An Inverse Modeling Constrained Multi-Objective Evolutionary Algorithm Based on Decomposition

Lucas R. C. Farias[1][2][3] and Aluizio F. R. Araújo[1]

Abstract— This paper introduces the inverse modeling constrained multi-objective evolutionary algorithm based on decomposition (IM-C-MOEA/D) for addressing constrained real-world optimization problems. Our research builds upon the advancements made in evolutionary computing-based inverse modeling, and it strategically bridges the gaps in applying inverse models based on decomposition to problem domains with constraints. The proposed approach is experimentally evaluated on diverse real-world problems (RWMOP1-35), showing superior performance to state-of-the-art constrained multi-objective evolutionary algorithms (CMOEAs). The experimental results highlight the robustness of the algorithm and its applicability in real-world constrained optimization scenarios.

I. INTRODUCTION

A multi-objective optimization problem (MOP) is characterized by a set of Pareto optimal solutions, representing a compromise between the different objectives. Such solutions form the Pareto set (PS) in the decision space with the correspondent Pareto front (PF) in the objective space. Various optimization problems in real-world applications hold a number of conflicting objectives and multiple complex constraints, such as the robot gripper optimization problem, water resources management, gearbox design, and process synthesis problem [1]. This kind of problem is denoted as constrained MOP (CMOP).

CMOPs bring more challenges than their unconstrained counterparts due to the coexistence of multiple objectives and constraints. The CMOP is defined as [2]:

$$
\begin{aligned}
\min \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}) \; f_2(\mathbf{x}) \dots f_m(\mathbf{x}))^T \\
\text{s.t.} \quad & \begin{cases} g_j(\mathbf{x}) \leq 0, \, j = 1, \dots, l \\ h_j(\mathbf{x}) = 0, \, j = l+1, \dots, k \\ \mathbf{x} = (x_1 \; x_2 \dots x_d)^T \in \Omega \end{cases}
\end{aligned} \quad (1)
$$

where $\mathbf{x}$ is a d-dimensional real-valued decision vector in the decision space $\mathbb{R}^d$, and $\mathbf{f}(\mathbf{x})$ is an m-dimensional objective vector in the objective space $\mathbb{R}^m$; $\Omega \subset \mathbb{R}^n$ defines the feasible region of the decision space; $g_j(\mathbf{x}) \leq 0$ is the j-th inequality constraint; $h_j(\mathbf{x}) = 0$ is the $(j-l)$-th equality constraint; and l and $(k-l)$ denote the number of inequality and equality constraints.

In recent decades, multi-objective evolutionary algorithms (MOEAs) have emerged as a popular approach for generating a set of non-dominated solutions in a single run to solve MOPs [2]. Existing MOEAs can be classified according to their selection methods: Pareto-based, decomposition-based, and indicator-based approaches [2]–[4]. These MOEAs are effective solvers for unconstrained multi-objective optimization problems. However, when applied to CMOPs, they need a constraint handling technique (CHT), a selection mechanism that can handle constraints. CMOPs are more prevalent in practical application scenarios than unconstrained MOPs. Thus, considerable effort has been devoted to designing CHTs and specific mechanisms to promote the emergence and development of various CMOEAs [5].

While MOEAs often require population diversity to store non-dominated solutions within the population or in an external archive, studies with model-based MOEAs aim to loosen such a diversity requirement. Model-based approaches focus on constructing a probabilistic model in the decision space throughout the evolutionary process [3]. Despite their effectiveness, the model-based MOEAs still rely on maintaining a candidate solution set, such as an external archive, to preserve non-dominated solutions obtained during the search. Another strategy to relax the diversity necessity involves constructing a regression model to identify solutions enhancing the diversity in the final generation [3], [6].

The just mentioned alternatives inspired Cheng et al. [3] to introduce an MOEA using Gaussian process-based inverse modeling (IM-MOEA). It stands out as a Pareto-based evolutionary algorithm based on an inverse model that maps the objective space into the decision space. This approach is promising when the decision-making process is essential for solving MOPs. The inverse mapping is an optimization process that becomes an optimal allocation of solutions within the Pareto front, simplifying the identification of suitable solutions. The optimal allocation refers to an approximation set of the PF that provides maximum information about the underlying PF, and it releases adherence to a uniform distribution of Pareto-optimal solutions [6].

Subsequent research has expanded upon the foundation laid by IM-MOEA [4], [6]–[11]. These works propose improvements such as adaptive weight vectors in population clustering [7], [9], E-IM-MOEA utilizes an external population and a biased reproduction operator [7], IM-MOEA-RF replaces the Gaussian process with Random Forest [8], and recently some works have extended inverse models based on Pareto dominance to deal with CMOP [4], [10], [11]. IM-MOEA/D proposes a decomposition-based approach tested as competitive on large-scale many-objective problems [6]. Decomposition-based inverse modeling to deal with CMOPs has yet to be proposed to the best of our knowledge.

