subproblems;
   K unit direction vectors: $v^1,...,v^K$;
   S: the size of each subpopulation;
   Q: a set of individual solutions and their F-values;
   T_c: the control generation for $\varepsilon(k)$;
   ε_0: parameter setting in ε-domination;
   T_{max}: the maximum generation.

Output: a set of non-dominated and feasible solutions.
1 **Initialization:**
2 Decompose a population into K sub-populations
   $(P_1, ..., P_K)$, each sub-population contains S individuals
   according to *AllocationSubPop(Q, S, K)*;
3 Set $r_k = 1.0$, $\epsilon_r = 1e - 3$, *PushStage = true*;
4 **while** $gen \leq T_{max}$ **do**
5   **for** $k \leftarrow 1$ **to** K **do**
6   **foreach** $x \in P_k$ **do**
7   Randomly choose y from P_k;
8   Apply genetic operators on x and y to
        generate a new solution z;
9        $R := R \cup \{z\}$;
10   **end**
11      $Q := R \cup (\cup_{k=1}^{K} P_k)$;
12   Use Q to set $P_1, ..., P_K$ according to Eq. (14);
13   **end**
14   **if** $gen >= l$ **then**
15   Calculate the rate of change of the ideal rz_k and
       nadir points rn_k as introduced in [16],
       respectively;
16      $r_k \equiv \max\{rz_k, rn_k\}$;
17   **end**
18   **if** $gen < T_c$ **then**
19   **if** $r_k \leq \epsilon_r$ and *PushStage == true* **then**
20   *PushStage = false*;
21        $\varepsilon(gen) = \varepsilon(0) = maxViolation$;
22   **else**
23   Update $\varepsilon(gen)$ according to Eq. (12);
24   **end**
25   **else**
26      $\varepsilon(k) = 0$;
27   **end**
28   **for** $k \leftarrow 1$ **to** K **do**
29   **if** $|P_k| \leq S$ **then**
30   randomly select $S - |P_k|$ solutions from Q
        and add them to P_k.
31   **else**
32   **if** *PushStage == true* **then**
33   result = *PushSubproblems(P_k, S)*;
34   **else**
35   result =
         PullSubproblems(P_k, gen, T_{max}, ε_0);
36   **end**
37   **end**
38   **end**
39    $gen = gen + 1$;
40 **end**
41 Find all the non-dominated and feasible solutions and
   output them.

---

are randomly selected from Q and added to P_k. If the size of sub-population P_k is greater than S, then S solutions are selected by using the PPS framework. More specifically, at the push search stage, S individual solutions are selected by employing non-dominated sorting method without considering any constraints, as illustrated in line 33. At the pull search stage, S individual solutions are selected by using an improved epsilon constraint-handling approach, as illustrated in line 35. The generation counter is updated at line 39. At line 41, a set of non-dominated and feasible solutions is output.

3.3. Computational complexity of each tested CMOEA

The computation complexity of NSGA-II-CDP is $O(MN^2)$ [27], while the computation complexity of MOEA/D-CDP is $O(MNT)$ [28]. Where M is the number of objectives, N is the population size, and T is the size of the neighbourhood in MOEA/D-CDP. MOEA/D-ACDP has the same computation complexity as that of MOEA/D-CDP. CM2M decomposed a MOP into K subproblems, and non-dominated sorting is employed in each subproblem. Therefore, the computation complexity of CM2M is $O(KMS^2) = O(MN^2/K)$, where S is the size of a subpopulation. The computational complexity of PPS-MOEA/D is the same as that of MOEA/D-CDP, and the computational complexity of PPS-M2M is the same as that of CM2M. Therefore, PPS-MOEA/D has a computational complexity with $O(MNT)$, while the computation complexity of PPS-M2M is $O(MN^2/K)$.

4. Experimental study

To evaluate the performance of the proposed PPS-MOEA/D and PPS-M2M, four other CMOEAs, including MOEA/D-ACDP [17], MOEA/D-CDP [15], NSGA-II-CDP [14], CM2M [18], PPS-MOEA/D [16] are used to make comparisons. The detailed parameters are listed as follows:

1. The mutation probability $P_m = 1/n$ (n denotes the dimension of the decision vector). The distribution index in the polynomial mutation is set to 20.
2. DE parameters: $CR = 1.0$, $f = 0.5$.
3. Population size: $N = 200$. Neighborhood size: $T = 20$.
4. Halting condition: each algorithm runs for 30 times independently, and stops when $400,000$ function evaluations are reached.
5. Probability of selecting individuals from its neighborhood: $\delta = 0.9$.
6. The max number of solutions updated by a child: $n_r = 2$.
7. Parameter setting in PPS-MOEA/D: $Tc = 1600$, $\alpha = 0.95$, $\tau = 0.05$, $cp = 2$, $l = 20$.
8. Parameter setting in PPS-M2M: $Tc = 1600$, $\alpha = 0.95$, $\tau = 0.05$, $cp = 2$, $l = 20$. The number of sub-regions $K = 10$.
9. Parameter setting in CM2M: $K = 10$.

