A Multi-Objective Evolutionary Algorithm based on Parallel Coordinates

Raquel Hernández Gómez
Computer Science Department
CINVESTAV-IPN
Mexico City, México
rhernandez@computacion.cs.cinvestav.mx

Carlos A. Coello Coello
Computer Science Department
CINVESTAV-IPN
Mexico City, México
ccoello@cs.cinvestav.mx

Enrique Alba Torres
Dpto. de Lenguajes y Ciencias de la Computación
University of Málaga
Málaga, Spain
eat@lcc.uma.es

ABSTRACT

Multi-Objective Evolutionary Algorithms (MOEAs) are powerful tools for solving a wide range of real-world applications that involve the simultaneous optimization of several objective functions. However, their scalability to many-objective problems remains as an important issue since, due to the large number of non-dominated solutions, the search is guided solely by the diversity criterion. In this paper, we propose a novel MOEA that incorporates a density estimator based on a visualization technique called Parallel Coordinates. Using this approach, a graph is represented by a digital image, where a pixel identifies the level of overlapping line segments and those individuals covering a wide area of the image have a high probability of survival. Experimental results indicate that our proposed approach, called Multi-objective Optimizer based on Value Path (MOVAP), outperforms existing algorithms based on clustering (SPEA2), crowding distance (NSGA-II), reference points (NSGA-III) and the hypervolume indicator (HypE) on most of the problems of the WFG test suite for five and seven objectives, while its performance in low dimensionality remains competitive.

Keywords

Multi-objective optimization; genetic algorithms; selection

1. INTRODUCTION

A Multi-objective Optimization Problem (MOP) is defined as follows:

$$\text{Minimize} \quad \vec{F}(\vec{x}) := (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x})) \tag{1}$$

$$\text{subject to} \quad \vec{x} \in \mathcal{S}, \tag{2}$$

where $\vec{x}$ is the *vector of decision variables*, $\mathcal{S} \subset \mathbb{R}^n$ is the *feasible region set* and $\vec{F}(\vec{x})$ is the vector of m (≥ 2) *objective functions* ($f_i : \mathbb{R}^n \to \mathbb{R}$). The aim is to seek from among the set of all values which satisfy the constraint functions defined in equation (2) the particular set $\vec{x^*}$ which yields the optimum values of all the objective functions.

Multi-Objective Evolutionary Algorithms (MOEAs) have been successfully applied to solve MOPs, thus becoming increasingly popular in recent years [4]. Their main components are a population of individuals $P = \{(\vec{x}, \vec{y}) \colon \vec{x} \in \mathcal{S}, \vec{y} = \vec{F}(\vec{x})\}$; the operators: parent selection (π), variation (ν) and survival selection (σ); as well as the iterative rule $P^{t+1} = \sigma\left(\nu(\pi\{P^t\}), P^t\right)$. The final population of a MOEA is known as approximation set.

The main aims of a MOEA are: 1) to achieve convergence to the Pareto optimal front,[1] 2) a uniform distribution along the Pareto front and 3) a good spread of solutions. The latter two are closely related, generally applied in objective space ($\mathbb{R}^m$), and denoted by *diversity*. Convergence is usually achieved by means of *Pareto dominance*,[2] while diversity is handled through a density estimator.

During the last two decades, a number of density estimators have been proposed, such as clustering [21] and crowding distance [7]. Even though they are scalable to any number of objectives, their proximity to the Pareto optimal front is poor, aggravating in many-objective problems ($m \geq 4$) [12, 16]. This is mainly due to the fact that most or all solutions in the population quickly become non-dominated with respect to the rest, and the best individuals are identified by the density estimator. Thus, in some cases good locally non-dominated solutions in terms of proximity toward the Pareto optimal front might be filtered out at the expense of keeping good solutions in terms of diversity, but which may be distant from the Pareto optimal front [1]. Another example of a density estimator is the hypervolume indicator [19], which leads to sets of solutions whose spread along the Pareto front is maximized (although this does not guarantee a uniform distribution). Nevertheless, the computational cost of the hypervolume increases exponentially on the number of objectives [3], which makes it prohibitive for many-objective problems. Finally, another diversity strategy is the use of a set of reference points [6], which should be uniformly distributed in objective space. When using the most recommended method for generating such points (Simplex-Lattice

GECCO '16, July 20-24, 2016, Denver, CO, USA
© 2016 ACM. ISBN 978-1-4503-4206-3/16/07...$15.00
DOI: http://dx.doi.org/10.1145/2908812.2908856

[1] The Pareto optimal front is defined as $POF := \{\vec{F}(\vec{x}) \in \mathbb{R}^m \colon \vec{x} \in \mathcal{S}, \nexists \vec{y} \in \mathcal{S}, \vec{y} \prec \vec{x}\}$.

[2] A solution $\vec{x} \in \mathcal{S}$ dominates a solution $\vec{y} \in \mathcal{S}$ ($\vec{x} \prec \vec{y}$), if and only if $\forall i \in \{1, \ldots, m\}$, $f_i(\vec{x}) \leq f_i(\vec{y})$ and $\exists j \in \{1, \ldots, m\}$, $f_j(\vec{x}) < f_j(\vec{y})$.

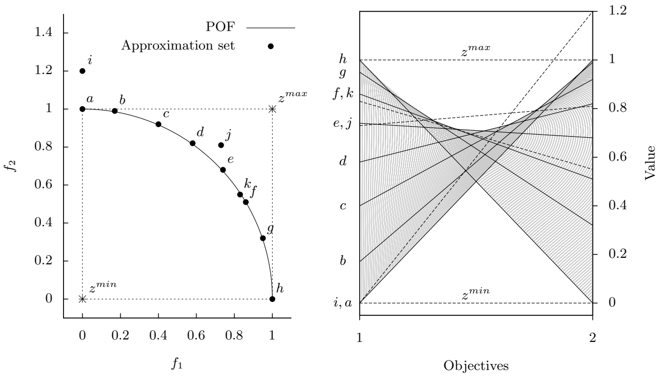


Figure 1: An example of approximation set in two-dimensional objective space (left) and its corresponding Parallel-Coordinates graph (right).

