# A Novel Local Search Mechanism Based on the Reflected Ray Tracing Method Coupled to MOEA/D

Miriam Pescador-Rojas
CINVESTAV-IPN,
Computer Science Department
Av. IPN 2508, C. P. 07360,
Col. San Pedro Zacatenco,
Mexico City
Email: pescador@computacion.cs.cinvestav.mx

Carlos A. Coello-Coello
CINVESTAV-IPN,
Computer Science Department,
Av. IPN 2508, C. P. 07360,
Col. San Pedro Zacatenco,
Mexico City
Email: ccoello@cs.cinvestav.mx

***Abstract*—In recent years, the use of decomposition methods has become a popular choice to solve complex multi-objective optimization problems. One possible approach to enhance the performance of a decomposition method is to couple it to a local search engine that speeds up convergence to the true Pareto front of a problem. Here, we propose a new local search strategy for continuous search spaces, which is based on *the law of reflection of light*. Our proposed approach computes reflected rays in order to explore and exploit promising subregions of the search space. The local search engine uses information of the Penalty Boundary Intersection method to create a scene with three elements: 1) a light source defined by the solution which is farthest from the ideal point, 2) a hyperspherical structure built by non-dominated neighboring solutions and 3) a set of reflected rays located in decision variable space. Our proposed local search engine is coupled to MOEA/D (a popular decomposition-based multi-objective optimizer) and is able to outperform three state-of-the-art multi-objective evolutionary algorithms, particularly in multifrontal problems.**

## I. Introduction

Multi-objective optimization problems (MOPs) are very common in real-world applications, and involve the solution of problems that have two or more (often conflicting) objectives, which we aim to optimize at the same time. The use of evolutionary algorithms to solve MOPs has become a very popular research area, mainly in the last 15 years [4]. The so-called multi-objective evolutionary algorithms (MOEAs) have become a frequently used tool to solve highly complex MOPs in recent years.

Being a very active research area, modern MOEAs have been enhanced with several types of mechanisms that aim to improve their performance. One of such mechanisms is the use of local search, which has been very popular in discrete search spaces, but has been less common (until very recently) in continuous search spaces. This is because is less intuitive to design neighborhood structures in continuous search spaces and most of the current research in this area has focused on the use of gradient-based methods [3] or direct search methods [21], which act as local search engines. The goal is, however, the same as when dealing with discrete search spaces: to improve the quality of the solutions generated by a MOEA, at an affordable computational cost.

In continuous search spaces, a local search engine needs to have a proper search direction (given, for example, by the gradient of the objective function being optimized) as well as a step size that defines for how long will the engine move in that specific direction. In MOPs, the process is more complex, because even the use of gradient information will provide different search directions for each objective, and it is necessary to estimate a proper descent direction from the combination of this information [12]. When coupling a local search engine to a MOEA, it is necessary to establish which solutions (from those generated by the MOEA) will serve as the starting point for the local search engine, and for how many iterations the engine will be applied. Evidently, it is crucial to keep a proper balance between the global search engine and the local search engine, in order to keep the overall computational cost within a value that is acceptable to the user. All of these topics have been studied and are still being investigated within the so-called Multi-Objective Memetic Algorithms [6].

The goal of this paper is to introduce a new local search engine called reflected ray tracing based on some well-known concepts from the law of reflection of light. The proposed local search mechanism is coupled to the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [20], which is a state-of-the-art MOEA.

The remainder of this paper is organized as follows. Section II provides some basic concepts from multi-objective optimization and the theory of reflection of light, which are required for a better understanding of the rest of the paper. The previous related work is briefly described in Section III and our proposed approach is provided in Section IV. Our experimental setup and results are presented in Section V. Finally, our conclusions and some possible future research paths are provided in Section VI.

## II. Basic Concepts

### A. Multi-Objective Optimization

We are interested in solving problems of the type[1]:

$$\min_{x \in \mathcal{F}} \vec{f}(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})] \quad (1)$$

where $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., k$ are the objective functions and $x \subset \mathbb{R}^n$ is the feasible region set. To describe the concept of optimality in which we are interested, we will introduce next a few definitions.

**Definitions.** (a) Given two vectors $\vec{u}, \vec{v} \in \mathbb{R}^k$, we say that $\vec{u} \leq \vec{v}$ if $u_i \leq v_i$ for all $i = 1, ..., k$. And that $\vec{u} < \vec{v}$ if $u_i \leq v_i$ and $\exists j \in \{1, \ldots, k\}$, $u_j < v_j$.

(b) Given two vectors $\vec{u}, \vec{v} \in \mathbb{R}^k$, we say that $\vec{u}$ **dominates** $\vec{v}$ (denoted by $\vec{u} \prec \vec{v}$) iff $\vec{u} < \vec{v}$.

(c) We say that a vector of decision variables $\vec{x}^* \in \mathcal{F}$ ($\mathcal{F}$ is the feasible region) is **Pareto optimum** if there does not exist another $\vec{x} \in \mathcal{F}$ such that $\vec{f}(\vec{x}) \prec \vec{f}(\vec{x^*})$.

(d) The **Pareto Optimal Set** is defined by $\mathcal{P}^* = \{\vec{x} \in \mathcal{F} | \vec{x} \text{ is Pareto optimum}\}$ and its image $\vec{f}(P^*)$ is called the **Pareto front**.

Thus, we wish to determine the Pareto optimal set from the set $\mathcal{F}$ of all the decision variable vectors.

