COARSE-EMOA: An Indicator-based Evolutionary Algorithm for Solving Equality Constrained Multi-objective Optimization Problems

Jesús L. Llano García[a,*], Raúl Monroy[a], Víctor Adrián Sosa Hernández[a], Carlos A. Coello Coello[b]

[a]*Tecnologico de Monterrey, School of Engineering and Sciences, Av. Lago de Guadalupe Km 3.5, Atizapán de Zaragoza, Edo. Mexico 52926, MEXICO*
[b]*CINVESTAV-IPN (Evolutionary Computation Group), Department of Computer Science, Av. IPN No. 2508, Mexico City 07360, MEXICO*

Abstract

Many real-world applications involve dealing with several conflicting objectives which need to be optimized simultaneously. Moreover, these problems may require the consideration of limitations that restrict their decision variable space. Evolutionary Algorithms (EAs) are capable of tackling Multi-objective Optimization Problems (MOPs). However, these approaches struggle to accurately approximate a feasible solution when considering equality constraints as part of the problem due to the inability of EAs to find and keep solutions exactly at the constraint boundaries. Here, we present an indicator-based evolutionary multi-objective optimization algorithm (EMOA) for tackling Equality Constrained MOPs (ECMOPs). In our proposal, we adopt an artificially constructed reference set closely resembling the feasible Pareto front of an ECMOP to calculate the Inverted Generational Distance of a population, which is then used as a density estimator. An empirical study over a set of benchmark problems each of which contains at least one equality constraint was performed to test the capabilities of our proposed COnstrAined Reference SEt - EMOA (COARSE-EMOA). Our results are compared to those obtained by six other EMOAs. As will be shown, our proposed COARSE-EMOA can properly approximate a feasible solution by guiding the search through the use of an artificially constructed set that approximates the feasible Pareto front of a given problem.

Keywords: Multi-Objective Optimization, Performance Indicators, Constrained Optimization, Evolutionary Algorithms

*Corresponding author

Email addresses: `jesus_llg@tec.mx` (Jesús L. Llano García), `raulm@tec.mx` (Raúl Monroy), `vsosa@tec.mx` (Víctor Adrián Sosa Hernández), `ccoello@cs.cinvestav.mx` (Carlos A. Coello Coello)

1. Introduction

Many real-world problems require the optimization of multiple objective functions concurrently [1]. Amongst these problems, some of the most commonly found include Portfolio selection where the return of the portfolio needs to be maximized, keeping the risk at a minimum [2], environmental engineering where the costs of monitoring the purity of water need to be in balance with the accuracy of the estimates obtained by testing [3] and power dispatch problems, where the cost of producing energy needs to be reduced while also trying to maintain the environmental impact of using cheap fuels at a minimum [4].

Moreover, these objectives can potentially conflict with each other, requiring the consideration of a set of trade-offs among them. The conflict that exists among objective functions often results in the lack of a single optimal solution capable of minimizing (or maximizing) all of them. Instead, a set of possible solutions appears, each focusing more on optimizing a particular objective function [1].

Over the last two decades, Evolutionary Algorithms (EAs) have been studied due to their known efficiency and robustness when solving Multi-objective Optimization Problems (MOPs) [1, 5]. It is the simplicity of implementing these techniques, and their flexibility in many different scenarios which has made EAs quite popular in recent years [6]. Today, a wide variety of Evolutionary Multi-objective Optimization Algorithms (EMOAs) capable of solving highly complex MOPs are available.

Despite the wide availability of EMOAs currently available, when faced with problems that require searching inside a restricted domain, the situation is entirely different [7]. Although some constraint-handling techniques have been proposed for EMOAs [8], most of them focus only on inequality constraints and disregard Equality Constrained MOPs (ECMOPs). This is mainly due to EMOAs' inability to find and adequately search over the constraint boundaries which makes it difficult for an EMOA to guarantee the generation of enough feasible solutions [9].

As a way to cope with constrained MOPs, many methods have been proposed ranging from the transformation of constraints into additional objectives to the incorporation of local search techniques and heuristics within EMOAs [10, 11, 12, 13, 14, 15, 16, 17, 18]. Nevertheless, many result in the inclusion of additional conditions for the problem to be solvable, by computing internal optimization problems or the calculation of gradients to obtain feasible solutions [14, 19]. Moreover, in many situations, the existing methodologies and algorithms cannot achieve a full set of feasible solutions, some of them attaining no more than a 50% success rate [20].

Here, we propose the COnstrAined Reference SEt - EMOA (COARSE-EMOA) as an alternative to handle ECMOPs. Our approach consists of an indicator-based EMOA that artificially generates a reference set that tries to approximate the PF of an ECMOP at each generation, using it to calculate the individual contribution to the Inverted Generational Distance [21] (IGD) as a way to preserve those individuals closer to the feasible Pareto front, pushing the

2

population towards it. We extend a technique based on the approach presented by Zapotecas et al. [22] for approximating the Pareto front of any given problem while considering its equality constraints. We show that our proposed technique is capable of guiding the search towards the feasible Pareto front of an ECMOP in most cases. The contributions of this work are outlined as follows:

1. A technique for constructing an approximation of the PF of an equality constrained multi-objective problem is presented. This procedure allows the synthesis of a reference set that facilitates the calculation of the IGD from a population to the best-found approximation of the feasible PF.

2. The COARSE-EMOA is developed incorporating the previous technique as a selection procedure for individual survivability. This drives the population of the algorithm to move closer to the feasible region and to the PF of the problem during the search process.

3. A comparison of the COARSE-EMOA against six algorithms used for constrained optimization is performed on existing equality constrained MOPs. The results show that the COARSE-EMOA can find feasible solutions that reach the boundary of the constraints by exclusively focusing on improving the performance indicator of its population.

The remainder of this paper is organized as follows. Section 2 provides some basic concepts adopted within the area of multi-objective optimization. Previous works focused on the study and solution of ECMOPs are also presented in this section. Section 3 describes our proposal for synthesizing a reference set to approximate the PF of a problem to consider feasible solutions. Section 4 describes an indicator-based EMOA that uses our technique to tackle ECMOPs. Our experimental results and methodology are presented in Section 5 as well as a short discussion of our findings. Lastly, Section 6 presents our conclusions and some paths for future research.

2. Preliminaries and Background

For this work we focus on the resolution of problems of the following type:

$$\begin{aligned}
\min_{\mathbf{x} \in \mathbb{R}^n} \quad & F(\mathbf{x}) \\
\text{s.t.} \quad & h_j(\mathbf{x}) = 0 \; j = 1, \ldots, q
\end{aligned} \tag{1}$$

where $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ is a vector formed by k objective functions, $f_i : S \subseteq \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, k$ is the ith objective function, $\mathbf{x}$ is a vector of decision variables that exist in $\mathbb{R}^n$ and S represents the feasible search space given by:

$$S := \{\mathbf{x} : h_j(\mathbf{x}) = 0, \; j = 1, \ldots, q\} \tag{2}$$

The following definitions are relevant to understand some of the concepts that will be referred to throughout the rest of the document in regards to MOPs.

3

Definition 1. a) Let $\mathbf{x}, \mathbf{y} \in S$ be two decision vectors. Then $\mathbf{x}$ is said to *dominate* $\mathbf{y}$ (denoted by $\mathbf{x} \prec \mathbf{y}$), if and only if: $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{y})$ and $\mathbf{F}(\mathbf{x}) \neq \mathbf{F}(\mathbf{y})$ [1].

b) The same vector $\mathbf{x}$ is said to *weakly dominate* a vector $\mathbf{y}$ (denoted by $\mathbf{x} \preceq \mathbf{y}$) if: $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{y})$ [1].

c) A point $\mathbf{x}^* \in S$ is said to be *Pareto optimal* if there is no $\mathbf{x}$ that dominates it. The phrase *Pareto Optimal* is taken to mean with respect to the entire decision space unless otherwise specified [1].

d) For a given MOP, the *Pareto optimal set* of $\mathbf{F}(\mathbf{x})$ can be defined as $P^* := \{\mathbf{x} \in S \,|\, \neg \exists \, \mathbf{y} \in S \, F(\mathbf{y}) \prec F(\mathbf{x})\}$ [1]

Further, we recall the concept of active constraints, which is key in the treatment of ECMOPs.

Definition 2. (Active constraints [23]) Given a feasible point $\mathbf{x}$, an inequality constraint g_i is said to be active if $g(\mathbf{x}) = 0$ and inactive at $\mathbf{x}$ if $g(\mathbf{x}) < 0$. All equality constraints (h_j) are always active.

2.1. Performance Indicators

Obtaining an approximation to the Pareto front is not enough; we need to assess how good it is. Furthermore, if we want to compare the performance of several algorithms over the same problem, we require a standard performance indicator to do so. Many different performance indicators have been proposed for assessing the quality of an approximation of the Pareto front [24, 25, 26, 27, 28].

Here, we present formally those that will be used throughout this work. We will start with the Inverted Generational Distance, which is a key element of our proposed approach, followed by the Averaged Hausdorff distance and the Feasibility ratio, which are two performance indicators employed to evaluate the final approximations obtained by our algorithm and to compare our results to those obtained by six more contrained EMOAs.

2.1.1. Inverted Generational Distance

It is well known that the Generational Distance (GD) [24] presents sensitivity to the size of the approximation front and to the number of non-dominated solutions in it [21]. The Inverted Generational Distance (IGD) proposed by Coello and Cortés [21] intends to solve this issue while retaining similar properties as its counterpart. It measures the distance between a discretization (Z) of the true PF of a problem and the approximation set (A) obtained by an algorithm. This indicator tells us how far the approximation is to the real solution of a problem and it is defined as follows:

$$IGD(A, Z) = \frac{1}{|Z|} \left(\sum_{z \in Z} \min_{a \in A} d(z, a)^p \right)^{1/p} \tag{3}$$

where $d(z, a)$ is the Euclidean distance from z to the closest element $a \in A$.