Design (SLD) [17]), the cardinality of this set becomes very large as the number of objectives increases.

In this paper, we address this issue and propose a novel MOEA that incorporates a density estimator based on the Parallel Coordinates [11] (also called Value-Path [14]). This technique has been frequently used for visualizing results in multi-objective optimization, specially in high-dimensional spaces. Its applicability varies from the identification of differences and similarities between alternatives, guidance in selecting solutions, monitoring the progress of an optimization run, assessment of the relative performance of different algorithms [14, 18] to, more recently, guidance during the search in bio-inspired meta-heuristics [9].

Value-Path is built in the 2-dimensional plane, where m copies of the real line $\mathbb{R}$ are placed perpendicular to the x-axis and a point in $\mathbb{R}^m$ is represented by a polygonal line[3] with vertices on the parallel axes. In Figure 1, we show an example of this graph with its corresponding Pareto front, composed of eleven non-dominated solutions. The basic idea of our proposal is based on the following observations:

1. The POF is represented by the shaded area, which we named *trade-off area.*

2. The boundaries between the white regions and the trade-off area, which we call *Pareto coordinates*, give us a hint of the shape of the POF.

3. As the number of uniformly distributed solutions in the approximation set increases, the coverage of the trade-off area becomes better.

4. Those individuals intersecting the upper-Pareto coordinate are beyond the POF.

Therefore, the 2D-graphs of each distinct pair of objective functions are transformed into a digital image[4] and this information is extracted in order to assign a contribution to each individual. The aim of this paper is to propose a density estimator that can handle many-objective problems at an affordable computational cost. To the best of our knowledge, this is the first MOEA that incorporates automatic image analysis [8] in its search mechanism.

The remainder of this paper is organized as follows. In Section 2, we briefly review some previous related work. Section 3 is devoted to the description of our proposed algorithm. In Section 4, we present a comparative study using the Zitzler-Deb-Thiele (ZDT) [20] and the Walking-Fish-Group (WFG) [10] test suites. Finally, Section 5 provides our conclusions and some paths of future work.

2. PREVIOUS RELATED WORK

In this section, we briefly review some optimizers that have a selection mechanism based on Pareto dominance plus a density estimator that is applied in objective space. A more comprehensive review can be found in [4, 13].

The Strength Pareto Evolutionary Algorithm 2 (SPEA2) [21] assigns a fitness value to each individual by considering its density, the number of solutions that it dominates as well as the number of solutions that dominate it. Such fitness takes part in the processes of parent and survival selection. The density is based on the clustering technique of the k^{th} nearest neighboring method, having a computational complexity of $\mathcal{O}(|P|^2(\log|P|+m))$.[5] In addition, an archive of non-dominated solutions computed so far is maintained within a (pre-defined) limit, and periodically combined with the current population.

The Nondominated Sorting Genetic Algorithm II (NSGA-II) [7], ranks an individual according to the number of solutions that dominate it. Thus, non-dominated members obtain the first rank. Next, to discern between two individuals of the same hierarchy, the one that is located in a less crowded region is preferred. In this case, crowding is calculated as the average distance of the two nearest neighbors that surround the particular solution corresponding to each objective, and it has a complexity of order $\mathcal{O}(|P|m\log|P|)$. The rank and the crowding distance induce a total order, which is used in both parent and survival selection schemes.

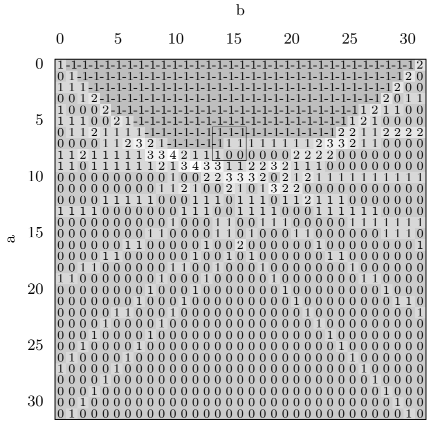
NSGA-III [6] is an improved version of its predecessor that is able to deal with many-objective problems. Thus, the crowding distance is replaced by a niching strategy (of complexity $\mathcal{O}(|P|^2m + m^3)$), that requires a set of well-spread reference points, which can be supplied by the user or generated by the Simplex-Lattice Design (SLD) method [17]. The population is normalized and associated with the lines passing through the reference points and the origin. Those individuals having the closest perpendicular distance to segregated lines are chosen for the next generation. Unlike NSGA-II, the parent selection is conducted by random sampling.

Another interesting proposal is the Hypervolume Estimation Algorithm (HypE) [2], which ranks the population as NSGA-II, and uses a density estimator based on the individual contributions to the hypervolume indicator. When the number of objectives is small (≤ 3), the exact hypervolume is calculated in $\mathcal{O}(|P|^m + m|P|\log|P|)$; otherwise, it is estimated in $\mathcal{O}(|P|ms)$ using Monte Carlo sampling, where a predefined number of samples (s) of objective vectors are randomly drawn and the portion of objective vectors that

[3] A polygonal line or polyline is a connected series of line segments.

[4] The term *image* refers to a two dimensional light intensity function $g(a, b)$ where a and b denote spatial coordinates and the value of g at any point (a, b) is proportional to the brightness (or gray level) of the image at that point. A *digital image* is an image that has been discretized in both its spatial coordinates and brightness [8].

[5] $|P|$ denotes the population size.



	0					5					10					15					20					25					30		
0	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2
	0	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	0
	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	0	0
	0	0	1	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	0	1	1
	1	0	0	0	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	2	1	1	0	0
5	1	1	1	0	0	2	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	2	1	0	0	0	0
	0	1	1	2	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	2	1	1	2	2	2	2
	0	0	0	0	1	1	2	3	2	1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	2	3	3	2	1	1	0	0	0	0
	1	1	2	1	1	1	1	1	3	3	4	2	1	1	1	0	0	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
	1	1	0	1	1	1	1	1	1	2	1	3	4	3	3	1	1	2	2	3	2	1	1	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	2	2	3	3	3	2	0	2	1	2	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	1	1	2	1	0	0	2	1	0	1	3	2	2	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	0	1	1	2	1	1	1	0	0	0	0	0	0	0
	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1
15	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0
20	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0
	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
25	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
30	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Figure 2: Digital image of the Value Path of Fig. 1.

are dominated by a specific solution represents an estimate for the hypervolume contribution of this solution. This approach is considered in the two selection schemes.