### B. The Multiobjective Evolutionary Algorithms based on decomposition.

One of the most useful strategies for solving Multiobjective Optimization Problems (MOPs) are Multi-Objective Evolutionary Algorithms (MOEAs) based on decomposition (MOEA/D) [20], which adopt a set of weight vectors and a scalarizing function to transform an MOP into $n$ subproblems. Penalty Boundary Intersection (PBI) is a scalarizing function commomly used in MOEA/D because is able to generate a set of solutions more uniformly than other scalarizing functions. Unfortunately, PBI requires an user-defined penalty parameter. PBI is formulated as follows:

$$f^{pbi}(\mathbf{f} : \mathbf{w}, \mathbf{z}, \theta) = d_1 + \theta d_2$$
$$\text{where } d_1 = \frac{\left\| (\mathbf{f} - \mathbf{z})^T \mathbf{w} \right\|}{\|\mathbf{w}\|}$$
$$\text{and } d_2 = \left\| \mathbf{f} - \left( \mathbf{z} + d_1 \frac{\mathbf{w}}{\|\mathbf{w}\|} \right) \right\|$$

$\mathbf{w}$ is a set of uniform weigth vectors. $d_1$ minimizes the distance to the ideal vector with the purpose of obtaining a convergence measure, $d_2$ minimizes the distance to the reference point in order to obtain uniformity. The balance between convergence and uniformity is handled by the penalty parameter $\theta$.

Several options are presented in [10], [14], [17], [19] to modify the penalty parameter $\theta$, depending on the geometrical properties of the Pareto front and the number of objectives. In fact, different values of $\theta$ can be used at different stages of the evolutionary process.

### C. The Reflection of Light

In this work we are interested in the model of *specular reflection of light*. This model describes the propagation of light as a ray that strikes a smooth surface and bounces off in one direction. The ray follows a trajectory that forms two equal angles: *the incidence angle* $\theta_i$, which is formed by an incident ray and a normal vector perpendicular to the surface and *the reflection angle* of $\theta_r$, which is formed by a normal vector and a reflected ray. For more details see Figure 2. This is known as *the law of reflection*.

We focus on the reflection of light at curved surfaces because we can trace a set of rays from any point on a curved surface and generate a corresponding image point [7]. This method, which is commonly used in computer graphics simulates the ilumination and reflection of ligth over three-dimensional objects.

Our motivation arises from the generalization of the reflection model to $n$ dimensions. Several authors [2], [8], [16] have studied the properties of geometrical models in high-dimensionality. In [15], two geometric genetic operators were introduced for solving single-objective problems. They used *conformal geometric algebra*[2] to generate inversions and reflections in a hypersphere.

## III. Previous Related Work

There have been a few proposals for coupling a local search engine to MOEA/D, although most of them were specifically conceived for discrete search spaces. Such proposals include the use of local search engines based on ant colony optimization [11], simulated annealing [13], and guided local search [1].

Two similar proposals of a local search engine coupled to MOEA/D in continuous search spaces are $MOEA/D + LS$ [18] and $MOEA/D + LSII$ [19]. Both approaches adopt a mathematical programming technique (Nelder and Mead's algorithm, which is a direct search method). These approaches use a neighborhood formed by a geometrical structure called simplex, and then apply reflection, expansion and contraction movements in order to generate new solutions. One difference between $MOEA/D + LS$ and $MOEA/D + LSII$ is that in the former, the local search is randomly applied to some of the solutions obtained by MOEA/D. In $MOEA/D + LSII$, local search is applied to the solutions that are at the extreme points and in the knee of the Pareto front. In this case, the search is first directed to the extremes of the Pareto front and once they have been approximated, the local search focuses on obtaining solutions that are close to the knee of the front.

These approaches have two main limitations: (1) they do not work properly with multi-frontal problems, and (2) a special

[1]Without loss of generality, we will assume only minimization problems.

[2]This is a Clifford algebra for performing rigid Euclidean transformations (isometry) such as rotations, translations and reflectios.

procedure is required to build the simplex, so that certain geometrical constraints are properly fulfilled.

Here, we propose a viable alternative for performing an effective and efficient local search. Also, we remark some relevant aspects that we considered when designing our local search engine. Finally, we also indicate how to exploit some useful information provided by MOEA/D.

## IV. Our Proposed Approach

We propose an algorithm that combines the decomposition approach called MOEA/D-PBI [20] (our global search engine) with a new local search engine based on the law of reflection of light.

Our proposed approach is called MOEA/D-Ray and it consists of three main components: 1) The global search procedure that applies genetic operators to generate new solutions and solves the scalarization subproblem; 2) the selection strategy which chooses a set of solutions to build a neighborhood structure for the local search mechanism; and 3) the local search engine that exploits and explores the neighborhood to generate new solutions.

These components work together to speed up the convergence process towards the Pareto optimal front (see Figure 1). In the following sections, we will describe in detail the components of our local search mechanism and we will explain how it was coupled to MOEA/D.