4.1. Performance metric

To evaluate the performance of the CMOEAs, a popular metric hypervolume (HV) [29] is adopted. HV is a popular performance metric which reflects the closeness of the achieved non-dominated solutions to the true PF. A larger HV value represents that the algorithm achieves a non-dominated set closer to the true PF. The definition of HV is as follows:

$$HV(S) = VOL(\bigcup_{x \in S}[f_1(x), z_1^r] \times \cdots [f_m(x), z_m^r]) \qquad (15)$$

where $VOL(\cdot)$ is the Lebesgue measure, m denotes the number of objectives, and $z^r = (z_1^r, \cdots, z_m^r)^T$ is a user-defined reference point in the objective space. A non-dominated sorting on all the obtained results of each algorithms, is carried out to achieve a non-dominated set as a reference PF. The reference point is placed at 1.2 times the distance to the nadir point of the reference PF. A larger value of HV may indicate better performance regarding diversity and/or convergence.

4.2. Experimental results

4.2.1. Performance comparisons

We conduct 30 independent runs for each CMOEA to solve the optimization problem of the teaching manipulator design. The statistical results of HV values are listed in Table 4. Wilcoxon's rank sum test at a 0.05 significance level is performed between PPS-MOEA/D and each of the other five CMOEAs. In terms of HV metric, the performance of PPS-MOEA/D is significantly better than those of PPS-M2M, NSGA-II-CDP, MOEA/D-ACDP, MOEA/D-CDP, CM2M on the optimization problem of the teaching manipulator design.

The distributions of HV values for the six algorithms in the 30 independent runs are shown in Fig. 9. We can observe that PPS-MOEA/D has the highest median of HV value and the smallest interquartile range (IQR), which also indicates that PPS-MOEA/D outperforms the other five algorithms in solving the design optimization problem of the teaching manipulator.

Table 5 shows the HV results of PPS-MOEA/D, MOEA/D-CDP, PPS-M2M, and CM2M. Wilcoxon's rank sum test at a 0.05 significance level is performed between PPS-MOEA/D and MOEA/D-CDP, and between PPS-M2M and CM2M, respectively. From Table 5, we can observe that PPS-MOEA/D is significantly better than MOEA/D-CDP in the HV metric, and PPS-M2M performs significantly better than CM2M in the HV metric. PPS-based methods (PPS-MOEA/D and PPS-M2M) outperform their counterparts (MOEA/D-CDP and CM2M) without adopting PPS framework, which demonstrates the superiority of the PPS framework for solving the design optimization of the teaching manipulator.

Non-dominated solutions achieved by each algorithm on the design optimization of the teaching manipulator with the best HV values during the 30 independent runs are plotted in Fig.10. It can be observed that most of the achieved solutions in PPS-MOEA/D are located on the reference PF, while the populations of the other five CMOEAs are trapped in local optima.

Table 4: HV results of the six CMOEAs on the optimization problem of the teaching manipulator design. Wilcoxon's rank sum test at a 0.05 significance level is performed between PPS-MOEA/D and each of the other five CMOEAs. † represents that the performance of the corresponding algorithm is significantly worse than that of PPS-MOEA/D.

	PPS-MOEA/D	PPS-M2M	NSGA-II-CDP	MOEA/D-ACDP	MOEA/D-CDP	CM2M
mean	**217.88**	157.17†	61.62†	184.13†	190.32†	109.73†
std	5.13	36.61	57.99	17.65	18.00	30.04

Table 5: HV results of the two PPS-based CMOEAs and their counterparts without PPS framework. Wilcoxon's rank sum test at a 0.05 significance level is performed between PPS-MOEA/D and MOEA/D-CDP, and between PPS-M2M and CM2M, respectively. † represents that the performance of the corresponding algorithm is significantly worse than that of PPS-MOEA/D or PPS-M2M.

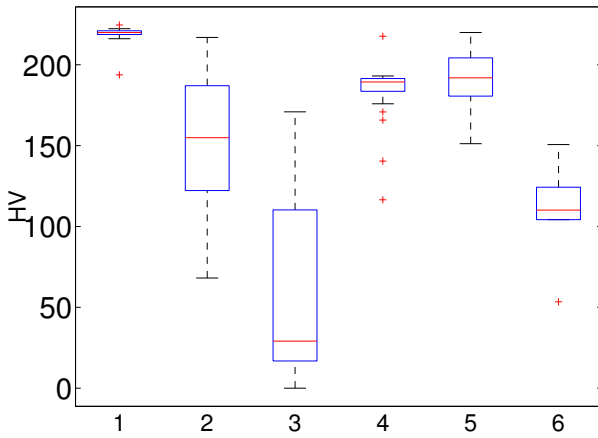|	PPS-MOEA/D	MOEA/D-CDP	‖		PPS-M2M	CM2M
mean	**217.88**	190.32†	‖	mean	**157.17**	109.73†
std	5.13	18.00	‖	std	36.61	30.04

Figure 9: HV box plot of PPS-MOEA/D and the other five CMOEAs on the optimization problem of the teaching manipulator design in 30 independent runs. To facilitate the display of the box plot, the number 1 - 6 represent PPS-MOEA/D, PPS-M2M, NSGA-II-CDP, MOEA/D-ACDP, MOEA/D-CDP, CM2M, respectively.