2.1.2. The Averaged Hausdorff Distance

Presented by Schütze et al. in [25], the Averaged Hausdorff Distance (Δ_p) is an indicator for assessing the performance of EMOAs which is composed of the GD_p and the IGD_p, which are two slight modifications to the original GD and IGD indicators. Thus, Δ_p is defined as shown below:

Given an approximation A and a discretization of the PF, Z, then:

$$GD_p(A, Z) = \left(\frac{1}{|A|} \sum_{a \in A} \min_{z \in Z} d(a, z)^p \right)^{1/p} \tag{4}$$

is the averaged semi-distance between A and Z, and:

$$IGD_p(A, Z) = \left(\frac{1}{|Z|} \sum_{z \in Z} \min_{a \in A} d(z, a)^p \right)^{1/p} \tag{5}$$

is the averaged semi-distance between Z and A. Then Δ_p is defined as:

$$\Delta_p(A, Z) = \max\left(IGD_p(A, Z), GD_p(A, Z) \right) \tag{6}$$

2.1.3. Feasibility Ratio

Given that EMOAs rarely consider constraints, performance indicators also rarely consider them as part of their definition. The *Feasibility ratio* is an indicator that offers the possibility of determining how far an algorithm is from achieving a feasible state (where all individuals are feasible).

Introduced in [15], this indicator measures the percentage of feasible solutions in a population. This measure is trivial to compute and is defined as follows:

$$F_r(A) = \frac{A_F}{\mu}, \tag{7}$$

being A_F the number of feasible solutions in an approximation (A) and μ its size.

2.2. Related Work

In the literature, many approaches have been designed to tackle constrained MOPs [8, 29, 30, 31]. Examples of such approaches come from the incorporation of techniques like the adoption of modified definitions of Pareto optimality, employing methods to repair solutions and the hybridization of EMOAs. Given the difficulty of handling equality constraints, many authors still seek to develop alternatives for the resolution of ECMOPs.

2.2.1. Penalty Functions

Penalty functions are perhaps the most common approach for tackling constrained optimization problems using evolutionary algorithms. In this approach, the main aim is to incorporate the level of constraint violation as a way to punish the fitness value of all infeasible individuals [32]. This penalization results

in the search being guided towards the feasible region by the improved survival of individuals that are not only closer to optimality but also to feasibility.

Two types of penalties exist in the field of classical optimization: interior and exterior [32, 33]. For exterior penalties, we start with infeasible solutions and from there, the penalty helps the algorithm to move towards the feasible region. In the case of interior penalties, we need to start within the feasible region of the decision space and then look for the optimum. This is because the exterior penalty causes the constraint boundaries to act as barriers, stopping elements from ever leaving the feasible region [33].

2.2.2. Transforming Equality Constraints with Feasibility Thresholds

Many EMOAs even after the incorporation of penalty functions, present important problems when handling equality constraints thanks to their inability to search precisely for, and remain within, the boundary between the feasible and infeasible regions [34] of the active constraints. As an alternative to handle this and to allow the algorithm to search in the proximity of the boundary allowing a better implementation of some general constraint handling techniques, several authors have proposed a straightforward and useful method to approximate equality constraints [10, 35, 36, 37].

The approach works by determining a feasibility threshold around the feasibility boundary of any equality constraints close to zero. This way, for any practical matter, the constraint remains equal to zero within a small region surrounding the boundary. This transformation opens up the possibility of exploring the usage of other techniques for constraint handling not necessarily designed to work with equality constraints.

2.2.3. The Constrained-domination principle

Another technique for constraint handling in evolutionary algorithms comes from the redefinition of the Domination Principle that many EMOAs already use to select individuals. An example of this is the Constrained-Domination Principle (CDP), which is a technique for constraint handling proposed by Deb et al. [13] as part of the NSGA-II and further explored within the adaptive NSGA-III algorithm [38]. The CDP presents a slight modification to the definition of Pareto domination allowing for an algorithm to select individuals taking into consideration their feasibility and not only optimality.

Using this principle, feasible individuals have better possibilities of surviving selection than those individuals that are not feasible. This method is expected to lead the search in the direction of the feasible space as with each generation, more and more individuals should reach feasibility while also preserving non-dominance. Thanks to the straightforward implementation of this strategy, its incorporation in practically any Pareto-based EMOA is rather simplistic. However, the use of techniques like this one, even after the definition of a feasibility threshold, does not secure that enough feasible solutions will be found to properly approximate the PF and the Pareto set of an ECMOP, given the difficulty of finding new feasible solutions [34].

2.2.4. Ensemble of Constraint Handling Techniques

To effectively solve a constrained problem, an algorithm requires to be able to use the information present even in infeasible individuals, for which the appropriate constraint-handling technique and its corresponding parameters are required. Given that in most scenarios it is difficult to select a specific technique capable of effectively solving a particular problem, Mallipeddi and Suganthan proposed in [39] an Ensemble of Constraint Handling Techniques (ECHT).

In this approach, several constraint-handling methods are selected and each is given a population. Each population corresponding to a method produces its offspring and evaluates them. Then, the obtained individuals not only compete with members of all populations in the ensemble but also with the members of their own population. This way an offspring may be rejected by its population but could be accepted by others according to the characteristics of the other constraint handling methods. This allows the algorithm to retain a balance between diversity and feasibility of individuals. However, the performance of the ensemble highly depends on the proper determination of parameters for each of its internal techniques, which could present an issue.

2.2.5. Dynamic Problem transformation

A contemporary approach for solving constrained MOPs is to transform the problem by incorporating the constraint violation of individuals as one more of the objectives to optimize. Jiao et al. in [40] propose a method to transform a constrained many-objective optimization problem (CMaOP) into a dynamic CMaOP. This allows an evolutionary algorithm to optimize objectives and processing constraints simultaneously.

Their approach reformulates a k-objective constrained problem into an equivalent $k + 1$-objective constrained problem. Moreover, the technique relaxes the constraint boundary to make all individuals ϵ-feasible at the beginning of the run, and it is shrunk dynamically as the search goes on. As a last way to ensure the preference of ϵ-feasible individuals over those that are not ϵ-feasible a variation of the CDP is adopted by the authors to perform the individual selection. By incorporating constraint violation as an objective and relaxing the boundary, the algorithm throughout the evolutionary process can preserve the diversity while searching for the optimal PF through even infeasible regions.

2.2.6. Repairing Infeasible Solutions

Other approaches aim to improve the number of feasible solutions that an algorithm can find. Some rely on repair mechanisms to improve the quality of their solutions, while others seek to refine their search process by incorporating techniques that are not typically found in evolutionary algorithms. A popular example of a repair mechanism for handling ECMOPs can be found in the work of Saha and Ray [14]. These authors present a repair algorithm based on the concept of the Most Probable Point of failure (MPP), which is commonly used in the context of reliability-based optimization of structures [41].

Their approach is used to help an EMOA to obtain a feasible approximation of the Pareto front in an ECMOP aiming to reach a complete feasible popu-

lation by the end of a run. However, the inclusion in this case of an internal optimization problem that needs to be solved to repair individuals implies an increased time complexity for the algorithm used to approximate the solution.

2.2.7. Constraint handling at different stages.

Through this section, we have seen a variety of promising evolutionary algorithms capable of tackling constrained problems. While these algorithms are capable of achieving competitive performance there are cases where this performance deteriorates, especially when presented with highly complex feasible solution spaces.

In an attempt to address this issue, Ma et al. in [16] present the Multi-Stage Constrained EMOA (MSCMO), where constraints are added one after the other and handled at different stages of the evolutionary process. Considering that having a small number of constraints simplifies the solution space, allowing individuals to converge towards the feasible region more easily while preserving diversity.

Furthermore, the authors present a strategy for sorting the constraints to be handle at each stage. This is done as a way to improve the convergence of the algorithm in terms of feasibility. Starting from an evolved approximation of the unconstrained Pareto front obtained by the algorithm in the first stage, the strategy determines which constraints to handle according to the infeasible rate of the current approximation. The constraints that obtain a higher infeasible rate are considered in the first stages.

2.2.8. Coevolution for constraint handling

While evolutionary algorithms have demonstrated outstanding capabilities while tackling multi-objective optimization problems, Constrained MOPs are particularly challenging for these algorithms. The inclusion of constraints that reduce the solution space hinders the ability of EMOAs for retaining diversity while looking for valid solution candidates.

Tian et al. in [17] present a coevolutionary framework specifically designed to be used in constrained multi-objective optimization. The Coevolutionary Constrained Multi-objective Optimization framework (CCMO) evolves two populations at the same time, independently from one another. One focused only on solving a CMOP and the second focusing on solving a helper problem derived from the original. The helper problem here consists of the original CMOP without any constraint.

Both populations collaborate by sharing the offspring generated by each. In this case, since the helper problem is easier to solve than the original CMOP individuals in the second population preserve diversity, while individuals in the first population give priority to feasibility. These characteristics allow the offspring generated by the helper population to improve the capability for the main population to approach the PF, helping it escape local optimums and feasible but non-optimal regions.

2.2.9. The Tri-Goal Evolution Framework

The real challenge of constrained multi and many-objective optimization can be generalized to find the balance between solution quality, diversity, and feasibility. For an EMOA to properly solve a CMOP it needs to maintain a certain balance between solutions that are feasible but have worse objective values and solutions that achieve better objective values at the cost of being infeasible. This balance is needed for the preservation of the quality and diversity of individuals in a population.

Inspired by the above Zhou et al. present the Tri-Goal Evolution (TiGE) Framework for Constrained Many-Objective Optimization [18]. Here three indicator values are used to measure the quality, diversity and feasibility of individuals are estimated and used to measure the worth of each individual. Further, the authors propose TiGE-2 that uses what they named a two-level balance scheme.

The individual selection of TiGE-2 starts by applying the non-dominated sorting strategy. On the first level, the sorting is based on the minimization of the normalized individual objective values and a modified crowding degree estimator, trying to balance the quality and diversity of individuals. On the second level, the sorting strategy is then applied on the second level, where individuals are ordered from best to worst based on the minimization of their previous rank and the constraint violation, aiming at finally balancing quality and diversity and feasibility.