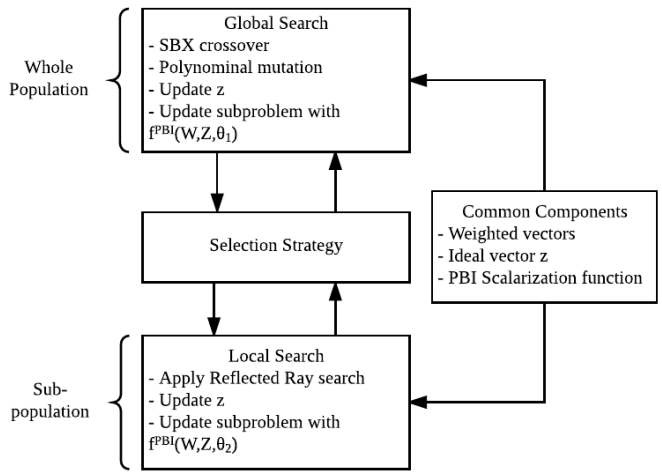



Fig. 1. Main components of our proposed local search mechanism.

### A. Selection Strategy.

When the PBI approach is applied, we monitor the change of the distance $d_1$ for every individual of the population as $\delta(d_{i,1}) = d_{i,1}^{old} - d_{i,1}^{cur}, \forall i = 1, \dots N$, where $d_{i,1}^{old}$ is the distance of the $i^{th}$ individual from the previous population and $d_{i,1}^{cur}$ is the distance of the $i^{th}$ individual from the current population. Then, we select the 50% of solutions with smaller $\delta(\vec{d_1})$. Thus, we choose from this subset the 50% of solutions with the smallest $f^{pbi}$. Here, the goal is to identify 25% of the worst solutions with null or small improvement of the scalar optimization subproblem. This subset is called $L$. For more details see Algorithm 1.

### B. Building the Hyperspherical Neighborhood

MOEA/D computes the Euclidean distances between any two weighted vectors and then works out the $T$ closest weighted vectors to each weighted vector. For each $i = 1, \dots, N$, set $B(i) = \{i_1, ...i_T\}$ where $\vec{w}_{i1}, \dots, \vec{w}_{iT}$ are the $T$ closest weighted vectors to $\vec{w}_i$ and $N$ is the number of subproblems. For each solution $k$ in $L$, we randomly take two nondominated solutions $\vec{s}_{k,p}$ and $\vec{s}_{k,q}$, where $p$ and $q$ are indexes from $B(k)$. Then, a hyperspherical neighborhood at decision variables space is built using equation (2).

$$r^2 = \sum_{i=1}^{n} x_i - c_i \tag{2}$$

where $\vec{c}$ is a center point, and $r$ is the radius. The average between $\vec{s}_{k,p}$ and $\vec{s}_{k,q}$ determines $\vec{c}$ and $r$ is computed by the Euclidean distance between $\vec{c}$ and $\vec{s}_{k,p}$. Both $\vec{s}_{k,p}$ and $\vec{s}_{k,q}$ are points located on the surface of the hypersphere.

### C. Reflected Ray Tracing

In order to guide the search to promising regions, we create a scene that simulates rays of light emitted from an origin point termed *the light source* (one solution from the set $L$) to an intersection point over the hypersphere (solution $\vec{s}_{k,p}$ or $\vec{s}_{k,q}$).

Two possible cases can be presented in the scene: the light source can be located either inside or outside of the hypersphere, which means that the local search can either exploit or explore the spherical neighborhood, respectively. In case of exploitation, the intersection point $\vec{i}$ corresponds to the original point $\vec{s}_{k,p}$ or $\vec{s}_{k,p}$, see Figure 2b. Otherwise, we need to compute the intersection of a light ray and a hypersphere, which involves equation (2) and *the parametric function of the ray* given by equation (3).

$$\vec{x} = \vec{o} + t\vec{d} \tag{3}$$

where $\vec{o}$ is an origin point ($\forall k \in L$). $\vec{d}$ is a direction vector (intersection point $\vec{i}$), and $t$ is the distance between $\vec{x}$ and $\vec{o}$. Replacing equation (2) in equation (3) we obtain the following equation:

$$t = -(\vec{o} - \vec{c}) \cdot \vec{d} \pm \sqrt{((\vec{o} - \vec{c}) \cdot \vec{d})^2 - ((\vec{o} - \vec{c})^2 - r^2)} \tag{4}$$

This implies solving a quadratic equation that depends on $t$. We use the smallest value of $t$ because it is the first intersection of the light ray over the hypersphere (see Figure 2c). The direction of the reflected light ray $\vec{R}$ is computed using:

$$\vec{R} = \vec{u} - (2\vec{u} \cdot \vec{n})\vec{n} \tag{5}$$

where $\vec{n}$ is the unit normal vector to the surface, $\vec{u}$ is the direction of the light ray trajectory, (i.e., the vector from $\vec{o}$ to $\vec{i}$).

Given the unit reflection rays, we scale them to a size equal to the hypersphere's radius multiplied by a $\lambda$ factor. Here, the goal is to avoid a strong disturbance over decision variables in order to refine the search in continuous space. Once the new points are obtained, we update the ideal vector values and subproblems using PBI's scalarizing function. The penalty value is changed ($\theta = 10$) to mantain a good uniformity along the Pareto front, as we previously recommended in Section II-B.

We repeat the above procedure for a pre-determined number of evaluations ($els$). Then, we obtain a subset of new candidate solutions (called $I$) that consists of *the intersection points, and the reflection rays*. Then, we select only the nondominated solutions from this subset and we add them to a new population, see Algorithm 2. Figure 2 illustrates a simple example of our method in two-dimensional objective space and two-dimensional decision variable space. Algorithm 3 shows the complete MOEA/D-Ray algorithm.

