Impact of Reformulation for the Real-World Constrained Multi-Objective Problems on Evolutionary Algorithms

Jinglin Wang*
12432701@mail.sustech.edu.cn
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, Guangdong, China

Hongyu Zhu*
12432733@mail.sustech.edu.cn
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, Guangdong, China

Yang Nan
nany@mail.sustech.edu.cn
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, Guangdong, China

Hisao Ishibuchi†
hisao@sustech.edu.cn
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, Guangdong, China

ABSTRACT

The real-world constrained multi-objective optimization problems (CMOPs) pose significant challenges for evolutionary multi-objective optimization (EMO) algorithms due to complex constraints. This paper explores the impact of reformulation strategies on EMO algorithms, focusing on equality constraint substitution and cumulative variable introduction. Experiments on 21 real-world CMOPs and their reformulated versions using NSGA-II, CCMO, and PPS-MOEA/D show that reformulation enhances feasibility rates and solution quality, particularly for low-feasibility and high-dimensional problems. Reformulating problems simplifies the search space, allowing algorithms like NSGA-II and CCMO to achieve significantly higher feasibility rates. These results underscore the effectiveness of problem reformulation in alleviating the difficulty of solving real-world CMOPs and provide valuable insights for improving EMO algorithm performance on complex problems.

CCS CONCEPTS

• **Evolutionary algorithm**; • **Computaing methodologies**; • **Mathematics of computing**; • **Theory of computation**;

KEYWORDS

Evolutionary multi-objective optimization (EMO), Problem Reformulation, Real-world problems

ACM Reference Format:
Jinglin Wang, Hongyu Zhu, Yang Nan, and Hisao Ishibuchi. 2025. Impact of Reformulation for the Real-World Constrained Multi-Objective Problems on Evolutionary Algorithms. In *Genetic and Evolutionary Computation Conference (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3712255.3726689

*Both authors contributed equally to this research.
†Corresponding author.

GECCO '25 Companion, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1464-1/2025/07
https://doi.org/10.1145/3712255.3726689

1 INTRODUCTION

In recent years, standard artificially generated test problems (e.g., DAC-SMOP [9], LIR-CMOP [2], and MW [8]) have been widely used to evaluate the performance of evolutionary multi-objective optimization (EMO) algorithms. These test problems are designed to introduce large infeasible regions, complex Pareto front shapes, and specific constraint characteristics, simulating the challenging nature of constrained multi-objective optimization problems (CMOPs). Many artificial test problems contain large infeasible regions in the objective space, requiring EMO algorithms to employ mechanisms like PPS-MOEA/D [3] to navigate these regions and reach the Pareto front. While valuable for algorithm development, these problems have limitations.

Real-world CMOPs (e.g., RE [11] and RCM [7]) differ significantly from artificially generated test problems. In real-world problems, feasible regions are more compact and evenly distributed, in contrast to the large infeasible regions found in artificial problems [10]. Real-world problems generally do not feature large infeasible regions between initial solutions and the Constrained Pareto Front (CPF), allowing optimization algorithms to converge to the Pareto front more quickly without complex traversal mechanisms.

However, the real-world problems are not always easy to be solved. For certain real-world CMOPs, such as RCM23, RCM24 and RCM30–35, existing EMO algorithms have a low success rate in finding feasible solutions. Even more critically, for some real-world problems, such as RCM22, RCM24, and RCM36–46, EMO algorithms may fail entirely to locate any feasible solutions [7]. These phenomena highlight the inherent complexity of constraints and the misalignment between algorithm design and real-world problem characteristics.

Despite challenges, real-world problems offer opportunities for advanced optimization. Refining problem formulation can greatly ease solving complex tasks. For example, Cheng He et al. used weighted variables and indicator functions to reduce large-scale multi-objective problems to low-dimensional single-objective ones, enhancing efficiency [5]. They also integrated reformulation with decomposition to design more effective EMO algorithms [4]. Similarly, Ruwang Jiao et al. reformulated multi-objective feature selection as a constrained problem, improving classification performance [6]. Such strategies simplify problem-solving without altering core problem nature.

Currently, most research focuses on improving algorithms, with little attention given to problem formulation or reformulation. Problem formulation is often seen as a fixed precondition and rarely analyzed systematically. This limited perspective may hinder algorithm performance on real-world problems. Therefore, exploring the impact of problem reformulation on EMO algorithms' ability to solve real-world CMOPs is both valuable and practical. This paper aims to investigate the role of problem reformulation in improving EMO algorithms through theoretical and experimental research. The core questions addressed are:

- **Problem reformulation strategies**: What reformulation strategies can be applied to real-world CMOPs?
- **Impact of problem reformulation on EMO algorithms**: How do different problem reformulation strategies affect the difficulty of solving problems using EMO algorithms?