2.2.10. Hybridizing EMOAs for Constraint Handling

A recent approach from the work of Cuate et al. [19] is the $\epsilon-$NSGA-II/PT, which considers the use of a hybrid EMOA for handling equality constraints. The authors propose the usage of a continuation strategy to refine the approximation obtained by an EMOA. Starting with a modified version of the NSGA-II that incorporates the $\epsilon-$constraint strategy proposed in [37] the algorithm obtains a small set of feasible KKT points that are extended into a better approximation of the feasible Pareto set of the problem through the application of a mathematical continuation technique called the Pareto Tracer (PT) [42]. The incorporation of the PT to refine the approximation obtained by the EMOA forces the algorithm to lose its gradient-free properties, given that the technique requires the calculation of the gradients of the functions to work.

3. Building a Feasible Reference Set for Calculating the IGD

Motivated by the necessity of finding ways for EMOAs to be able to handle constraints and, most importantly, to be able to tackle ECMOPs, we extend a technique to artificially construct a reference that closely resembles the shape and location of the feasible PF of a problem given a set of constraints. As a result of this technique, we obtain a reference surface that allows using a performance indicator, such as the IGD, as a tool for selecting individuals which generationally shorten the distance between the approximation and the feasible front.

To construct the reference surface mentioned above, we first need to establish the characteristics of the PF to be found. Zapotecas-Martinez et al. [22] state that given a particular MOP, there is a very high probability that its Pareto front draws a curve in the objective space. Following this assumption, the authors relate the PF of a problem to a curve belonging to the following family:

$$\{(y_1)^\alpha + \ldots + (y_k)^\alpha = 1 : y_j \in [0,1], \alpha \in (0,\infty)\} \tag{8}$$

The main benefit of using this particular family of curves is that according to an index α, a curve with various degrees of convexity or concavity can be described. This behavior follows that given $\alpha < 1$, the resulting curve will be convex and when $\alpha > 1$ the curve will be concave. An interesting case is seen when $\alpha = 1$, where the resulting curve describes a line equivalent to a set of evenly distributed weight vectors when normalized within [0,1]. Using this curve family, we can decompose the problem of obtaining an adequate reference into three simpler subproblems:

1. To generate a set of evenly distributed weight vectors as a starting set.
2. To select from the population the set that minimizes each objective function and the point that describes the maximum bulge of the front, known as the knee.
3. To find an adequate α value to transform the set of weight vectors into a surface that passes through the points of the previous step.

We use the ideas from [22] as a starting point. However, we introduce the addition of a penalty factor inside the function used to determine the extreme and knee points at each generation. Moreover, we propose that searching for the correct value of α used to define the reference curvature can be transformed into the problem of determining if a point is inside a polygon. The last problem is easily solvable exploiting the relationship that exists between a Delaunay triangulation and a convex hull [43].

3.1. Constrained Achievement Function

The first step in approximating the curve of the PF of the problem is to identify the Convex Hull of Individual Minima (CHIM) of the problem. While doing this represents a challenge on its own, there is a way to identify CHIM candidate points from within the population of the algorithm at each generation, improving the approximation of the PF as the quality of solutions also improves. Identifying these points is comparable to finding a set of minimizers for each objective function from the problem, known as the extremes of the front from which the following definition arises:

Definition 3. Let $\mathbf{x}_j^*$ be the global minimizers of $f_j(\mathbf{x})$, $j = 1,\ldots,k$ and $\mathbf{F}_j^*(\mathbf{x}_j^*)$, $j = 1 \ldots, k$. Let Φ be a square matrix of order k whose j^{th} column is $\mathbf{F}_j^* - \mathbf{F}^*$. Then, the convex hull of individual minima is defined by the set of points in $\mathbb{R}^k$ that are convex combinations of $\mathbf{F}_j^*$ [22, 44], e.g. CHIM $= \left\{\phi\beta : \beta \in \mathbb{R}^k : \sum_{i=1}^k \beta_i = 1, \ \beta_i \geq 0, i = 1,\ldots,k\right\}$.

10

In the above definition, $\mathbf{F}_j^*(\mathbf{x}_j^*)$ denotes the Utopian vector defined by the global minima values of each objective function f_j. From **Definition 3**, it is possible to find individuals that closely resemble the extremes and knee point of the PF, from any approximation obtained by an algorithm at a given time step.

Thus, let us consider $Q = P_t \cup \mathbf{q}$ as the current approximation to the real PF achieved by an evolutionary algorithm. Then, we state the extremes (individual minima) of the PF (ξ^i) with the following achievement function presented in [22].

$$\xi^i = \arg\min_{x \in P} \left(\max_{j=1}^{k} \left((f_j(\mathbf{x}) - f_j^*)/e_j^i \right) \right) \tag{9}$$

where $\mathbf{e^i} = (e_1^i, \ldots, e_k^i)$ is the canonical basis in $\mathbb{R}^k$ and f_j^* is the current minimum for the jth objective function found along the search process. This achievement function allows us to also identify the maximum bulge (knee) formed by the convex hull of the population. This can be achieved by finding a solution candidate $\mathbf{x}$ such that it minimizes a Tchebycheff problem. To identify such a solution using the achievement function we employ the weight vector $\lambda_j = (\lambda_1 = \frac{1}{k} \ldots \lambda_k = \frac{1}{k})$ instead of $\mathbf{e^i}$.

We have found that for the particular case of ECMOPs, by using a penalty factor is possible to select as extremes a set of points that both lie closest to the feasible space and minimize an objective function. Due to its simplicity, we decided to apply a quadratic penalty [23] to the achievement function, resulting in the following:

$$\xi_p^i = \arg\min_{x \in P} \left(\max_{j=1}^{k} \left((f_j(\mathbf{x}) - f_j^*)/e_j^i \right) + \frac{1}{2u} \sum_{m=1}^{q} c_m(\mathbf{x})^2 \right) \tag{10}$$

where each $c_m(\mathbf{x})$ represents the constraint violations of an individual, where u is a penalty factor as close as possible to zero that determines the severity of the penalty applied to each reference point. The closer u is to zero, the harder the severity of the constraints applied to the function.

The importance of u comes from the necessity of applying enough pressure from the constraint violation to the achievement function; otherwise, one of two situations could happen. On the one hand, if the pressure is too high, it could lead to select only one individual minimum point, which prevents the construction of an adequate reference set. On the other hand, if the pressure is not enough, it can result in the reference set behaving like the unconstrained version of the problem.

We propose the use of an exterior penalty method, in contrast to a barrier method (i.e., interior penalty method) because we are only dealing with equality constraints. This means that the feasible set of constraints has an empty interior, and hence any barrier method is useless. An interior penalty method would cause an algorithm to ignore solutions close to the feasible boundaries [23, 45].

The effects of this new function are contrasted against those of the original unconstrained version in Figure 1. We can appreciate how different are the

11

selected sets and how our approach minimizes the distance from the feasible region. This would mean that the reference surface constructed using this set will be in proximity to the feasible space, and even more, the location of the points shows that the shape of the curve will be similar to that of the feasible region.
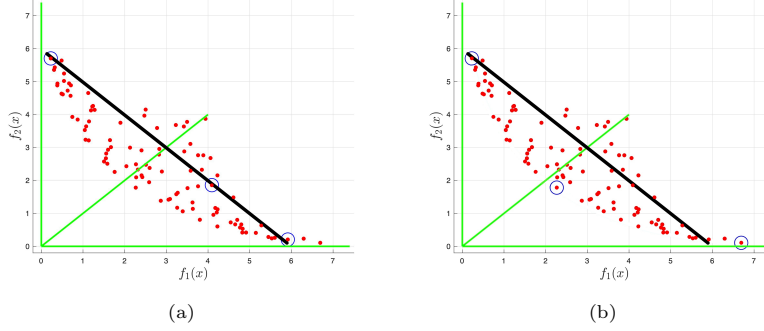

Figure 1: We present, in these figures, the result of evaluating a random population over ξ_p, figure (a), and over the original achievement function, figure (b). We have used (blue solid) circles to ease the identification of those elements selected by each function. In each figure, we show both the knee and the extreme points of the current population; notice that, for figure (a), these points minimize the distance from the feasible region denoted by the black diagonal.

3.2. Finding Alpha

Once the extremes and knee of the Pareto front have been identified, we can proceed to search for the factor α. This factor is used to curve the set of evenly distributed weights to the shape of the PF. The determination of α can be easily performed by browsing the range $[0, \infty]$ in small increments until the reference surface passes through the maximum bulge described by the CHIM (the knee point). An appropriate increment value for α needs to be selected as a large value can cause the search to overestimate the curvature of the surface, and a step that is too small can lead to an increment in the number of values to be tested, resulting in a time increase to find α.

We propose to curtail the number of α values tested during the process, performing a triangulation to determine if the curve of the PF we are looking for is concave or convex, which is essential for approximating it. This is done by constructing a polygon (convex hull) using the extreme points and the Utopian point that the extremes draw and determining if the knee vector lies within. Examples of the drawn polygons and knee point location can be seen in Figure 2.
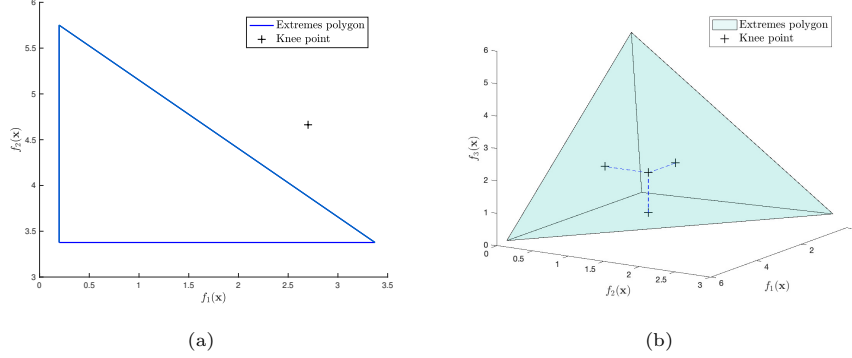
12

Figure 2: In the figures, we show two examples of polygons constructed using individual minima points. The algorithm then uses the knee point (black cross) to determine the curvature of the PF. In (a) the point lies outside the polygon for two objectives, exemplifying a concave PF, while in (b) the point lies inside a polygon, this time for three objectives, meaning the PF is convex.