**Algorithm 1** Selection strategy

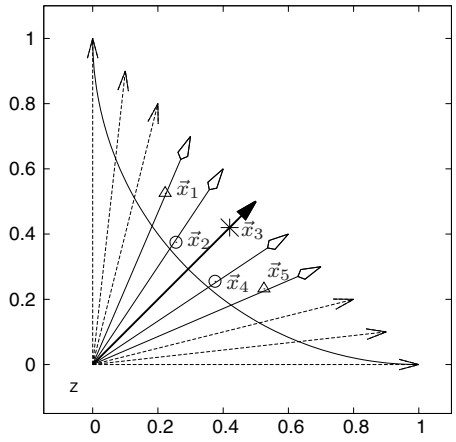
**Input.** $\delta$: record of change of the distance $d_1$ for each individual
$\theta_2$: penalty parameter
$w$: weight vectors
$z$: ideal vector
$pop$: current population
**Output.** $L$: indexes of a subpopulation

Sort individuals of $pop$ considering values of $\delta$
Set $A \leftarrow pop[i \cdots n/2]$,
Sort the indexes of $A$ considering the $f^{PBI}(\mathbf{f} : \mathbf{w}, \mathbf{z}, \theta_2)$ value
Set $L \leftarrow A[i, \cdots n/2]$

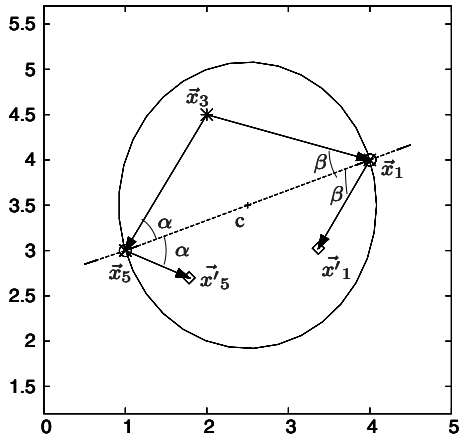
**Algorithm 2** Local search mechanism

**Input.** $\lambda$: scale factor
$L$: indexes of subpopulation
**Output:** $N$: a subset of new candidate solutions

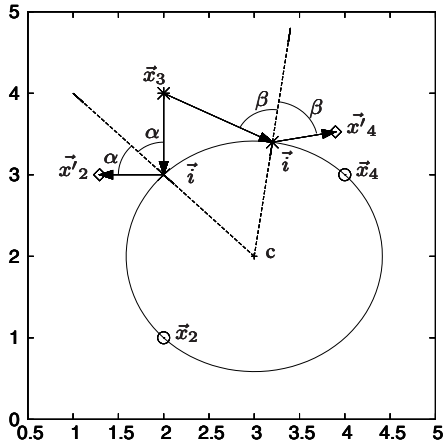
**for** $k \in L$ **do**
  **Step 1. Build hyperspherical neighborhood**
  Select (randomly) two non-dominated solutions ($\vec{s}_{k,p}$ and $\vec{s}_{k,q}$) from neighborhood of $k$, $B(k)$
  Compute the hypersphere's center $\vec{c}$ as the Euclidean distance between $\vec{s}_{k,p}$ and $\vec{s}_{k,q}$
  Compute the radius $r$ as the Euclidean distance between $\vec{c}$ and $\vec{s}_{k,p}$
  **Step 2. Apply reflected ray search**
  Set the light source as the individual with index $k$ in the current population $pop$. $\vec{o} = pop[k]$
  **Step 2.1** Compute intersection point $\vec{i}$ using equation (4)
  **if** $\vec{i}$ is different to $\vec{s}_{k,p}$ or $\vec{s}_{k,q}$ **then**
    Add $\vec{i}$ to set $N$
  **end if**
  **Step 2.2** Compute unit normal vector at surface (direction vector: $\vec{c}$ to $\vec{i}$)
  **Step 2.1** Compute all reflected rays $R$ using equation (5)
  **for** each reflected rays **do**
    **Step 2.3. Scale reflected rays**. $\lambda \cdot R_j$
  **end for**
  Add $N \leftarrow R$
  **Step 3. Update population.** Update ideal vector $\vec{z}$
  Update subproblem $k$, $f^{pbi}(\mathbf{f} : \mathbf{w}, \mathbf{z}, \theta_2)$
**end for**




a) Example of weighted vectors in objective space. $\vec{x}_3$ is a candidate solution to define light source because is far away from the ideal vector $\vec{z}$. $\vec{x}_1$, $\vec{x}_2$, $\vec{x}_4$ and $\vec{x}_5$ are solutions in the same neighborhood that minimize every scalarazing subproblem.




b) The case when the light source $\vec{x}_3$ is inside of the spherical neighborhood, $\vec{x'}_1$ and $\vec{x'}_5$ are the new solutions computed by reflected ray tracing.




c) The case when the light source $\vec{x}_3$ is outside of the spherical neighborhood, $\vec{x'}_2$ and $\vec{x'}_4$ are the new solutions computed by reflected ray tracing.

Fig. 2. Geometrical procedure of our local search based on reflected ray tracing. Figure a) shows the distribution of 5 solutions in objetive space. Figure b) and c) present scenes of reflected ray tracing in decision variable space.

**Algorithm 3** Reflected ray tracing coupled to MOEA/D

**Input:** A stopping criterion
$pop\_size$: population size
$\{w_1, ..., w_n\}$: A well-distributed set of weighted vectors
$T$: the number of weight vectors in the neighborhood of each weighted vector
$els$: the maximum number of evaluations for the local search.
$\rho$: local search rate
**Output:** $EP$: the nondominated solutions found during the search
$P$: the final population found by MOEA/D.

