Inflationary Differential Evolution for Constrained Multi-Objective Optimisation Problems

Gianluca Filippi[0000−0001−7393−905X] and
Massimiliano Vasile[0000−0001−8302−6465]

James Weir Building, 75 Montrose St, Glasgow G1 1XJ
g.filippi@strath.ac.uk, massimiliano.vasile@strath.ac.uk

Abstract. In this paper we review several parameter-based scalarisation approaches used within Multi-Objective Optimisation. We propose then a proof-of-concept for a new memetic algorithm designed to solve the Constrained Multi-Objective Optimisation Problem. The algorithm is finally tested on a benchmark with a series of difficulties.

Keywords: Constrained multi-objective optimisation · Scalarisation · Memetic algorithm

1 Introduction

Many real-word problems involve several competing objectives that have to be concurrently optimised. Most everyday decisions are based on intuition and common sense. However areas as engineering, physics, economics, etc. require more rigorous mathematical modelling and programming [25, 20, 14]. This paper deals with Multi-Objective Optimisation Problems (MOPs) and in particular with deterministic and continuous Constrained Multi-Objective Optimisation Problems (CMOPs) [22].

There are mainly three approaches for Multi-Objective Optimisation (MOO) [22, 13]. The *a posteriori* methods, based on the definition of a partial order, calculate a set of equally valuable solutions. The decision maker then, informed of this trade-off, chooses within the set. In the *a priori* methods the decision maker is required to specify additional preferences to define a total order between different options, for example by defining an utility function. The optimisation eventually finds a single minimal solution. The *interactive* methods finally require feedback and preferences from the user multiple time during the execution of the algorithm. We are here interested in the posterior approaches for which the whole set of possible solutions can be generated by two algorithmic methods: the direct multi-objective approach or the parameter-based scalarisation procedure. For the former, the interested reader can find useful information in [8, 6] while we dedicate this paper to the latter. By scalarisation we mean that the different objectives are aggregated and then a Single-Objective Optimisation Problem (SOP) is solved. By using different parameters of the aggregation function finally the MOP is translated to a number of SOPs and the set of optimal solutions is reconstructed [17].

We propose the use of Evolutionary Computation (EC) for the solution of the scalarisation problem. This methodology has indeed become popular showing excellent performance. Many dialects of EC have been developed and in this paper we present an advancement of the memetic algorithm Multi–Population Adaptive Inflationary Differential Evolution Algorithm (MP-AIDEA) [10] where Weighted Chebyshev Scalarisation (WCS) is combined with Pascoletti-Serafini Scalarissation (PSS) together with a novel constraint handling approach.

A review of the possible approaches to Constrained Optimsation Problem (COP) in general is in [11] and to penalty functions in particular is in [23]. We propose here an indirect approach with an adaptive exterior penalty function for hard constraint handling.

The assessment of the quality of a MOO algorithm is a delicate matter. Useful indications on how to categorise difficulties in MOPs have been described in [7]. A benchmark based on these information has been defined in [26] while the complexity introduced by a constrained search space has been included in [9]. Taking inspiration from [5] we finally extend the test cases in [26] introducing constraint functions that disconnect the objective space.

The paper is structured as follows. Section 2 presents an overview of basic concepts about MOP and MOO. In particular, Section 2.1 presents the criteria used to order different solutions, Section 2.2 defines the optimisation problem that is analysed in the following of the paper and Section 2.3 presents the normalisation procedure. Section 3 reviews the most common and promising approaches for parameter-based scalarisation. Section 4 describes our approach. Section 5 presents the benchmark and the algorithm tuning. Section 6 gives the results. Section 7 finally concludes.

2 Basic Concepts

We start by giving some basic definitions from MOO that will be used in the following.

2.1 Ordering Criteria

Consider the two generic non empty sets $\mathbb{K} \subset \mathbb{R}^s$ and $\mathbb{S} \subset \mathbb{R}^s$, with $s \in \mathbb{N}$.

Definition 1 (Cone)**.** The set $\mathbb{K}$ is called a cone if $\mathbf{k} \in \mathbb{K}, \lambda \geq 0 \implies \lambda\mathbf{k} \in \mathbb{K}$. Pointedness of $\mathbb{K}$ means that $\mathbb{K} \cap -\mathbb{K} = \{0_{\mathbb{R}}\}$. The set $\mathbb{S}$ is said to be bounded below with respect to the cone $\mathbb{K}$ if there exist $\mathbf{s} \in \mathbb{R}^s$ such that $\mathbb{S} \subset \mathbf{s} + \mathbb{K}$.