The close relationship that exists between convex hulls and the Delaunay triangulation simplifies the process of determining if a point lies within the boundaries of a convex hull. By lifting the point to a paraboloid one dimension higher and searching for the facet projected by the triangulation that is farthest below it, we can determine if the point lies within the area of the convex hull [46, 47]. To find if the point is inside the polygon, we used a directed search to identify the facet that contains the point as implemented by the $Delaunay.find\_simplex$ function of $scipy$ [48].

After determining if the knee is inside or outside the polygon, the range $\alpha \in [0, \infty]$ is split into two. If the knee lies inside the polygon, the PF to approximate is convex and $\alpha \in [0, 1]$; otherwise, the PF is concave and $\alpha \in [1, \infty]$. Departing from the simplest case where $\alpha = 1$, which is equivalent to having a well-distributed set of weight vectors, we need to move in small steps of 0.05 as determined in [22], towards 0 or ∞ to transform those weights into an appropriate curvature. This can be seen in Figure 3 and Figure 4. Pareto dominance for this case is particularly useful to identify when an adequate α value has been found. To search for α, given a convex PF, we continue moving its value closer to zero until the curve it describes dominates the knee point. In the opposite case, we increase α until at least one point of the curvature is dominated by the knee point.
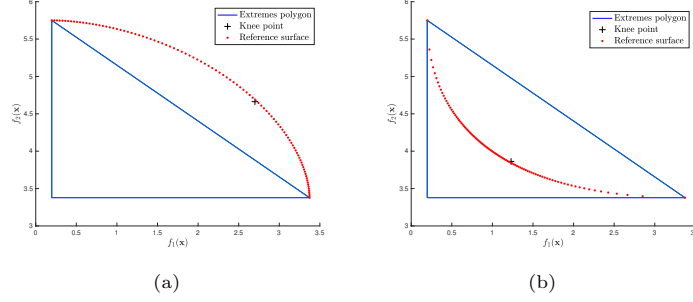
13

(a) (b)

Figure 3: Figures (a) and (b) exemplify a reference surface constructed to approximate a concave and a convex PF, respectively. The procedure draws the curves according to the extremes and knee points of a particular population identified beforehand.
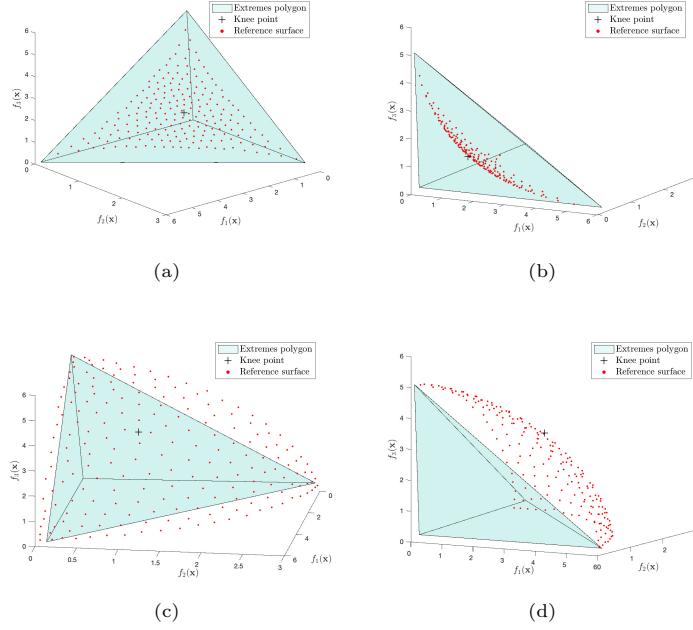

(a) (b)

(c) (d)

Figure 4: From the figures above, we present in (a) and (b) two views of the same 3-D reference surface. This surface approximates the curvature of a convex PF. While (c) and (d) present two views of a 3-D reference surface constructed to approximate a concave PF. In both cases, the surface is generated based on individuals identified as the current minima of the problem.

14

4. Constrained Reference Set - EMOA

₄₃₀   Here, we describe the structure of the COARSE-EMOA, an evolutionary algorithm intended for dealing with ECMOPs that uses the Inverted Generational Distance as part of its selection mechanism. The COARSE-EMOA does not need any information regarding the PF of the problem for calculating this indicator. Furthermore, it incorporates the previously mentioned method of
₄₃₅ synthesizing a feasible reference set as an attempt to approximate an adequate discretization of the front. This allows the algorithm to lead the search of individuals in direction of the reference set and towards an approximation of the feasible PF of the problem.

Algorithm 1 COnstrAined Reference SEt - EMOA

Input: stopping criterion (Number of function evaluations);
μ : population size
Output: P_t : Resulting approximation to the PF.

$t \leftarrow 0$
$P_t \leftarrow \{\mathbf{x}_1, \ldots, \mathbf{x}_\mu\}$ ▷ Initialize a population with μ random individuals.
$Evaluate(P_t)$ ▷ The population is evaluated.
$fe \leftarrow \mu$ ▷ Starts with μ Function Evaluations
while $fe < Stop\ criterion$ **do**
    $ind \leftarrow generate(P_t)$ ▷ Generate one new individual.
    $P_{t+1} \leftarrow reduce(P_t \cup \{ind\})$ ▷ Select the μ best individuals.
    $fe \leftarrow fe + 1$ ▷ One individual is evaluated at each generation.
end while

The COARSE-EMOA (Algorithm 1) starts by initializing a population P_t
₄₄₀ ($t = 0$) of μ randomly generated individuals. With each passing generation, the algorithm creates a new individual (ind) selecting two different individuals from P to act as parents. The determined pair is recombined by applying Simulated Binary Crossover (SBX) [49], and the offspring has a chance of being mutated with Polynomial-Based Mutation (PBM) [50]. Each ind has the possibility of
₄₄₅ becoming a member of the population P_{t+1}, if and only if replacing an individual with it, leads to a better-evaluated population. This process is repeated until a preset number of function evaluations f_e are performed.

Algorithm 2 $reduce(Q)$

Input: Q : Population to be reduced
Output: Q' : Population after reduction

$s = \arg\max_{\mathbf{x} \in Q}(IGD(Q\backslash\{\mathbf{x}\}, R))$ ▷ calculate the contribution to the IGD for all $\mathbf{x} \in Q$
$Q' = Q\backslash s$
return Q'

15

The reduction procedure we employ is described in Algorithm 2. It selects the μ best individuals from $Q = P_t \cup \{ind\}$ based on the contribution of each individual to the IGD. Given that the size of Q is $\mu + 1$, the new population discards only one element. For selecting which individuals are preserved, the following Definition is essential.

Definition 4. (Contribution to the IGD [22]) Let Z be a discrete approximation of the PF of a particular MOP. The contribution of a solution $x \in Q$ to the IGD indicator is defined as:

$$\Psi(x, Q, Z) = IGD(Q \backslash \{x\}, Z) \tag{11}$$

Unlike other algorithms that combine performance indicators and dominance in their selection process, such as RIB-EMOA or SMS-EMOA, our proposed COARSE-EMOA does not incorporate a non-dominated sorting procedure before calculating the contribution of individuals to the indicator. The main reason for only using the contribution of individuals is to study the capabilities of the proposed technique at properly approximating the PF of the problem using reference sets synthesized directly from the population. The effects of including a sorting procedure, like for example one based on the constraint-domination principle described in 2.2.3, provides a line of future research. The IGD was selected in the first instance over other indicators as it is not affected by the population size, and it provides balanced insight of the convergence, spread, and distribution of the algorithm according to the works of Bezerra et al. [51].

Given **Definition 4**, if removing an individual from Q does not cause any change to the IGD, or its removal reduces the current IGD, that individual is then deemed inferior. The worst individual from each generation is then discarded; ties are solved randomly. This allows those individuals that have better quality to survive each generation while reducing the distance that exists between the individuals and the artificial approximation of the PF which gets closer to the real feasible PF of the problem, as the algorithm progresses.

This reduction in the distance between the population and a better and feasible reference set causes the final population to lie and remain within the active region of the constraints. This results in individuals being unable to escaping the active region making this technique particularly suitable for working with equality constrained MOPs.

4.1. Computational Complexity

Studying the behavior of the algorithm during experimentation exhibits that COARSE-EMOA performance depends entirely on the speed of calculating the individual contribution to the performance indicator. Nevertheless calculating the IGD is significantly faster than calculating other performance indicators at $O(mMN)$ where m is the number of objectives, M is the number of reference points and N is the size of solution set [52]. This is very reasonable when compared with respect to other algorithms like the NSGA-III where the non-dominated sorting has a complexity of $O(mN^2)$.

Given that the COARSE-EMOA only generates one individual at each generation after the initial population, the complexity of calculating the individual contribution to the indicator remains $O(mMN)$. It is important to mention, that changing the number of offspring obtained at each generation would invariably increase this complexity.

5. Experimental Design and Results

As a way to illustrate that the COARSE-EMOA is capable of solving EC-MOPs, we evaluate the ability of the algorithm to obtain a good approximation to the PF over a set of benchmark ECMOPs. Moreover, we compare the obtained results with respect to those obtained by the ϵ-NSGA-II/PT, the CCMO, the MSCMO, the TiGE-2, the C-NSGA-II and the C-NSGA-III. It is worth noting that the last two methods incorporate the constrained-domination principle described in Section 2.2.

The ϵ-NSGA-II/PT, the CCMO, the MSCMO, and the TiGE-2 were chosen as a way to test the COARSE-EMOA with respect to other approaches from the state-of-the-art. We selected the C-NSGA-II and C-NSGA-III as a way to contrast a common constraint-handling technique, in this case the CDP, against our proposed constraint achievement function. Additionally, C-NSGA-III was selected given the good performance observed in [53]. The aim was to show the capabilities of our technique for guiding the search towards the feasible PF of a constrained problem.

The set of experiments performed consisted of a total of 20 independent runs of each algorithm per problem [19, 54, 55]. Each approximation was evaluated using two performance indicators to assess their quality and to quantify the number of feasible individuals found by the algorithms. The experiments were run in a Windows server with two Intel Xenon CPU E5-2670 v3 processors running at 2.30GHz with a total of 48 logic processors, and 1 TB of RAM.

