Constraint Handling in Multiobjective Optimization

Tutorial presented at the IEEE World Congress on Computational Intelligence, WCCI 2022, Padua, Italy, 18–23 July 2022

Bogdan Filipič, Aljoša Vodopija

Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

{bogdan.filipic, aljosa.vodopija}@ijs.si

Presenters

Bogdan Filipič is a senior researcher and head of Computational Intelligence Group at the Department of Intelligent Systems of the Jožef Stefan Institute, Ljubljana, Slovenia, and associate professor of Computer Science at the Jožef Stefan International Postgraduate School. His research interests are in artificial intelligence, evolutionary computation and randomized optimization.

Aljoša Vodopija is a research assistant at the Department of Intelligent Systems of the Jožef Stefan Institute, Ljubljana, Slovenia, and a final-year Ph.D. student of Information and Communication Technologies at the Jožef Stefan International Postgraduate School. In his doctoral research, he focuses on constrained multiobjective optimization with evolutionary algorithms.

Contents

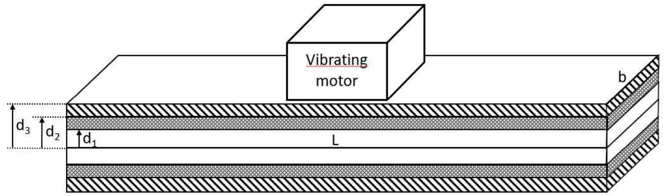
Introduction

Background

- Optimization problems often include both **multiple objectives and constraints**
- **Multiobjective evolutionary algorithms (MOEAs)** – a natural extension of EAs for solving **multiobjective optimization problems (MOPs)**
- Dealing with **constrained multiobjective optimization problems (CMOPs)** long ignored – believed that **constraint handling techniques (CHTs)** for single-objective problems can easily be incorporated into MOEAs
- Recent shift of research focus towards CMOPs

Motivating Example (i)

Vibrating platform (Messac 1996)



- Engineering design problem
- Design variables: d_1, d_2, d_3, b, L
- Task: maximize the fundamental frequency of the platform, minimize its cost

Motivating Example (ii)

Objectives

- f_1 ... fundamental frequency

$$f_1(d_1, d_2, d_3, b, L) = \frac{\pi}{2L^2}\left(\frac{EI}{\mu}\right)^{1/2}$$

$$EI = \frac{2b}{3}\left[E_1 d_1^3 + E_2(d_2^3 - d_1^3) + E_3(d_3^3 - d_2^3)\right]$$

$$\mu = 2b\left[\rho_1 d_1 + \rho_2(d_2 - d_1) + \rho_3(d_3 - d_2)\right]$$

- f_2 ... cost

$$f_2(d_1, d_2, d_3, b) = 2b\left[c_1 d_1 + c_2(d_2 - d_1) + c_3(d_3 - d_2)\right]$$

Motivating Example (iii)

Constraints

- Boundary constraints

$$0.01 \leq d_1 \leq 0.6$$
$$0.01 \leq d_2 \leq 0.6$$
$$0.01 \leq d_3 \leq 0.6$$
$$0.35 \leq b \leq 0.5$$
$$3 \leq L \leq 6$$

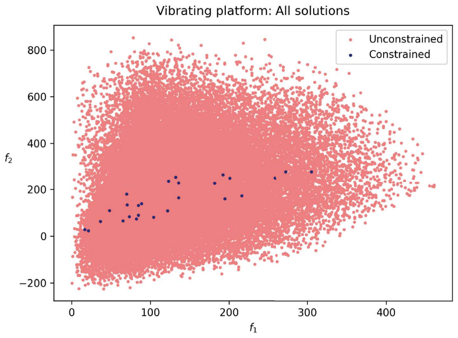
- Inequality constraints

$$0 \leq d_2 - d_1 \leq 0.01$$
$$0 \leq d_3 - d_2 \leq 0.01$$
$$\mu L \leq 2800$$

Motivating Example (iv)

Some problem characteristics

- 5 design variables
- 2 objectives
- 5 constraints
- feasibility ratio* $< 10^{-5}$



*Estimated empirically through solution sampling. Denotes the proportion of feasible solutions among the sampled solutions.

Challenges of Constrained Multiobjective Optimization

- Need to handle both objectives and constraints
- Feasibility ratio can be low
- Objectives and constraints may or may not be correlated
- Feasible region can be disconnected
- etc.

Prerequisites: CMOP Formulation

Constrained multiobjective optimization problem (CMOP):

$$\begin{aligned} \text{minimize} \quad & f_m(x), \quad m = 1, \ldots, M \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, I \\ & h_i(x) = 0, \quad i = I+1, \ldots, I+J \end{aligned}$$

where

- $x = (x_1, \ldots, x_n)$ ... decision vector
- $S \subseteq \mathbb{R}^n$ ... decision space
- $f_m : S \rightarrow \mathbb{R}$ ... objective functions
- $g_i : S \rightarrow \mathbb{R}$ ... inequality constraint functions
- $h_i : S \rightarrow \mathbb{R}$ ... equality constraint functions

Prerequisites: Constraint Violation

The equality constraints are usually reformulated into inequality constraints:

$$g_i(x) = |h_i(x)| - \epsilon \leq 0, \quad i = I+1, \ldots, I+J$$

where $\epsilon > 0$ is a user-defined tolerance value (e.g. 10^{-4})

Constraint violation for a single constraint:

$$v_i = \max(g_i(x), 0)$$

Overall constraint violation for all constraints combined:

$$v(x) = \sum_{i=1}^{I+J} v_i(x)$$

Constraint Handling Techniques (CHTs)

CHTs for Single-Objective Optimization

- Penalty functions
- Solution repair
- Separation of objectives and constraints
- Other approaches

Penalty Functions (i)

Idea: transform a constrained problem into an unconstrained one by adding penalty terms to the objective function:

$$f'(x) = f(x) + \sum_{i=1}^{I} p_i \cdot \max(g_i(x), 0) + \sum_{i=I+1}^{I+J} q_i \cdot |h_i(x)|$$

where

- $f'(x)$ ... modified objective function
- p_i ... penalty factors for inequality constraints
- q_i ... penalty factors for equality constraints

Penalty Functions (ii)

Variants

- Death penalty
- Static penalty
- Dynamic penalty
- Adaptive penalty
- Adjustments and modifications of these variants

Penalty Functions (iii)

- Most popular CHT
- Issue: Setting the penalty factors
- Penalties too low: The algorithm spends a lot of time exploring the infeasible region
- Penalties too high: The algorithm may have difficulties detecting the optimum when it is located at the border of the feasible region

Solution Repair

- Idea: Introduce a procedure for converting infeasible solutions to feasible ones
- Repaired solutions can be used for evaluation only, or can replace the original solutions in the population (Lamarckian evolution)
- Problem-dependent, a specific procedure needed for each problem
- Suitable when repair is easy and of low computational cost