This paper is structured as follows: Section2 discusses real-world CMOPs and suitable reformulation strategies; Section3 presents experiments on multiple EMO algorithms applied to both original and reformulated problems, analyzing the effect of reformulation on performance; Section 4 concludes and suggests future research directions.

2 PROBLEM REFORMULATION STRATEGIES

In this section, we explain the problem reformulation strategies mentioned in Section 1. Two reformulation strategies applied to the RCM problem set are introduced here.

2.1 Equality Constraint Variable Substitution Strategy

In optimization problems, equality constraint variable substitution is a common reformulation technique aimed at reducing the number of decision variables and equality constraints by leveraging variable dependencies. This method simplifies the problem's dimensionality, making it easier to solve. Using **RCM22** as an example, this subsection illustrates how to apply equality constraint variable substitution in a CMOP.

Equality constraints in CMOPs are typically expressed as:

$$h_i(x) = 0, \quad i = 1, 2, \ldots, k. \tag{1}$$

By analyzing these constraints, we identify dependencies among variables (e.g., x_i as a function of others) and replace variables accordingly. The benefits of this approach include:

(1) **Reduced dimension and equality constraints:** Eliminating redundant variables and constraints simplifies the optimization problem.
(2) **Preserved mathematical equivalence:** The reformulated problem remains theoretically equivalent to the original.

However, additional inequality constraints may arise to ensure that substituted variables satisfy their box constraints. Depending on the problem's characteristics, some constraints may be implicitly satisfied, while others must be explicitly maintained.

Using **RCM22** as an example, this process of reformulation and the treatment of additional constraints are discussed in detail. Objective Functions:

$$f_1(x) = -9x_1 - 15x_2 + 6x_3 + 16x_4, \tag{2}$$

$$f_2(x) = 10(x_5 + x_6). \tag{3}$$

Subject to:

$$\begin{cases} h_1(x): & x_7 + x_8 - x_4 - x_3 = 0, \\ h_2(x): & x_1 - x_5 - x_7 = 0, \\ h_3(x): & x_2 - x_6 - x_8 = 0, \\ h_4(x): & x_7x_9 + x_8x_9 - 3x_3 - x_4 = 0, \\ g_1(x): & x_7x_9 + 2x_5 - 2.5x_1 \le 0, \\ g_2(x): & x_8x_9 + 2x_6 - 1.5x_2 \le 0. \end{cases} \tag{4}$$

The box constraints:

$$0 \le x_1, x_3, x_4, x_5, x_6, x_8 \le 100, \quad 0 \le x_2, x_7, x_9 \le 200. \tag{5}$$

From the equality constraints h_2 and h_3, we derive:

$$x_2 = x_6 + x_8, \tag{6}$$

$$x_7 = x_1 - x_5. \tag{7}$$

By substituting x_2 and x_7, the problem is simplified as follows:

- **Reduced Decision Variables:** The dimension is reduced from nine to seven.
- **Additional Inequality Constraints:** The range of substituted variables must be preserved:

$$0 \le x_6 + x_8 \le 200, 0 \le x_1 - x_5 \le 200. \tag{8}$$

Among these, $0 \le x_6 + x_8 \le 200$ and $x_1 - x_5 \le 200$ are implicitly satisfied. Therefore, the only additional constraint is:

$$g_3(x): \quad x_1 - x_5 \ge 0. \tag{9}$$

The reformulated version of **RCM22** is as fellow:

Objective Functions:

$$f_1(x) = -9x_1 - 15(x_6 + x_8) + 6x_3 + 16x_4, \tag{10}$$

$$f_2(x) = 10(x_5 + x_6). \tag{11}$$

Subject to:

$$\begin{cases} h_1(x): & x_7 + x_8 - x_4 - x_3 = 0, \\ h_2(x): & x_7x_9 + x_8x_9 - 3x_3 - x_4 = 0, \\ g_1(x): & (x_1 - x_5)x_9 + 2x_5 - 2.5x_1 \le 0, \\ g_2(x): & x_8(x_9 - 1.5) + 0.5x_6 \le 0, \\ g_3(x): & x_1 - x_5 \ge 0. \end{cases} \tag{12}$$