Definition 2 (Dominance)**.** A point $\mathbf{s} \in \mathbb{S}$ is said to be K-minimal for $\mathbb{S}$ if $(\mathbf{s} - \mathbb{K}) \cap \mathbb{S} = \{\mathbf{s}\}$. It is instead defined weakly K-minimal if $(\mathbf{s} - \text{int}(\mathbb{K})) \cap \mathbb{S} = \{\mathbf{s}\}$ where $\text{int}(\mathbb{K})$ is the interior of $\mathbb{K}$. It is finally defined properly K-minimal (in the sense of Benson [2]) if it is a minimal point for $\mathbb{S}$ and also $0_{\mathbb{R}}$ is a minimal point of cl(cone($\mathbb{S}+\mathbb{K}-\{\mathbf{s}\}$)) where cl($\mathbb{S}$) is the closure of $\mathbb{S}$. The set of all the (weakly) K-minimal points is called the (weakly) efficient set $(\varepsilon_w(\mathbb{S}))$ $\varepsilon(\mathbb{S})$.

In the case $\mathbb{K} = \mathbb{R}^m_+$ the K-minimal points are also called Edgeworth-Pareto (EP)-minimal points and the K-dominance become the more famous Pareto dominance.

2.2 Problem Statement

The scalarisation approaches presented in the paper are applied to the following CMOP:

$$\begin{aligned} &\text{minimise} && \mathbf{f}(\mathbf{x}) = [f_1, f_2, ..., f_m]^T \\ &\text{subject to} && c_i(\mathbf{x}) \leq 0, \quad i = 1, ..., n \\ & && \mathbf{x} \in \mathbb{X} \end{aligned} \tag{1}$$

with $\mathbb{X} \subset \mathbb{R}^n$ the parameter space, $m, n \in \mathbb{N}$, $m \geq 2$ and $\mathbb{Y} = \{\mathbf{f}(\mathbf{x})$ s.t. $\mathbf{x} \in \mathbb{X}$, $g_j(\mathbf{x}) \leq 0$, $j = 1, ..., n\}$ the feasible objective space. We require that f_i and c_j are locally $\mathcal{C}^2$.

In the following we will assume that the closed convex pointed cone $\mathbb{K}$ introduces an anti-symmetric partial order $\leq_K$ in the objective space $\mathbb{Y}$. The cone $\mathbb{K}$ is then used to define the efficient set in the objective space $\varepsilon(\mathbb{Y})$ and the corresponding efficient set in the parameter space $\varepsilon(\mathbb{X})$.

2.3 Normalisation

In case of prior knowledge about the reference points $\mathbf{z}^*$ (best) and $\mathbf{z}^{**}$ (worst), the objective functions $\mathbf{f}$ can be normalised in order to reduce the difference in the order of magnitude between the components f_i:

$$\bar{\mathbf{f}} = \frac{\mathbf{f} - \mathbf{z}^*}{\mathbf{z}^{**} - \mathbf{z}^*}. \tag{2}$$

$\mathbf{z}^*$ and $\mathbf{z}^{**}$ can be defined as reference solutions by the decision maker. However $\mathbf{z}^*$ usually corresponds to the ideal point $\mathbf{z}_{ideal}$ or to the utopian point $\mathbf{z}_{\text{utopian}}$ while $\mathbf{z}^{**}$ corresponds to the nadir point $\mathbf{z}_{nadir}$. A visualisation of $\mathbf{z}_{\text{ideal}}$, $\mathbf{z}_{\text{utopian}}$ and $\mathbf{z}_{\text{nadir}}$ is in Fig. 1 for a MOP with two objective functions f_1 and f_2. They are theoretic points that collapse the extreme behaviour of the different solutions in the Pareto front. $\mathbf{z}_{\text{ideal}}$ is the combination of the best solutions for the different objectives. $\mathbf{z}_{\text{nadir}}$ represents instead the worst possible combination of points. $\mathbf{z}_{\text{utopian}}$, is finally defined by means of an ϵ from $\mathbf{z}_{\text{ideal}}$. The points $\mathbf{z}_{\text{ideal}}$ and $\mathbf{z}_{\text{nadir}}$ will be used in the following of the paper while $\mathbf{z}_{\text{utopian}}$ has been here introduced for the sake of completeness.

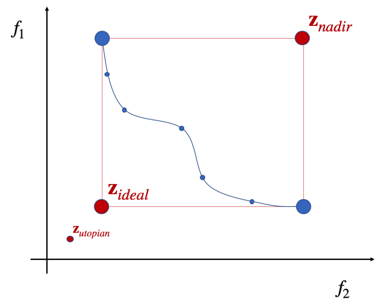


Fig. 1: Representation of utopian, ideal and nadir points for a generic bi-objective optimisation problem.

3 Review of Scalarisation Strategies

This section reviews the most important parameter-based scalarisation approaches: Epsilon-Constraint Scalarisation (ECS), Weighted-Sum Scalarisation (WSS), Benson Scalarisation (BS), WCS and PSS.