Bearing in mind the potential of inverse modeling for

[1]Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil, {lrcf,aluizioa}@cin.ufpe.br

[2]Universidade Católica de Pernmabuco, Recife, Brazil, lucas.farias@unicap.br

[3]Centro de Estudos e Sistemas Avançados do Recife, Recife, Brazil, lrcf@cesar.org.br

solving MOPs, we think it can do well for CMOPs since it acts directly in the objective space, allowing a better distribution mapping of the constraint behavior than in decision space. In this context, we propose the inverse modeling constrained multi-objective evolutionary algorithm based on decomposition (IM-C-MOEA/D) that incorporates a constraint handling technique that adjusts its search strategy based on the presence and severity of constraints, actively guiding the optimization process to enable the solutions' feasibility and improving convergence toward the Pareto front [4], [12]. To validate our approach, we conducted experiments using a set of real-world CMOPS, namely RWMOP1-RWMOP35 [1], in which the number of objectives ranges from 2 to 5, the number of variables varies between 2 and 30, and has up to 29 constraints. Our experimental results suggest the superior performance of the proposed method compared to six state-of-the-art MOEAs: paired offspring generation-based constrained evolutionary algorithm (POCEA) [13], large-scale multi-objective competitive swarm optimization algorithm (LMOCSO) [14], reference vector guided evolutionary algorithm (RVEA) [15], many-objective evolutionary algorithm based on dominance and decomposition (MOEA/DD) [16], IM-MOEA [3], and IM-C-MOEA, an extended version of IM-MOEA to deal with constraints that we also proposed.

The fundamental concept is elaborated in Section 2, followed by a detailed description of the primary components of the IM-C-MOEA/D. Section 3 presents the experimental results of the proposed algorithm in comparison to six other CMOEAs on 35 real-world instances. Finally, Section 4 provides the conclusion of the study and outlines future work.

II. IM-C-MOEA/D

This section discusses the details of the inverse modeling constrained multi-objective evolutionary algorithm based on decomposition (IM-C-MOEA/D). A high-level scheme is initially shown in Figure 1. The main items that compose it are the following:

- Initialization: As most decomposition-based MOEAs do, we initialize the individuals of the population, P, weight vectors, λ, and the reference point, z. In particular, the weight vectors are initially generated using the Das and Dennis sampling (Subsection II.A).
- Population partition: The subpopulations are created by the k-means algorithm, followed by a tournament selection of solutions in the subpopulation (Subsection II.B).
- Inverse model: Each subpopulation has an inverse model, a mapping from the objective space into the decision space (Subsection II.C).
- Constraint handling: The constraint handling procedure to compare viable and unviable solutions for replacement purposes is described in Subsection II.D.
- Decomposition-based global replacement: The survivor selection is based on decomposition using a global replacement, which associates each new solution with the most suitable weight vector (Subsection II.E).
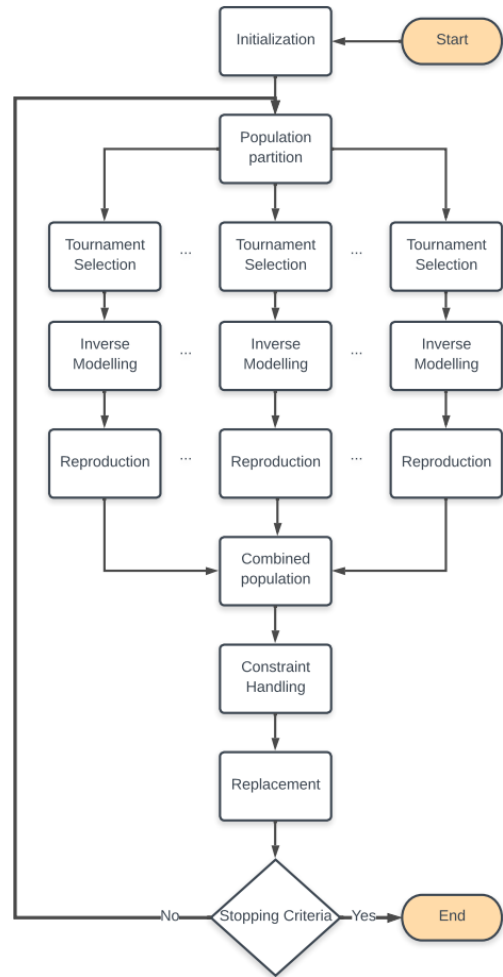

Fig. 1. Flowchart of execution of the IM-C-MOEA/D.

- Stopping Criteria: The maximum number of generations or fitness evaluations.

The details of the IM-C-MOEA/D components and their computational complexity (Subsection II.F) are below.

A. Generating Weights

The weight vectors are generated uniformly distributed over a unit simplex using the method proposed by Das and Dennis. This approach is present in most variants of decomposition-based MOEAs [17]. The number of weight vectors to be generated, N, is calculated as $N = \binom{H+m-1}{m-1}$, where H is the number of divisions of each axis and m is the number of objectives. The value of H cannot be lower than m to avoid creating intermediate points. Moreover, generating weight vectors may become computationally costly for more than three objectives [18].

B. Population Partition

IM-MOEA/D builds the inverse models (IMs) in the reproduction stage; firstly, the k-means algorithm creates clusters according to the position in the objective space [6]. Then, IM-C-MOEA/D applies a binary tournament to select solutions, with priority to feasible ones, in all clusters [12]. There is an IM for each group.