**Step 1. Initialization:**
Set $E_p \leftarrow \emptyset$
Compute the Euclidean distance between any two weighted vectors and then work out the $T$ closest weighted vectors to each weighted vector.
**for** $i \leftarrow 1$ to $N$ **do**
    Set $B(i) = \{i_i, ...i_T\}$ where $w_{i1}, ..., w_{iT}$ are the $T$ closest weighted vectors to $w_i$
**end for**
Set the ideal vector, $\vec{z^*} = (+\infty, \ldots, +\infty)$
Generate an initial population $P = \{x_1, ....x_N\}$ randomly.
Set $FV^i = F(x_i)$
**while** the stopping criterion is not satisfied **do**
    **for** $i \leftarrow 1$ to $N$ **do**
        **Step 2. Evolution**. Randomly select two indexes $k, l$ from $B(i)$
        Apply crossover from $x_k$ and $x_l$ and then generate a new solution $y$.
        Apply the mutation operator on $y$ to generate $y^{'}$
        **Step 3. Update of** $z$:
        **for** $j = 1, \ldots, k$ **do**
            Update subproblem $i$, $f^{pbi}(\mathbf{f} : \mathbf{w}, \mathbf{z}, \theta_1)$
        **end for**
        **Step 4. Update of Neighboring Solutions**
        **for** $j \in B(i)$ **do**
            **if** $f^{pbi}(y'|w_j, z) \leq f^{pbi}(x|w_j, z)$ **then**
                set $x_j = y^{'}$ and $FV^j = F(y')$
                **Step 5. Monitor the change of the value of** $d_1$ set $\delta = d_1' - d_1$. (If it's the first generation, then $\delta = d_1$)
            **end if**
        **end for**
        **if** $random(0, 1) \leq \rho$ **then**
            **Step 6. Selection strategy.** Select the initial solutions $L$, for applying the local search according to Algorithm 1.
            **Step 7. Local search mechanism.** Apply local search according to Algorithm 2 and obtain the set $N$.
        **end if**
        **Step 8. Update of** $EP$: Remove from $EP$ all the vectors dominated by $F(y')$.
        Add $F(y')$ to $EP$, if no vectors in $EP$ dominate $F(y')$
    **end for**
**end while**

## V. Experimental Results

In order to validate our proposed approach, we adopted three different benchmark functions:

- The *Zitzler-Deb-Thiele (ZDT)* test suite [22]. We used 30 decision variables for ZDT1 to ZTD3, while ZDT4 and ZDT6 were tested using 10 decision variables.
- The *Deb-Thiele-Laumanns-Zitzler (DTLZ)* test suite [9] defined by seven MOPs. Here, DTLZ1 was tested with 7 decision variables; DTLZ2 to DTLZ6 employed 12 decision variables and DTLZ7 adopted 22. For all problems, we tested the algorithms using three objective functions.
- The *Walking-Fish-Group (WFG)* test suite [23]. Here, we employed 24 decision variables and three objective functions in all cases.

**Comparison.** We compared our proposed MOEA/D-Ray with respect to the same version of MOEA/D [20] without the use of local search, and with respect to MOEA/D+LSII [19][3] and NSGA-II [5][4].

**Parameters settings**. The parameters adopted by the algorithms compared are summarized in Table I. For a fair comparison we used the same set of weighted vectors generated by the Simplex Lattice Design method. Also, we used PBI and $T = 20$ (size of neighborhood) for all algorithms based on MOEA/D. Regarding the population size, $pop\_size$, we used 100 individuals in bi-objective problems, and 210 individuals in three-objective problems. We performed 20,000 function evaluations ($nfe$) in bi-objective MOPs and 30,000 in three-objective MOPs. The parameters $\eta_c$ and $P_c$, represent the crossover index and rate for Simulated Binary Crossover (SBX), $\eta_m$ and $P_m$, represent the mutation index and the mutation rate for Polynomial-Based Mutation, respectively. The above mentioned parameters were set the same in the four algorithms for a fair comparison. Such values are the same that were proposed in the original version of the MOEA/D [20]. $els$ is the maximum number of evaluation functions employed by the local search in MOEA/D+LS-II and MOEA/D-Ray. The parameters $sls$, $\alpha$, $\beta$, $\gamma$ used by the local search in MOEA/D-LSII are the ones proposed by their authors.

TABLE I
Parameters settings for each algorithm

| Parameter | NSGA-II | MOEA/D | MOEA/D NM | MOEA/D Ray |
|---|---|---|---|---|
| $popSize$ | 100 / 210 | | | |
| $nfe$ | 20,000 / 30,000 | | | |
| $T$ | 20 | 20 | 20 | 20 |
| $\eta_c$ | 20 | 20 | 20 | 20 |
| $P_c$ | 0.9 | 0.9 | 0.9 | 0.9 |
| $\eta_m$ | 20 | 20 | 20 | 20 |
| $P_m$ | 1/N | 1/N | 1/N | 1/N |
| $sls$ | – | – | 0.001 | – |
| $els$ | – | – | 300 | 200 |
| $rls$ | – | – | 15 | – |
| $alpha$ | – | – | 1.0 | – |
| $beta$ | – | – | 2.0 | – |
| $gamma$ | – | – | 0.5 | – |
| PBI $\theta_1$ | – | 5 | 5 | 5 |
| PBI $\theta_2$ | – | – | 10 | 10 |
| $\rho$ | – | – | – | 0.6 |
| rate | – | – | – | 0.3 |