Separation of Objectives and Constraints

- In contrast to penalty functions, these techniques handle objectives and constraints separately
- Examples:
  - Superiority of feasible solutions: Always assign a higher fitness to feasible solutions than to infeasible ones
  - Multiobjective optimization approach: $K+1$ objectives where K is the number of constraints
  - Coevolution: evolve two interacting populations

Other Approaches

- Special representations and operators
- Hybrid techniques
- Ensembles of CHTs
- Landscape-aware constraint handling: Using the concept of violation landscape (Malan 2018; Malan and Moser 2019)

CHTs for Multiobjective Optimization

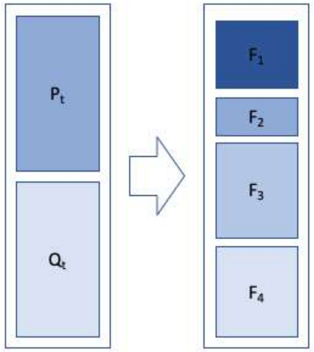
- CHTs incorporated in Nondominated sorting genetic algorithm II (NSGA-II)
- CHTs incorporated in Multiobjective evolutionary algorithm based on decomposition (MOEA/D)
- Advanced techniques

CHTs incorporated in NSGA-II

- Constrained dominance principle (CDP)
- Stochastic ranking (SR)
- Penalty function

NSGA-II

- **The most frequently used algorithm in constrained multiobjective optimization**
- CHT usually incorporated within the sorting procedure



NSGA-II: CDP (Deb, Pratap, Agarwal, et al. 2002)

A solution x is said to **constrained-dominate** a solution y, if any of the following conditions is true:

- Solution x is feasible and solution y is not
- Solutions x and y are both infeasible, but solution x has a smaller overall constraint violation than y
- Solutions x and y are feasible and solution x dominates solution y

The most commonly used CHT in constrained multiobjective optimization (Z. Ma and Y. Wang 2019)

NSGA-II: SR (Geng et al. 2006)

Stochastic ranking selection:

- Feasible solutions are compared based on the dominance relation
- Infeasible solutions are compared either based on on the overall constraint violation or dominance relation
- **The comparison criterion is randomly selected**

IS-MOEA:

- Based on NSGA-II and stochastic ranking selection
- Uses the **infeasible elitists preservation**

NSGA-II: Penalty Function (Woldesenbet et al. 2009)

Transform the objective functions into:

$$f_i'(x) = \begin{cases} f_i(x) & \text{if } x \text{ is feasible} \\ v(x) & \text{if } x \text{ is infeasible and } \rho_F(P) = 0 \\ p_i(x) + d_i(x) & \text{if } x \text{ is infeasible and } \rho_F(P) \neq 0 \end{cases}$$

where

$$p_i(x) = (1 - \rho_F(P))v(x) + \rho_F(P)f_i(x)$$

and

$$d_i(x) = \sqrt{f_i(x)^2 + v(x)^2}$$

CHTs incorporated in MOEA/D

- CDP
- SR
- ϵ-constraint (Epsilon)
- Improved ϵ-constraint (IEpsilon)

MOEA/D

The original problem is decomposed into multiple subproblems

The **Tchebycheff aggregation function** is the most widely used decomposition approach in constrained multiobjective optimization

A subproblem is defined as follows:

$$\text{minimize } g(x \mid \lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i \, |f_i(x) - z_i^*|\}$$

where z^* is an approximation for the ideal point and λ a weight vector

Idea: The aggregation function can be seen as a fitness of the subproblem $\rightarrow$ Easy to incorporate CHTs for single-objective optimization

MOEA/D-DE (i)

MOEA/D-DE:

- Employs **differential evolution** (DE) operator for generating new solutions
- Limits the maximal number of solutions replaced by a better child solution, n_r

The most interesting part of MOEA/D-DE is the **update phase**

Algorithm 1: Update neighboring solutions N with x

```
c ← 0;
while c < n_r and N ≠ ∅ do
    randomly pick y ∈ N;
    if g(x) < g(y) then
        y ← x, c ← c + 1;
    end
    N ← N − {y};
end
```


MOEA/D-DE (ii)

Algorithm 2: Update neighboring solutions N with x

```
c ← 0;
while c < n_r and N ≠ ∅ do
    randomly pick y ∈ N;
    if g(x) < g(y) then
        y ← x, c ← c + 1;
    end
    N ← N − {y};
end
```

Algorithm 3: Update neighboring solutions N with x

```
c ← 0;
while c < n_r and N ≠ ∅ do
    randomly pick y ∈ N;
    if x ⪯ y then
        y ← x, c ← c + 1;
    end
    N ← N − {y};
end
```


MOEA/D: CDP and SR (Jan et al. 2013)

MOEA/D-CDP:

$$x \preceq_{\text{CDP}} y \Leftrightarrow \begin{cases} g(x) < g(y) & \text{if } v(x) = v(y) = 0 \\ v(x) < v(y) & \text{otherwise} \end{cases}$$

MOEA/D-SR:

$$x \preceq_{\text{SR}} y \Leftrightarrow \begin{cases} g(x) < g(y) & \text{if } v(x) = v(y) = 0 \text{ or rand} < p \\ v(x) < v(y) & \text{otherwise} \end{cases}$$

MOEA/D: ϵ-Constraint Technique (Asafuddoula et al. 2012)

MOEA/D-Epsilon:

$$x \preceq_{\epsilon} y \Leftrightarrow \begin{cases} g(x) < g(y) & \text{if } v(x) = v(y) \text{ or } (v(x) \leq \epsilon \text{ and } v(y) \leq \epsilon) \\ v(x) < v(y) & \text{otherwise} \end{cases}$$

The ϵ value is updated in each generation:

$$\epsilon = \bar{v} \cdot \rho_F(P)$$

where

$$\bar{v} = \frac{1}{|P|} \sum_{x \in P} v(x)$$

MOEA/D: Improved ϵ-Constraint Technique (Fan, W. Li, Cai, Huang, et al. 2019)

MOEA/D-IEpsilon:

The ϵ value is updated in each generation:

$$\epsilon(t) = \begin{cases} v(x^\theta) & \text{if } t = 0 \\ (1-\tau)\epsilon(t-1) & \text{if } \rho_F(P) < \alpha \text{ and } t < T_c \\ (1+\tau)v_{\max} & \text{if } \rho_F(P) \geq \alpha \text{ and } t < T_c \\ 0 & \text{if } t \geq T_c \end{cases}$$

where τ, α, T_c are user-defined parameters and $v(x^\theta)$ is the overall constraint violation of the top θ-th individual in the initial population

Advanced Techniques

- Ensembles
- Multiple phase techniques
- Multiple population techniques
- Hybrids
- Coevolution