4.2.2. Analysis of the results

To study the characteristics of the teaching manipulator design optimization problem, $3,000,000$ solutions are generated as shown in Fig. 11, where $600,000$ solutions are generated randomly, and the rest of the solutions are generated by PPS-MOEA/D, PPS-M2M, NSGA-II-CDP, MOEA/D-ACDP, MOEA/D-CDP and CM2M. The approximated landscape of the teaching manipulator design optimization problem is a narrow strip. Some feasible and infeasible solutions are located on the same regions in the objective space. Its PF is discrete, which consists of two different segments as shown in Fig. 11. Therefore, the teaching manipulator design optimization problem is diversity-hard, which means it is difficult to achieve the whole PF for a CMOEA.

MOEA/D-based CMOEAs decompose a CMOP into a set of single constrained optimization problems, which have an intrinsic capability to maintain the diversity of working population. Because the teaching manipulator design optimization problem is diversity-hard, MOEA/D-based CMOEAs perform

better than NSGA-II-based CMOEAs, which can also be observed from Table 4. For PPS-MOEA/D and PPS-M2M, PPS-MOEA/D adopts the MOEA/D framework, while PPS-M2M employs the NSGA-II framework in each sub-region. The performance of PPS-MOEA/D is better than that of PPS-M2M in the case of design optimization of the teaching manipulator.

Another feature of the teaching manipulator design optimization problem is that the unconstrained PF is close to its constrained PF, and the ratio of feasible to infeasible solutions is small, as illustrated by Fig. 11. To be more specific, *AB* represents the distance between the unconstrained PF to the constrained PF, and *CD* denotes the length of infeasible regions in front of the constrained PF, as illustrated in Fig. 11. Since *AB* is much smaller than *CD*, it is easy for the PPS-based algorithms to pull the infeasible solutions to the constrained PF. CMOEAs without PPS framework have to get across large infeasible region. Therefore, this feature presents more difficulty for the CMOEAs without PPS framework. Compared with CMOEAs without PPS framework, the PPS-based algorithms first find the unconstrained without considering any constraints. Then at the pull search stage, infeasible solutions located on the unconstrained PF are pulled to the constrained PF. In contrast, CMOEAs without PPS framework must traverse large infeasible regions to approach the constrained PF. Therefore, the PPS-based CMOEAs performs better than their counterparts without adopting PPS framework.

From the above discussion, we can conclude that PPS-MOEA/D outperforms the other five CMOEAs (PPS-M2M, NSGA-II-CDP, MOEA/D-ACDP, MOEA/D-CDP and CM2M) significantly on the teaching manipulator optimization problem. PPS-based methods (PPS-MOEA/D and PPS-M2M) outperform their counterparts (MOEA/D-CDP and CM2M) without adopting PPS framework, which demonstrates the superiority of the PPS framework for solving the teaching manipulator design optimization problem.