We chose the parameters of each EMOA in such a way that we could make a fair comparison among them. To ensure that the final approximations contained the same number of candidate solutions, we set population sizes for each algorithm as $\mu = 100$ for problems with two objectives and $\mu = 300$ for the problem with three objectives [15]. The use of a larger population size in the problem with three objectives was for the algorithms to achieve a better distribution in the larger surface described by the PF in three dimensions [5].

The distribution indexes for the evolutionary operators (Simulated Binary Crossover (SBX) and Polynomial-Based Mutation (PBM)) adopted by our approach and the others were set as $distc = 20$ and $distm = 20$, respectively [13]. We kept the crossover probability as $pc = 0.9$ and the mutation probability as $pm = \frac{1}{\mu}$ where μ is the size of the population. These parameter values have been found appropriate for a wide variety of problems by many different authors [5, 56, 57]. Finally, we set the total number of function evaluations in such a way that it did not exceed 20,000 for the bi-objective problems and 150,000 for the tri-objective test problem [15].

17

The only parameter that is unique to our algorithm is the penalty factor u, which is used to determine the pressure that the constraint violation applies when looking for the feasible minima points of the problem. In this case, we have set a fixed parameter $u = 1 \times 10^{-6}$ for all the problems. As defined in [23], u needs to be as close as possible to zero so that it exerts enough pressure. However, initial empirical tests performed with the COARSE-EMOA over the problems with $u = 1 \times 10^{-7}$ resulted in the algorithm not identifying the necessary points for generating a reference at the very beginning of the search. This happens as there are not enough individuals sufficiently close to the feasibility boundary that are candidates to represent the extremes and knee point of the PF.

5.1. Test Problems

All seven algorithms, including the COARSE-EMOA, were tested over a set of Equality Constrained MOPs. Starting with the CZDT functions proposed in [14], which consist of a set of problems derived from the well-known ZDT test suite [58]. To further test the capabilities of the algorithms in particular on ECMOPs whose Pareto set location is altered by the constraints, we adopted a set of problems proposed by Cuate et al. [15], namely the Eq1-ZDT1, the Eq2-ZDT1, the Eq1-Quad and Eq2-Quad. Lastly, as a way to test the capabilities of the algorithms while handling PF that have been degenerated by constraints we include the first problem of the Eq-DTLZ suite proposed in [53].

Table 1 describes the different features that each of the mentioned test functions possesses. Each one allows the assessment of the performance of EMOAs under different conditions known to be challenging for them.

It is essential to mention that the constrained version of the ZDT test suite has a unique characteristic. While all the problems contain one equality constraint, its addition does not alter the feasibility of the true Pareto front of the problem, resulting in the same shape for both the constrained and unconstrained versions. Nevertheless, these problems pose a considerable challenge for evolutionary algorithms because of the difficulties associated with the constraints incorporated into them.

18

Table 1: Test problems used for experimentation

Problem	# of Objectives	# of Constraints	Characteristics
CZDT1	2	1	The problem has a convex Pareto front
CZDT2	2	1	The problem is the non-convex counterpart of CZDT1.
CZDT3	2	1	The problem has a discontinuous Pareto front consisting of several convex pieces.
CZDT4	2	1	The problem contains 21^9 local Pareto fronts, resulting in a multifrontal problem.
CZDT6	2	1	The problem presents two difficulties: First, the Pareto-optimal solutions are not uniformly distributed over the PF with more solutions located, where $f_1(\mathbf{x})$ is near one. Second, the number of solutions near the Pareto front is scarce. This density increases as the population gets farther away from the front.
Eq1-ZDT1	2	1	The problem is based on the ZDT1, here the constraint establishes the feasible region within the perimeter of a hyper-cylinder in the search space
Eq2-ZDT2	2	1	The problem is similar to the Eq1-ZDT1, it differs on the addition of box constraints, which results in a disconnected PF.
Eq1-Quad	3	1	The problem consists of three quadratic objective functions with the addition of one equality constraint. A modification of a problem found in [59].
Eq2-Quad	3	2	The problem is similar to the previous one with the addition of a second constraint. The Pareto set of this problem then consists of two connected components that can be both expressed by curves (and which are hence 1-dimensional).
Eq-DTLZ1	3	1	The problem is based on the DTLZ1 formulation. However, it includes one equality constraint in such a way that it reduces the dimensionality of the Pareto set and front by one.

The problems mentioned above are state-of-the-art examples of test problems that include equality constraints in their formulation [15, 53]. To the best of our knowledge, no other benchmarks currently exist that include only MOPs with equality constraints.

5.2. Measures for Assessing Performance

We are interested in quantifying two particular aspects of the approximations: (1) the number of feasible Pareto candidate solutions that the algorithms were able to find and (2) how good was each algorithm at approximating the feasible PF of the problems. For this, two different performance indicators were employed to evaluate the approximations.

To assess the percentage of feasible solutions found in the final approximations of the algorithms, we calculated the F_r as described in Section 2.1.3. A larger value obtained in this indicator means that more feasible solutions were found. This means that the techniques used to handle constraints were successful in guiding the search towards the feasible region of the problem.

To determine how close an approximation is to represent the feasible PF of the problems, the indicator Δ_p was used. This indicator provides a simple yet effective way to compare the results of the algorithms, providing a ranking that matches convergence, spread and distribution [25, 51]. The smaller the value obtained by this indicator, the closer an approximation is to the feasible PF. It is possible to use this indicator as we are working with benchmark problems whose feasible PF is known.

5.3. Performance of COARSE-EMOA

We will now describe the obtained results of applying the COARSE-EMOA over the 8 Equality Constrained Multi-objective Optimization Problems previously indicated. The results of the experiments are described in Table 2.

These results indicate that the COARSE-EMOA was able to obtain populations with more than 50% feasible individuals in most cases, even in the presence of discontinuous Pareto fronts, as is the case of CZDT3 and Eq2-ZDT1. However, the results also indicate that the approach used to generate the reference set presented sensitivity to multifrontal problems. This was the case when the COARSE-EMOA was unable to obtain a single feasible individual in CZDT4 and obtained an average Δ_p of 3.0401. The existence of local fronts throughout the search space caused the algorithm to be unable to construct a reference surface that corresponded with the global PF of the problem leading to stagnation of the individuals in a region far from it. An interesting thing to mention is that while the global minima of the problem were not reached in CZDT6, still 76% of the solutions were feasible. In the case of Eq1-Quad, Eq2-Quad and Eq-DTLZ1 the COARSE-EMOA was unable to cover the entirety of the PF. Nevertheless, the algorithm was able to find solutions with great proximity to the PF out of which 52%, 45% and 100% were feasible, respectively.

Table 2: Averaged Hausdorff distance (Δ_p) and Feasibility ratio (F_r) results of the COARSE-EMOA. The values were obtained over 20 independent test runs on each problem.

| Problem	Average Δ_p	St. Deviation	Median	Average $F_r(\%)$	St. Deviation	Median
CZDT1	0.0008	0.0003236	0.0006	0.73	0.0543357	0.76
CZDT2	0.0018	0.0005414	0.0017	0.62	0.0421776	0.63
CZDT3	0.0205	0.0007957	0.0204	0.54	0.0928615	0.56
CZDT4	3.0401	1.3131170	2.9625	0.00	0.00	0.00
CZDT6	0.0702	0.02799403	0.0692	0.73	0.2188509	0.76
Eq1-ZDT1	0.0009	0.0003848	0.0009	0.78	0.0727874	0.77
Eq2-ZDT1	0.0020	0.0005748	0.0020	0.70	0.1360902	0.70
Eq1-QUAD	0.2603	0.0214872	0.2608	0.52	0.0764229	0.54
Eq2-QUAD	2.3168	1.2085106	2.2913	0.45	0.0982110	0.46
Eq-DTLZ1	0.0282	0.0692024	0.02801	1.00	0.0000000	1.00

To better understand the results, we graphically show the best runs of the COARSE-EMOA for each problem in Figures 5 and 6. The blue surface shows the feasible PF of each problem while the red dots represent the final population of that particular run. The COARSE-EMOA was able to compute a set of points with a relatively good spread over the Pareto set in the majority of the problems; this is the case for CZDT1, CZDT2, CZDT3, Eq1-ZDT1 and Eq2-ZDT2. In the case of Figure 5(d) (problem CZDT4), we can see that no point close to the PF was obtained, due to the COARSE-EMOA getting stuck in a local front.
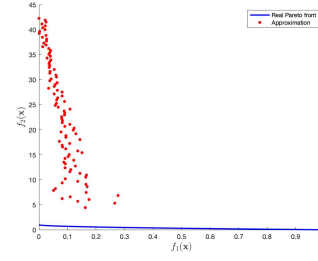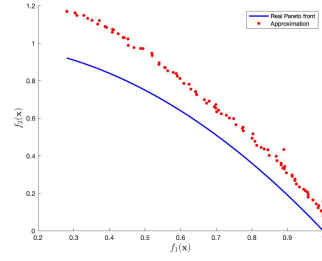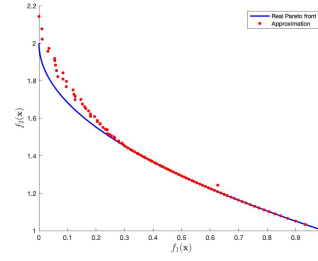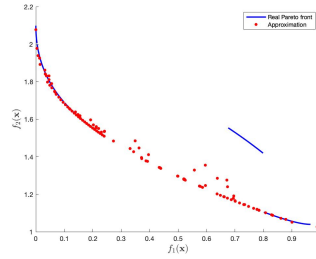
20
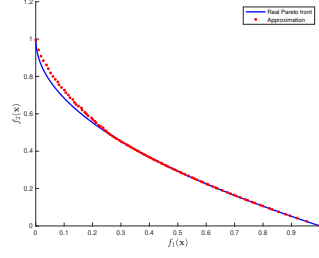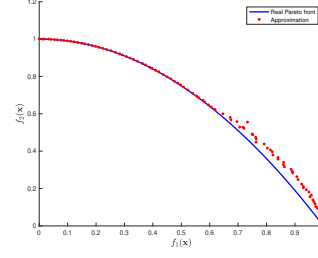
(a) CZDT1.