[3]For MOEA/D, MOEA/D-Ray and MOEA/D+LSII, we adopted our own framework which is available at: http://computacion.cs.cinvestav.mx/~pescador/codes.html

[4]We used the source code of NSGA-II available at: http://www.iitk.ac.in/kangal/codes.shtml

In order to determine the most appropriate setting for $\lambda$ (step size) and $\rho$ (local search rate), we tested several combinations of values. Parameter $\lambda$ was set within the range [0.2 to 2.0], varying at steps of 0.2, and $\rho$ was tested in the range [0.1 to 1.0] at steps of 0.1. We experimented using DTLZ1 with 10 decision variables and 3 objective functions, and we performed 30 independent executions for every combination of values for these two parameters. Then, the hypervolume indicator was computed based on the correlations found for each parameter instance. For the experiments reported here we found the best results with $\lambda = 0.6$ and $\rho = 0.3$.

### A. Experimental Methology

All MOEAs were tested using the same PC and software. Our PC has an Intel Core i5 processor (2.6 GHz), 8 GB RAM and the operating system OS-X. In Table II we compared the execution time for each algorithm based on MOEA/D, showing that our proposed MOEA/D-Ray is faster than MOEA/D+LSII for all problems and does not have a considerable time difference with respect to the original MOEA/D.

TABLE II
EXECUTION TIME FOR MOEA/D, MOEA/D+LSII AND MOEA/D-RAY (IN SECONDS) ALL ALGORITHMS WERE IMPLEMENTED IN C++ AND WERE EXECUTED ON A PC WITH THE SAME HARDWARE AND SOFTWARE.

| MOP | MOEA/D | MOEA/D+LSII | MOEA/D-Ray |
|---|---|---|---|
| ZDT1 | 0.9745 | 1.6094 | 1.2127 |
| ZDT2 | 1.0510 | 1.6209 | 1.3237 |
| ZDT3 | 0.9696 | 1.7990 | 1.3855 |
| ZDT4 | 0.9870 | 1.6445 | 1.5456 |
| ZDT6 | 0.9526 | 1.7201 | 1.6042 |
| DTLZ1 | 1.4300 | 2.6683 | 2.0855 |
| DTLZ2 | 1.3714 | 2.6328 | 2.0335 |
| DTLZ3 | 1.4321 | 2.7061 | 2.0170 |
| DTLZ4 | 1.4077 | 2.5641 | 1.8207 |
| DTLZ5 | 1.3802 | 2.1688 | 1.7995 |
| DTLZ6 | 1.3269 | 1.9438 | 1.7764 |
| DTLZ7 | 1.3770 | 2.2040 | 1.8812 |
| WFG1 | 1.4677 | 2.8971 | 1.9134 |
| WFG2 | 1.4113 | 2.8316 | 1.8767 |
| WFG3 | 1.4222 | 2.7747 | 1.6620 |
| WFG4 | 2.0113 | 2.9811 | 2.5685 |
| WFG5 | 2.1218 | 3.0918 | 2.5482 |
| WFG6 | 2.2843 | 3.0034 | 2.5869 |
| WFG7 | 1.7058 | 3.2843 | 2.5575 |
| WFG8 | 1.5424 | 3.5776 | 2.7119 |
| WFG9 | 1.8179 | 3.3040 | 2.8062 |

**Performance measure**. For comparing our results, we adopted the hypervolume indicator (HV) [23] to assess both convergence and maximum spread. In Table III, we provide the reference points adopted for each of the test problems used. We performed 30 independent runs for each algorithm and problem instance. Next, we show the average and standard deviations of the performance measure adopted in our study. The two best values of the hypervolume indicator are shown in gray scale in Table III, where the darker tone corresponds to the best value. Additionally, we also applied the Wilcoxon rank sum test with a confidence level of 95%. Table IV shows the value of $p$, which is the probability of observing that the null hypothesis is true. A value $H = 1$ indicates that we reject the null hypothesis of equal medians with a 5% significance level, which means that the results are statistically significant.

We analyzed the convergence of all the MOEAs based on decomposition analyzed in our comparative study. This analysis showed that our proposed approach was the fastest, as illustrated in Figure 4 for DTLZ1.

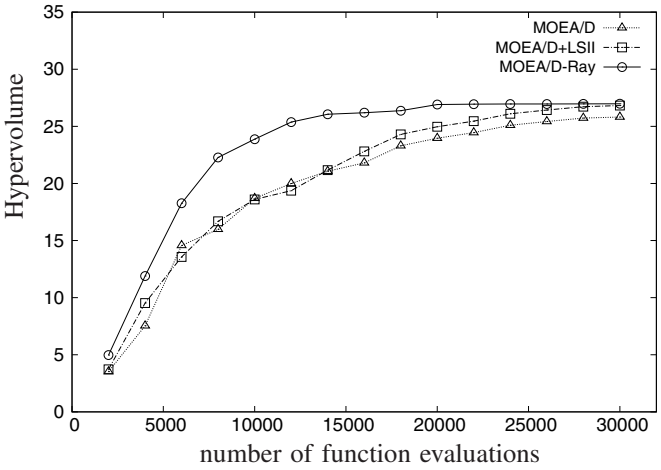



Fig. 3. Convergence plot for DTLZ1. MOEA/D-Ray is faster than both the original MOEA/D and MOEA/D-LSII.