C. Inverse Modeling

Conventional model-based algorithms aim to estimate the distribution of the candidate solutions in the decision space. In contrast, inverse model-based algorithms are built to represent the inverse mapping from the objective space to the decision space. After constructing the IMs, evenly distributed candidate solutions can be directly sampled from the objective space and mapped into the decision space [3]. However, estimating the inverse mapping from the m-dimensional objective space to the d-dimensional decision space can be challenging. Therefore, a multivariate IM is decomposed into several univariate regression models:

$$P(X|Y) \approx \prod_{i=1}^{d} (P(x_i|f_j) + \epsilon_{j,i}), \tag{2}$$

where $j = 1, 2, ..., m$, $i = 1, 2, ..., d$, $P(x_i|f_j)$ is an univariate model that represents the inverse mapping from objective function f_j to decision variable x_i, and $\epsilon_{j,i}$ is an error term. We consider that $\epsilon_{j,i} \sim \mathcal{N}(0, (\sigma_n)^2)$ is an additive Gaussian noise. Consequently, each univariate model, together with the error term achieved using the Gaussian process, is a way for modeling both the global regularity and the local randomness in the distribution of the candidate solutions during the search [3].

D. Constraint Handling Approach

IM-C-MOEA/D extends IM-MOEA/D [6] incorporating a constrained handling approach suggested by Jain and Deb [4], [12]. This modification enables the proposed IM-C-MOEA/D to handle CMOPs effectively. Therefore, in IM-C-MOEA/D, when a new candidate solution $\mathbf{y}$ (an offspring) is compared with a randomly selected neighbor $\mathbf{x}$, the constraint violations (CV) of both solutions are verified before applying solution substitution based on the Tchebycheff metric. Depending on the feasibility of solutions, the following four scenarios can occur:

1) If $\mathbf{y}$ is infeasible and $\mathbf{x}$ is feasible, then $\mathbf{x}$ remains unchanged;
2) If $\mathbf{y}$ is feasible and $\mathbf{x}$ is infeasible, then $\mathbf{x}$ is replaced by $\mathbf{y}$;
3) If both $\mathbf{y}$ and $\mathbf{x}$ are infeasible, then $\mathbf{x}$ is replaced by $\mathbf{y}$ if the constraint violation of the latter is lower than that of the former;
4) If both solutions are feasible, $\mathbf{x}$ is replaced by $\mathbf{y}$ when $\mathbf{y}$ has a better Tchebycheff metric value than $\mathbf{x}$.

Such modifications prioritize feasible and small-CV solutions in the population without inserting any new parameter to the algorithm [12].

E. Decomposition-Based Replacement

The Tchebycheff (TCH) decomposition is defined as:

$$\text{minimize } g^{TCH}(\mathbf{x}|\lambda, \mathbf{z}) = \max_{1 \le j \le m} (\lambda_j |f_j(\mathbf{x}) - z_j|),$$
$$\text{subject to } \mathbf{x} \in \Omega \tag{3}$$

where m is the number of objectives and $\mathbf{z}$ is the utopian reference point, i.e. $z_j = \min \{f_j(\mathbf{x})|\mathbf{x} \in \Omega\}$, for every

---

Algorithm 1: IM-C-MOEA/D

1 Initialize the population P and a weight vectors set λ;
2 Define $T = 0.1N$;
3 Determine the T neighbors in $B(i)$ for each $\lambda_i \in \lambda$;
4 Calculate the reference point $\mathbf{z}$ based on the population P;
5 for *each Gen* $= 1 ... Gen_{max}$ **do**
6   Partition the population $P(Gen)$ using k-means to create subparent populations $P_1(Gen)$, ..., $P_K(Gen)$;
7 $O \leftarrow \emptyset$;
8   **for** *each* $k \in \{1, ..., K\}$ **do**
9   Apply the tournament selection operator to $P_k(Gen)$
10   Build the inverse model through the Gaussian process using subpopulation $P_k(Gen)$;
11   Sample candidate solutions $O_k(Gen)$ in the objective space and map them back to the decision space using the inverse models;
12   Perform mutation on $O_k(Gen)$;
13 $O \leftarrow O \cup O_k(Gen)$;
14 $\mathbf{z} \leftarrow min(\mathbf{z}, \mathbf{f}(O(Gen)))$;
15   **for** *each* $\mathbf{o} \in O(Gen)$ **do**
16   Find the most appropriate weight vector λ_i for $\mathbf{o}$ using Equation (4);
17   Calculate the constraint violation (CV) of $\mathbf{o}$ and each solution in $B(i)$;
18   **for** *each* $j \in B(i)$ **do**
19   **if** $g^{TCH}(\mathbf{o}|\lambda_j, \mathbf{z}) \le g^{TCH}(P_j|\lambda_j, \mathbf{z})$ *and* $CV(\mathbf{o}) \le CV(P_j)$ **then**
20 $P_j \leftarrow \mathbf{o}$;

21 return P;

$j = 1, ..., m$. The m-dimensional weight vector is defined as $\lambda = (\lambda_1 ... \lambda_m)^T$, $\sum_1^m \lambda_j = 1$ and $\lambda_j \ge 0$, for all $j \in 1, ..., m$. Different Pareto-optimal solutions can be obtained by altering weight vectors using the TCH approach [17].

After generating a new solution $\mathbf{o}$, the global replacement scheme determines the most suitable weight vector i using the TCH decomposition:

$$i = \arg \min_{1 \le k \le N} \left\{ g^{TCH}(\mathbf{o} \mid \lambda_k, \mathbf{z}) \right\}. \tag{4}$$

The algorithm then verifies the neighborhood of the λ_i, searching for solutions that can be improved by the offspring $\mathbf{o}$. If $g^{TCH}(\mathbf{o} \mid \lambda_j, \mathbf{z})$ is better than $g^{TCH}(\mathbf{x}_j \mid \lambda_j, \mathbf{z})$, the current solution $\mathbf{x}_j$ in the T-neighborhood of the weight vector λ_j is replaced by $\mathbf{o}$ [17].