We consider a generic preference vector $\boldsymbol{\omega} = [\omega_1, ..., \omega_m]^T$ for the objective functions $\mathbf{f} = [f_1, f_2, ..., f_m]^T$ and a generic reference point $a = \mathbf{z}^*$. $\boldsymbol{\omega}$ and a can be either defined a priory by the decision maker or (as stated in Section 1 and used in Section 4) made varying in order to reconstruct the entire efficient set.

The scalarisation methods are compared in Table 1 as in [19] where the following criteria have been considered: the possibility to use different ordering cones, the necessity or not of boundedness and convexity conditions, the provability for obtaining properly efficient solutions, the use of reference and preference information and the introduction by the method of additional constraint functions.

Table 1: Characteristics of six scalarisation methods

Method	WSS	ECS	BS	WCS	PSS	CS
Ordering cone	any	$\mathbb{R}^m_+$	$\mathbb{R}^m_+$	$\mathbb{R}^m_+$	any	any
Boundedness from below	$-$	$-$	$-$	$+$	$-$	$-$
Convexity	$+$	$-$	$-$	$-$	$-$	$-$
Proof of properly efficient solutions	$+$	$-$	$-$	$-$	$-$	$+$
Preference weights	$+$	$-$	$-$	$+$	$-$	$+$
Reference points	$-$	$-$	$-$	$-$	$+$	$+$
Additional constraints or variables	$-$	$+$	$+$	$+$	$+$	$-$

3.1 Epsilon Constraint Scalarisation

The ECS was introduced by Haimes et al. in 1971 [18]. In this approach, one of the functions in $\mathbf{f}$ in Eq. (1) is maintained as the objective while the remaining functions are treated as inequality constraints

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X}} f_i \\ &s.t. \quad f_k \leq \epsilon_k \quad k \in \{1, ..., m\} \setminus \{i\} \\ &\qquad\quad c_j \leq 0 \quad \forall j \in \{1, ..., n\} \end{aligned} \tag{3}$$

The boundedness from below for the ECS is not an essential condition. However, the set of thresholds ϵ_k has to be decided carefully by the decision maker. A wrong selection, indeed, could bring to a not finite optimal solution or to an infeasible solution. The ECS can be applied only in the case when the ordering cone equals $\mathbb{R}^m_+$. The method does not require convexity condition on the problem under consideration. It generates weakly efficient solutions and does not provide conditions for generating properly efficient solutions. Decision maker's preferences, namely weights of objectives and reference points, are not taken into account. Finally, the problem size increases due to adding the constraints.

3.2 Weighted-Sum Scalarisation

The WSS was suggested by Gass and Saaty [16] in 1955 and it is probably the most commonly used scalarization technique for MOP. Here the Eq. (1) translates to:

$$\min_{\mathbf{x}\in\mathbb{X}} \sum_{i=1}^{n} \omega_i f_i \tag{4}$$

As for the ECS the boundedness below is not required but in that case the weights $\boldsymbol{\omega}$ have to be chosen carefully. Weakly and properly efficient solutions are guaranteed under the convexity condition. Weights of objectives are used but reference points are not considered. The method does not introduce additional constraints.

3.3 Benson's Scalarisation

The method was introduced in [1]. Here an initial guess $\mathbf{x}_0$ is given by the decision maker. The sum of the deviations l_i is maximised to find a new dominating point:

$$\begin{aligned} &\max_{\mathbf{x}\in X} \textstyle\sum_{i=1}^{n} l_i \\ &s.t. \quad f_i(\mathbf{x}_0) - l_i - f_i(\mathbf{x}) = 0 \quad i = 1, ..., m \\ &\qquad\quad l \geq 0 \\ &\qquad\quad c_j \leq 0 \quad \forall j = 1, 2, ..., n \end{aligned} \tag{5}$$

The BS requires the ordering cone $\mathbb{K}$ to equal $\mathbb{R}^m_+$. The boundedness below is not a requirement, however if the condition is not satisfied, more attention has to be put on the selection of $\mathbf{x}_0$. There is no necessity for the problem to be convex. BS provides necessary and sufficient conditions to converge to efficient solutions, but not to properly efficient solutions. Preferences from the decision maker are not taken into account. Finally, besides functions c_j, additional constraints are considered.

3.4 Weighted Chebyshev Scalarisation

The idea of the WCS is first presented in [3]. The Eq. (1) translates to:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X}} \|\mathbf{f}-\mathbf{z}_{\text{ideal}}\|_\infty^\omega \\ &s.t. \quad c_j \leq 0 \qquad \forall j=1,2,...,n \end{aligned} \tag{6}$$

where $\|\mathbf{f}-\mathbf{z}_{ideal}\|_\infty^\omega$ is the weighted Chebyshev distance $\max_i\{\omega_i(f_i-z_{ideal,i})\}$ between $\mathbf{f}(\mathbf{x}) \in \mathbb{Y}$ and the ideal point $\mathbf{z}_{\text{ideal}}$.