Hu and Yen [9] have been the only ones to propose density estimators based on Value Path, embedding them with a multi-objective particle swarm algorithm. The basic principle is that non-dominated solutions are ranked according to the height of intersection between the parallel axes and the polylines. In contrast with our proposal, this algorithm is sensitive to objective ordering.

3. OUR PROPOSED APPROACH

In this section, we describe a new MOEA called Multi-objective Optimizer based on Value Path (MOVAP), which uses Pareto dominance as its primary search engine and a density estimator based on Parallel Coordinates. The latter is applied to parent and survival selection.

The core idea is to create a digital image containing the Parallel Coordinates of each distinct pair of objective functions (see Algorithm 1). These sub-graphs are attached next to each other and only normalized individuals are considered (line 5). In this case, the digital image is represented as a matrix, where its dimensions are fixed, depending on the number of objectives (m), the population size ($|P|$) and the resolution parameter (γ) (lines 1-4). An element of this array (*pixel*) corresponds to the number of intersecting polylines (line 15) and the region above the upper-Pareto coordinate is identified with the value minus one (lines 17-21). Therefore, the gray levels oscillate in the range $[-1, |P|]$.

In Algorithm 1, we adopt the notation $p.\vec{y}$ to refer to the objective vector of an individual p. The total number of sub-graphs is given in line 1, and the equation of a line segment is given in line 11. In the worst case, the computational complexity of this algorithm is $\mathcal{O}(|P|^2m^2)$. Furthermore, in Figure 2, we show an example of a digital image using a resolution parameter $\gamma = 3$.

The next step is to determine the density of an individual (see Algorithm 2). For this purpose, all pixels, as well as their boundaries of a polyline are inspected (lines 13-27). The boundary of a pixel q is limited to the eight points that are at a unit distance from it, denoted by $N_8(q)$ (see, e.g., coordinate $(15, 7)$ from Figure 2). This set of neighbors must fall inside the current sub-graph (denoted by c).

During the inspection process, the number of edged, empty and filled pixels are counted (lines 14-17). The overall density is updated according to their values, in such a way that highly dense points (lines 18 and 27) or those intersecting the upper-Pareto coordinate (lines 19 and 20) are penalized, whereas isolated ones (line 22 and 25), belonging to the trade-off area, are rewarded.

Algorithm 1 Build PC-image

```
Require: Population $P$, resolution $\gamma$, objectives $m$
Ensure: Image $I_{A\times B}$, parameter $\theta$
 1: $s \leftarrow m(m-1)/2$
 2: $\theta \leftarrow \gamma|P|/s$
 3: $A \leftarrow \theta, B \leftarrow \theta s$
 4: $I \leftarrow \mathbf{0}_{A\times B}$
 5: for all $\{p \in P : p.\vec{y} \in [0,1]^m\}$ do
 6:     $c \leftarrow 0$
 7:     for all $i \in \{1,\ldots,m-1\}$ do
 8:         for all $j \in \{i+1,\ldots,m\}$ do
 9:             $m \leftarrow \frac{P.y_j - P.y_i}{\theta - 1}$
10:             for all $k \in \{0,\ldots,\theta-1\}$ do
11:                 $y \leftarrow mk + P.y_i$
12:                 $a \leftarrow \lfloor (A-1)(1-y) \rceil$
13:                 $b \leftarrow \theta c + k$
14:                 if $a \in [0, A)$ then
15:                     $I[a,b] \leftarrow I[a,b] + 1$
16:             $c \leftarrow c + 1$
17: for all $b \in \{0,\ldots,B\}$ do
18:     $a \leftarrow 0$
19:     while $a < A$ and $I[a,b] = 0$ do
20:         $I[a,b] \leftarrow -1$
21:         $a \leftarrow a + 1$
22: return $I$
```

Once the matrix has been processed, the population density is normalized (line 31). Thus, extreme solutions[6] obtain zero (the best) value and the most crowded gets one (the worst value). The density of individuals lying outside the interval $[0, 1]$ is equal to the Euclidean norm of the objective vector. The complexity of this procedure is of order $\mathcal{O}(|P|^2m^2)$.

The proposed density estimator is coupled with a steady state genetic algorithm, where one offspring is created at each generation. In the following paragraph, we describe the main loop of MOVAP, given by Algorithm 3.

First, the population is initialized, either randomly or from a previously computed solution (line 3). At each iteration, a binary tournament selection is performed, based on the population density (line 5). Thus, isolated individuals have a high probability of mating. Next, an offspring is created using the variation operators (line 6). The new individual is added to the current population and a normalization procedure (explained later on) is invoked (lines 7 and 8). If there are individuals outside the region of interest, the one with the highest norm is removed from the population. Otherwise, the population is ranked using the non-dominated sorting procedure of NSGA-II [7] (lines 10-13). If the last front consists of one individual (lines 14 and 15), then it is eliminated. Otherwise, the digital image is built for calculating the density estimator, and the individual with the highest value is discarded (lines 17-19).