F. Algorithm Framework

The pseudo-code for IM-C-MOEA/D is illustrated in Algorithm 1. IM-C-MOEA/D initializes the population and weight vectors, determines the neighborhood structure for each weight vector, and calculates the ideal reference point

(lines 1-4). The population is partitioned into K groups using k-means (line 6), and new individuals are generated for each partition (lines 7-13). The global replacement is applied to identify the most suitable weight vector, and the individuals in the neighborhood of the most suitable weight vector are examined using the constraint handling approach and decomposition function (lines 14-20). The algorithm stops when the maximum number of generations is reached. The maximum computational complexity of IM-C-MOEA/D is determined by the time cost of population partitioning using k-means ($O(N^2)$), reproduction by inverse modeling ($O(N^3)$), and global replacement ($O(mN^2)$). The overall computational complexity is $O(N^3)$.

III. Validation of the Algorithm

We introduce the benchmark problems used for the validation experiments and the parameter settings used in all CMOEAs. Then, we present the performance metric for assessing the quality of the approximate Pareto-optimal solutions generated by each competing algorithm. At last, the experimental results are described and analyzed.

A. Experimental Setup

The validation process was designed to evaluate the performance of the IM-C-MOEA/D on a range of real-world problems, RWMOP1 to RWMOP35 [1]. The features are detailed in columns 2 to 7 of Table I. In particular, the definition of the population size and number of function evaluations in each problem was extracted from the original RWMOP paper [1]. The problems are categorized into Mechanical Design, Chemical Engineering, Process, Design and Synthesis, and Power Electronics.

IM-C-MOEA/D and IM-C-MOEA were implemented in the PlatEMO, a MATLAB-based platform in which the other CMOEAs codes are available. The experiments run in a computer with 32 gigabytes of RAM and a 3.60GHz 8-core Intel Core i9-9900KF processor.

We used the parameters informed by the literature for the CMOEAs. The variation operators for MOEA/DD [16] and RVEA [15] are SBX and polynomial mutation [12], with both distribution indices set to 20 and the probabilities of crossover and mutation set to 1.0 and $1/d$ (d is the number of decision variables). The penalty parameter α of APD in LMOCSO [14] is set to 2. In POCEA [13], the parameter K is set to 5. For the inverse models, IM-MOEA [3], IM-MOEA/D [6], IM-C-MOEA, and IM-C-MOEA/D, the two main parameters are the number of groups K set to 10, and the model group size L, set to 3 or 2 when the problem has only two decision variables.

The performance of each algorithm is evaluated through the widely-used metric, namely the hypervolume (HV) [19]. **HV**: Given a reference point $\mathbf{z} = (z_1 \ldots z_n)^T$ dominated by all Pareto-optimal solutions, the HV of a set of solutions P is defined as the volume of the objective space dominated by all solutions in P, bounded by $\mathbf{z}$:

$$HV(P, \mathbf{z}) = Vol\left(\bigcup_{\mathbf{x} \in P}[\mathbf{f}(\mathbf{x}), \mathbf{z}]\right), \qquad (5)$$

where $Vol(\cdot)$ denotes the Lebesgue measure, and $[\mathbf{f}(\mathbf{x}), \mathbf{z}]$ represents the hyperrectangle formed by the fitness of the solution $\mathbf{x}$ and the reference point $\mathbf{z}$. The larger the HV, the better the approximation quality of P. We use a reference point 10% higher than the upper bound of the PF for our experiments. To reduce the computational complexity of determining HV if $m > 4$, we use the Monte Carlo method with 1,000,000 sampling points to approximate its value.

The tests were run 30 times independently, and each performance mean and standard deviation were recorded. The Wilcoxon rank-sum test was conducted to analyze the results with a significance level 0.05. The findings were interpreted with symbols +, -, and $\approx$, meaning that the result by another MOEA is significantly better, worse, or statistically equivalent to the results obtained by IM-C-MOEA/D.

B. Experiments

Tables I and II present the experimental results of IM-C-MOEA/D compared with the chosen CMOEAs for solving the RWMOP1-35 [1]. For each test problem, the number of objectives (m) and decision variables (d) vary from 2 to 5 and 2 to 30. The maximum number of constraints is 29.

Table I presents the HV values achieved by the original models (IM-MOEA and IM-MOEA/D) and their versions for handling constraints: IM-C-MOEA/D and IM-C-MOEA share features regarding the constraint violation criterion as Jain and Deb [4], [12] have proposed. The experimental results show that MOEAs with constraint treatments performed better in 26 out of 35 problems. The performance was statistically equivalent for the remaining nine instances.

Table II presents experimental results achieved by the IM-C-MOEA/D and five state-of-the-art CMOEAs used for solving real-world multi-objective problems, RWMOP1-35. Out of the six algorithms, IM-C-MOEA/D achieved the best performance in 11 problems, followed by IM-C-MOEA in 6 problems, RVEA and POCEA in 4 problems each, MOEA/DD in 3 problems, and LMOCSO in 2 problems.