Advanced Techniques: Two-Phase Framework (Liu et al. 2019)

Two-phase framework:

1. **First phase:** Solve a constrained single-objective problem

$$\begin{aligned} \text{minimize} \quad & f'(x) = \sum_{i=1}^{M} f_i(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, I+J \end{aligned}$$

2. **Second phase:** Apply constrained multiobjective optimization on the original problem **starting with solutions obtained in the first phase**

ToP:

- Differential evolution in the first phase
- NSGA-II (CDP) or IDEA in the second phase

Advanced Techniques: Push and Pull Search (Fan, W. Li, Cai, H. Li, et al. 2019b)

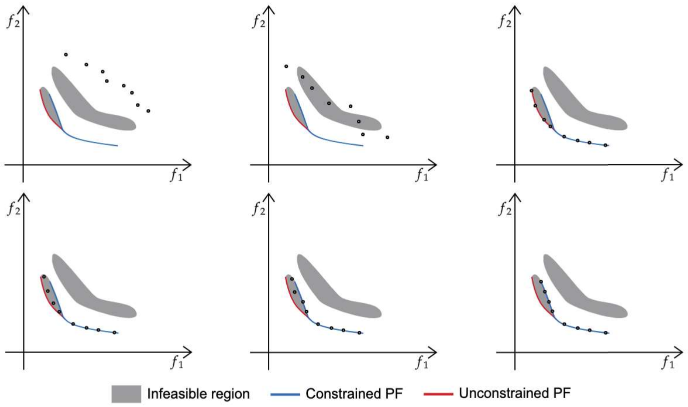
Search is divided into **two stages**:

1. **Push** ignores constraints
2. **Pull** handles infeasible solutions

PPS-MOEA/D:

- Push stage: MOEA/D-DE
- Pull stage: MOEA/D-IEpsilon
- **Parameters for the pull stage assessed in the push stage**

Advanced Techniques: Push and Pull Search (Fan, W. Li, Cai, H. Li, et al. 2019b)



Advanced Techniques: Two-Archive Evolutionary Algorithm (K. Li et al. 2019)

Two complementary archives:

- **Convergence archive:** Maintain the convergence and feasibility of the evolution process
- **Diversity archive:** Maintain the diversity of the evolution process
- A restricted mating mechanism combines parents from the two archives

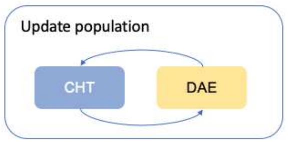
C-TAEA:

- Based on MOEA/D and M2M framework (decomposition of the original multiobjective optimization problem into multiple simpler subproblems)

Advanced Techniques: Detect-and-Escape Strategy (Zhu et al. 2020)

If a “suboptimal” area is **detected**, **escape**:

1. Feasible subregion → search areas which **dominate the current one**
2. Infeasible area → **improve diversity**



MOEA/D-DAE:

- Based on MOEA/D and ϵ-constraint CHT
- Only one detect-and-escape cycle is allowed

Advanced Techniques: Coevolutionary Framework (Tian, T. Zhang, et al. 2021)

Two populations:

1. One population is solving the **original problem**
2. The other one is solving a **helper problem**—a simpler problem derived from the original one

CCMO:

- Coevolutionary framework incorporated into NSGA-II
- Helper problem: original problem without constraints

Advanced Techniques: Two-Rankings Approach (Z. Ma, Y. Wang, and Song 2021)

Two rankings:

1. The first ranking is based on **CDP** (R^{c})
2. The second ranking is based on **Pareto dominance** (R^{p})

The two rankings are then combined into a single ranking:

$$R^{\mathrm{ToR}} = \alpha R^{\mathrm{c}} + (1-\alpha) R^{\mathrm{p}}$$
$$\alpha = \frac{1+\rho_F(P)}{2}$$

ToR-NSGA-II:

- Two-rankings approach integrated into the NSGA-II framework

Advanced Techniques: Additional (i)

- **c-DPEA**: Dual-population-based evolutionary algorithm (M. Ming et al. 2021)
- **CMOEA-MS**: Two-stage constrained multiobjective optimization (Tian, Y. Zhang, et al. 2021)
- **MSCMO**: Multi-stage evolutionary algorithm for constrained multiobjective optimization (H. Ma et al. 2021)
- **POCEA**: Paired offspring generation-based evolutionary algorithm (He et al. 2021)

Advanced Techniques: Additional (ii)

- **DPSEA**: Evolutionary algorithm with a dynamic population size (B. Wang et al. 2022)
- **ICMA**: Indicator-based constrained multiobjective algorithm (Yuan et al. 2022)
- **TriP**: Tri-population based coevolutionary framework (F. Ming et al. 2022)
- **TSTI**: Two stage evolutionary algorithm based on three indicators (Dong et al. 2022)

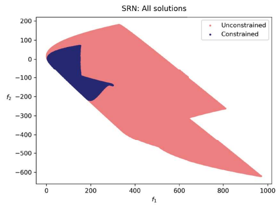
Test Problems

Overview

- Artificial test problems
- Artificial test suites
- Real-world test problems based on mathematical models
- Real-world test problems based on simulation

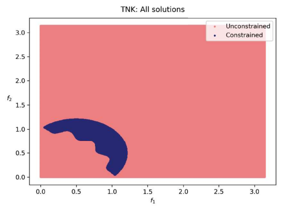
Artificial Test Problems: SRN (Srinivas et al. 1995)

$$
\begin{aligned}
\text{min. } & f_1(x) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2 \\
\text{min. } & f_2(x) = 9x_1 - (x_2 - 1)^2 \\
\text{s.t. } & g_1(x) = x_1^2 + x_2^2 - 225 \leq 0 \\
& g_2(x) = x_1 - 3x_2 + 10 \leq 0 \\
& x_1, x_2 \in [-20, 20]
\end{aligned}
$$



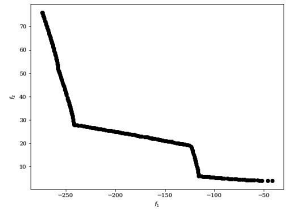
Artificial Test Problems: TNK (Tanaka et al. 1995)

$$
\begin{aligned}
\text{min. } & f_1(x) = x_1 \\
\text{min. } & f_2(x) = x_2 \\
\text{s.t. } & g_1(x) = x_1^2 + x_2^2 - 1 - 0.1\cos(16\tan^{-1}\frac{x}{y}) \geq 0 \\
& g_2(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \leq 0 \\
& x_1, x_2 \in [0, \pi]
\end{aligned}
$$