The box constraints:

$$0 \le x_1, x_3, x_4, x_5, x_6, x_8 \le 100, \quad 0 \le x_9 \le 200. \tag{13}$$

This strategy is also applicable to RCM23, 24, 28, and RCM36–46. For RCM22–28, the substituted variables often satisfy additional inequality constraints implicitly, reducing problem complexity. In RCM24 and 28, all such constraints are satisfied, significantly simplifying the problem. However, in RCM36–46, due to complex equality constraints, the introduced inequalities cannot be implicitly satisfied and must be retained. Although substitution reduces decision variables and equality constraints, it often increases the number of inequalities, potentially maintaining or even increasing overall problem difficulty.

2.2 Cumulative Variable Reformulation Strategy

In optimization problem reformulation, beyond reducing decision variable dimensions via equality constraint substitution, another effective approach is to introduce intermediate variables based on cumulative relationships (e.g., summation or product).This method captures inter-variable dependencies, simplifies constraints, and enhances clarity. Compared to traditional methods, cumulative variable reformulation reduces explicit constraints and offers a more intuitive problem representation.

For CMOPs like RCM30–35, which have numerous inequality constraints and strict monotonicity, cumulative variables simplify constraint expressions without changing objective functions. This section demonstrates the cumulative variable reformulation using **RCM30**, providing a detailed analysis of the strategy's advantages.

For the original model of **RCM30**, the objective functions consist of two parts:

$$f_1 = \frac{\left(\sum_k (k^{-4}) \left(\sum_{i=1}^{25} s(i)\cos(kx_i)\right)\right)^2}{\sqrt{\sum_k k^4}}, \tag{14}$$

$$f_2 = \left(m - \sum_{i=1}^{25} s(i)\cos(x_i)\right)^2. \tag{15}$$

where:

- x_i are the decision variables, with box constraints $0 < x_i < \frac{\pi}{2}$, $i = 1, 2, \ldots, 25$;
- $s(i)$ and m are given parameters related to the problem characteristics;
- $\sum_k k^{-4}$ is a normalization factor determined by the exponential nature of the weight k.

The constraints of the original problem are as follows:

$$x_{i+1} - x_i > 10^{-5}, \quad i = 1, 2, \ldots, 24. \tag{16}$$

The box constraints:

$$0 < x_i < \frac{\pi}{2}, \quad i = 1, 2, \ldots, 25. \tag{17}$$

These constraints ensure strict monotonically increasing properties between variables.

In the reformulation of **RCM30**, the constraints on monotonically increasing variables x_i are implicitly satisfied by summing variables t_i. To ensure intermediate variables y_i meet the original bounds, a normalized variable $t_{26} \in (0, \frac{\pi}{2} - 24 \times 10^{-5})$ is introduced. The main steps are as follows:

Step 1: Calculate the normalization factor α:

$$\alpha = \frac{t_{26}}{\sum_{i=1}^{25} t_i}. \tag{18}$$

This equation computes the normalization factor α, defined as t_{26} divided by the sum of $t_1, t_2, \ldots, t_{25}$. It scales the accumulated relationships among variables, ensuring the generated intermediate variables y_i satisfy the original problem's box constraints during normalization.

Step 2: Normalize the variables $t_1, t_2, \ldots, t_{25}$ to obtain $\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{25}$:

$$\hat{t}_i = \alpha t_i, \quad i = 1, 2, \ldots, 25. \tag{19}$$

This equation normalizes the original variables $t_1, t_2, \ldots, t_{25}$ by multiplying them by the normalization factor α to obtain new variables $\hat{t_1}, \hat{t_2}, \ldots, \hat{t_{25}}$. This transformation ensures that the new variables fall within the desired range, and the scaling is tied to the value of t_{26}, preventing excessively large values.

Step 3: To implicitly satisfy the constraints in the original problem model, the values of $\hat{t}_i$, $i = 2, 3, \ldots, 25$, should greater than 10^{-5}:

$$\hat{t}_i = \hat{t}_i + 10^{-5}, \quad i = 2, 3, \ldots, 25. \tag{20}$$

This equation implicitly satisfies the original constraint requiring adjacent variables to differ by more than 10^{-5}. By adding 10^{-5} to all $\hat{t}_i$ except $\hat{t}_1$, the resulting y_i meet this condition. Since 10^{-5} is added to 24 values, t_{26} is bounded by $\left(0, \frac{\pi}{2} - 24 \times 10^{-5}\right)$, ensuring y_i remain within the original box constraints $\left(0, \frac{\pi}{2}\right)$.