IM-C-MOEA/D was found to be better in the unconstrained problem (RWMOP 9), with many objectives (RWMOP 11), and in the context of low inequality constraints in general (RWMOP 1-21, 25-29). IM-C-MOEA performed closest to its version with decomposition in terms of performance, excelling in finding feasible solutions to problems with equality constraints (RWMOP 23) and many constraints (RWMOP 35). POCEA performed better on problems with equality restrictions (RWMOP 1, 22-24). RVEA and MOEA/DD performed better on problems with more inequality restrictions and decision variables (RWMOP 30-35).

The performance of RWMOPs 28, 33, and 34 can be attributed to the inherent features of these problem sets. RWMOP 28 involves the simultaneous management of inequality and equality constraints. Furthermore, RWMOP 33 and 34 involve 30 decision variables and 29 inequality constraints. Such scenarios present challenges in identifying feasible

TABLE I

MEAN AND STANDARD DEVIATION OF THE HYPERVOLUME ON RWMOP1-35. BEST PERFORMANCES ARE HIGHLIGHTED. SETTINGS FOR THE POPULATION SIZE (N), THE NUMBER OF OBJECTIVES (m), DECISION VARIABLES (d), THE MAXIMUM OF FITNESS EVALUATION (FE), AND QUANTITY OF INEQUALITY (ng) AND EQUALITY (nh) CONSTRAINTS.

Problem	N	m	d	ng	nh	FE	IM-C-MOEA	IM-MOEA	IM-C-MOEA/D	IM-MOEA/D
Mechanical Design Problems										
Pressure Vessel Design	80	2	4	2	2	20000	4.6051e-1 3.13e-2 +	0.0000e+0 (0.00e+0)	5.8560e-1 7.39e-3 +	0.0000e+0 (0.00e+0)
Vibrating Platform Design	80	2	5	5	0	20000	3.6766e-1 7.79e-3 +	0.0000e+0 (0.00e+0)	3.3757e-1 3.56e-2 +	0.0000e+0 (0.00e+0)
Two Bar Truss Design	80	2	3	3	0	20000	8.6356e-1 6.18e-3 +	1.9427e-1 (2.74e-1)	7.1513e-1 4.62e-2 +	0.0000e+0 (0.00e+0)
Welded Beam Design	80	2	4	4	0	20000	7.8057e-1 2.14e-2 +	2.9166e-1 (2.62e-1)	5.0765e-1 9.09e-2 +	0.0000e+0 (0.00e+0)
Disc Brake Design	80	2	4	4	0	20000	4.2608e-1 7.55e-4 +	2.6687e-1 (1.78e-2)	4.2795e-1 8.56e-4 +	2.3416e-1 (6.83e-3)
Speed Reducer Design	80	2	7	11	0	20000	2.7533e-1 2.31e-4 +	0.0000e+0 (0.00e+0)	2.7468e-1 3.49e-4 +	0.0000e+0 (0.00e+0)
Gear Train Design	80	2	4	1	0	20000	4.7110e-1 1.89e-3 +	4.6305e-1 (2.97e-3)	4.7906e-1 2.08e-3 +	4.7157e-1 (5.36e-3)
Car Side Impact Design	105	3	7	9	0	26250	2.3507e-2 3.31e-4 +	2.0817e-2 (8.62e-4)	2.3871e-2 4.29e-4 +	1.6289e-2 (1.46e-3)
Four Bar Plane Truss	80	2	4	0	0	20000	3.6739e-1 8.50e-3 ≈	3.6587e-1 (7.52e-3)	4.0321e-1 2.05e-3 +	3.3149e-1 (1.72e-2)
Two Bar Plane Truss	80	2	2	2	0	20000	8.3546e-1 (4.69e-3) ≈	8.3677e-1 3.71e-3	8.4480e-1 5.81e-4 +	8.3696e-1 (2.18e-3)
Water Resources Management	212	5	3	7	0	53000	8.4717e-2 2.04e-3 +	7.1942e-2 (3.29e-3)	9.0599e-2 1.47e-3 +	7.5898e-2 (2.86e-3)
Simply Supported I-beam Design	80	2	4	1	0	20000	4.9200e-1 1.73e-2 +	2.0577e-1 (1.20e-1)	4.0147e-1 5.92e-2 +	2.1557e-4 (1.18e-3)
Gear Box Design	105	3	7	11	0	26250	8.6440e-2 2.76e-4 +	1.1527e-3 (6.31e-3)	8.7439e-2 1.99e-4 +	0.0000e+0 (0.00e+0)
Multiple Disk Clutch Brake Design	80	2	5	8	0	20000	5.3033e-1 1.75e-2 +	1.3503e-1 (1.15e-1)	4.9617e-1 3.39e-2 +	4.4961e-2 (3.41e-2)
Spring Design	80	2	3	8	0	20000	4.9915e-1 6.04e-3 +	0.0000e+0 (0.00e+0)	2.8940e-1 7.85e-2 +	0.0000e+0 (0.00e+0)
Cantilever Beam Design	80	2	2	2	0	20000	7.1653e-1 1.01e-2 +	6.1425e-1 (7.77e-2)	5.6697e-1 7.51e-2 +	1.1608e-1 (3.79e-3)
Bulk Carrier Design	105	3	6	9	0	26250	1.9777e-1 2.98e-2 +	1.6645e-2 (6.33e-2)	1.9138e-1 4.20e-2 +	1.9777e-2 (5.91e-2)
Front Rail Design	80	2	3	3	0	20000	3.8177e-2 3.28e-4 +	3.6332e-2 (8.30e-4)	4.0302e-2 8.05e-5 +	3.9935e-2 (2.12e-4)
Multi-product Batch Plant	105	3	10	10	0	26250	3.2244e-1 6.73e-3 +	0.0000e+0 (0.00e+0)	3.2377e-1 4.52e-3 +	0.0000e+0 (0.00e+0)
Hydro-static Thrust Bearing Design	80	2	4	7	0	20000	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
Crash Energy Management for High-speed Train	80	2	6	4	0	20000	3.1130e-2 2.39e-4 +	3.1048e-2 (1.61e-4)	3.1405e-2 6.63e-5 +	3.0760e-2 (7.36e-6)
Chemical Engineering Problems										
Haverly's Pooling Problem	80	2	9	2	4	20000	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
Reactor Network Design	80	2	6	1	4	20000	1.1034e-1 1.59e-1 +	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
Heat Exchanger Network Design	105	3	9	0	6	26250	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
Process, Design, and Synthesis Problems										
Process Synthesis Problem	80	2	2	2	0	20000	2.1984e-1 2.41e-3 +	1.7608e-1 (1.07e-1)	2.3916e-1 1.36e-3 +	2.3282e-1 (1.85e-5)
Process Synthesis and Design Problem	80	2	3	1	1	20000	1.6593e-1 2.45e-2 +	0.0000e+0 (0.00e+0)	1.7561e-1 1.94e-2 +	0.0000e+0 (0.00e+0)
Process Flow Sheeting Problem	80	2	3	3	0	20000	1.9524e+2 2.44e+2 +	1.0780e-1 (3.31e-1)	1.4589e+5 4.60e+5 +	5.2170e+2 (2.30e+3)
Two Reactor Problem	80	2	7	4	4	20000	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
Process Synthesis Problem	80	2	7	9	0	20000	6.9427e-1 1.18e-1 +	0.0000e+0 (0.00e+0)	7.5814e-1 3.19e-2 +	0.0000e+0 (0.00e+0)
Power Electronics Problems										
Synch. Optimal PWM of 3-level Inverters	80	2	25	24	0	80000	3.4240e-2 1.07e-1 +	0.0000e+0 (0.00e+0)	1.0268e-1 2.36e-1 +	0.0000e+0 (0.00e+0)
Synch. Optimal PWM of 5-level Inverters	80	2	25	24	0	80000	1.5029e-1 5.78e-2 +	0.0000e+0 (0.00e+0)	2.0911e-2 1.15e-1 +	0.0000e+0 (0.00e+0)
Synch. Optimal PWM of 7-level Inverters	80	2	25	24	0	80000	5.5420e-1 1.53e-1 +	0.0000e+0 (0.00e+0)	9.9896e-2 2.59e-1 +	0.0000e+0 (0.00e+0)
Synch. Optimal PWM of 9-level Inverters	80	2	30	29	0	80000	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
Synch. Optimal PWM of 11-level Inverters	80	2	30	29	0	80000	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
Synch. Optimal PWM of 13-level Inverters	80	2	30	29	0	80000	2.7882e-2 7.35e-2 +	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
								26/0/9		26/0/9