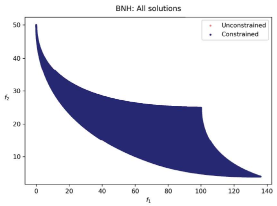
Artificial Test Problems: OSY (Osyczka et al. 1995)

$$
\begin{aligned}
\text{max. } & f_1(x) = 25(x_1 - 2)^2 + (x_2 - 2)^2 + \\
& (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2 \\
\text{min. } & f_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \\
\text{s.t. } & g_1(x) = x_1 + x_2 - 2 \geq 0 \\
& g_2(x) = 6 - x_1 - x_2 \geq 0 \\
& g_3(x) = 2 - x_2 + x_1 \geq 0 \\
& g_4(x) = 2 - x_1 - 3x_2 \geq 0 \\
& g_5(x) = 4 - (x_3 - 3)^2 - x_4 \geq 0 \\
& g_6(x) = (x_5 - 3)^2 + x_6 - 4 \geq 0 \\
& x_1, x_2, x_6 \in [0, 10], x_3, x_5 \in [1, 5], x_4 \in [0, 6]
\end{aligned}
$$



Artificial Test Problems: BNH (Binh et al. 1997)

$$\begin{aligned}
\text{min. } & f_1(x) = 4(x_1^2 + x_2^2) \\
\text{min. } & f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2 \\
\text{s.t. } & g_1(x) = (x_1 - 5)^2 + x_2^2 - 25 \leq 0 \\
& g_2(x) = (x_1 - 8)^2 - (x_2 + 3)^2 - 7.7 \geq 0 \\
& x_1 \in [0, 5], x_2 \in [0, 3]
\end{aligned}$$



Artificial Test Problems: Issues with SRN, TNK, OSY, BNH

Issues:

- Low dimensionality
- Not hard to solve
- Complexity/difficulty not tunable

→ Further proposals: Frameworks for constructing harder tunable problems

Artificial Test Suites: CTP (Deb, Pratap, and Meyarivan 2001)

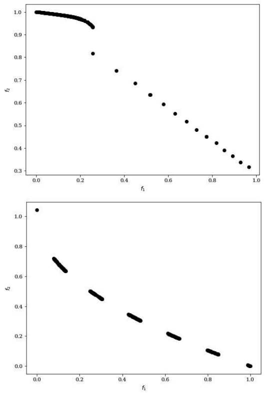
Constrained test problems (CTPs)

Scalable number of decision variables and **tunable constraint difficulties**

Two kinds of difficulty:

- Difficulty in the vicinity of PF
- Difficulty in the entire search space

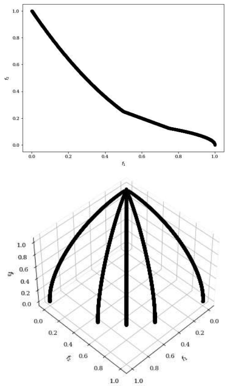
8 bi-objective CMOPs including 1–2 constraints



Artificial Test Suites: CF (Q. Zhang et al. 2008)

Constrained multiobjective test problems from the CEC 2009 Special Session and Competition (CFs)

10 problems with 2 or 3 objectives and 1 or 2 constraints



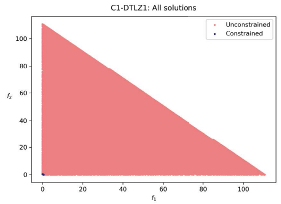
Artificial Test Suites: C-DTLZ (Jain et al. 2014)

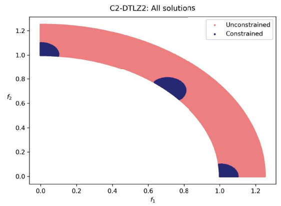
Constrained DTLZ problems (C-DTLZs)

Three types of CMOPs:

- C1: unconstrained PF still optimal, barrier in approaching PF
- C2: only parts of unconstrained PF feasible
- C3: unconstrained PF no longer optimal

6 scalable CMOPs in the number of objectives and constraints





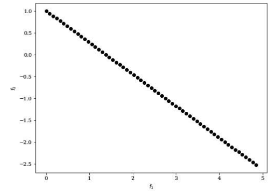
Artificial Test Suites: NCTP (J. Li et al. 2016)

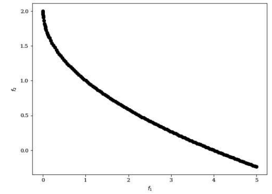
New constrained test problems (NCTPs)

An extension of CTPs:

- Difficulty of convergence is increased
- Infeasible region is increased by an additional constraint

18 bi-objective CMOPs with 1 or 2 constraints





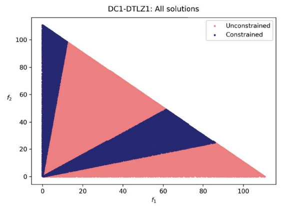
Artificial Test Suites: DC-DTLZ (K. Li et al. 2019)

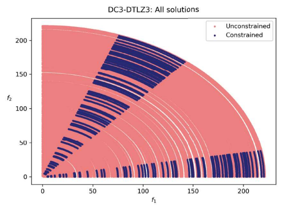
Constrained DTLZ problems where constraints act in the decision space (DC-DTLZs)

Three types of constraints:

- DC1: several infeasible segments
- DC2: unconstrained PF still optimal, barrier in approaching PF
- DC3: decision space consists of several feasible regions

6 scalable CMOPs in the number of objectives and constraints





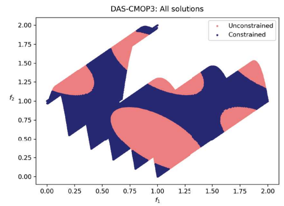
Artificial Test Suites: DAS-CMOP (Fan, W. Li, Cai, H. Li, et al. 2019a)

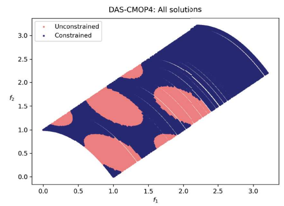
Difficulty-adjustable and scalable CMOPs (DAS-CMOPs)

Test problem kit considering basic difficulty types:

- T1: diversity hardness
- T2: feasibility hardness
- T3: convergence hardness

9 CMOPs of increasing hardness, scalable in the number of objectives

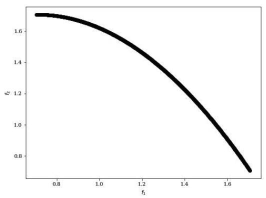


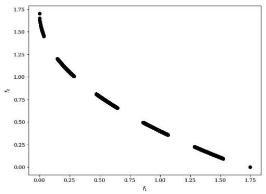


Artificial Test Suites: LIR-CMOP (Fan, W. Li, Cai, Huang, et al. 2019)

Large infeasible region CMOPs (LIR-CMOPs)

14 CMOPs with 2 or 3 objectives and 2 or 3 constraints





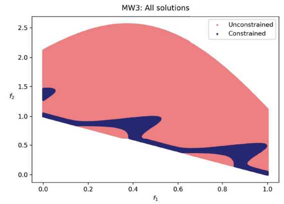
Artificial Test Suites: MW (Z. Ma and Y. Wang 2019)