The linearisation is often considered:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X}, t\in\mathbb{R}} t \\ &s.t. \qquad \omega_i(f_i - z_{\text{ideal},i}) \leq t, \ \forall i=1,2,...,m \\ &\qquad\qquad c_j \leq 0, \qquad\qquad \forall j=1,2,...,n \end{aligned} \tag{7}$$

The WCS requires the cone $\mathbb{K}$ to be $\mathbb{R}_+^m$. The bondedness below is a necessary condition for the existence of $\mathbf{z}_{\text{ideal}}$. Instead the convexity assumption is not needed. The method assures generation of weakly efficient solutions and efficient solutions. However it is not guaranteed to generate properly efficient solutions. The preference vector $\boldsymbol{\omega}$ over the objective space is considered. The ideal point could be considered as a special case for the reference point. However the solutions are not guaranteed to be close to the reference point. In the linearised version, the size of the problem is increased by new constraints.

3.5 Pascoletti-Serafini Scalarisation

A first description of the PSS is given by Gerstewitz in [4]. As stated in [12], the PSS is a generalisation of ECS, WSS and WCS and it can be represented as:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X}} t \\ &s.t. \quad a + tr - f(x) \in \mathbb{K} \\ &\qquad c_j \leq 0, \qquad\qquad \forall j=1,2,...,n \end{aligned} \tag{8}$$

Eq. (8) can be interpreted as the process where the ordering cone $\mathbb{K}$ is moved in the direction $-r$ along the line $a+tr$ minimising the intersection $(a+tr-\mathbb{K})\cap f(\mathbb{X})$ until it becomes the empty set.

An arbitrary ordering cone can be adopted. The boundedness below and the convexity are not required conditions. The method guarantees to get at least weakly efficient solutions but it does not provide conditions to generate properly efficient solutions. It does use reference points but not preference vectors. Finally it uses additional functional constraints.

3.6 Conic Scalarisation

The Conic Scalarisation (CS) method was first introduced by Gasimov in [15] where beside the preference weighted vector $\boldsymbol{\omega}$ and the reference point $\mathbf{a}$, the augmentation parameter $\boldsymbol{\alpha}$ is considered:

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{X}} \sum_i \omega_i(f_i - a_i) + \alpha \sum_i |\omega_i(f_i - a_i)| & \\ s.t. \quad c_j \leq 0 & \forall j = 1, 2, ..., n \end{array} \tag{9}$$

As stated in [19], CS is a generalisation of WSS, BS and PSS. An arbitrary ordering cone can be used. The boundedness below is not an essential condition. No convexity is required. There are also conditions that guarantee to generate properly efficient minimal points. Preference and reference information is used. Finally, no additional constraints are required.

4 Memetic Strategy for the Constrained Scalarisation

We make the reasonable assumptions that the MOP is bounded below, which is usually satisfied for engineering problems, that no other cone than $\mathbb{R}^m_+$ is necessary and that we are interested in efficient solutions and not necessary in proper efficient solutions. For these reasons we have implemented in the memetic optimiser MP-AIDEA [10] a combination of WCS and PSS in order to solve CMOPs with the scalarisation approach.

4.1 Scalarisation Approach

We briefly describe here the extension of MP-AIDEA [10] highlighting the differences that have been introduced. The general structure of the algorithm is summarised in Algorithm 1. A set of N_{pop} different populations with n_{pop} elements each are first initialised: either a first guess is used or they are defined randomly. The optimisation process then hybridises the Differential Evolution (DE) step (line 3) where the N_{pop} populations are evolved and the local search (line 6) where their best candidate solutions are refined. The number of local refinements is adapted within MP-AIDEA allowing them to be run only if the converged solution in the DE is outside the basins of attraction of the previous recorded local minima which depend on the distances between previous best solutions of the DE and best solutions of the local search. More information about this point can be found in [10]. The DE is then locally and globally restarted (line 10) until the maximum number of evaluations $n_{\text{feval,max}}$ (considering both DE and local search) of the objective function is achieved (termination condition in line 2). In particular, the number of local restarts n_{LR} for each population and the corresponding dimension of the bubble δ_{local} where the starting vector is initialised are both auto-adapted within the algorithm. The radius of the bubble for the global restart δ_{global} and the convergence threshold ρ of DE are instead defined by the user.