solutions, highlighting the need for effective constraint-handling techniques and optimization strategies to address such constrained optimization situations.

IM-MOEA/D was designed to deal with large-scale (up to 600 decision variables) and many objectives (up to 6) optimization problems. Adding a constraint handling technique, IM-C-MOEA/D can extend to real-world CMOPs.

IV. CONCLUSION

This paper proposed an inverse modeling CMOEA based on decomposition (IM-C-MOEA/D) for dealing with CMOPs with multiple or many objectives. This model uses a k-means scheme for clustering within the objective space, the selection criterion based on a global replacement to choose the most appropriate weight vector from the population, and a constraint handling approach to deal with CMOPs. IM-C-MOEA/D reached competitive performance compared to 6 state-of-the-art CMOEAs in most of the 35 real-world CMOPs [1] with different inequality and equality constraints.

IM-C-MOEA/D is an improved version of IM-MOEA/D, inheriting favorable features such as mapping the objective space to the decision space. Despite the achievements, we can observe limitations in the IM-C-MOEA/D, which raise possibilities for future lines of research:

1) Adapting the weight vectors might improve the diversity of the solutions over the evolutionary process since the shape of the Pareto front directly affects the performance of MOEAs based on decomposition;
2) IM-C-MOEA/D has various parameters. Thus, it may be worth investigating the influence of parameter setting on the model performance;
3) Exploring constraint-handling techniques to reach feasible solutions in problems that involve combinations of inequality and equality constraints and numerous decision variables.

REFERENCES

[1] A. Kumar, G. Wu, M. Z. Ali, Q. Luo, R. Mallipeddi, P. N. Suganthan, and S. Das, "A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results," *Swarm and Evolutionary Computation*, vol. 67, p. 100961, 2021.

[2] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, and K. C. Tan, "A survey on evolutionary constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 2, pp. 201–221, 2023.

[3] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using gaussian process-based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[4] J. Shen, H. Dong, P. Wang, J. Li, and W. Wang, "An inverse model-guided two-stage evolutionary algorithm for multi-objective optimization," *Expert Systems with Applications*, vol. 225, p. 120198, 2023.

TABLE II

Mean and standard deviation of the Hypervolume metric on RWMOP1-35. Best performances are Highlighted.