(b) CZDT2.
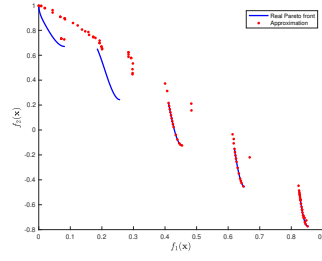
(c) CZDT3.

(d) CZDT4.

(e) CZDT6.

(f) Eq1-ZDT1.

(g) Eq2-ZDT1.

Figure 5: In the Sub-figures, we show the Pareto front approximations for the CZDT, Eq1-ZDT1 and Eq2-ZDT1 obtained by the COARSE-EMOA. The red dots represent the solution approximation, while the blue lines represent the real Pareto front of the problem.
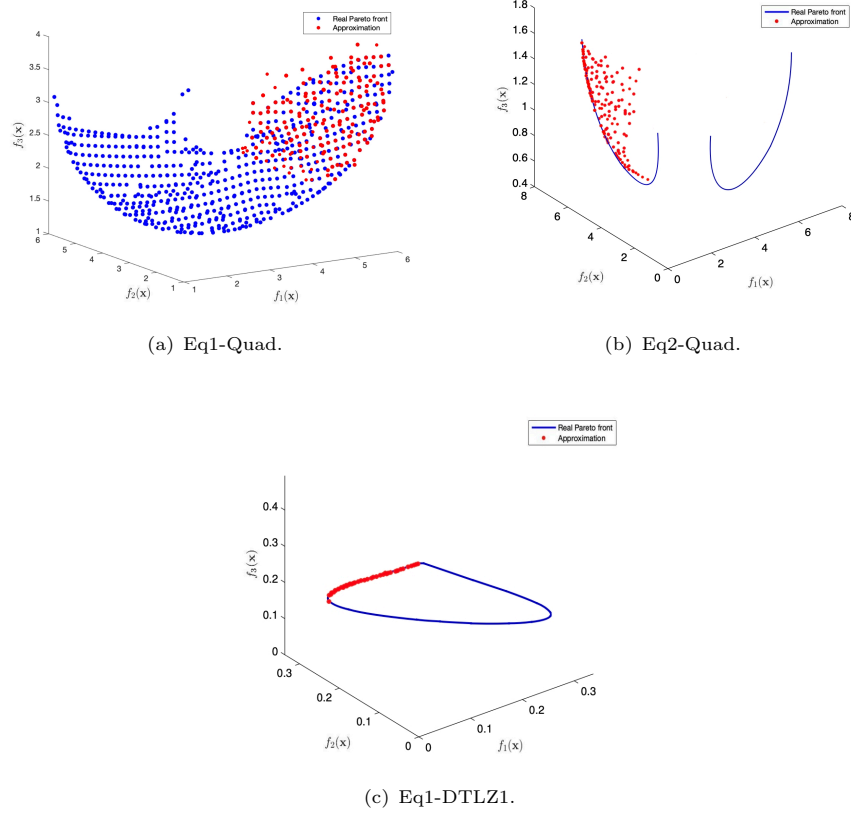
21

(a) Eq1-Quad.

(b) Eq2-Quad.

(c) Eq1-DTLZ1.

Figure 6: The graphs show a Pareto front approximation for Eq1-Quad, Eq2-Quad and Eq1-DTLZ1 using the COARSE-EMOA. The red dots represent the solution approximation, while the blue areas represent the real Pareto front of the problem.

5.4. Comparison of COARSE-EMOA vs Other Algorithms

Continuing with our experiments,we compared the COARSE-EMOA performance against six state-of-the-art algorithms using our set of ten problems. The results obtained are detailed in Table 3. We have highlighted the best results obtained on the Δ_p indicator with **boldface** text. We only consider feasible solutions while evaluating the quality of the approximations. Results with gray background highlight approximations composed exclusively of feasible individuals. The lack of results in Table 3 for the Δ_p column means that the associated algorithm was unable to generate even a single feasible individual.

Using the algorithms' output, we performed a Wilcoxon's rank-sum with a significance level of 0.05 [60]. We obtained the $p-$value by comparing the results of the COARSE-EMOA against ϵ-NSGA-II/PT, CCMO, MSCMO, TiGE-2, C-NSGA-III and C-NSGA-II when possible. Problems marked with (*) reflect a statistically significant result based on the Wilcoxon rank test. For the case of

22

algorithms CCMO, MSCMO, and TiGE-2 implementations were obtained from the Evolutionary multi-objective optimization platform (PlatEMO) presented by Tian et al. in [61]

The results above show that $\epsilon-$NSGA-II/PT obtained only entirely feasible approximations to the PF. It is followed by MSCMO that achieved fully feasible populations in 5 of the 10 problems. While COARSE-EMOA was unable to generate populations of exclusively feasible solutions, the algorithm still achieved an average of 60.7% feasibility overall. The Feasibility ratios obtained by the COARSE-EMOA show that our proposed technique for constructing a reference is able, in most cases (seven out of eight problems), to approximate the feasible PF of a problem while still taking into consideration equality constraints.

Furthermore, the results demonstrate that regardless of the ratios of feasibility, the COARSE-EMOA approximations present better distribution and convergence according to Δ_p, when compared against the other algorithms. This is supported by the COARSE-EMOA achieving the best overall performance in 4 out of the 10 problems, according to the indicator.

We found that our approach is capable of overcoming different obstacles such as discontinuous Pareto fronts and non-uniform search spaces. However, the reference surface construction is highly sensitive to the presence of local fronts. This is evident from the results obtained for the case of the multifrontal problem (CZDT4), where the COARSE-EMOA was unable to escape from local fronts causing the algorithm to stagger.

According to our results, COARSE-EMOA obtains a similar and even better individuals distribution over the PF supported by Δ_p values, presented in Figure 8. These performance values indicate that the COARSE-EMOA has beneficial convergence properties comparable to those of other state-of-the-art algorithms. Nevertheless, the COARSE-EMOA was unable to surpass the feasibility ratio of the ϵ-NSGA-II/PT, CCMO, MSCMO, and TiGE-2 as seen in Figure 7.

As mentioned before, the COARSE-EMOA is capable of tackling ECMOPs without extra information. This means that no gradient information for the incorporation of local search or continuation techniques is necessary to achieve an approximation of the feasible PF of an ECMOP, nor a transformation is required to obtain a set of feasible solutions. The underlying idea of using an EMOA, and any stochastic technique for the matter, comes from the impossibility of guaranteeing that gradient information is obtainable for the problem to be optimized.

Table 3: Resulting Δ_p and Feasibility Ratio of the COARSE-EMOA and six more algorithms. The values were obtained using 20 independent runs on each problem.

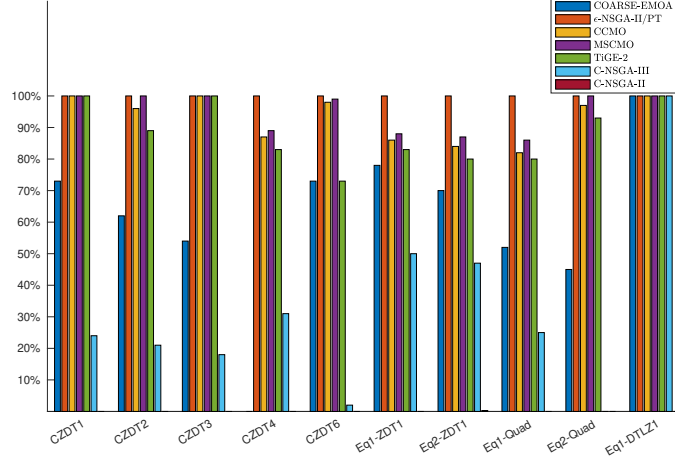| Problem | COARSE-EMOA | | ϵ-NSGA-II/PT | | CCMO | | MSCMO | | TiGE-2 | | C-NSGA-III | | C-NSGA-II | |
	Δ_p (Mean / Std. Dev.)	F_r	Δ_p (Mean / Std. Dev.)	F_r	Δ_p (Mean / Std. Dev.)	F_r	Δ_p (Mean / Std. Dev.)	F_r	Δ_p (Mean / Std. Dev.)	F_r	Δ_p (Mean / Std. Dev.)	F_r	Δ_p (Mean / Std. Dev.)	F_r
CZDT1(*)	**0.0008** / **0.0003**	0.73	0.0038 / 0.0002	1.00	0.0015 / 0.0003	1.00	0.0013 / 0.0005	1.00	0.0034 / 0.0017	1.00	0.6086 / 0.0161	0.24	- / -	0.00
CZDT2(*)	**0.0018** / **0.0005**	0.62	0.0038 / 0.0014	1.00	0.0023 / 0.0021	0.96	0.025 / 0.0032	1.00	0.0066 / 0.0009	0.89	0.6214 / 0.0116	0.21	- / -	0.00
CZDT3(*)	0.0205 / 0.0007	0.54	0.0156 / 0.0164	1.00	**0.0150** / **0.0089**	1.00	0.0155 / 0.0080	1.00	0.0161 / 0.0097	1.00	0.8812 / 0.0129	0.18	- / -	0.00
CZDT4	- / -	0.00	**0.0031** / **0.0016**	1.00	0.0237 / 0.0156	0.87	0.0179 / 0.0135	0.89	0.0338 / 0.0076	0.83	0.8129 / 0.0554	0.31	- / -	0.00
CZDT6(*)	0.0702 / 0.0279	0.73	0.0884 / 0.0180	1.00	0.0782 / 0.0340	0.98	**0.0695** / **0.0482**	0.99	0.0954 / 0.1472	0.73	1.2133 / 0.0393	0.02	- / -	0.00
Eq1-ZDT1(*)	**0.0009** / **0.00038**	0.78	0.0158 / 0.0015	1.00	0.0063 / 0.0022	0.86	0.0054 / 0.0015	0.88	0.0144 / 0.0127	0.83	0.6543 / 0.1053	0.50	- / -	0.00
Eq2-ZDT1(*)	**0.0020** / **0.0005**	0.70	0.1251 / 0.0428	1.00	0.0493 / 0.0642	0.84	0.0823 / 0.1070	0.87	0.1319 / 0.0368	0.80	0.6624 / 0.1235	0.47	2.3760 / 1.6357	0.003
Eq1-QUAD(*)	0.2603 / 0.0214	0.52	0.1261 / 0.0043	1.00	0.1517 / 0.0081	0.82	**0.0991** / **0.0115**	0.86	0.5051 / 0.0351	0.80	0.5714 / 0.0931	0.25	- / -	0.00
Eq2-QUAD(*)	2.3168 / 1.2085	0.45	1.9969 / 1.0378	1.00	1.2383 / 0.0019	0.97	**1.1005** / **0.0027**	1.00	1.3107 / 0.0083	0.93	- / -	0.00	- / -	0.00
Eq-DTLZ1(*)	0.0282 / 0.0692	1.00	**0.02381** / **0.0010**	1.00	0.0627 / 0.0024	1.00	0.0523 / 0.0024	1.00	0.1351 / 0.0094	1.00	0.1877 / 0.0731	1.00	- / -	0.00