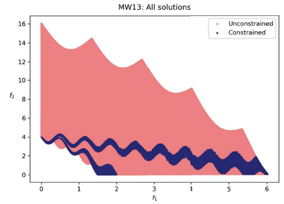
Ma and Wang problems (MWs)

Four types of CMOPs:

- Type I: unconstrained PF remains feasible
- Type II: constrained PF is a part of the unconstrained PF
- Type III: constrained PF consists of a part of the unconstrained PF and part of a boundary
- Type IV: unconstrained PF no longer optimal

11 bi-objective CMOPs and 3 scalable in the number of objective with 1–4 constraints





Artificial Test Suites: Others

- **DOC**: Constrained multiobjective optimization problems with constraints in the decision and objective space (Liu et al. 2019)
- **Eq-DTLZ** and **Eq-IDTLZ**: Benchmark for equality constrained multiobjective optimization (Cuate et al. 2020)
- **CLSMOP**: Constrained large-scale mutliobjective optimization problems (He et al. 2021)

Real-World Test Problems Based on Mathematical Models

Real-world constrained multiobjective optimization problems (RCMs) from CEC 2021 Special Session and Competition and GECCO 2021 Competition[1]

A collection of real-world test problems based on mathematical models:

- Mechanical design problems
- Chemical engineering optimization problems
- Process synthesis optimization problems
- Power systems optimization problems

50 problems with 2–34 variables, 2–5 objectives, and 1–29 constraints

[1] https://www3.ntu.edu.sg/home/epnsugan/index_files/CEC2021/CEC2021-1.htm

Real-World Test Problems Based on Simulations (i)

Mazda benchmark problem[2]:

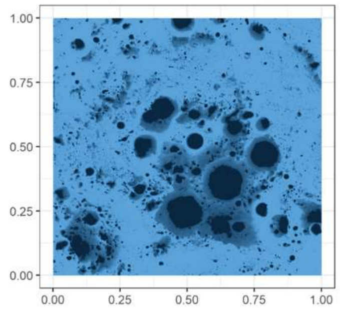
- Based on a real-world car structure design
- 222 design variables
- 2 objectives:
  - Minimization of the total weight of various cars
  - Maximization of the number of common gauge parts among various cars
- 54 constraints

[2] http://ladse.eng.isas.jaxa.jp/benchmark/

Real-World Test Problems Based on Simulations (ii)

Lunar lander landing site selection[3]:

- 2 design variables: coordinates x, y
- 3 objectives:
  - Total communication time
  - Continuous shade days
  - Landing point inclination angle
- 2 constraints
  - Max. continuous shade days
  - Max. landing point inclination angle



[3] http://www.jpnsec.org/files/competition2018/EC-Symposium-2018-Competition-English.html

Real-World Test Problems Based on Simulations (iii)

Wind turbine design problem[4]:

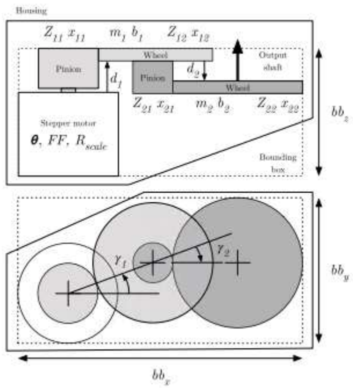
- Based on a real-world wind turbine design
- 32 design variables
- 5 objectives:
  - Annual power production
  - Average annual cost
  - Tower base load
  - Blade tip speed
  - Fatigue damage
- 22 constraints

[4] http://www.jpnsec.org/files/competition2019/EC-Symposium-2019-Competition-English.html

Real-World Test Problems Based on Simulations (iv) (Picard et al. 2021)

Multiobjective design of actuators (MODAct):

- Design of electro-mechanical actuators
- 20 CMOPs with 20 design variables
- 2–5 objectives:
  - Total cost (min.)
  - Torque excess (max.)
  - Harmonic mean of safety factors (max.)
  - Elec. to mech. energy conversion (max.)
  - Transmission ratio (min.)
- 7–10 constraints

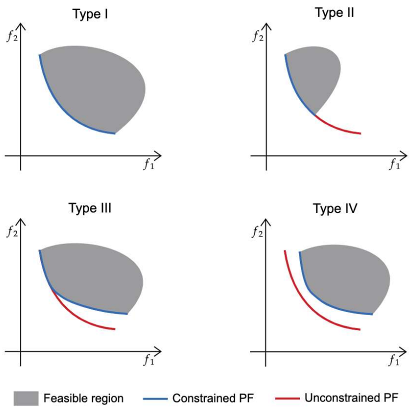


Problem Characterization

Overview

- Type of CMOPs
- Pareto front shapes
- Problem landscapes

Type of CMOPs (Z. Ma and Y. Wang 2019)

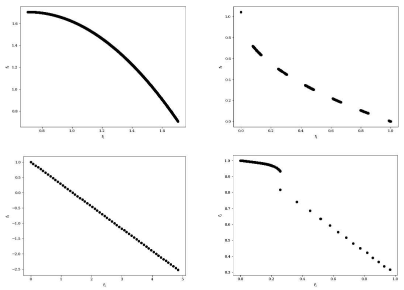


Test Suite Comparison: Type of CMOPs

Type	CTP	CF	C-DTLZ	NCTP	DC-DTLZ	DAS-CMOP	LIR-CMOP	MW
I	✓		✓	✓	✓	✓	✓	✓
II	✓	✓	✓		✓	✓	✓	✓
III		✓		✓		✓	✓	✓
IV	✓		✓	✓			✓	✓

Pareto Front Shapes

- Linear/Convex/Concave
- Connected/Disconnected/Discrete
- Mixed



Test Suite Comparison: Pareto Front Shapes

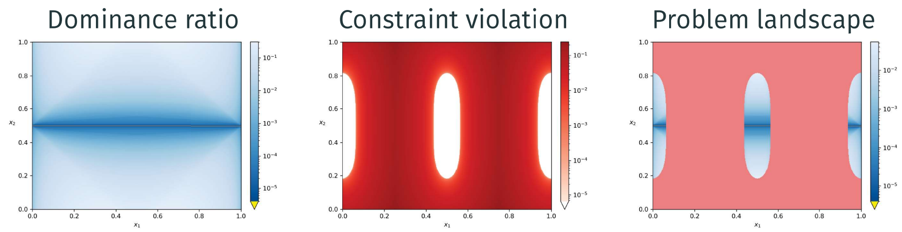
Type	CTP	CF	C-DTLZ	NCTP	DC-DTLZ	DAS-CMOP	LIR-CMOP	MW
Linear	✓	✓	✓	✓	✓	✓		✓
Convex	✓	✓	✓	✓		✓	✓	
Concave		✓	✓		✓	✓	✓	✓
Conn.	✓		✓	✓	✓	✓	✓	✓
Disconn.	✓	✓	✓		✓	✓	✓	✓
Discrete	✓	✓		✓		✓	✓	✓
Mixed	✓	✓		✓		✓	✓	✓