Problem	MOEADD	RVEA	LMOCSO	POCEA	IM-C-MOEA	IM-C-MOEA/D
Mechanical Design Problems						
RWMOP1	9.5415e-2 (1.36e-2) −	5.8494e-1 (6.41e-3) ≈	1.6634e-1 (3.78e-1) −	5.9008e-1 (5.44e-3) +	4.6051e-1 (3.13e-2) −	5.8560e-1 (7.39e-3)
RWMOP2	1.6658e-1 (1.38e-1) −	1.2969e-1 (1.07e-1) −	0.0000e+0 (0.00e+0) −	1.0722e-1 (1.07e-1) −	3.6766e-1 (7.79e-3) +	3.3757e-1 (3.56e-2)
RWMOP3	1.8434e-1 (1.68e-1) −	1.3640e-1 (1.32e-1) −	1.2892e-1 (1.63e-1) −	2.7468e-1 (2.55e-1) −	8.6356e-1 (6.18e-3) +	7.1513e-1 (4.62e-2)
RWMOP4	1.6516e-1 (1.68e-1) −	1.2465e-1 (2.52e-1) −	2.0180e-1 (2.77e-1) −	7.8000e-2 (1.90e-1) −	7.8057e-1 (2.14e-2) +	5.0765e-1 (9.09e-2)
RWMOP5	4.2741e-1 (1.37e-3) ≈	4.1089e-1 (7.09e-3) −	2.3936e-1 (6.63e-4) −	4.1987e-1 (4.97e-3) −	4.2608e-1 (7.55e-4) −	4.2795e-1 (8.56e-4)
RWMOP6	2.7637e-1 (4.00e-4) +	2.7520e-1 (1.97e-4) +	3.7843e-3 (2.07e-2) −	2.7163e-1 (2.53e-3) −	2.7533e-1 (2.31e-4) −	2.7468e-1 (3.49e-4)
RWMOP7	4.7792e-1 (1.89e-3) −	4.7329e-1 (2.52e-3) −	4.8097e-1 (1.87e-4) +	4.7703e-1 (2.58e-3) −	4.7110e-1 (1.89e-3) −	4.7906e-1 (2.08e-3)
RWMOP8	1.8588e-2 (1.98e-3) −	2.4442e-2 (6.91e-4) +	2.1393e-2 (8.22e-4) −	1.0487e-3 (4.00e-3) −	2.3507e-2 (3.31e-4) −	2.3871e-2 (4.29e-4)
RWMOP9	5.2956e-2 (4.20e-16) −	4.0183e-1 (1.72e-3) −	5.4659e-2 (1.11e-3) −	3.6764e-1 (8.82e-3) −	3.6739e-1 (8.50e-3) −	4.0321e-1 (2.05e-3)
RWMOP10	7.8343e-2 (1.61e-14) −	8.3741e-1 (3.01e-3) −	1.3234e-1 (1.24e-2) −	8.3582e-1 (3.81e-3) −	8.3546e-1 (4.69e-3) −	8.4480e-1 (5.81e-4)
RWMOP11	3.5929e-2 (9.13e-3) −	6.3899e-2 (5.51e-3) −	1.7825e-2 (3.39e-3) −	6.5821e-2 (4.27e-3) −	8.4717e-2 (2.04e-3) −	9.0599e-2 (1.47e-3)
RWMOP12	8.6529e-2 (5.03e-2) −	3.7746e-1 (1.61e-2) −	1.9747e-1 (1.21e-1) −	1.0165e-1 (1.32e-1) −	4.9200e-1 (1.73e-2) +	4.0147e-1 (5.92e-2)
RWMOP13	7.1196e-2 (5.16e-3) −	8.7233e-2 (1.94e-4) ≈	3.6925e-2 (3.16e-2) −	2.5287e-2 (3.66e-2) −	8.6440e-2 (2.76e-4) −	8.7439e-2 (1.99e-4)
RWMOP14	5.3019e-1 (2.89e-2) +	5.9811e-2 (2.95e-2) −	2.8898e-1 (1.54e-1) −	6.4845e-2 (2.82e-2) −	5.3033e-1 (1.75e-2) +	4.9617e-1 (3.39e-2)
RWMOP15	1.5010e-1 (1.02e-1) −	4.9408e-1 (8.41e-2) +	0.0000e+0 (0.00e+0) −	4.8424e-1 (1.34e-1) +	4.9915e-1 (6.04e-3) +	2.8940e-1 (7.85e-2)
RWMOP16	7.9114e-2 (6.47e-5) −	7.3781e-1 (5.49e-3) +	2.3627e-1 (2.04e-2) −	6.9917e-1 (1.51e-2) +	7.1653e-1 (1.01e-2) +	5.6697e-1 (7.51e-2)
RWMOP17	2.5019e-1 (1.27e-2) +	1.8445e-1 (4.02e-2) ≈	1.5358e-1 (1.08e-1) ≈	0.0000e+0 (0.00e+0) −	1.9777e-1 (2.98e-2) ≈	1.9138e-1 (4.20e-2)
RWMOP18	3.7327e-2 (2.82e-4) −	4.0146e-2 (9.09e-5) ≈	4.0171e-2 (7.79e-5) ≈	4.0252e-2 (8.40e-5) ≈	3.8177e-2 (3.28e-4) −	4.0302e-2 (8.05e-5)
RWMOP19	2.7191e-1 (2.07e-2) −	2.5669e-1 (2.26e-2) −	0.0000e+0 (0.00e+0) −	1.1002e-3 (6.03e-3) −	3.2244e-1 (6.73e-3) ≈	3.2377e-1 (4.52e-3)
RWMOP20	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
RWMOP21	2.9893e-2 (1.27e-3) −	3.1280e-2 (3.05e-4) −	0.0000e+0 (0.00e+0) −	2.9334e-2 (4.27e-5) −	3.1130e-2 (2.39e-4) −	3.1405e-2 (6.63e-5)
Chemical Engineering Problems						
RWMOP22	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	2.8788e-1 (4.32e-1) +	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
RWMOP23	1.6921e-1 (1.60e-1) +	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	1.0249e+0 (2.08e-1) +	1.1034e-1 (1.59e-1) +	0.0000e+0 (0.00e+0)
RWMOP24	3.3333e-2 (1.83e-1) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	2.0518e+5 (7.65e+5) +	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
Process, Design, and Synthesis Problems						
RWMOP25	9.0909e-2 (6.38e-16) −	2.3564e-1 (9.68e-4) −	2.3450e-1 (1.05e-3) −	2.3502e-1 (1.90e-3) −	2.1984e-1 (2.41e-3) −	2.3916e-1 (1.36e-3)
RWMOP26	1.4428e-1 (3.49e-2) −	1.2908e-1 (7.21e-3) −	3.0328e-3 (1.66e-2) −	1.1034e-1 (6.18e-2) −	1.6593e-1 (2.45e-2) −	1.7561e-1 (1.94e-2)
RWMOP27	5.8817e+0 (1.30e+1) −	4.3310e+4 (1.57e+4) −	1.9076e+6 (6.94e+6) +	1.9727e+4 (1.19e+4) ≈	1.9524e+2 (2.44e+2) −	1.4589e+5 (4.60e+5)
RWMOP28	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
RWMOP29	7.3212e-1 (8.56e-2) ≈	5.9826e-1 (1.40e-1) −	0.0000e+0 (0.00e+0) −	2.4476e-1 (2.21e-1) −	6.9427e-1 (1.18e-1) −	7.5814e-1 (3.19e-2)
Power Electronics Problems						
RWMOP30	1.7041e-1 (2.75e-1) ≈	3.3266e-1 (3.11e-1) +	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	3.4240e-2 (1.07e-1) ≈	1.0268e-1 (2.36e-1)
RWMOP31	2.1301e-1 (2.87e-1) +	1.6258e-1 (2.67e-1) +	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	1.5029e-2 (5.78e-2) ≈	2.0911e-2 (1.15e-1)
RWMOP32	3.8724e-1 (3.79e-1) +	3.3306e-1 (3.55e-1) +	0.0000e+0 (0.00e+0) −	4.4262e-2 (1.53e-1) ≈	5.5420e-2 (1.53e-1) ≈	9.9896e-2 (2.59e-1)
RWMOP33	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
RWMOP34	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
RWMOP35	3.2693e-2 (1.26e-1) ≈	4.8730e-2 (1.29e-1) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	2.7882e-2 (7.35e-2) +	0.0000e+0 (0.00e+0)
+/-/≈	6/19/10	8/16/11	2/22/11	6/19/10	10/13/12	