[6] The set of all m points in $\mathbb{R}^m$ that yield the best value of one objective function with the lowest norm.

```
Algorithm 2 Calculate population density
Require: Population $P$, image $I_{A \times B}$, parameter $\theta$, extreme solutions $E$, objectives $m$
Ensure: Density values $D_{|P| \times 1}$
$D \leftarrow \mathbf{0}_{|P| \times 1}$
$v_{min} \leftarrow \infty$, $v_{max} \leftarrow -\infty$
for all $\{p \in (P \setminus E) : p.\vec{y} \in [0,1]^m\}$ do
    $c \leftarrow 0$
    for all $i \in \{1, \ldots, m-1\}$ do
        for all $j \in \{i+1, \ldots, m\}$ do
            $m \leftarrow \frac{P.y_j - P.y_i}{\theta - 1}$
            for all $k \in \{0, \ldots, \theta - 1\}$ do
                $y \leftarrow mk + P.y_i$
                $a \leftarrow \lfloor (A-1)(1-y) \rceil$
                $b \leftarrow \theta c + k$
                if $a \in [0, A)$ then
                    $S \leftarrow N_8(I[a,b])$
                    $n_{edge} \leftarrow \mid \{s \in S \colon s = -1\} \mid$
                    $n_{empty} \leftarrow \mid \{s \in S \colon s = 0\} \mid$
                    $n_{filled} \leftarrow 8 - n_{edge} - n_{empty}$
                    $n_{sum} \leftarrow \sum_{s \in S, s > 0} s$
                    $D[p] \leftarrow D[p] + (A - a) I[a,b]$
                    if $n_{empty} > 0$ and $n_{edge} > 0$ then
                        $D[p] \leftarrow D[p] + (a+1) n_{edge} n_{empty}$
                    else
                        $D[p] \leftarrow D[p] - (a+1)(n_{edge} + n_{empty})$
                    if $n_{filled} > 0$ then
                        if $m > 2$ and $(b \mod 2) = 0$ then
                            $D[p] \leftarrow D[p] - (A - a) n_{filled}$
                        else
                            $D[p] \leftarrow D[p] + (A - a) n_{sum} / n_{filled}$
            $c \leftarrow c + 1$
    $v_{min} \leftarrow \min\{v_{min}, D[p] - 1\}$
    $v_{max} \leftarrow \max\{v_{max}, D[p]\}$
Normalize $\forall p \in (P \setminus E)$:
    $$D[p] \leftarrow \begin{cases} \frac{D[p] - v_{min}}{v_{max} - v_{min}} & \text{if } p \in [0,1]^m \\ \|p.\vec{y}\| & \text{otherwise} \end{cases}$$
return $D$
```

```
Algorithm 3 Main loop of MOVAP
Require: MOP, stopping criterion, image resolution $\gamma$
Ensure: Approximation set $P$
$i \leftarrow 1$
$D \leftarrow \mathbf{0}_{|P| \times 1}$
Initialize population $P^i$
while termination condition is not fulfilled do
    Perform parent selection
    Create an offspring $o$
    $P^i \leftarrow P^i \cup \{o\}$
    $[P', E] \leftarrow$ Normalize objectives $(P^i, m)$
    $Q \leftarrow \{p \in P' : p.\vec{y} \notin [0,1]^m\}$
    if $Q \neq \emptyset$ then
        $r \leftarrow \arg\max_{\vec{q} \in Q} \|q.\vec{y}\|$
    else
        $\{F_1, \ldots, F_k\} \leftarrow$ Non-dominated sorting$(P^i)$
        if $|F_k| = 1$ then
            $r \leftarrow F_k$
        else
            $[I, \theta] \leftarrow$ Build PC-image $(P', \gamma, m)$
            $D \leftarrow$ Calculate pop. density $(P', I, \theta, E, m)$
            $r \leftarrow \arg\max_{p \in F_k} D[p]$
    Reduce population $P^{i+1} \leftarrow P^i \setminus \{r\}$
    $i \leftarrow$i + 1
return $P^i$
```

```
Algorithm 4 Normalize objectives
Require: Population $P$, objectives $m$
Ensure: Population $P'$, extreme points $E$
 1: $v_i \leftarrow \mid \min(\{0\} \cup \{p.y_i : p \in P\}) \mid$, $\forall i \in \{1, \ldots, m\}$
 2: if $\vec{v} \neq \vec{0}$ then
 3:     $P' \leftarrow \{p.\vec{y} + \vec{v} : p \in P\}$
 4: Update the minimum reference point $\vec{z}^{min}$
 5: for all $i \in \{1, \ldots, m\}$ do
 6:     $e \leftarrow \arg\min_{p \in P'} \frac{p.y_i}{\|p.\vec{y}\|}$
 7:     $E \leftarrow E \cup \{e\}$
 8:     $z_i^{max} \leftarrow e_i$ {see Figure 1}
 9: $p.\vec{y} \leftarrow \frac{p.\vec{y} - \vec{z}^{min}}{\vec{z}^{max} - \vec{z}^{min}}$, $\forall p \in P'$
10: return $P'$, $E$
```

The normalization done in Algorithm 4 works in objective space and serves for three purposes: it translates vectors to the origin, if their coordinates are negative (lines 1-3); it finds the extreme points (lines 5-8); and it normalizes the population (line 9). It is noteworthy that if there are several candidates parallel to one axis, the solution with the lowest norm is preferred. Additionally, each component of $\vec{z}^{min}$ corresponds to the best found objective value (line 4).

In the following, we determine the complexity of this approach. Parent selection is performed in $\mathcal{O}(1)$, as well as the offspring generation and population reduction. The normalization and verification of individuals inside the region of interest is done in $\mathcal{O}(|P|m)$, each. The nondominated sorting is of $\mathcal{O}(|P|^2 m)$. As seen before, the building of the image and the density calculation can be performed in $\mathcal{O}(|P|^2 m^2)$. Therefore, the overall complexity of MOVAP at each iteration is $\mathcal{O}(|P|^2 m^2)$ with a maximum storage of $\mathcal{O}(|P|^2)$.

Re-taking our previous example of Figure 1, we provide the density values of the eleven non-dominated individuals in Table 1. The best individuals are the extreme solutions a and h, whereas the three candidates to be removed are i, j and k. This is because they are outside the region of interest, beyond the Pareto Front or too close to other individuals. Moreover, the special pattern of solution j, where it intersects the upper-Pareto coordinate, is recognized in the bounded pixels of Figure 2. Finally, the concave geometry of the approximation set can be appreciated as a valley-like curve in the upper-Pareto coordinate. Here, it is worth mentioning that when the geometry is convex, the lower-Pareto coordinate takes a mountain-like curved shape, while when it is linear, the upper-Pareto coordinate forms a triangle. In high dimensionality, if the same pattern is repeated for each pair of objectives, we can deduce that the Pareto front adopts such form. Otherwise, it corresponds to a mixed shape.