Problem Landscape

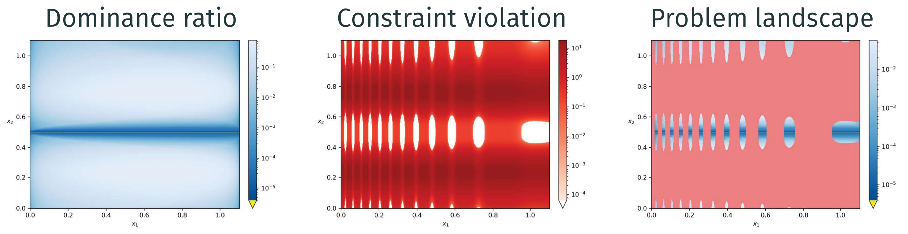
Constrained multiobjective problem landscape, $\mathcal{L}(S, f, v, d)$:

- $S \subseteq \mathbb{R}^n$ ... decision space
- $f : S \to \mathbb{R}^M$ ... objective vector function
- $v : S \to \mathbb{R}$ ... overall constraint violation function
- $d : S \times S \to \mathbb{R}$ ... distance metric

Example: C2-DTLZ2



Example: MW6



Exploratory Landscape Analysis: State of the Art (i)

Multiobjective optimization:

- Limited studies in the combinatorial (Verel et al. 2013; Daolio et al. 2017; Liefooghe, Daolio, et al. 2020) and continuous context (Liefooghe, Verel, et al. 2021)
- Initial attempts to **visualize bi-objective continuous problems** (Fonseca 1995; Kerschke, H. Wang, et al. 2016; Kerschke and Grimme 2017; Schäpermeier et al. 2021)

Constrained single-objective optimization:

- Preliminary study on the **characterization of constrained single-objective optimization problems** (Malan, Oberholzer, et al. 2015)
- Incorporation of these characteristics to **guide the constraint handling** (Malan 2018; Malan and Moser 2019)

Exploratory Landscape Analysis: State of the Art (ii)

Constrained multiobjective optimization:

- Three preliminary studies on exploratory landscape analysis exist in the literature (Picard et al. 2021; Alsouly et al. 2022; Vodopija et al. 2022)

Exploratory Landscape Analysis (i) (Picard et al. 2021)

Goals:

- Analyze the **effect of constraints** on search and objective spaces
- Measure the **feasibility ratio**
- Quantify the **relationship between objectives and constraints**
- Measure the **disjointedness of feasible regions**

Methods:

- Uniform sampling
- Random walk

Exploratory Landscape Analysis (ii) (Alsouly et al. 2022)

Goals:

- Explore the **relationship between MOEA performance and CMOP characteristics**
- Analyze the ***y*-distribution of problem landscapes**
- Analyze the **interaction between constraints and objectives**
- Measure the **ruggedness** of problem landscapes
- Explore the **connectedness of Pareto fronts and sets**

Methods:

- Uniform sampling
- Random walk

Exploratory Landscape Analysis (iii) (Vodopija et al. 2022)

Goals:

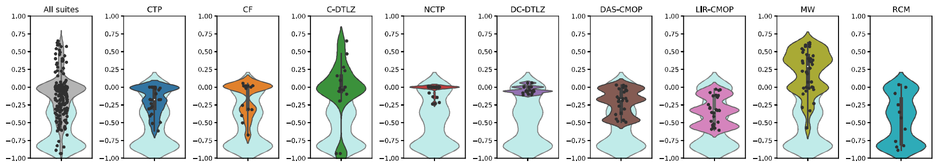
- Assess the existing test suites of CMOPs
- Measure **correlations between objectives and constraints**
- Identify feasible **subregions and basins**
- Characterize the **local structure** of violation landscapes

Methods:

- Space-filling sampling
- Random and adaptive walk
- Information content

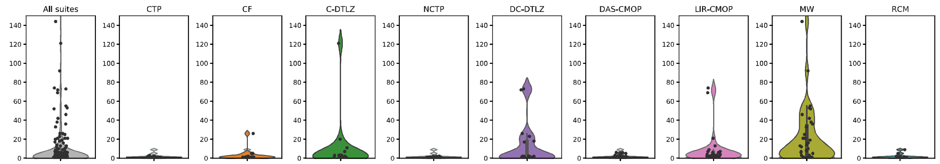
Test Suite Comparison: Correlations

Correlations between objectives and constraints



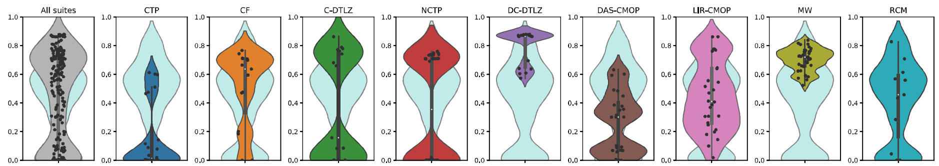
Test Suite Comparison: Feasible Subregions

Number of feasible subregions



Test Suite Comparison: Ruggedness

Maximum information content



A Note on Present Test Suites

- There are too **many Type I and Type II** CMOPs in the existing suites (Tanabe et al. 2017)
- Pareto front shapes of the artificial test problems are **unrealistically regular** (Ishibuchi et al. 2019)
- The existing artificial test problems **fail to satisfactorily represent some real-world problem characteristics**

Performance Assessment

Performance Indicators

Any popular performance indicator for multiobjective optimization can be adapted for CMOPs by removing infeasible solutions

The most frequently used indicators in the literature are:

- Hypervolume (HV)
- Generational distance (GD and GD^+)
- Inverted generational distance (IGD and IGD^+)
- Epsilon indicator (EPS)

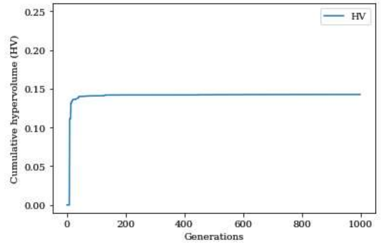
It is very important to use **archives** or **cumulative indicators**

Constraint-Related Measures

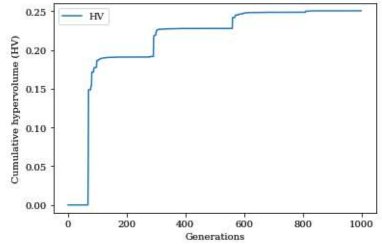
- Minimum of overall constraint violations
- Mean of overall constraint violations
- Feasibility ratio