4.2.3. The Design of the Teaching Manipulator

The achieved reference PF obtained by six CMOEAs and the original design suggested by a human expert are shown in Fig. 12. Each solution in the reference PF dominates the original
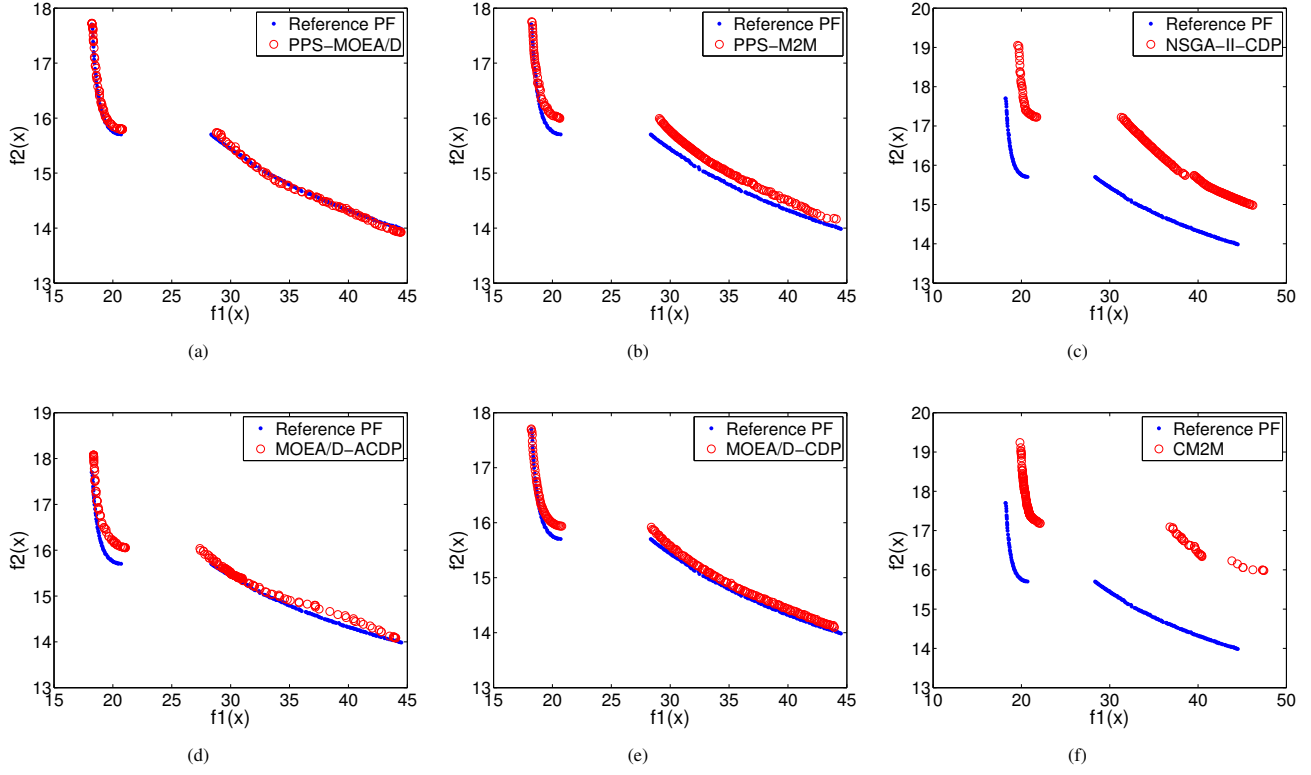
Figure 10: Non-dominated solutions achieved by each algorithm on the design optimization of the teaching manipulator with the best HV values during the 30 independent runs are plotted.
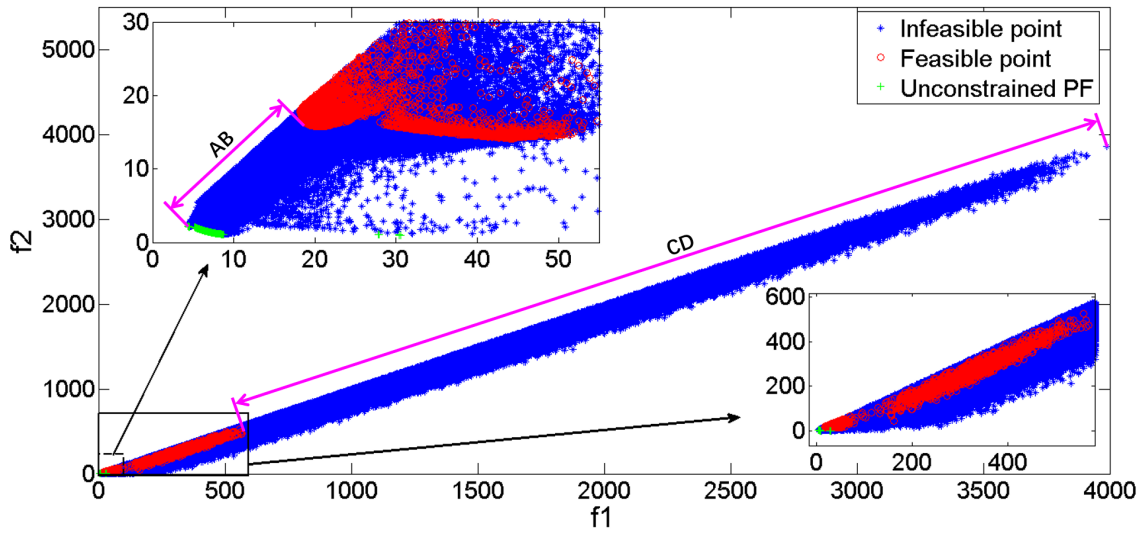

Figure 11: The distribution of solutions of the teaching manipulator design optimization problem in the objective space.

design. The performance of the teaching manipulator is improved significantly by using the proposed PPS-MOEA/D and PPS-M2M. Since the reference PF is divided into two parts, the endpoints of each part are selected as representative cases, as illustrated in Fig. 12. The detailed comparison of the original design by the human expert and the four representative solutions in the reference PF are listed in Table. 6. The values of k, b and T_3 of case A, B are significantly different from those of case C, D. A possible reason is that different constraints are activated in each different segment of the reference PF.

4.2.4. Limitation of PPS

It is worth noting that PPS-based algorithms are not suitable for solving CMOPs whose unconstrained PFs are degenerated, which means the dimension of the unconstrained PFs is less than their constrained counterpart. One possible reason is that, at the end of the push stage, the populations of PPS-based algorithms are converged to the unconstrained PFs. Since the unconstrained PFs are degenerated, the diversity of the populations in PPS-based algorithms is lost, and it is very difficult to pull the populations to the constrained PFs whose dimensions are greater than these of their unconstrained counterpart.

5. Conclusion

This paper builds up a model of a teaching manipulator for recording and setting trajectories for manipulators in production lines. The design of the teaching manipulator is formulated as a constrained multi-objective optimization problem (CMOP). Two kinds of PPS-based algorithms, including PPS-MOEA/D [16] and PPS-M2M, are employed to solve the formulated CMOP. It is worth noting that PPS-M2M is a new algorithm which combines PPS with M2M decomposition method to solve CMOPs. To be more specific, PPS-M2M also divides the search process into two different stages — push and pull search, as done in PPS-MOEA/D [16]. At the push search stage, a CMOP is decomposed into a set of simple CMOPs. Each simple CMOP corresponds to a sub-population and is solved by using NSGA-II without considering any constraints, which also can help each sub-population get across infeasible regions efficiently. Some constrained information can be probed and collected to help guide the parameter settings of the constraint-handling mechanisms in the pull search stage. At the beginning of the pull search stage, infeasible solutions in each sub-population are pulled to the feasible and non-dominated regions by employing an improved epsilon constraint-handling mechanism. At the last ten percentages of the maximum generations, each sub-population is merged into one population which is evolved by using the improved epsilon constraint-handling method and the epsilon-dominance mechanism. The experimental results indicate that PPS-MOEA/D outperforms the other five CMOEAs (PPS-M2M, NSGA-II-CDP, MOEA/D-ACDP, MOEA/D-CDP and CM2M) significantly on the teaching manipulator optimization problem. PPS-based methods (PPS-MOEA/D and PPS-M2M) outperform their counterparts (MOEA/D-CDP and CM2M) without adopting PPS framework, which demonstrates the superiority of the PPS framework for solving real-world optimization problems.