4. EXPERIMENTAL RESULTS

In this section, we investigate the effectiveness of MOVAP, not only in many-objective artificial problems (of 5D and 7D), but also in low dimensionality. For this purpose, we present a comparative study that includes the algorithms

Table 1: Example data

Solution	$\vec{F}$	Density
a	(1e-12, 1.00)	0.0000
b	(0.17, 0.99)	0.0003
c	(0.40, 0.92)	0.2491
d	(0.58, 0.82)	0.4747
e	(0.74, 0.68)	0.4753
f	(0.86, 0.51)	0.6107
g	(0.95, 0.32)	0.2067
h	(1.00, 0.00)	0.0000
i	(0.00, 1.20)	1.2000
j	(0.73, 0.81)	1.0000
k	(0.83, 0.55)	0.6819

Table 2: Parameters adopted in our study

m	WFG		$\lvert P\rvert$	NSGA-III	MOVAP
	n	k		SLD	γ
2	24	4	100	99	3
3	24	4	120	14	2
5	47	8	196	4,5	2
7	71	12	210	4,5	2

SPEA2, NSGA-II, NSGA-III and HypE; all which were developed in EMO Project, our framework for Evolutionary Multi-Objective Optimization. Next, we describe the test problems, experimental settings and the performance measure adopted, closing with the discussion of the results.

For two objectives, we select the five real-valued problems of the Zitzler-Deb-Thiele (ZDT) set [20] and for larger dimensionality, we use the Walking-Fish-Group (WFG) [10] benchmark. In all the cases, the optimum is known and their Pareto front geometries are diverse (e.g., convex, concave, linear, degenerated, disconnected, etc.). In the WFG benchmark, properties such as non-separability, multi-modality, deceptiveness and bias, are preserved as we increase the number of objective functions, making them harder to solve for MOEAs.

The decision variables (n) for ZDT1-3 was set to 30 and for ZDT4,6 was set to 10. In the case of WFG, the variables and position-related parameter (k) are specified in Table 2. The number of function evaluations was set to 40,000 and 50,000 for the ZDT and WFG test problems, respectively.

All the MOEAs were implemented using real-numbers encoding and their parameters were identical (see Table 2). The variation operators were polynomial-based mutation and simulated binary crossover (SBX) [5] in all algorithms. As suggested in [6], the crossover rate and its distribution index were set to 0.9 and 20, for 2 and 3 objectives, and 1.0 and 30 for many-objective problems. The mutation rate and its distributed index was set to $1/n$ and 20, respectively. For NSGA-III, the set of weight vectors was generated using the SLD method [17]. For HypE, the number of sampling points was fixed to 20,000.

In MOVAP, the image resolution was empirically determined, being independent of the problem to be solved. For this purpose, we correlated its behavior with the hypervolume indicator, and we found the average optimum values of Table 2. We observed that as this value increases, the overlapping level among polylines is minimum, and even though the individuals are not well distributed, the density estimator reflects the contrary.

For comparing results, we adopted the hypervolume indicator,[7] because it rewards both convergence towards the Pareto front as well as the maximum spread of the solutions obtained. The reference points used were $(1.1, \ldots)$ for the ZDT test suite, $(3, 5, 7, \ldots)$ for the instances WFG1 and WFG3; and $(2.2, 4.2, 6.2, \ldots)$ for the rest of the problems. We also used the Value Path for inspecting diversity.

We performed 30 independent runs of each of the five MOEAs compared on all the test instances adopted. With the aim of comparing the performance of all algorithms among themselves in a pairwise fashion, the Wilcoxon rank sum test (one-tailed) with the Bonferroni correction [15] was applied to the hypervolume indicator values.

Experimental results appear in Table 3 and some examples of the approximation sets, corresponding to the median values, are depicted in Figures 3 and 4. For comparison purposes in the ZDT benchmark, the approximation sets are plotted with a vertical shift.

With seven and five objectives, the best algorithm was MOVAP, obtaining the highest hypervolume values and significantly outperforming the other algorithms: HypE, NSGA-III, NSGA-II and SPEA2. Only in WFG2 (a problem with disconnected geometry) for seven objectives MOVAP was second, without being significantly surpassed by NSGA-II. The second best optimizer was NSGA-III, which was able to get very close to the Pareto optimal fronts. However, it produced very poor diversity. On the other hand, HypE encouraged spread over distribution, being unable to cover the complete Pareto front. NSGA-II and SPEA2, in general, experimented some stagnation during the search, being incapable of reaching the optimal solution.

In three and two objectives, MOVAP ranked second, producing similar results to HypE (which ranked first) and significantly outperforming SPEA2, NSGA-II and NSGA-III in almost all cases. This behavior was expected, since HypE is using the exact hypervolume in its search mechanism and we are using the same performance indicator for comparing results. Nonetheless, we found that HypE focused more on the spread than on the distribution of solutions, while MOVAP favored distribution over spread. For example, in ZDT2, which has a concave Pareto front, MOVAP was able to find good representatives near the extreme points (see Figure 3). Moreover in WFG8, a non-separable and biased problem, for three objectives, MOVAP outperformed HypE. Only in WFG2 MOVAP could not perform significantly better than the other algorithms. With respect to NSGA-III, it produced a more uniform distribution than SPEA2 and NSGA-II, obtaining the best results in WFG9 (a multi-modal, deceptive, non-separable and biased problem) for three objectives. Finally, SPEA2 obtained slightly better results than NSGA-II, standing out by the diversity of its solutions.

In summary, we observed that MOVAP produced much better solutions near the Pareto optimal front than NSGA-II and SPEA2 in low and high dimensionality. With respect to NSGA-III it had better diversity and in comparison with HypE, MOVAP was competitive in low dimensionality and produced much better results for five and seven objectives. For this reason, we believe that our proposed approach is a promising alternative for solving MOPs, in both low and high dimensionality.