Figure 4 shows the Pareto front in ZDT4 obtained in the median execution for each of the MOEAs based on decomposition.

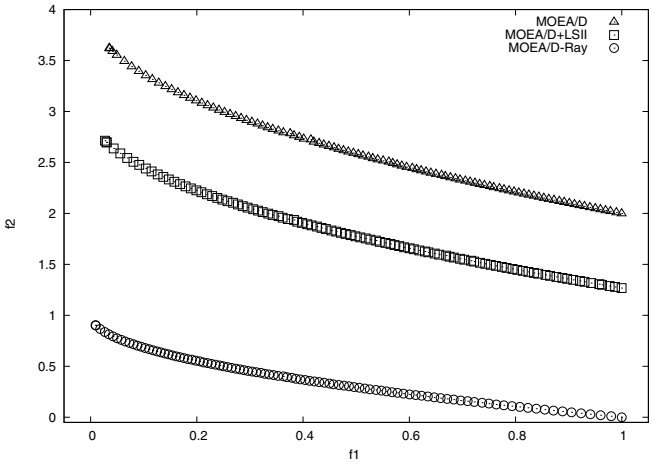



Fig. 4. Pareto front for ZDT4 obtained by MOEA/D, MOEA/D+LSII and MOEA/D-Ray. MOEA/D-Ray is able to converge to the Pareto optimal front.

### B. Discussion of Results

As shown by our results, our proposed MOEA/D-Ray is a good choice to solve bi-objective and three-objective MOPs. Our proposed approach was able to obtain better results than MOEA/D, MOEA/D-LSII and NSGA-II, in most cases. According to Wilcoxon's rank sum test, we corroborate that

TABLE III

THE STATISTICAL RESULTS (MEAN AND STANDARD DEVIATION) OF THE HYPERVOLUME VALUES OBTAINED BY NSGA-II, MOEA/D, MOEA/D+LSII AND MOEA/D-RAY. THE VALUES IN PARENTHESES CORRESPOND TO THE STANDARD DEVIATIONS. THE TWO BEST VALUES ARE SHOWN IN GRAY SCALE, WHERE THE DARKER TONE CORRESPONDS TO THE BEST VALUE. THE BEST RESULTS HAVE A SIGNIFICANCE LEVEL OF $\alpha = 0.5$

| MOP | NSGA-II | MOEA/D | MOEA/D+LSII | MOEA/D-Ray | Reference |
|---|---|---|---|---|---|
| ZDT1 | 3.591618 (0.331818) | 3.608082 (0.258752) | 1.918301 (1.975182) | 3.659684 (0.001113) | $(2.0, 2.0)^T$ |
| ZDT2 | 3.258176 (0.499599) | 3.259610 (0.440483) | 1.820356 (2.980399) | 3.326285 (0.001818) | $(2.0, 2.0)^T$ |
| ZDT3 | 4.646560 (1.352253) | 4.686198 (1.009701) | 2.879301 (2.459046) | 4.814008 (0.001251) | $(2.0, 2.0)^T$ |
| ZDT4 | 11.603338 (0.180104) | 6.684286 (7.941478) | 6.981166 (8.828981) | 11.654082 (0.019547) | $(2.0, 6.0)^T$ |
| ZDT6 | 3.041577 (0.002865) | 3.001439 (0.059057) | 2.770564 (1.799824) | 3.326633 (0.017445) | $(2.0, 2.0)^T$ |
| DTLZ1 | 26.256424 (0.343029) | 25.007872 (17.861199) | 24.536081 (15.126729) | 26.972470 (0.007736) | $(3.0, 3.0, 3.0)^T$ |
| DTLZ2 | 7.393638 (0.007769) | 7.433595 (0.000300) | 7.062171 (0.204876) | 7.427342 (0.002048) | $(2.0, 2.0, 2.0)^T$ |
| DTLZ3 | 15.500045 (66.354695) | 21.032095 (34.369513) | 17.692667 (22.637649) | 23.428327 (13.229819) | $(3.0, 3.0, 3.0)^T$ |
| DTLZ4 | 6.636596 (6.195573) | 6.934670 (4.933657) | 4.505939 (4.357881) | 7.400575 (0.062163) | $(2.0, 2.0, 2.0)^T$ |
| DTLZ5 | 1.214578 (0.023905) | 1.205595 (0.007449) | 0.650153 (0.592119) | 1.220499 (0.000480) | $(1.0, 1.0, 2.0)^T$ |
| DTLZ6 | 0.335902 (0.994035) | 4.426891 (2.058689) | 4.426885 (2.151937) | 6.052581 (0.084691) | $(2.0, 2.0, 2.0)^T$ |
| DTLZ7 | 12.561307 (11.537606) | 13.370739 (0.079531) | 14.812108 (14.551490) | 13.368943 (0.153535) | $(2.0, 2.0, 7.0)^T$ |
| WFG1 | 43.939962 (14.760476) | 42.648792 (17.294853) | 42.548805 (18.596301) | 42.771111 (6.940847) | $(3.0, 5.0, 7.0)^T$ |
| WFG2 | 33.986503 (11.837966) | 33.668455 (11.586520) | 26.000000 (32.45865) | 40.525098 (16.507582) | $(2.2, 4.2, 6.2)^T$ |
| WFG3 | 82.365507 (9.414187) | 85.642500 (7.936499) | 86.667626 (13.212936) | 84.743678 (1.527415) | $(3.0, 5.0, 7.0)^T$ |
| WFG4 | 24.230959 (1.858749) | 22.582676 (1.464781) | 24.548152 (1.100759) | 26.062369 (1.745179) | $(2.2, 4.2, 6.2)^T$ |
| WFG5 | 14.477720 (1.095130) | 14.425945 (0.974080) | 26.138973 (301.162955) | 24.694191 (1.356886) | $(2.2, 4.2, 6.2)^T$ |
| WFG6 | 24.805127 (2.034386) | 21.043143 (2.891076) | 25.087495 (98.297523) | 24.287705 (1.947224) | $(2.2, 4.2, 6.2)^T$ |
| WFG7 | 11.935860 (6.194971) | 12.634786 (7.023479) | 12.775412 (6.156326) | 27.167782 (1.419601) | $(2.2, 4.2, 6.2)^T$ |
| WFG8 | 22.490566 (3.353735) | 12.785893 (2.853772) | 12.726598 (2.455577) | 22.010472 (1.308116) | $(2.2, 4.2, 6.2)^T$ |
| WFG9 | 20.552339 (5.599485) | 13.446701 (5.212999) | 13.99828 (6.300886) | 23.326568 (5.443270) | $(2.2, 4.2, 6.2)^T$ |