Example: Hypervolume

MOEA/D-CDP

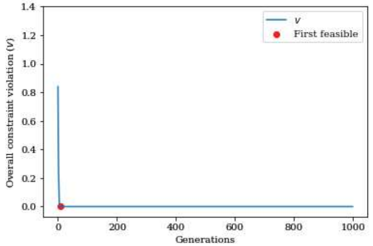


MOEA/D-IEpsilon

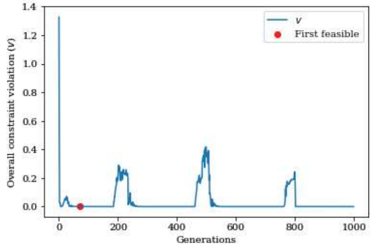


Example: Overall Constraint Violation

MOEA/D-CDP

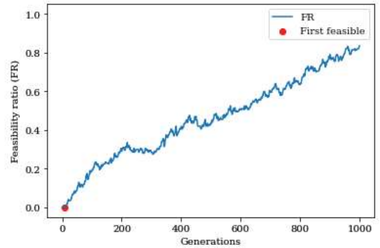


MOEA/D-IEpsilon

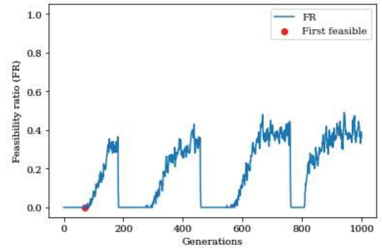


Example: Feasibility Ratio

MOEA/D-CDP



MOEA/D-IEpsilon



Software for Constrained Multiobjective Optimization

Overview

- Python
- R
- Matlab
- Java

Python (i)

pymoo: Multi-objective Optimization in Python[5]

- Algorithm implementations: NSGA-II (CDP), NSGA-III (CDP), C-TAEA
- CMOP implementations: TNK, OSY, BNH, CTP, DAS-CMOP, MW, MODAct, 3 RCMs
- Performance assessment: HV, GD, GD^+, IGD, IGD^+
- Additional: Solution repair when constraints are analytically expressed and visualization techniques

[5] https://pypi.org/project/pymoo/

Python (ii)

jMetalPy: Python Version of the JMetal Framework[6]

- Algorithm implementations: NSGA-II (CDP), NSGA-III (CDP), MOEAD-IEpsilon
- CMOP implementations: SRN, TNK, OSY, BNH, LIR-CMOP
- Performance assessment: HV, GD, IGD, EPS
- Additional: Statistical analysis and visualization techniques

[6] https://pypi.org/project/jmetalpy/

Python (iii)

deap: Distributed Evolutionary Algorithms in Python[7]

- Constraint handling by delta penalty approach, closest valid penalty approach, or island approach

pygmo: Parallel Optimization for Python[8]

- Advanced algorithms for hypervolume calculation

[7] https://pypi.org/project/deap/
[8] https://pypi.org/project/pygmo/

R

mco: Multiple Criteria Optimization Algorithms and Related Functions[9]

- Algorithm implementations: NSGA-II (CDP)
- CMOP implementations: BNH
- Performance assessment: HV, GD, EPS

MOEADr: Component-Wise MOEA/D Implementation[10]

- Constraint handling by penalty function approach, violation-based ranking
- Performance assessment: HV, IGD

[9] https://cran.r-project.org/web/packages/mco/index.html
[10] https://cran.r-project.org/web/packages/MOEADr/index.html

Matlab

PlatEMO: Evolutionary Multi-objective Optimization Platform[11]

- Algorithm implementations: NSGA-II (CDP), NSGA-III (CDP), ToP, PPS-MOEA/D, C-TAEA, MOEA/D-DAE, CCMO
- CMOP implementations: CF, C-DTLZ, DC-DTLZ, LIR-CMOP, MW, DOC
- Performance assessment: HV, GD, IGD
- Additional: Statistical analysis, visualization techniques and GUI

[11] https://github.com/BIMK/PlatEMO

Java

MOEAFramework: A Free and Open Source Java Framework for Multiobjective Optimization[12]

- Algorithm implementations: NSGA-II (CDP)
- CMOP implementations: SRN, TNK, OSY, BNH, CF, C-DTLZ
- Performance assessment: HV, GD, IGD
- Additional: Statistical analysis and visualization techniques

jMetal: A Framework for Multi-objective Optimization with Metaheuristics[13]

- Same functionalities as jMetalPy

[12] https://github.com/MOEAFramework/MOEAFramework
[13] https://github.com/jMetal/jMetal

Conclusions

Summary

- Increasing interest in **constrained multiobjective optimization**
- Many new **techniques, test suites, and software** proposed in the last years
- **Problem characterization** is now gaining interest

Open Issues and Future Research Directions

- Advances in **exploratory landscape analysis** for CMOPs
- Artificial **test suites** reflecting real-world problem characteristics
- Comprehensive **algorithm performance assessment** in solving CMOPs
- **Assessment of the recently proposed CHTs** on real-world problems

Acknowledgment

- We acknowledge financial support from the Slovenian Research Agency (Young researcher program, research core funding no. P2-0209 "Artificial Intelligence and Intelligent Systems", and research project no. N2-0254 "Constrained Multiobjective Optimization Based on Problem Landscape Analysis").
- We thank Dr. Tea Tušar from the Jožef Stefan Institute for providing several figures included in this tutorial.

References

- Alsouly, H., M. Kirley, and M. A. Muñoz (2022). "An instance space analysis of constrained multi-objective optimization problems". arXiv:2203.00868.
- Asafuddoula, M. et al. (2012). "An adaptive constraint handling approach embedded MOEA/D". In: *2012 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, pp. 1–8.
- Binh, T. T. and U. Korn (1997). "MOBES: A multi-objective evolution strategy for constrained optimization problems". In: *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 176–182.
- Cuate, O. et al. (2020). "A benchmark for equality constrained multi-objective optimization". In: *Swarm and Evolutionary Computation* 52, p. 100619.

- Daolio, F. et al. (2017). "Problem features versus algorithm performance on rugged multiobjective combinatorial fitness landscapes". In: *Evolutionary Computation* 25.4, pp. 555–585.
- Deb, K., A. Pratap, S. Agarwal, et al. (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197.
- Deb, K., A. Pratap, and T. Meyarivan (2001). "Constrained test problems for multi-objective evolutionary optimization". In: *Evolutionary Multi-Criterion Optimization (EMO 2001)*. Springer, pp. 284–298.
- Dong, J. et al. (2022). "A two-stage evolutionary algorithm based on three indicators for constrained multi-objective optimization". In: *Expert Systems with Applications* 195, p. 116499.