It is notable that few works have been done by employing CMOEAs to solving real-world CMOPs. This paper provides a feasible method. In the future, we will investigate multi-scenario optimization [30] and conduct additional experiments with different trajectories to test the performance of PPS-based CMOEAs. Besides, we will study surrogate-assisted methods in the framework of PPS to solve problems with expensive fitness evaluation, which is widely existed in the real-world optimization problems.

Appendix .1. Kinematic modeling

The transformation matrix between two adjacent joints from frame i to frame $i + 1$ based on the D-H convention is given as follows:

$$R_{i+1}^i = Rot_z(\theta_i) \cdot Trans_z(d_i) \cdot Trans_x(a_i) \cdot Rot_x(\alpha_i) \qquad (.1)$$

where $Rot_i(\bullet) \in SE(3)$ is a homogeneous rotational transformation matrix about axis i, while $Trans_i(\bullet) \in SE(3)$ is a homogeneous translation transformation matrix about axis i. SE(3) represents the Special Euclidean Group[31], which can represent an arbitrary rigid transformation including translations and rotations. Multiplying all the link transformation matrices yields the total transformation matrix from the base of the teaching manipulator to the end-effector.

Appendix .2. Dynamic modeling

In this study, the method of Lagrange is utilized to establish the dynamic model of the teaching manipulator. The Lagrange equation is given as follows:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}}) - \frac{\partial L}{\partial q} = J^T f_{opr}(t) + \tau_b - \tau_{frc}(\dot{q}) \qquad (.2)$$

$$L = K - U = \sum_{i=1}^{6}(K_i - U_i) \qquad (.3)$$

where L is the Lagrangian, $q \in \mathbb{R}^6$ and $\dot{q} \in \mathbb{R}^6$ are the generalized angular position and velocity, respectively. $J \in \mathbb{R}^{6 \times 6}$ is the Jacobian matrix of the teaching manipulator with respect to the reference inertial frame, while $f_{opr}(t) = [F_x, F_y, F_z, n_x, n_y, n_z]$ is the operating force and torque vector of human operator, in which F_x, F_y, F_z are the components of the operating force, and n_x, n_y, n_z are the components of the operating torques. The magnitude of the operating torque is tiny, because the three joints of the spherical wrist are not equipped with friction disks and the counterweight A can help keep Joint 5 in balance. Thus, rather than the magnitude of operating torque, the magnitude of the operating force should be considered as a focus in design. $\tau_b \in \mathbb{R}^6$ is the moment vector of force acting on joints by the balancer. $\tau_{frc}(\dot{q}) \in \mathbb{R}^6$ is a friction moment vector of the friction disks on joints. Since frictional moments always resist the rotation of joints, the sign of the term $\tau_{frc}(\dot{q})$ is always negative.
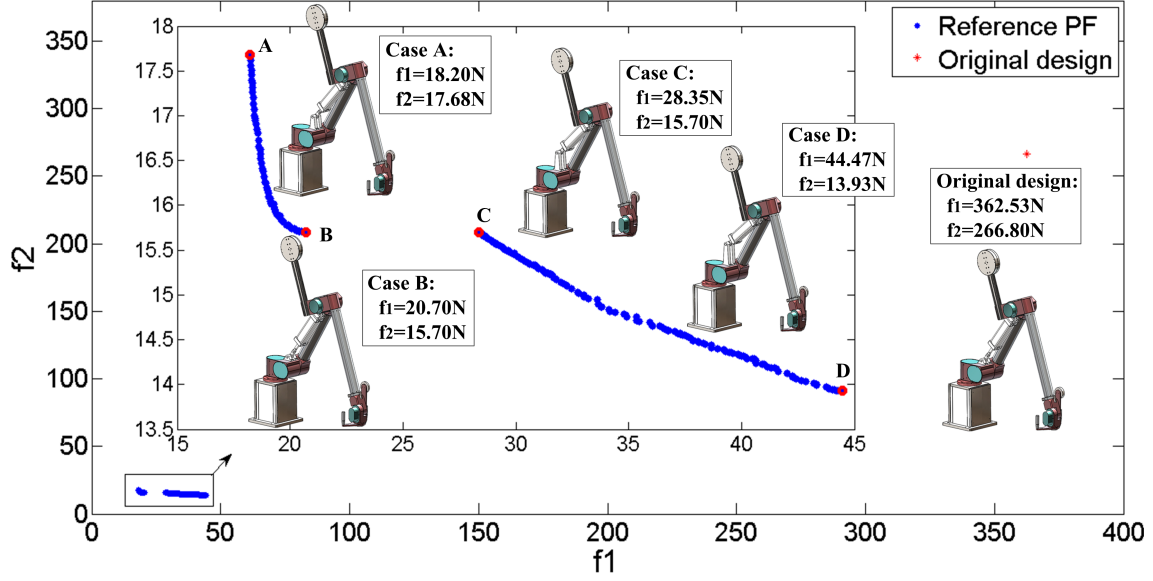
Figure 12: The four endpoints of the reference PF are selected as the representative cases to be compared with the original design by the human expert. The CAD models of the four solutions are drawn out with the different appearance to the original design.