[5] J. Yuan, H.-L. Liu, Y.-S. Ong, and Z. He, "Indicator-based evolutionary algorithm for solving constrained multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 379–391, 2021.

[6] L. R. C. Farias and A. F. R. Araújo, "IM-MOEA/D: An inverse modeling multi-objective evolutionary algorithm based on decomposition," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 462–467.

[7] Y. Lin, H. Liu, and Q. Jiang, "Dynamic reference vectors and biased crossover use for inverse model based evolutionary multi-objective optimization with irregular pareto fronts," *Applied Intelligence*, vol. 48, no. 9, pp. 3116–3142, 2018.

[8] P. Gholamnezhad, A. Broumandnia, and V. Seydi, "An inverse model-based multiobjective estimation of distribution algorithm using random-forest variable importance methods," *Computational Intelligence*, 2020.

[9] Z. Zhang, S. Liu, W. Gao, J. Xu, and S. Zhu, "An enhanced multi-objective evolutionary optimization algorithm with inverse model," *Information Sciences*, vol. 530, pp. 128–147, 2020.

[10] A. L. da Costa Oliveira, R. Gusmão, and A. Britto, "Learning decision variables in many-objective optimization problems," *IEEE Latin America Transactions*, vol. 21, no. 11, pp. 1157–1163, 2023.

[11] A. L. da Costa Oliveira, A. Britto, and R. Gusmão, "A framework for inverse surrogate modeling for fitness estimation applied to multi-objective evolutionary algorithms," *Applied Soft Computing*, vol. 146, p. 110672, 2023.

[12] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach,"

[13] *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2013.

[13] C. He, R. Cheng, Y. Tian, X. Zhang, K. C. Tan, and Y. Jin, "Paired offspring generation for constrained large-scale multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 3, pp. 448–462, 2020.

[14] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, "Efficient large-scale multiobjective optimization based on a competitive swarm optimizer," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3696–3708, 2019.

[15] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[16] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2014.

[17] Z. Wang, Q. Zhang, M. Gong, and A. Zhou, "A replacement strategy for balancing convergence and diversity in moea/d," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 2132–2139.

[18] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[19] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.