Figure 7: The Average F_r obtained by each of the algorithms over the test problems is shown. Each bar represents the feasibility ratio average of the COARSE-EMOA, $\epsilon-$NSGA-II, CCMO, MSCMO, TiGE-2, C-NSGA-III and C-NSGA-II, respectively.
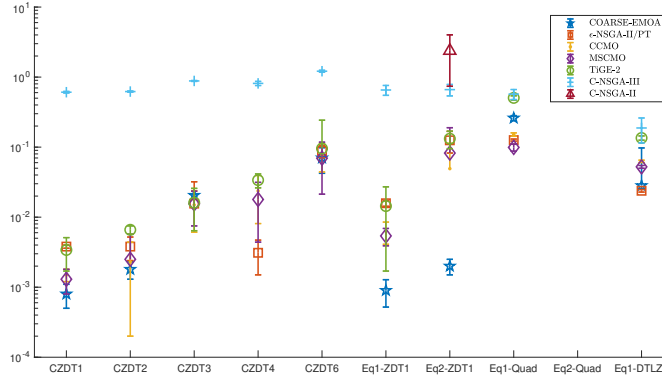

Figure 8: A semi-log scatter plot of the average Δ_p obtained by each algorithm over our set of test problems. The blue stars represent the average and standard deviation of Δ_p of the COARSE-EMOA. The red squares represent the same information for $\epsilon-$NSGA-II/PT, while the yellow dots do so for CCMO. The purple diamonds represent the values of MSCMO, and the green circles the values of TiGE-2. C-NSGA-III and C-NSGA-II are represented by the light blue crosses and the burgundy triangles, respectively.

25

6. Conclusions and Future Work

We have presented an EMOA capable of solving Equality Constrained Multi-objective Optimization Problems. Its capabilities of tackling ECMOPs come mainly from the addition of a quadratic penalty integrated into an achievement function based on Tchebycheff for approximating the shape and location of the Pareto Front (PF) of a problem. This new achievement function allowed the algorithm to remain entirely based on the minimization of a performance indicator through evolution while considering feasibility.

According to the observed behavior of the COARSE-EMOA and the obtained results, our proposed approach can successfully synthesize reference sets that approximate the unknown feasible Pareto Front (PF) of an ECMOP as the algorithm progresses. Each reference set aims at minimizing a penalty associated with the constraint violation of individuals. This allowed the algorithm to evolve its population towards the boundary of the equality constraints and, most importantly, to remain there, which is something that EMOAs naturally have problems achieving [34].

While the COARSE-EMOA did not obtain a complete feasible PF approximation on all the problems, on average more than half of its resulting individuals were valid solution candidates, as reflected by the average feasibility ratio. Furthermore, the results in Δ_p indicate that COARSE-EMOA obtained a better-distributed population closer to the feasible PF of the problem in contrast to the approximations obtained by the other algorithms over four of the test functions.

As a path for future work, we have considered an attractive approach to explore the possibility of incorporating different methods to extend the COARSE-EMOA. The first extension contemplates the incorporation of a sorting procedure based on the constrained domination principle. Such a procedure could reduce the number of times the algorithm calculates the contribution to the performance indicator of each individual and possibly reduce the overall time complexity of the COARSE-EMOA. Another path takes into consideration the convergence properties shown by the COARSE-EMOA, the incorporation of a technique to refine its approximation could help assure that a full feasible state is achieved and could even help to improve the quality of the obtained approximation in terms of coverage of the Pareto front.

We also consider the possibility of exploring different curve families that could provide the means of constructing a reference set with a more flexible structure. Other families of curves could offer the possibility of handling unexpected changes in the front found in problems with irregular PF shapes, seeking to improve all in all the performance of any algorithm incorporating the reference construction technique.

This research focused on demonstrating the capabilities of the COARSE-EMOA when dealing with two and three-objective problems, each with one or two equality constraints. This limitation arises from the geometrical nature of the proposed procedure to generate the reference surface, in specific the transformation of a lattice of points to fit within those individuals considered to be

extreme and knee candidates.

By researching and implementing a generalization for a higher dimensionality of this procedure, the basic structure of the COARSE-EMOA should allow it to be applied in Equality Constrained Many-objective optimization problems. Future analyses on the behavior of the algorithm are needed where such a procedure is presented and studied over Many-objective optimization problems with at least one constraint, examples of which have been introduced in [53, 61].

Acknowledgments

References

[1] C. A. Coello Coello, G. B. Lamont, D. A. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation), Springer-Verlag, Berlin, Heidelberg, 2006.

[2] T. Krink, S. Paterlini, Multiobjective optimization using differential evolution for real-world portfolio optimization, Computational Management Science 8 (1) (2011) 157–179. doi:10.1007/s10287-009-0107-6.

[3] P. M. Reed, B. S. Minsker, A. H. Valocchi, Why Optimize Long Term Groundwater Monitoring Design? A Multiobjective Case Study of Hill Air Force Base, 2001, pp. 1–8. arXiv:https://ascelibrary.org/doi/pdf/10.1061/40569\%282001\%29502, doi:10.1061/40569(2001)502.

[4] E. Tsoi, Kit Po Wong, Chun Che Fung, Hybrid GA/SA algorithms for evaluating trade-off between economic cost and environmental impact in generation dispatch, in: Proceedings of 1995 IEEE International Conference on Evolutionary Computation, Vol. 1, 1995, pp. 132–.

[5] K. Deb, D. Kalyanmoy, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Inc., USA, 2001.

[6] P. A. Vikhar, Evolutionary algorithms: A critical review and its future prospects, in: 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), 2016, pp. 261–265.

[7] C. A. Coello Coello, Treating constraints as objectives for single-objective evolutionary optimization, Engineering Optimization 32 (3) (2000) 275–308. arXiv:https://doi.org/10.1080/03052150008941301, doi:10.1080/03052150008941301.

27

[8] Z. Fan, Y. Fang, W. Li, J. Lu, X. Cai, C. Wei, A Comparative Study of Constrained Multi-Objective Evolutionary Algorithms on Constrained Multi-Objective Optimization Problems, in: 2017 IEEE Congress on Evolutionary Computation (CEC'2017), IEEE Press, San Sebastián, Spain, 2017, pp. 209–216, iSBN 978-1-5090-4601-0.

[9] E. Mezura-Montes, C. A. Coello Coello, Constraint-Handling in Nature-Inspired Numerical Optimization: Past, Present and Future, Swarm and Evolutionary Computation 1 (4) (2011) 173–194.

[10] C. A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer Methods in Applied Mechanics and Engineering 191 (11) (2002) 1245 – 1287. `doi:10.1016/S0045-7825(01)00323-1`.

[11] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, USA, 2004.

[12] E. Mezura-Montes, C. A. Coello Coello, A survey of constraint-handling techniques based on evolutionary multiobjective optimization.

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

[14] A. Saha, T. Ray, Equality constrained multi-objective optimization, in: 2012 IEEE Congress on Evolutionary Computation, 2012, pp. 1–7. `doi:10.1109/CEC.2012.6256109`.

[15] O. Cuate, A. Ponsich, L. Uribe, S. Zapotecas-Martínez, A. Lara, O. Schütze, A new hybrid evolutionary algorithm for the treatment of equality constrained MOPs, Mathematics 8 (1) (2019) 7. `doi:10.3390/math8010007`.

[16] H. Ma, H. Wei, Y. Tian, R. Cheng, X. Zhang, A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints, Information Sciences 560 (2021) 68–91. `doi:https://doi.org/10.1016/j.ins.2021.01.029`.
URL `https://www.sciencedirect.com/science/article/pii/S0020025521000566`

[17] Y. Tian, T. Zhang, J. Xiao, X. Zhang, Y. Jin, A coevolutionary framework for constrained multiobjective optimization problems, IEEE Transactions on Evolutionary Computation 25 (1) (2021) 102–116. `doi:10.1109/TEVC.2020.3004012`.

[18] Y. Zhou, M. Zhu, J. Wang, Z. Zhang, Y. Xiang, J. Zhang, Tri-goal evolution framework for constrained many-objective optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems 50 (8) (2020) 3086–3099. `doi:10.1109/TSMC.2018.2858843`.

[19] O. Cuate, L. Uribe, A. Ponsich, A. Lara, F. Beltran, A. R. Sánchez, O. Schütze, A new hybrid metaheuristic for equality constrained bi-objective optimization problems, in: K. Deb, E. Goodman, C. A. Coello Coello, K. Klamroth, K. Miettinen, S. Mostaghim, P. Reed (Eds.), Evolutionary Multi-Criterion Optimization, Springer International Publishing, Cham, 2019, pp. 53–65.

[20] A. Barkat Ullah, R. Sarker, C. Lokan, Handling equality constraints in evolutionary optimization, European Journal of Operational Research 221 (3) (2012) 480 – 490. `doi:https://doi.org/10.1016/j.ejor.2012.01.047`.

[21] C. A. Coello Coello, N. C. Cortés, Solving multiobjective optimization problems using an artificial immune system, Genetic Programming and Evolvable Machines 6 (2) (2005) 163–190. `doi:10.1007/s10710-005-6164-x`.