[7] The hypervolume is equal to the sum of all the rectangular areas of a non-dominated set (A), bounded by some reference point.

Table 3: Median and standard deviation of the hypervolume indicator for 2, 3, 5 and 7 objectives. In each case, the outperformance relation among algorithms is shown, using a significance level of $\alpha = 0.5$ (for example, SPEA2 performs significantly better than HypE on WFG1). The two best values are shown in gray scale, where a darker tone corresponds to the best value.

m	Problem	SPEA2 (1)	NSGA-II (2)	NSGA-III (3)	HypE (4)	MOVAP (5)
7	WFG1	6.18e+05(7.97e+3) 4	6.49e+05(9.21e+3) 1,4	6.71e+05(1.33e+4) 1,2,4	5.32e+05(1.96e+4) –	8.36e+05(1.64e+4) 1,2,3,4
	WFG2	7.88e+05(7.98e+3) 4	8.11e+05(5.74e+3) 1,3,4	7.96e+05(4.65e+4) 4	7.24e+05(6.64e+4) –	8.09e+05(6.77e+3) 1,3,4
	WFG3	1.06e+06(8.32e+4) –	1.32e+06(1.87e+4) 1,3,4	1.05e+06(3.21e+4) –	1.13e+06(3.48e+4) 1,3	1.42e+06(6.50e+3) 1,2,3,4
	WFG4	4.65e+05(1.89e+4) 2,4	3.15e+05(1.69e+4) 4	5.72e+05(8.12e+3) 1,2,4	2.76e+05(1.59e+4) –	6.52e+05(5.12e+3) 1,2,3,4
	WFG5	4.31e+05(1.79e+4) 2,4	3.25e+05(1.85e+4) 4	5.52e+05(5.52e+3) 1,2,4	2.94e+05(1.60e+4) –	6.14e+05(3.70e+3) 1,2,3,4
	WFG6	4.24e+05(2.17e+4) 2,4	3.34e+05(2.26e+4) 4	5.64e+05(1.58e+4) 1,2,4	2.68e+05(2.64e+4) –	5.99e+05(1.40e+4) 1,2,3,4
	WFG7	3.68e+05(2.36e+4) 2,4	3.25e+05(1.91e+4) 4	5.99e+05(8.52e+3) 1,2,4	2.94e+05(1.63e+4) –	6.65e+05(4.40e+3) 1,2,3,4
	WFG8	3.42e+05(2.34e+4) 2,4	3.23e+05(1.48e+4) 4	4.69e+05(4.10e+4) 1,2,4	2.42e+05(1.72e+4) –	5.05e+05(4.43e+3) 1,2,3,4
	WFG9	4.14e+05(1.87e+4) 2,4	2.78e+05(1.78e+4) 4	4.71e+05(2.52e+4) 1,2,4	2.32e+05(1.74e+4) –	4.82e+05(1.81e+4) 1,2,4
5	WFG1	3.60e+03(8.17e+1) 4	3.71e+03(7.79e+1) 1,3,4	3.67e+03(4.94e+1) 1,4	2.82e+03(1.17e+2) –	4.57e+03(1.07e+2) 1,2,3,4
	WFG2	4.61e+03(2.56e+2) 4	4.63e+03(2.19e+2) 4	4.61e+03(2.26e+2) 4	4.24e+03(3.00e+2) –	4.64e+03(3.78e+2) 4
	WFG3	6.19e+03(1.65e+2) 4	6.93e+03(1.11e+2) 1,3,4	6.33e+03(1.02e+2) 4	5.55e+03(1.55e+2) –	7.34e+03(5.36e+1) 1,2,3,4
	WFG4	2.72e+03(6.26e+1) 2,4	2.34e+03(8.39e+1) 4	3.11e+03(4.16e+1) 1,2,4	1.69e+03(9.10e+1) –	3.38e+03(2.12e+1) 1,2,3,4
	WFG5	2.63e+03(5.72e+1) 2,4	2.39e+03(7.30e+1) 4	2.94e+03(2.33e+1) 1,2,4	1.96e+03(1.33e+2) –	3.19e+03(1.31e+1) 1,2,3,4
	WFG6	2.55e+03(5.85e+1) 2,4	2.21e+03(8.35e+1) 4	2.97e+03(5.63e+1) 1,2,4	1.80e+03(1.36e+2) –	3.21e+03(4.93e+1) 1,2,3,4
	WFG7	2.44e+03(6.60e+1) 2,4	2.19e+03(1.34e+2) 4	3.22e+03(2.94e+1) 1,2,4	1.82e+03(1.10e+2) –	3.45e+03(1.62e+1) 1,2,3,4
	WFG8	2.03e+03(7.13e+1) 2,4	1.88e+03(4.94e+1) 4	2.39e+03(4.80e+1) 1,2,4	1.48e+03(1.24e+2) –	2.66e+03(2.56e+1) 1,2,3,4
	WFG9	2.47e+03(8.43e+1) 2,4	2.09e+03(1.17e+2) 4	2.63e+03(1.21e+2) 1,2,4	1.75e+03(1.68e+2) –	2.75e+03(5.47e+1) 1,2,3,4
3	WFG1	4.75e+01(2.65e+0) 3	4.41e+01(2.12e+0) 3	4.22e+01(2.93e+0) –	5.66e+01(1.62e+0) 1,2,3,5	5.25e+01(2.32e+0) 1,2,3
	WFG2	5.20e+01(3.62e+0) 2,3	5.15e+01(3.73e+0) –	5.14e+01(4.02e+0) –	5.34e+01(4.21e+0) 1,2,3,5	4.43e+01(4.13e+0) –
	WFG3	7.28e+01(3.22e-1) –	7.45e+01(3.61e-1) 1,3	7.28e+01(3.81e-1) –	7.59e+01(2.19e-1) 1,2,3,5	7.51e+01(3.71e-1) 1,2,3
	WFG4	2.67e+01(2.41e-1) 2	2.51e+01(4.26e-1) –	2.78e+01(1.09e-1) 1,2	2.97e+01(4.66e-2) 1,2,3,5	2.88e+01(1.13e-1) 1,2,3
	WFG5	2.54e+01(2.03e-1) 2	2.42e+01(2.93e-1) –	2.59e+01(1.09e-1) 1,2	2.74e+01(9.65e-1) 1,2,3,5	2.68e+01(1.25e-1) 1,2,3
	WFG6	2.51e+01(3.69e-1) 2	2.39e+01(4.60e-1) –	2.59e+01(3.08e-1) 1,2	2.77e+01(2.68e-1) 1,2,3,5	2.71e+01(3.16e-1) 1,2,3
	WFG7	2.74e+01(2.53e-1) 2	2.62e+01(3.61e-1) –	2.84e+01(7.93e-2) 1,2	2.98e+01(2.01e-2) 1,2,3,5	2.92e+01(3.78e-2) 1,2,3
	WFG8	2.21e+01(2.03e-1) 2	2.11e+01(2.94e-1) –	2.30e+01(1.60e-1) 1,2	2.34e+01(2.81e-1) 1,2,3	2.40e+01(1.22e-1) 1,2,3,4
	WFG9	2.36e+01(1.02e+0) 2,4	2.25e+01(8.25e-1) 4	2.58e+01(1.33e+0) 2,4	2.17e+01(1.75e+0) –	2.40e+01(1.44e+0) 2,4
2	ZDT1	8.71e-01(3.10e-4) 2	8.70e-01(5.70e-4) –	8.71e-01(3.36e-5) 1,2	8.72e-01(2.79e-5) 1,2,3,5	8.72e-01(8.58e-5) 1,2,3
	ZDT2	5.38e-01(5.40e-4) 2	5.37e-01(4.66e-4) –	5.38e-01(5.15e-5) 1,2	5.39e-01(2.59e-5) 1,2,3,5	5.38e-01(3.69e-5) 1,2,3
	ZDT3	1.33e+00(9.72e-4) 3	1.33e+00(1.18e-3) 3	1.33e+00(4.01e-4) –	1.33e+00(1.52e-2) 1,2,3,5	1.33e+00(3.39e-2) 1,2,3
	ZDT4	8.68e-01(1.94e-3) –	8.68e-01(1.35e-3) –	8.69e-01(1.64e-3) –	8.71e-01(5.63e-4) 1,2,3,5	8.70e-01(2.03e-3) –
	ZDT6	4.99e-01(1.14e-3) 3	4.99e-01(1.14e-3) 3	4.97e-01(1.59e-3) –	5.03e-01(3.01e-4) 1,2,3,5	5.03e-01(3.12e-4) 1,2,3