More details about the DE step are in Algorithm 2. The building blocks that make any Evolutionary Algorithm (EA) are: initialisation (line 2), variation (line 6), evaluation (lines 3 and 7), selection (line 8) and termination (line 4). In particular, the population at the first generation (G=1) is defined from Algorithm 1. Within the main loop (lines 4-10) all the agents at the current generation G are selected as parents and are subjected to the variation step for the definition of generation G+1. The two schemes DE/Rand/1/bin and DE/CurrentToBest/2/bin [24] have been implemented for the parent's variation. The best for each agent between the corresponding parent at generation G and offspring at generation G+1 is finally selected. The differences here introduced in the DE to solve the scalarisation problem affect only the evaluation of the fitness function of the candidate solutions. To translate the MOP presented in Eq. (1) to a single objective problem we propose, within the DE, to apply the WCS described in Section 3.4:

$$\begin{array}{lll} \min\limits_{\mathbf{x}\in\mathbb{X}} \max_i\{\omega_i(f_i - z_{\text{ideal},i})\} & \forall i = 1,2,...,m \\ s.t. \quad c_j \leq 0 & \forall j = 1,2,...,n \end{array} \tag{10}$$

when the problem is not normalised, and

$$\begin{array}{lll} \min\limits_{\mathbf{x}\in\mathbb{X}} \max_i\{\omega_i \bar{f}_i\} & \forall i = 1,2,...,m \\ s.t. \quad c_j \leq 0 & \forall j = 1,2,...,n \end{array} \tag{11}$$

when it is normalised. During the local search, instead, the PSS described in Eq. (8) is implemented because a differentiable fitness function is required. The following constrained minimisation problem is then considered

$$\begin{array}{lll} \min\limits_{\mathbf{x}\in\mathbb{X}, t\in\mathbb{R}} t & & \\ s.t. & \omega_i(f_i - z_i) \leq t, & \forall i = 1,2,...,m \\ & c_j \leq 0, & \forall j = 1,2,...,n \end{array} \tag{12}$$

when the problem is not normalised and

$$\begin{array}{lll} \min\limits_{\mathbf{x}\in\mathbb{X}, t\in\mathbb{R}} t & & \\ s.t. & \omega_i(\bar{f}_i - \bar{z}_i) \leq t, & \forall i = 1,2,...,m \\ & c_j \leq 0, & \forall j = 1,2,...,n \end{array} \tag{13}$$

when it is normalised. In Eqs. (12) and (13) z_i ($\bar{z}_i$) is the best candidate solution f_i ($\bar{f}_i$) obtained in the previous DE. As stated in [12], Eqs. (12) and (13) could be considered as a reformulation (a linearisation) of the WCS where an additional variable is introduced and where the direction $r_i = 1/\omega_i$. However we consider here a different reference than the ideal point $\mathbf{z}_{\text{ideal}}$.

4.2 Constraint Handling

Within the DE step and with reference to [11, 23] we propose the following indirect approach with an adaptive exterior penalty function for hard constraint

Algorithm 1 MP-AIDEA for the scalarisation problem

Initialisation
while $n_{\text{feval}} < n_{\text{feval,max}}$ **do**
  Run the DE step (Algorithm 2)
  for $p \in [1, 2, ..., N_{\text{pop}}]$ **do**
    if $\mathbf{x}_{p,\text{best}}$ not in the basin of attraction of previous solutions **then**
      Run local search (Eq. (12)) with $\mathbf{x}_{0,p} = \mathbf{x}_{\text{best},p}$, $t_{0,p} = 0$ and the reference vector $\mathbf{z}_p$: $\min_{\mathbf{x}\in\mathbb{X}, t\in\mathbb{R}} t$ s.t. $\omega_i(f_i(\mathbf{x}) - z_i) \leq t \wedge c_j(\mathbf{x}) \leq 0$, $\forall i = 1, 2, ..., m$, $\forall j = 1, 2, ..., n$
      update $\mathbf{x}_{p,\text{best}}$ from the local search.
    end if
  end for
  Initialise populations for local or global restart in the next DE step [10].
end while

Algorithm 2 DE step

for $p \in [1, 2, ..., N_{\text{pop}}]$ **do**
  Initialise (input) the genotype $\mathbf{x}_{p,q}^{(G)}$ for the p-population at generation $G = 1$ where $q = 1, 2, ..., n_{\text{pop}}$
  Evaluate the phenotype of each candidate solution: $f_{s,p,q}^{(G)}$ (Algorithm 3)
  while the population is not contracted **do**
    Select parents: all generation G;
    Variate the parent's genotype: two strategies randomly alternated (DE/Rand/1/bin, DE/CurrentToBest/2/bin) define generation G+1;
    Evaluate new candidates $f_{s,p,q}^{(G+1)}$ (Algorithm 3):
    Select between parents and children with a greedy criterion
    update generation: G = G+1.
  end while
  $\mathbf{x}_{p,\text{best}} = \arg\min_i f_{s,p,q}^{(\text{end})}(\mathbf{x}_{p,q})$;
  $\mathbf{z}_p = \{f_1(\mathbf{x}_{p,\text{best}}), f_2(\mathbf{x}_{p,\text{best}}), ..., f_m(\mathbf{x}_{p,\text{best}})\}$.
end for