- Fan, Z., W. Li, X. Cai, H. Huang, et al. (2019). "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions". In: *Soft Computing* 23.23, pp. 12491–12510.
- Fan, Z., W. Li, X. Cai, H. Li, et al. (2019a). "Difficulty adjustable and scalable constrained multiobjective test problem toolkit". In: *Evolutionary Computation* 28.3, pp. 339–378.
- — (2019b). "Push and pull search for solving constrained multi-objective optimization problems". In: *Swarm and Evolutionary Computation* 44, pp. 665–679.
- Fonseca, C. M. (1995). "Multiobjective genetic algorithms with application to control engineering problems". PhD thesis. University of Sheffield.

- Geng, H. et al. (2006). "Infeasible elitists and stochastic ranking selection in constrained evolutionary multi-objective optimization". In: *Simulated Evolution and Learning (SEAL)*. Springer, pp. 336–344.
- He, C. et al. (2021). "Paired offspring generation for constrained large-scale multiobjective optimization". In: *IEEE Transactions on Evolutionary Computation* 25.3, pp. 448–462.
- Ishibuchi, H., L. He, and K. Shang (2019). "Regular Pareto front shape is not realistic". In: *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, pp. 2034–2041.
- Jain, H. and K. Deb (2014). "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach". In: *IEEE Transaction on Evolutionary Computation* 18.4, pp. 602–622.

- Jan, M. A. and R. A. Khanum (2013). “A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D”. In: *Applied Soft Computing* 13.1, pp. 128–148.
- Kerschke, P. and C. Grimme (2017). “An expedition to multimodal multi-objective optimization landscapes”. In: *Evolutionary Multi-Criterion Optimization (EMO 2017)*. Springer, pp. 329–343.
- Kerschke, P., H. Wang, et al. (2016). “Towards analyzing multimodality of continuous multiobjective landscapes”. In: *Parallel Problem Solving from Nature (PPSN XIV)*. Springer, pp. 962–972.
- Li, J. et al. (2016). “A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization”. In: *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, pp. 4175–4182.

- Li, K. et al. (2019). “Two-archive evolutionary algorithm for constrained multiobjective optimization”. In: *IEEE Transactions on Evolutionary Computation* 23.2, pp. 303–315.
- Liefooghe, A., F. Daolio, et al. (2020). “Landscape-aware performance prediction for evolutionary multiobjective optimization”. In: *IEEE Transactions on Evolutionary Computation* 24.6, pp. 1063–1077.
- Liefooghe, A., S. Verel, et al. (2021). “Landscape features and automated algorithm selection for multi-objective interpolated continuous optimisation problems”. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO ’21)*. ACM, pp. 421–429.
- Liu, Z. and Y. Wang (2019). “Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces”. In: *IEEE Transactions on Evolutionary Computation* 23.5, pp. 870–884.

- Ma, H. et al. (2021). “A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints”. In: *Information Sciences* 560, pp. 68–91.
- Ma, Z. and Y. Wang (2019). “Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons”. In: *IEEE Transaction on Evolutionary Computation* 23.6, pp. 972–986.
- Ma, Z., Y. Wang, and W. Song (2021). “A new fitness function with two rankings for evolutionary constrained multiobjective optimization”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51.8, pp. 5005–5016.
- Malan, K. M. (2018). “Landscape-aware constraint handling applied to differential evolution”. In: *Theory and Practice of Natural Computing*. Springer, pp. 176–187.

- Malan, K. M. and I. Moser (2019). “Constraint handling guided by landscape analysis in combinatorial and continuous search spaces”. In: *Evolutionary Computation* 27.2, pp. 267–289.
- Malan, K. M., J. F. Oberholzer, and A. P. Engelbrecht (2015). “Characterising constrained continuous optimisation problems”. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, pp. 1351–1358.
- Messac, A. (1996). “Physical programming: Effective optimization for computational design”. In: *AIAA Journal* 14.1, pp. 149–158.
- Ming, F. et al. (2022). “A tri-population based co-evolutionary framework for constrained multi-objective optimization problems”. In: *Swarm and Evolutionary Computation* 70, p. 101055.

- Ming, M. et al. (2021). “A dual-population-based evolutionary algorithm for constrained multiobjective optimization”. In: *IEEE Transactions on Evolutionary Computation* 25.4, pp. 739–753.
- Osyczka, A. and S. Kundu (1995). “A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm”. In: *Structural Optimization* 10.2, pp. 94–99.
- Picard, C. and J. Schiffmann (2021). “Realistic constrained multiobjective optimization benchmark problems from design”. In: *IEEE Transactions on Evolutionary Computation* 25.2, pp. 234–246.
- Schäpermeier, L., C. Grimme, and P. Kerschke (2021). “To boldly show what no one has seen before: A dashboard for visualizing multi-objective landscapes”. In: *Evolutionary Multi-Criterion Optimization (EMO 2021)*. Vol. 12654. Springer, pp. 632–644.

- Srinivas, K. and K. Deb (1995). “Multiobjective function optimization using nondominated sorting genetic algorithms”. In: *Evolutionary Computation* 2.3, pp. 221–248.
- Tanabe, R. and A. Oyama (2017). “A note on constrained multi-objective optimization benchmark problems”. In: *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, pp. 1127–1134.
- Tanaka, M. et al. (1995). “GA-based decision support system for multicriteria optimization”. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp. 1556–1561.
- Tian, Y., T. Zhang, et al. (2021). “A coevolutionary framework for constrained multiobjective optimization problems”. In: *IEEE Transactions on Evolutionary Computation* 25.1, pp. 102–116.

- Tian, Y., Y. Zhang, et al. (2021). “Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization”. In: *IEEE Transactions on Cybernetics*, pp. 1–14.
- Verel, S. et al. (2013). “On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives”. In: *European Journal of Operational Research* 227.2, pp. 331–342.
- Vodopija, A., T. Tušar, and B. Filipič (2022). “Characterization of constrained continuous multiobjective optimization problems: A feature space perspective”. In: *Information Sciences* 607, pp. 244–262.
- Wang, B. et al. (2022). “Evolutionary algorithm with dynamic population size for constrained multiobjective optimization”. In: *Swarm and Evolutionary Computation* 73, p. 101104.

- Woldesenbet, Y. G., G. G. Yen, and B. G. Tessema (2009). “Constraint handling in multiobjective evolutionary optimization”. In: *IEEE Transactions on Evolutionary Computation* 13.3, pp. 514–525.
- Yuan, J. et al. (2022). “Indicator-based evolutionary algorithm for solving constrained multiobjective optimization problems”. In: *IEEE Transactions on Evolutionary Computation* 26.2, pp. 379–391.
- Zhang, Q. et al. (2008). *Multiobjective optimization test instances for the CEC 2009 special session and competition*. Technical report CES-487. The School of Computer Science and Electronic Engieering, University of Essex, Colchester, UK.
- Zhu, Q., Q. Zhang, and Q. Lin (2020). “A constrained multiobjective evolutionary algorithm with detect-and-escape strategy”. In: *IEEE Transactions on Evolutionary Computation* 24.5, pp. 938–947.