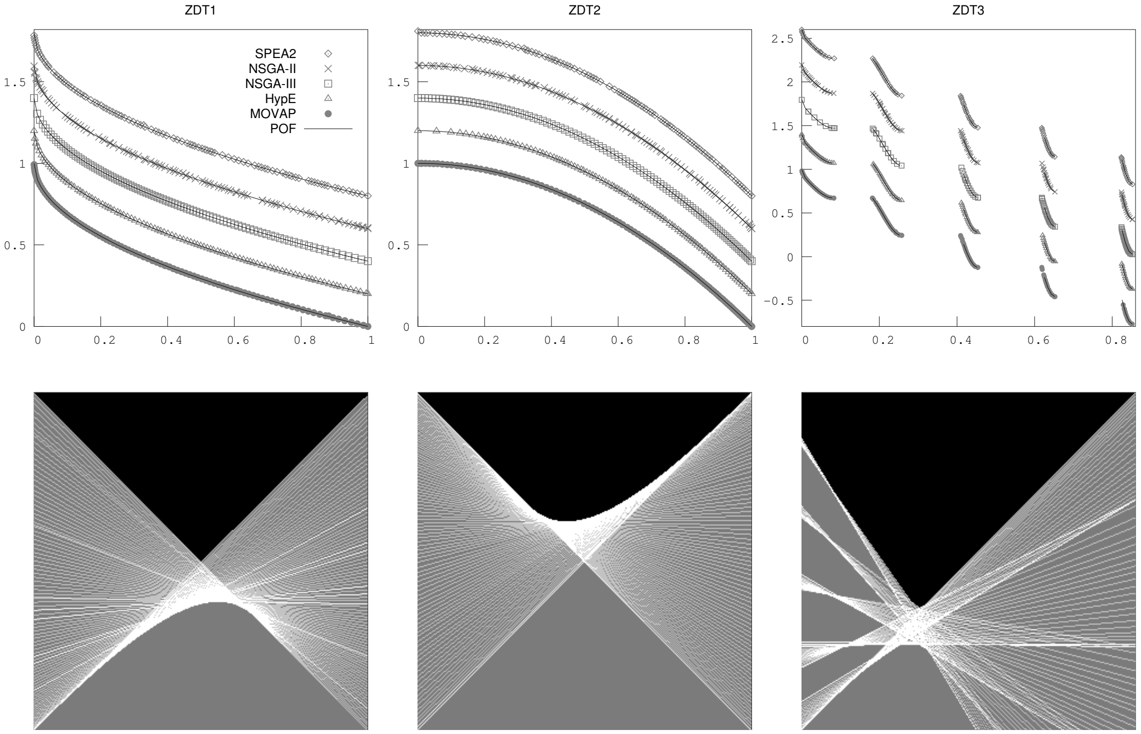


Figure 3: Pareto fronts produced by MOEAs on some problems of the ZDT test suite (top) and the corresponding digital images generated by MOVAP (bottom).