[22] S. Zapotecas-Martínez, V. A. Sosa-Hernández, H. Aguirre, K. Tanaka, C. A. Coello Coello, Using a family of curves to approximate the pareto front of a multi-objective optimization problem, in: T. Bartz-Beielstein, J. Branke, B. Filipič, J. Smith (Eds.), Parallel Problem Solving from Nature – PPSN XIII, Springer International Publishing, Cham, 2014, pp. 682–691.

[23] J. Nocedal, S. J. Wright, Numerical Optimization, 2nd Edition, Springer, New York, NY, USA, 2006.

[24] D. A. Van Veldhuizen, G. B. Lamont, On measuring multiobjective evolutionary algorithm performance, in: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), Vol. 1, 2000, pp. 204–211 vol.1. `doi:10.1109/CEC.2000.870296`.

[25] O. Schütze, X. Esquivel, A. Lara, C. A. Coello Coello, Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization, IEEE Transactions on Evolutionary Computation 16 (4) (2012) 504–522. `doi:10.1109/TEVC.2011.2161872`.

[26] H. Ishibuchi, H. Masuda, Y. Tanigaki, Y. Nojima, Modified distance calculation in generational distance and inverted generational distance, in: A. Gaspar-Cunha, C. Henggeler Antunes, C. A. Coello Coello (Eds.), Evolutionary Multi-Criterion Optimization, Springer International Publishing, Cham, 2015, pp. 110–125.

[27] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271. `doi:10.1109/4235.797969`.

[28] M. Pilegaard Hansen, A. Jaszkiewicz, Evaluating the quality of approximations to the non-dominated set (1998).

[29] Z. Fan, W. Li, X. Cai, K. Hu, H. Lin, H. Li, Angle-based Constrained Dominance Principle in MOEA/D for Constrained Multi-objective Optimization Problems, in: 2016 IEEE Congress on Evolutionary Computation (CEC'2016), IEEE Press, Vancouver, Canada, 2016, pp. 460–467, iSBN 978-1-5090-0623-9.

[30] X. Yu, X. Yu, Y. Lu, G. G. Yen, M. Cai, Differential Evolution Mutation Operators for Constrained Multi-Objective Optimization, Applied Soft Computing 67 (2018) 452–466.

[31] W. Ning, B. Guo, Y. Yan, X. Wu, J. Wu, D. Zhao, Constrained Multi-Objective Optimization Using Constrained Non-Dominated Sorting Combined with an Improved Hybrid Multi-Objective Evolutionary Algorithm, Engineering Optimization 49 (10) (2017) 1645–1664.

[32] O. Yeniay, Penalty function methods for constrained optimization with genetic algorithms, Mathematical and Computational Applications 10 (2005) 45–56. `doi:10.3390/mca10010045`.

[33] F. S. Lobato, V. Steffen, Treatment of Multi-objective Optimization Problem, Springer International Publishing, Cham, 2017, pp. 25–44. `doi:10.1007/978-3-319-58565-9_3`.

[34] Z. Michalewicz, M. S. Y, Evolutionary algorithms for constrained parameter optimization problems, Evolutionary Computation 4 (1996) 1–32.

[35] C. Xiao, J. Yin, X. Zhou, Z. Xue, M. Yi, W. Shu, Constrained multi-objective evolutionary algorithm based on decomposition for environmental/economic dispatch, in: 2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA), 2014, pp. 1–8.

[36] A. Homaifar, C. X. Qi, S. H. Lai, Constrained optimization via genetic algorithms, SIMULATION 62 (4) (1994) 242–253. `arXiv:https://doi.org/10.1177/003754979406200405`, `doi:10.1177/003754979406200405`.

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

[38] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach, IEEE Transactions on Evolutionary Computation 18 (4) (2014) 602–622. `doi:10.1109/TEVC.2013.2281534`.

[39] R. Mallipeddi, P. N. Suganthan, Ensemble of constraint handling techniques, IEEE Transactions on Evolutionary Computation 14 (4) (2010) 561–579. `doi:10.1109/TEVC.2009.2033582`.

[40] R. Jiao, S. Zeng, C. Li, S. Yang, Y.-S. Ong, Handling constrained many-objective optimization problems via problem transformation, IEEE Transactions on Cybernetics (2020) 1–14 `doi:10.1109/TCYB.2020.3031642`.

[41] X. Du, W. Chen, A most probable point-based method for efficient uncertainty analysis, Journal of Design and Manufacturing Automation 4 (1) (2001) 47–66. `arXiv:https://doi.org/10.1080/15320370108500218`, `doi:10.1080/15320370108500218`.

[42] A. Martín, O. Schütze, Pareto tracer: a predictor–corrector method for multi-objective optimization problems, Engineering Optimization 50 (3) (2018) 516–536. `arXiv:https://doi.org/10.1080/0305215X.2017.1327579`, `doi:10.1080/0305215X.2017.1327579`.

[43] M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, Computational Geometry: Algorithms and Applications, 3rd Edition, Springer-Verlag TELOS, Santa Clara, CA, USA, 2008.

[44] I. Das, Nonlinear multicriteria optimization and robust optimality, Ph.D. thesis, Rice University (1997).

[45] I. Griva, S. Nash, A. Sofer, Linear and Nonlinear Optimization: Second Edition, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2009.

[46] O. Devillers, S. Pion, M. Teillaud, Walking in a triangulation, in: Proceedings of the Seventeenth Annual Symposium on Computational Geometry, SCG '01, Association for Computing Machinery, New York, NY, USA, 2001, p. 106–114. `doi:10.1145/378583.378643`.

[47] E. P. Mücke, I. Saias, B. Zhu, Fast randomized point location without preprocessing in two- and three-dimensional delaunay triangulations, Computational Geometry 12 (1) (1999) 63 – 83. `doi:https://doi.org/10.1016/S0925-7721(98)00035-2`.

[48] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. `doi:https://doi.org/10.1038/s41592-019-0686-2`.

[49] R. Agrawal, K. Deb, R. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 9.

[50] K. Deb, D. Deb, Analysing mutation schemes for real-parameter genetic algorithms, International Journal of Artificial Intelligence and Soft Computing 4 (2014) 1–28. `doi:10.1504/IJAISC.2014.059280`.

[51] L. Bezerra, M. López-Ibáñez, T. Stützle, An empirical assessment of the properties of inverted generational distance on multi- and many-objective optimization, Vol. 10173, 2017, pp. 31–45. `doi:10.1007/978-3-319-54157-0_3`.

[52] X. Cai, Y. Xiao, M. Li, H. Hu, H. Ishibuchi, X. Li, A grid-based inverted generational distance for multi/many-objective optimization, IEEE Transactions on Evolutionary Computation 25 (1) (2021) 21–34. `doi:10.1109/TEVC.2020.2991040`.

[53] O. Cuate, L. Uribe, A. Lara, O. Schutze, A Benchmark for Equality Constrained Multi-Objective Optimization, Swarm and Evolutionary Computation 52, article Number: UNSP 100619.

[54] V. A. Sosa Hernández, A. Lara, H. Trautmann, G. Rudolph, O. Schütze, The Directed Search Method for Unconstrained Parameter Dependent Multi-objective Optimization Problems, Springer International Publishing, Cham, 2017, pp. 281–330. `doi:10.1007/978-3-319-44003-3_12`.

[55] V. A. Sosa Hernández, O. Schütze, H. Wang, A. Deutz, M. Emmerich, The Set-Based Hypervolume Newton Method for Bi-Objective Optimization, IEEE Transactions on Cybernetics 50 (5) (2020) 2186–2196.

[56] K. Deb, S. Agrawal, Understanding interactions among genetic algorithm parameters, in: in Foundations of Genetic Algorithms 5, Morgan Kaufmann, 1999, pp. 265–286.

[57] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, V. B. S. Prasath, Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach, Information 10 (12) (2019) 390. `doi:10.3390/info10120390`.

[58] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary computation 8 (2000) 173–95. `doi:10.1162/106365600568202`.

[59] A. Martín, O. Schütze, Pareto tracer: a predictor–corrector method for multi-objective optimization problems, Engineering Optimization 50 (3) (2018) 516–536. `arXiv:https://doi.org/10.1080/0305215X.2017.1327579`, `doi:10.1080/0305215X.2017.1327579`.

[60] W. Haynes, Wilcoxon Rank Sum Test, Springer New York, New York, NY, 2013, pp. 2354–2355. `doi:10.1007/978-1-4419-9863-7_1185`.

[61] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization, IEEE Computational Intelligence Magazine 12 (4) (2017) 73–87.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

AUTHORSHIP STATEMENT

Manuscript title: COARSE-EMOA: An Indicator-based Evolutionary Algorithm for Solving Equality

 Constrained Multi-objective Optimization Problems

_____

All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication before its appearance in the *Hong Kong Journal of Occupational Therapy*.

Authorship contributions
Please indicate the specific contributions made by each author (list the authors' initials followed by their surnames, e.g., Y.L. Cheung). The name of each author must appear at least once in each of the three categories below.

Category 1
Conception and design of study: J. L. Llano ,   R. Monroy   ,   V. A. Sosa ,   C. A. Coello ;

acquisition of data: J. L. Llano   , _____, _____, _____;

analysis and/or interpretation of data: J. L. Llano ,   R. Monroy ,   V. A. Sosa   , C. A. Coello .

Category 2
Drafting the manuscript: J. L. Llano   ,   R. Monroy   ,   V. A. Sosa   ,   C. A. Coello   ;

revising the manuscript critically for important intellectual content: R. Monroy , V. A. Sosa   ,

  C. A. Coello , _____.

Category 3
Approval of the version of the manuscript to be published (the names of all authors must be listed):

   J. L. Llano   ,   R. Monroy   ,   V. A. Sosa   ,   C. A. Coello   , _____,

_____, _____, _____, _____, _____.

Acknowledgements

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

This statement is signed by all the authors (*a photocopy of this form may be used if there are more than 10 authors***):**

Author's name (typed)	Author's signature	Date
Jesus L. Llano García		May 29, 2021
Raul Monroy		May 29, 2021
Victor A. Sosa Hernandez		May 29, 2021
Carlos A. Coello Coello		May 29, 2021