Algorithm 3 DE, Evaluation

for each q-agent in the population, with $q \in [1, 2, ..., n_{\text{pop}}]$ **do**
  $f_{s,p,q} = \max_i\{\omega_i(f_i(\mathbf{x}_{p,q}) - z_{\text{ideal},i})\}$, $i \in [1, 2, ..., m]$
  $c_{p,q} = \max_j\{c_j(\mathbf{x}_{p,q})\}$, $j \in [1, ..., n]$
end for
for each q-agent with $k \in [1, 2, ..., n_{\text{pop}}]$ **do**
  if $c_{p,q} > 0$ **then**
    $f_{s,p,q} = \max_i\{f_{s,p,q}\} + c_{p,q}$
  end if
end for

handling where hardness refers to the absolute satisfaction of the constraint. By 'indirect approach' we mean that the COP is translated to a Free Optimisation

Problem (FOP): this type of constraint handling is done before the EA run. The following mapping is used:

$$f_s(\mathbf{x}_{p,q}) = \begin{cases} f_s(\mathbf{x}_{p,q}) & \text{if } \max_i c_i(\mathbf{x}_p) \le 0 \\ \max_q\{f_s(\mathbf{x}_{p,q})\} + \max_j\{c_j(\mathbf{x}_{p,q})\} & \text{else} \end{cases} \tag{14}$$

where, for a generic population, f_s is the scalarised value of $\mathbf{f}$ for the given agent q in the population p, $\max_q\{f_s(\mathbf{x}_{p,q})\}$ is the maximum of f_s over the current population and $\max_j\{c_j(\mathbf{x}_{p,q})\}$ is the maximum constraint violation for the considered element q. Algorithm 3 summarises the fitness evaluation within the DE step including also the constraint handling.

For the local search (line 6 of Algorithm 1) instead the constraints in Eqs. (12) and (13) are directly handled within the nonlinear programming solver *fmincon* [21].

5 Testing Procedure

5.1 Benchmark

The test functions used in this paper have been selected from [26] where a benchmark for unconstrained MOPs is defined. A set of constraints $\mathbf{c}$ inspired by [5] has been further introduced to increase the complexity by disconnecting the feasible set $\mathbb{Y}$. A similar benchmark generation can be found in [9].

The general structure of each bi-objective optimisation problem is:

$$\begin{aligned} \text{minimise} \; & \mathcal{T} = [f_1, f_2]^T \\ \text{where} \; & f_1 = f_1(x_1) \\ & f_2 = g(x_2, ..., x_m)h(f_1(x_1), g(x_2, ..., x_m)) \\ s.t. \; & c_i \le 0, \;\; i = 1, ..., n \\ & \mathbf{x} \in \Omega \end{aligned} \tag{15}$$

where for all the test cases it is considered that $i \in \{1, 2\}$ and the constraint functions are:

$$\begin{aligned} c_1 &: 1.69x_1^2 + 1.01(gh)^2 - 2.6x_1(gh) - 0.02 \ge 0 \\ c_2 &: (x_1 - 0.5)^2 + (gh - 0.5)^2 - 0.5 \le 0. \end{aligned} \tag{16}$$

The objective functions $\mathcal{T}_{1,2,3}$ are presented in the following.

Test case 1 $\mathcal{T}_1$ has a convex Pareto front

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, ..., x_m) &= 1 + 9/(m-1)\textstyle\sum_{i=2}^m x_i \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned} \tag{17}$$

where $m = 30$ and $x_i \in [0, 1]$. The Pareto optimal front is at $g(\mathbf{x}) = 1$.

Test case 2 $\mathcal{T}_2$ is the non-convex counterpart of $\mathcal{T}_1$

$$\begin{array}{l} f_1(x_1) = x_1 \\ g(x_2, ..., x_m) = 1 + 9/(m-1)\sum_{i=2}^{m} x_i \\ h(f_1, g) = 1 - (f_1/g)^2 \end{array} \quad (18)$$

where $m = 30$ and $x_i \in [0, 1]$. The Pareto optimal front is at $g(\mathbf{x}) = 1$.

Test case 3 $\mathcal{T}_3$ presents the discreteness: the Pareto front is divided in several non continuous convex parts:

$$\begin{array}{l} f_1(x_1) = x_1 \\ g(x_2, ..., x_m) = 1 + 9/(m-1)\sum_{i=2}^{m} x_i \\ h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1) \end{array} \quad (19)$$

where $m = 10$ and $x_i \in [0, 1]$. The Pareto optimal front is at $g(\mathbf{x}) = 1$.