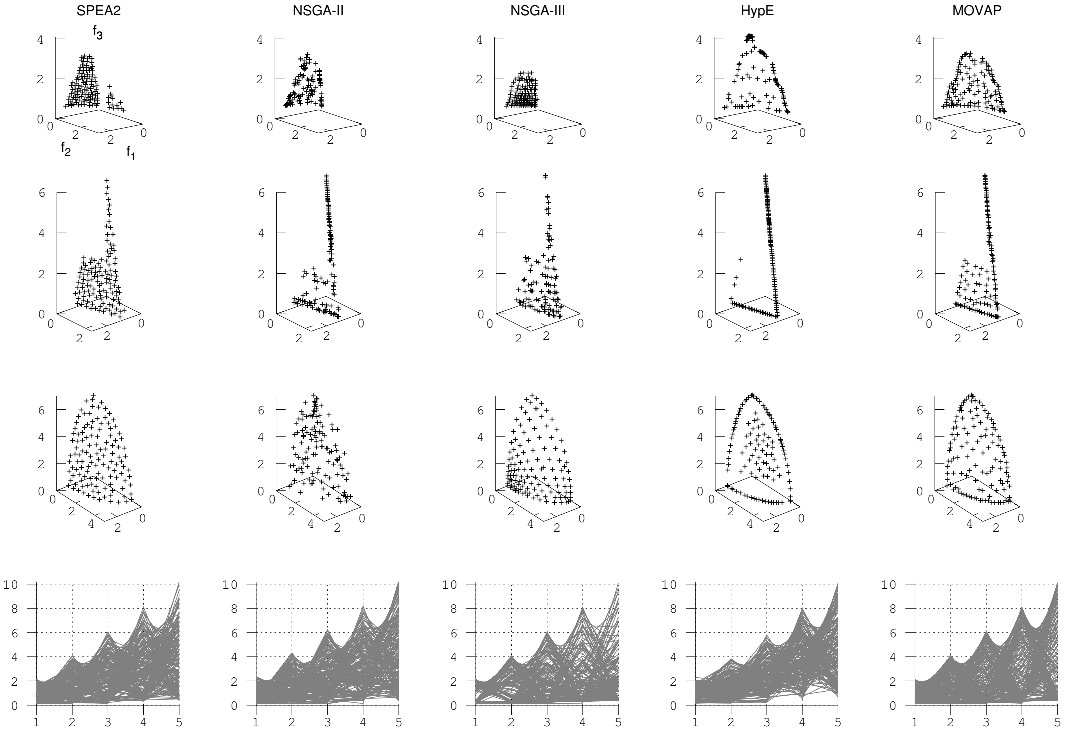


Figure 4: Approximation sets of MOEAs in WFG1, WFG3, WFG4 and WFG9 (from top to bottom).

5. CONCLUSIONS AND FUTURE WORK

This paper presents a new algorithm, called MOVAP (Multi-objective Optimizer based on Value Path), which uses image analysis concepts in its selection mechanism. The basic idea consists in discretizing the Parallel Coordinates graph and assigning a fitness value to each individual based on the density of its polylines. Experimental results indicate that the proposed approach significantly outperforms HypE, NSGA-III, NSGA-II and SPEA2 in more than 35% of the test instances, producing much better diversity of solutions, and exploring more regions of the search space in high-dimensionality than the MOEAs with respect to which it was compared. Whereas in low dimensionality, our proposed approach was competitive, producing very similar results to those generated by HypE. Moreover, the complexity of MOVAP is quadratic with respect to the number of objectives and the population size. Based on these preliminary results, we believe that our proposed approach is a suitable alternative for solving many-objective problems. As part of our future work, we are interested in studying the scalability of MOVAP beyond seven objectives and studying the properties of the proposed density estimator. Finally, although MOVAP does not require a set of reference points as NSGA-III, or a large number of sampling points as HypE, it is worth indicating that it needs a resolution parameter which, however, could be tuned during execution time.

Acknowledgments

The first author acknowledges support from CONACyT and CINVESTAV-IPN to pursue graduate studies in Computer Science. The second author gratefully acknowledges support from CONACyT project no. 221551. The third author's work is partially funded by the Spanish MINECO and FEDER project TIN2014-57341-R.[8]

6. REFERENCES

[1] S. F. Adra and P. J. Fleming. Diversity Management in Evolutionary Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):183–195, April 2011.

[2] J. Bader and E. Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76, Spring, 2011.

[3] K. Bringmann and T. Friedrich. Don't be Greedy when Calculating Hypervolume Contributions. In *FOGA '09: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 103–112, Orlando, Florida, USA, January 2009. ACM.

[4] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.

[5] K. Deb and R. B. Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9:115–148, 1995.

[6] K. Deb and H. Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, Aug 2014.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[8] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[9] W. Hu and G. G. Yen. Adaptive Multiobjective Particle Swarm Optimization Based on Parallel Cell Coordinate System. *IEEE Transactions on Evolutionary Computation*, 19(1):1–18, Feb 2015.

[10] S. Huband, P. Hingston, L. Barone, and L. While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, Oct 2006.

[11] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag New York, Secaucus, NJ, USA, 2009.

[12] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation, 2008. CEC 2008.*, pages 2419–2426, June 2008.

[13] C. Lücken, B. Barán, and C. Brizuela. A Survey on Multi-Objective Evolutionary Algorithms for Many-Objective Problems. *Computational Optimization and Applications*, 58(3):707–756, 2014.

[14] K. Miettinen. Survey of Methods to Visualize Alternatives in Multiple Criteria Decision Making Problems. *OR Spectr.*, 36(1):3–37, jan 2014.

[15] R. G. Miller. *Simultaneous Statistical Inference*. Springer-Verlag, New York, 1991.

[16] R. C. Purshouse and P. J. Fleming. On the Evolutionary Optimization of Many Conflicting Objectives. *IEEE Transactions on Evolutionary Algorithms*, 11(6):770–784, December 2007.

[17] H. Scheffé. Experiments with Mixtures. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 20:344–360, 1958.

[18] T. Tusar and B. Filipic. Visualization of Pareto Front Approximations in Evolutionary Multiobjective Optimization: A Critical Review and the Prosection Method. *IEEE Transactions on Evolutionary Computation*, 19(2):225–245, April 2015.

[19] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[20] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput.*, 8(2):173–195, jun 2000.

[21] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.

[8] http://moveon.lcc.uma.es