Table 6: The decision variables, objective and constraint values of the four representative solutions and the original design.

	m_A	L_A	m_B	L_B	k	H_b	b	T_1	T_2	T_3	f_1	f_2	c_1	c_2	c_3
Case A	0.86	0.20	11.64	0.55	**1897**	0.19	**1008**	0.00	0.25	**0.02**	18.20	17.68	1.17e-6	**2.41e-6**	6.23e-5
Case B	0.86	0.20	11.64	0.55	**682**	0.11	**1760**	5.76	5.61	**0.02**	20.70	15.70	8.49e-7	**3.06e-9**	1.08e-5
Case C	0.86	0.20	11.98	0.53	**1**	0.24	**770**	0.00	0.19	**11.03**	28.35	15.70	2.62e-5	**10.57**	7.46e-4
Case D	0.87	0.20	12.93	0.50	**0**	0.27	**615**	0.00	0.88	**25.81**	44.47	13.93	2.00e-3	**24.07**	5.91e-3
Original design	1.60	0.19	15.00	0.46	**3730**	0.12	**2500**	75.70	75.70	**75.70**	362.53	266.80	57.62	**72.48**	2.60e-3

For the ith link, the kinetic energy K_i and the potential energy U_i are given by Eq.(.4).

$$K_i = \frac{1}{2}m_i V_{ci}^T V_{ci} + \frac{1}{2}\omega_i^T I_i \omega_i \qquad (.4)$$

$$U_i = m_i g^T P_{ci} \qquad (.5)$$

where m_i is the mass of the ith link, V_{ci} is the element of linear velocity vector V and represents the linear velocity of the center of mass (COM) for link i, ω_i is the element of angular velocity vector ω and represents the angular velocity of link i. I_i denotes the inertia tensor in the inertial frame. For simplicity, we assume that the mass of all links are uniformly distributed, and the COMs are located at the center of the links. $g = [0, 0, -g_z]$ is the gravitational acceleration vector and P_{ci} is the position vector of the COM of link i, measured in the reference inertial coordinate. V and ω vector can be obtained from the equation as follows:

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = J\dot{q} \qquad (.6)$$

Integrating equations from Eq.(.3) to (.5), the dynamic model of the teaching manipulator is given as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_{frc}(\dot{q}) = J^T f_{opr}(t) + \tau_b \qquad (.7)$$

where $\ddot{q} \in \mathbb{R}^6$ is the generalized angular acceleration, $M(q) \in \mathbb{R}^{6\times6}$ is the inertia matrix, which is a symmetric and positive definite matrix. $G(q) \in \mathbb{R}^6$ is a six-dimension vector which represents the gravitational moments acting on joints. Moreover, the mass of counterweights and the length of the rods, which connect the counterweights and the corresponding joints, can change the distribution of mass and the inertia of the links, thus influencing the operating force. $C(q, \dot{q}) \in \mathbb{R}^{6\times6}$ is the centripetal and Coriolis matrix. In this matrix, the elements only involving $\dot{q}$ are centripetal which are related to the centripetal force of each joint, while the rest elements involving q are Coriolis which are related to the Coriolis force of each joint. The elements of which is given as follows:

$$c_{ij} = \frac{1}{2}\sum_{k=1}^{6}\left(\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{kj}}{\partial q_i}\right)\dot{q}_k \qquad (.8)$$

where m_{ij} is the element of the inertia matrix in the ith row and the jth column.

Appendix .3. Balancer model

The balancer consists of a pneumatic cylinder and a linear elastic spring. The pneumatic cylinder is designed to connect an air pump with stable pressure, thus we assume the pneumatic cylinder provides a constant force b in work. The spring with stiffness k is designed to be in its unloaded position when the

balancer is vertical to the plane of the horizon, i.e. $q_2 = \frac{\pi}{2}$. In this case, the length of the balancer is in the shortest state, which can be given by Eq. (.9)

$$L_{k0} = a_2 - H_b \qquad (.9)$$

If Joint 2 is in the angular position other than $\frac{\pi}{2}$, the balancer is elongated. The elongated distance of the balancer is equal to the compressed distance of the spring. Fig..13 (a) and (c) show the geometric relation of Link 2 and the balancer. The length of the balancer is determined by using the law of cosines.

$$L_k = \sqrt{H_b^2 + a_2^2 - 2H_b a_2 \cos(\pi/2 - q_2)} \qquad (.10)$$

The force of the spring can be given as follows:

$$F_s = k(L_s - L_{s0}) = k(L_k - L_{k0}) \qquad (.11)$$

where L_s and L_{s0} are the elongated length and the original length of the spring, respectively, as shown in Fig..13 (b).