Step 4: The intermediate variable y_i represents the cumulative sum of variables $\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{25}$:

$$y_i = \sum_{j=1}^{i} \hat{t}_j, \quad i = 1, 2, \ldots, 25. \tag{21}$$

This equation defines the intermediate variable y_i, which is the cumulative sum of the normalized variables $\hat{t_1}, \hat{t_2}, \ldots, \hat{t_{25}}$. By accumulating each normalized variable, we obtain a new variable y_i that represents the cumulative effect of the original variables. This cumulative sum implicitly satisfies the increasing relationship between the variables in the original problem, simplifying the constraints.

Thus, the reformulated version of **RCM30** can be expressed as:

Objective functions:

$$f_1 = \frac{\left(\sum_k (k^{-4}) \left(\sum_{i=1}^{25} s(i)\cos(ky_i)\right)\right)^2}{\sqrt{\sum_k k^4}}, \tag{22}$$

$$f_2 = \left(m - \sum_{i=1}^{25} s(i)\cos(y_i)\right)^2. \tag{23}$$

Where:

$$\begin{cases} \alpha = \frac{t_{26}}{\sum_{i=1}^{25} t_i}, \\ \hat{t}_i = \alpha t_i, \quad i = 1, 2, \ldots, 25, \\ \hat{t}_i = \hat{t}_i + 10^{-5}, \quad i = 2, 3, \ldots, 25, \\ y_i = \sum_{j=1}^{i} \hat{t}_j, \quad i = 1, 2, \ldots, 25. \end{cases} \tag{24}$$

Additionally, the box constraints are adjusted as follows:

$$0 < t_1, t_2, \ldots, t_{25} < 1, 0 < t_{26} < \frac{\pi}{2} - 24 \times 10^{-5}. \tag{25}$$

For the reformulated **RCM30**, since $t_1, t_2, \ldots, t_{25}$ are normalized and their minimum values are fixed, their ranges can be constrained to $(0, 1)$.

This strategy introduces intermediate variables based on the cumulative characteristics (e.g., summation) of normalized variables, implicitly satisfying monotonic constraints and reducing complexity. While adding an extra decision variable, it largely makes the solution space feasible, especially for high-dimensional problems, enhancing solvability and offering a novel modeling approach.

3 IMPACT OF PROBLEM REFORMULATION ON EMO ALGORITHMS

To evaluate the impact of problem reformulation on EMO algorithm performance, we conducted experiments on 21 RCM problems using NSGA-II [1], CCMO [13], and PPS-MOEA/D [3], implemented in PlatEMO [12] with default parameters. Both original and reformulated problems were tested under consistent settings: a population size of 120, a maximum of 1,200,000 function evaluations, and 21 independent runs to reduce randomness. All experiments were performed on a 64-core AMD EPYC server with 256GB memory and 30 parallel threads. Evaluation was based on two key metrics: Feasibility Rate (FR) and Hypervolume (HV), as reported in Table 2 and Table 3, respectively. The FR and HV values for the reformulated problems that perform better than the original versions are highlighted in gray, while FR and HV values for the original problems that perform better than the reformulated versions are highlighted in yellow. All the code and results data tables can be found at https://github.com/lin-12150/Reformulation-for-RWMOPs.

Table 1 presents results under varying equality constraint thresholds for RCM36–RCM46. For RCM22–RCM28, a threshold of $\epsilon = 10^{-4}$ was used. However, for high-dimensional RCM36–RCM46, no feasible solutions were found under this threshold. Experiments showed that feasible solutions were only obtained when $\epsilon = 1$ for RCM36–RCM39 and $\epsilon = 10^{-1}$ for RCM40–RCM46. These thresholds were adopted to ensure fair comparison between original and reformulated problems. For evaluation, all reformulated solutions were projected back to the original problem space to compute FR and HV.Table 2 shows that reformulation notably improves FR, especially in problems with poor feasibility. For instance, NSGA-II and CCMO failed to find any feasible solutions in original RCM22 and RCM24 (FR = 0), but both achieved an FR of 1 after reformulation. In RCM23, where FR was already 1, reformulation preserved this performance. For high-dimensional RCM36–RCM46, results were mixed. Reformulation improved NSGA-II's FR in RCM36 (from 0.4762 to 0.7619), but CCMO's FR remained unchanged. These outcomes suggest that while reformulation reduces variable count and constraint complexity, some algorithms still struggle with constraint satisfaction.Table 3 demonstrates that HV also benefits from reformulation. For example, NSGA-II and CCMO had no feasible solutions in original RCM22 but achieved HVs of 0.6038 and 0.9045, respectively, after reformulation. PPS-MOEA/D showed minimal HV change, reflecting its inherent robustness. In RCM30–RCM35, which use cumulative variable reformulation, FR and HV remained consistently high across all algorithms, validating the effectiveness of this strategy for moderate-difficulty problems.