5.2 Tuning

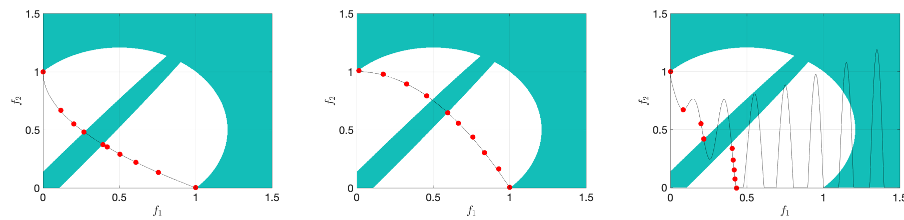
This section presents the tuning procedure applied to the modified version of MP-AIDEA presented in Section 4.1 and its results. The maximum number of function evaluations for each test problem has been fixed to $n_{\text{feval,max}} = 5e4$. The combination of the following parameters instead have been tuned: the number of populations $N_{\text{pop}} \in \{2, 4\}$, the number of agents in each populations $n_{\text{pop}} \in \{30, 45\}$, the dimension of the bubble for the global restart $\delta_{\text{global}} \in \{0.15, 0.25\}$ and the convergence threshold $\rho \in \{0.05, 0.15\}$ for the DE step. The efficient set of each problem has been uniformly discretised using 10 trigonometric couple of weights $w_{f1} = \frac{\cos\theta}{\cos\theta+\sin\theta}$ and $w_{f2} = \frac{\sin\theta}{\cos\theta+\sin\theta}$ with $\theta \in [0, \frac{\pi}{2}]$. Each combination of parameter setting and weights have been repeated 10 times. The results have been compared with the analytical Pareto front of $\mathcal{T}_{1,...,3}$ and finally the setting with the minimum average error has been selected. The tuning's results are presented in Table 2.

Table 2: MP-AIDEA tuning results

$\mathcal{T}$	N_{pop}	n_{pop}	δ_{global}	ρ
1	2	45	0.25	0.05
2	2	45	0.25	0.05
3	4	45	0.25	0.05

6 Results

The efficient sets for $\mathcal{T}_{1,2,3}$ are finally plotted in Fig. 2 for 10 equally spaced preferences weights $w_{f1} = \frac{\cos\theta}{\cos\theta+\sin\theta}$ and $w_{f2} = \frac{\sin\theta}{\cos\theta+\sin\theta}$ where $\theta \in [0, \frac{\pi}{2}]$. As it can be seen in the figures, the approach proposed in Section 4.1 is capable to find the efficient set for the MOP satisfying the constraint functions. In particular it has been noted that the implementation of the multi-population restart within MP-AIDEA is of fundamental importance for such problems, as this benchmark, that have a disconnected objective space.



(a) (b) (c)

Fig. 2: Efficient sets for $\mathcal{T}_1$ (a), $\mathcal{T}_2$ (b) and $\mathcal{T}_3$ (c). The shaded area represents the unfeasible domain. The black line is the sub-domain containing the global efficient set. Red points are the solution of the proposed method.

Acknowledgement

The work in this paper was supported by the H2020-MSCA-ITN-2016 UTOPIAE, grant agreement 722734.

7 Conclusions

In this paper we have presented a review of the most important scalarisation methods for MOPs highlighting the corresponding advantages and disadvantages. We have proposed then a new memetic approach for the solution of CMOPs. A combination of WCS and PSS has been implemented in the memetic optimiser MP-AIDEA in order to translate the MOP to a corresponding set of SOPs. A novel adaptive exterior penalty function has been used for the constraint handling. The approach has been tested demonstrating its capability of finding efficient points. Future steps will regard further analysis of the performance of the proposed algorithm with both comparison between different scalarisation techniques and different optimisation solvers. Finally, besides the parameter-based scalarisation approach, also the direct multi-objective selection will be considered.