Applying the law of sines yields the following equation.

$$\frac{H_b}{\sin\beta} = \frac{L_k}{\sin(\pi/2 - q_2)} \qquad (.12)$$

From Eq.(.12), the moment arm of the force of the balancer can be determined by the following equation.

$$r = a_2 \sin\beta = \frac{a_2 H_b \sin(\pi/2 - q_2)}{L_k} \qquad (.13)$$

From Eq.(.11) and (.13), we can determine the moment of the balancer acting on Joint 2.

$$\tau_{b,2} = r(F_s + b) \qquad (.14)$$

where $\tau_{b,2}$ is the second element of the moment vector τ_b. b is the constant pulling force of the pneumatic cylinder, direction of which is along the longitudinal direction of the cylinder. Since the moment of the balancer only acts on Joint 2, we have the following expression of the vector.

$$\tau_b = [0, \tau_{b,2}, 0, 0, 0, 0] \qquad (.15)$$

Appendix .4. Friction disk model and balance conditions

The friction disk plays an important role in keeping balance of the device. A friction disk model considers both kinetic and static friction. For the friction disk at Joint i, $(i = 1, 2, 3)$, we have the following equation.

$$\tau_{f,i} = \begin{cases} -\tau_{ex,i}, & \dot{q}_i = 0 \\ sgn(\dot{q}_i)T_i, & \dot{q}_i \neq 0 \end{cases} \qquad (.16)$$

where $\tau_{f,i}$ is the frictional moment at Joint i^{th}. $\tau_{ex,i}$ is the resultant moment from all other moments except the friction moment acting at Joint i. If $\dot{q}_i$ is 0, the friction disk provides a moment with the same size but opposite direction of $\tau_{ex,i}$ to keep the corresponding joint in equilibrium. If the joint rotates, the friction disk provides a kinetic friction moment to resist the motion, which is denoted by T_i (i = 1, 2, 3).

The teaching manipulator is required to maintain the angular position of Joint 1,2 and 3 if the device stops in any position of the trajectory. In this case, the angular velocities and accelerations of the three joints are zero, and the human operator stops driving the end-effector. Substitute $\dot{q} = 0$, $\ddot{q} = 0$, and $f_{opr}(t) = 0$ to Eq.(.7), we have Eq.(.17).

$$G(q) - \tau_b = \tau_{frc}(q) \qquad (.17)$$

where

$$\tau_{frc} = [\tau_{f,1}, \tau_{f,2}, \tau_{f,3}, 0, 0, 0], \qquad (.18)$$

The three friction disks installed at the Joint 1, 2 and 3 provide the frictional moment. The design of counterweights and the rods, as well as the position of the joints influence the mass distribution, and change the gravitational moment $G(q)$.

To keep balance, given any q_t in the trajectory during operation, the two following relations should be satisfied:

$$|G_2(q_t) - \tau_{b,2}(q_t)| \leq T_2 \qquad (.19)$$

$$|G_3(q_t)| \leq T_3 \qquad (.20)$$

where G_i is the ith component of the vector of gravity load. There is no need to consider balance on Joint 1, which is not sensitive to gravity, since the rotational axis is vertical to the plane of the horizon.

To keep Joint 5 in balance, we set a counterweight A attached to the Joint 5. From Eq. (.17), with no torque provided by the balancer and friction disks, for the Joint 5, we have the following equation.

$$G_5(q_t) = 0 \qquad (.21)$$

In a real world problem, it is difficult to satisfy such condition. We therefore transfer the equality constraint Eq. (.21) into an inequality constraint as follows:

$$|G_5(q_t)| \leq \epsilon \qquad (.22)$$

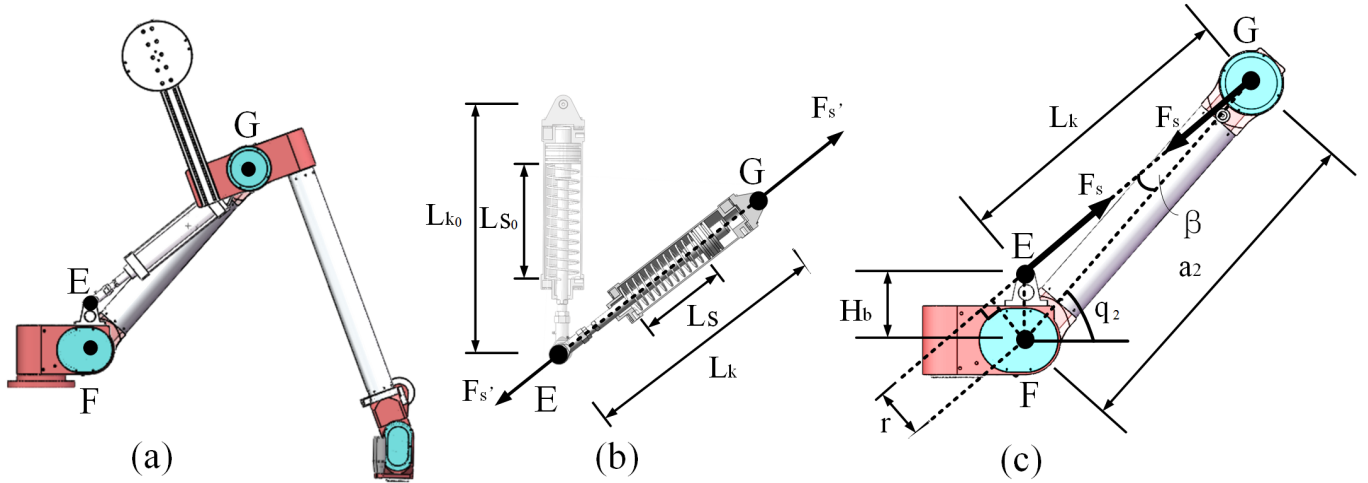where ε is a small enough positive constant.