In summary, problem reformulation—through equality constraint substitution or cumulative variables—significantly improves feasibility and solution quality in EMO algorithms. However, its effectiveness varies with problem complexity and algorithm characteristics, underscoring the importance of selecting appropriate reformulation strategies.

4 CONCLUSION

This paper discusses the role of problem reformulation in solving real-world constrained multi-objective optimization problems using EMO algorithms. The experimental results demonstrate that reformulation improves algorithm performance by reducing decision variables and simplifying constraints, especially for problems with low feasibility. For instance, in RCM22 and RCM24, the feasibility rates of both NSGA-II and CCMO improved from 0 to 1 after reformulation, highlighting the effectiveness of these strategies in complex problems.The performance comparison of different algorithms shows that PPS-MOEA/D remains stable, achieving high feasibility and HV in most cases, while NSGA-II and CCMO show significant improvements on medium-complexity problems. However, for high-dimensional problems like RCM36-46, while reformulation reduces decision variables, the new inequality constraints increase the difficulty of solving these problems.In conclusion, problem reformulation strategies are valuable for reducing problem-solving complexity and improving solution quality in EMO algorithms. Future research can explore additional reformulation methods for various problem types.

REFERENCES

[1] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 2, (Apr. 2002), 182–197.

[2] Zhun Fan, Wenji Li, Xinye Cai, Han Huang, Yi Fang, Yugen You, Jiajie Mo, Caimin Wei, and Erik Goodman. 2019. An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. *Soft Computing*, 23, 23, 12491–12510.

[3] Zhun Fan, Wenji Li, Xinye Cai, Hui Li, Caimin Wei, Qingfu Zhang, Kalyanmoy Deb, and Erik Goodman. 2019. Push and pull search for solving constrained multi-objective optimization problems. *Swarm and Evolutionary Computation*, 44, 665–679.

[4] Cheng He, Ran Cheng, Ye Tian, and Xingyi Zhang. 2020. Iterated problem reformulation for evolutionary large-scale multiobjective optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, 1–8.

[5] Cheng He, Lianghao Li, Ye Tian, Xingyi Zhang, Ran Cheng, Yaochu Jin, and Xin Yao. 2019. Accelerating large-scale multiobjective optimization via problem reformulation. *IEEE Transactions on Evolutionary Computation*, 23, 6, (Dec. 2019), 949–961.

[6] Ruwang Jiao, Bing Xue, and Mengjie Zhang. 2024. Solving multiobjective feature selection problems in classification via problem reformulation and duplication handling. *IEEE Transactions on Evolutionary Computation*, 28, 4, (Aug. 2024), 846–860.

[7] Abhishek Kumar, Guohua Wu, Mostafa Z. Ali, Qizhang Luo, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das. 2021. A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results. *Swarm and Evolutionary Computation*, 67, 100961.

[8] Zhongwei Ma and Yong Wang. 2019. Evolutionary Constrained Multiobjective Optimization: Test suite construction and performance comparisons. *IEEE Transactions on Evolutionary Computation*, 23, 6, 972–986.

[9] Alanna McNulty, Beatrice Ombuki-Berman, and Andries Engelbrecht. 2024. A comparative study of evolutionary algorithms and particle swarm optimization approaches for constrained multi-objective optimization problems. *Swarm and Evolutionary Computation*, 91, 101742.

[10] Yang Nan, Hisao Ishibuchi, Tianye Shu, and Ke Shang. 2024. Analysis of real-world constrained multi-objective problems and performance comparison of multi-objective algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 576–584.

[11] Ryoji Tanabe and Hisao Ishibuchi. 2020. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89, 106078–.

[12] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. 2017. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization[educational forum]. *IEEE Computational Intelligence Magazine*, 12, 4, 73–87.

[13] Ye Tian, Tao Zhang, Jianhua Xiao, Xingyi Zhang, and Yaochu Jin. 2021. A co-evolutionary framework for constrained multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 25, 1, 102–116.