References

1. Benson, H.P.: Existence of efficient solutions for vector maximization problems. Journal of Optimization Theory and Applications **26**(4), 569–580 (1978). https://doi.org/10.1007/BF00933152
2. Benson, H.P.: An improved definition of proper efficiency for vector maximization with respect to cones. Journal of Mathematical Analysis and Applications **71**(1), 232–241 (sep 1979). https://doi.org/10.1016/0022-247X(79)90226-9
3. Bowman, V.J.: On the Relationship of the Tchebycheff Norm and the Efficient Frontier of Multiple-Criteria Objectives. In: Thiriez H., Z.S. (ed.) Multiple Criteria Decision Making. Lecture Notes in Economics and Mathematical Systems (Operations Research), vol. 130, pp. 76–86. Springer-Verlag, Berlin, Heidelberg (1976). https://doi.org/10.1007/978-3-642-87563-2_5
4. Boţ, R.I., Grad, S.M., Wanka, G.: A general approach for studying duality in multiobjective optimization. Mathematical Methods of Operations Research **65**(3), 417–444 (2007). https://doi.org/10.1007/s00186-006-0125-x
5. Burachik, R.S., Kaya, C.Y., Rizvi, M.M.: A new scalarization technique to approximate pareto fronts of problems with disconnected feasible sets. Journal of Optimization Theory and Applications **162**(2), 428–446 (2014). https://doi.org/10.1007/s10957-013-0346-0
6. Burke, E.K., Graham, K.: Search methodologies: Introductory tutorials in optimization and decision support techniques. Springer US (jan 2014). https://doi.org/10.1007/978-1-4614-6940-7
7. Deb, K.: Multi-objective genetic algorithms: problem difficulties and construction of test problems. Evolutionary Computation **7**(3), 205–230 (1998). https://doi.org/10.1162/evco.1999.7.3.205
8. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction. In: Springer, L. (ed.) Multi-objective Evolutionary Optimisation for Product Design and Manufacturing, pp. 1–24 (2011). https://doi.org/https://doi.org/10.1007/978-0-85729-652-8_1
9. Deb, K., Pratap, A., Meyarivan, T.: Constrained test problems for multi-objective evolutionary optimization. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **1993**, 284–298 (2001). https://doi.org/10.1007/3-540-44719-9_20
10. Di Carlo, M., Vasile, M., Minisci, E.: Adaptive multi-population inflationary differential evolution. Soft Computing **24**(5), 3861–3891 (2019). https://doi.org/10.1007/s00500-019-04154-5
11. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing (Natural Computing Series). Springer (2003). https://doi.org/10.1007/978-3-662-44974-8, www.springer.com/series/
12. Eichfelder, G.: Scalarizations for adaptively solving multi-objective optimization problems. Computational Optimization and Applications **44**(2), 249–273 (2009). https://doi.org/10.1007/s10589-007-9155-4
13. Emmerich, M.T., Deutz, A.H.: A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Natural Computing **17**(3), 585–609 (2018). https://doi.org/10.1007/s11047-018-9685-y, https://doi.org/10.1007/s11047-018-9685-y
14. Epstein, J.M.: Why model? Journal of Artificial Societies and Social Simulation **11**(4), 12 (2008), http://jasss.soc.surrey.ac.uk/11/4/12.html

15. Gasimov, R.N.: Characterization of the Benson proper efficiency and scalarization in nonconvex vector optimization. In: M. Köksalan (ed.) Multiple Criteria Decision Making in the New Millennium, pp. 189–198. Springer-Verlag, Berlin, Heidelberg (2001). https://doi.org/10.1007/978-3-642-56680-6_17
16. Gass, S., Saaty, T.: The computational algorithm for the parametric objective function. Naval Research Logistics Quarterly **2**(1-2), 39–45 (1955). https://doi.org/10.1002/nav.3800020106
17. Gunantara, N.: A review of multi-objective optimization: Methods and its applications. Cogent Engineering **5**(1), 1–16 (2018). https://doi.org/10.1080/23311916.2018.1502242
18. Haimes, Y.Y., Lasdon, L.S., Wismer, D.A.: On a bicriterion formation of the problems of integrated system identification and system optimization. IEEE Transactions on Systems, Man and Cybernetics **SMC-1**(3), 296–297 (1971). https://doi.org/10.1109/TSMC.1971.4308298
19. Kasimbeyli, R., Ozturk, Z.K., Kasimbeyli, N., Yalcin, G.D., Erdem, B.I.: Comparison of some scalarization methods in multiobjective optimization: Comparison of Scalarization Methods. Bulletin of the Malaysian Mathematical Sciences Society **42**(5), 1875–1905 (sep 2019). https://doi.org/10.1007/s40840-017-0579-4, https://doi.org/10.1007/s40840-017-0579-4
20. Levins, R.: Strategies of abstraction. Biology and Philosophy **21**(5), 741–755 (2006). https://doi.org/10.1007/s10539-006-9052-8
21. MATLAB: 9.7.0.1216025 (R2019b). The MathWorks Inc., Natick, Massachusetts (2018)
22. Miettinen, K.M.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers Group, Dordrecht (2012)
23. Smith, A.E., Coit, D.W.: Penalty functions. In: Baeck, T., Fogel, D., Michalewicz, Z. (eds.) Handbook of Evolutionary Computation, chap. 5.2. Oxford University Press, Institute of Physics Publishing (1995). https://doi.org/10.1887/0750308958/b386c48
24. Storn, R., Price, K.: Differential Evolution – A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization **11**(4), 341–359 (1997). https://doi.org/10.1023/A:1008202821328, http://link.springer.com/10.1023/A:1008202821328
25. Wymore, A.W.: Model-Based Systems Engineering. C. Press (1993)
26. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evolutionary Computation **8**(2), 173–195 (2000). https://doi.org/10.1162/106365600568202