Figure .13: Force analysis of the Joint 2. (a) A teaching manipulator in one of the position. (b) Force analysis of balancer. (c) Geometric analysis of the arm of the pulling force acting on Link 2.

References

[1] H. Lipson, J. B. Pollack, Automatic design and manufacture of robotic lifeforms, Nature 406 (6799) (2000) 974.

[2] M. Tarkian, Design reuse and automation: on high level cad modeling for multidisciplinary design and optimization, Ph.D. thesis, Linköping University Electronic Press (2009).

[3] M. Tarkian, Design automation for multidisciplinary optimization: A high level cad template approach, Ph.D. thesis, Linköping University Electronic Press (2012).

[4] M. Pettersson, Design optimization in industrial robotics: Methods and algorithms for drive train design, Ph.D. thesis, Institutionen för ekonomisk och industriell utveckling (2008).

[5] M. Tarkian, J. Persson, J. Ölvander, X. Feng, Multidisciplinary design optimization of modular industrial robots by utilizing high level cad templates, Journal of Mechanical Design 134 (12) (2012) 124502.

[6] Y. Li, Q. Xu, GA-based multi-objective optimal design of a planar 3-DOF cable-driven parallel manipulator, in: Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on, IEEE, 2006, pp. 1360–1365.

[7] M. Pettersson, J. Ölvander, Drive train optimization for industrial robots, IEEE Transactions on Robotics 25 (6) (2009) 1419–1424.

[8] L. Zhou, S. Bai, M. R. Hansen, Design optimization on the drive train of a light-weight robotic arm, Mechatronics 21 (3) (2011) 560–569.

[9] R. Datta, K. Deb, Multi-objective design and analysis of robot gripper configurations using an evolutionary-classical approach, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, ACM, 2011, pp. 1843–1850.

[10] R. Datta, A. Jain, B. Bhattacharya, A piezoelectric model based multi-objective optimization of robot gripper design, Structural and Multidisciplinary Optimization 53 (3) (2016) 453–470.

[11] R. Datta, S. Pradhan, B. Bhattacharya, Analysis and design optimization of a robotic gripper using multiobjective genetic algorithm, IEEE Transactions on Systems, Man, and Cybernetics: Systems 46 (1) (2016) 16–26.

[12] L. Zhou, S. Bai, A new approach to design of a lightweight anthropomorphic arm for service applications, Journal of Mechanisms and Robotics 7 (3) (2015) 031001.

[13] K. Deb, Multi-objective optimization using evolutionary algorithms, Vol. 16, John Wiley & Sons, 2001.

[14] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm : NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[15] M. A. Jan, R. A. Khanum, A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D, Applied Soft Computing 13 (1) (2013) 128–148.

[16] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman, Push and pull search for solving constrained multi-objective optimization problems, Swarm and Evolutionary Computation 44 (2019) 665–679.

[17] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, E. Goodman, MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems, Applied Soft Computing 74 (2019) 621–633.

[18] H.-L. Liu, C. Peng, F. Gu, J. Wen, A constrained multi-objective evolutionary algorithm based on boundary search and archive, International Journal of Pattern Recognition and Artificial Intelligence 30 (01) (2016) 1659002.

[19] J. Denavit, A kinematic notation for lower-pair mechanisms based on matrices, ASME Journal of Applied Mechanics (1955) 215–221.

[20] M. W. Spong, S. Hutchinson, M. Vidyasagar, et al., Robot modeling and control, 2006.

[21] S. I. Valdez, S. Botello-Aceves, H. M. Becerra, E. Hernandez-Martinez, Comparison of a concurrent and a sequential optimization methodologies for serial manipulators using metaheuristics, IEEE Transactions on Industrial Informatics PP (99) (2018) 1–1.

[22] P. I. Corke, Robotics, Vision & Control: Fundamental Algorithms in MATLAB, 2nd Edition, Springer, 2017, ISBN 978-3-319-54413-7.

[23] K. Miettinen, Nonlinear Multiobjective Optimization, Vol. 12, Springer Science & Business Media, 1999.

[24] T. Takahama, S. Sakai, Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites, in: 2006 IEEE International Conference on Evolutionary Computation, 2006, pp. 1–8. doi:10.1109/CEC.2006.1688283.

[25] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, E. Goodman, An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions, Soft Computing (2017) 1–20.

[26] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, Evolutionary Computation 10 (3) (2002) 263–282.

[27] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.

[28] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.

[29] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.

[30] H. Wang, J. Doherty, Y. Jin, Hierarchical surrogate-assisted evolutionary multi-scenario airfoil shape optimization, in: Proceedings of the 2018 IEEE World Congress on Computational Intelligence (WCCI 2018), Institute of Electrical and Electronics Engineers (IEEE), 2018.

[31] B. Siciliano, O. Khatib, Springer handbook of robotics, Springer, 2016.