TABLE IV

WILCOXON'S RANK SUM FOR THE ZDT, DTLZ, AND WFG TEST PROBLEMS WITH A SIGNIFICANCE LEVEL OF 95%. A VALUE $H = 1$ INDICATES THAT WE REJECT THE NULL HYPOTHESIS, WHICH MEANS THAT THE RESULTS ARE STATISTICALLY SIGNIFICANT.

| MOP | NSGA-II & MOEA/D-Ray | | MOEA/D & MOEA/D-Ray | | MOEA/D+LSII & MOEA/D-Ray | |
|---|---|---|---|---|---|---|
| | p | H | p | H | p | H |
| ZDT1 | 3.0180e-11 | (1) | 0.0059 | (1) | 5.2152e-12 | (1) |
| ZDT2 | 3.0199e-11 | (1) | 0.3871 | (0) | 4.1079e-12 | (1) |
| ZDT3 | 3.0199e-11 | (1) | 0.6520 | (0) | 3.1602e-12 | (1) |
| ZDT4 | 8.3074e-8 | (1) | 2.9972e-11 | (1) | 2.9972e-9 | (1) |
| ZDT6 | 2.9897e-11 | (1) | 2.9897e-11 | (1) | 2.7887e-8 | (1) |
| DTLZ1 | 0.0643 | (1) | 3.0161e-11 | (1) | 1.2118e-12 | (1) |
| DTLZ2 | 2.9636e-11 | (1) | 2.9636e-11 | (1) | 3.0873e-11 | (1) |
| DTLZ3 | 8.2735e-7 | (1) | 0.1388 | (0) | 5.8006e-9 | (1) |
| DTLZ4 | 0.6626 | (0) | 0.0079 | (1) | 8.3729e-9 | (1) |
| DTLZ5 | 3.0199e-11 | (1) | 0.9411 | (0) | 3.0180e-11 | (1) |
| DTLZ6 | 3.0199e-11 | (1) | 3.0199e-11 | (1) | 3.0199e-11 | (1) |
| DTLZ7 | 0.1334 | (0) | 0.0303 | (1) | 0.4035 | (0) |
| WFG1 | 5.2622e-4 | (1) | 4.1097e-7 | (1) | 8.2872e-6 | (1) |
| WFG2 | 3.0161e-11 | (1) | 0.0070 | (1) | 3.1553e-6 | (1) |
| WFG3 | 3.0061e-11 | (1) | 2.8227e-8 | (1) | 1.1984e-8 | (1) |
| WFG4 | 3.0085e-11 | (1) | 1.6792e-4 | (1) | 1.1043e-4 | (1) |
| WFG5 | 3.0142e-11 | (1) | 0.2115 | (0) | 0.8073 | (0) |
| WFG6 | 2.9450e-11 | (1) | 0.0420 | (1) | 3.5990e-11 | (1) |
| WFG7 | 3.0104e-11 | (1) | 0.0163 | (1) | 0.0035 | (1) |
| WFG8 | 3.0199e-11 | (1) | 0.0468 | (1) | 0.1958 | (0) |
| WFG9 | 3.0142e-11 | (1) | 0.0056 | (1) | 1.8392e-8 | (1) |

our results are statistically significant, because the p-value was small, and our results indicate that the null hypothesis can be rejected at the 5% level. Additionally, our proposed approach is faster than MOEA/D+LSII and does not have a considerable time difference with respect to MOEA/D. Also, we could observe a good hypervolume value in multifrontal problems such as ZDT4, ZDT6, DTLZ1, DTLZ3, WFG2, WFG4, and WFG9. In these cases, the neighborhood structure built with the non-dominated solutions was able to improve the worst solutions. Unfortunately, MOPs with a disconnected Pareto front such as DTLZ7, WFG3, WFG6 were difficult for our local search mechanism. Here, our selection strategy was affected directly because the disconnected regions did not correlate with the neighborhood structure.

## VI. Conclusions and Future Work

We have presented a new local search method that is able to speed up the convergence of MOEA/D. Our proposed approach incorporates four main strategies: 1) we monitor the worst individuals and apply the local search in order to improve them, 2) our method uses information of good individuals (non-dominated solutions) located in the neighborhood of the solutions obtained so far by MOEA/D with the purpose of establishing a new spherical neighborhood in decision variable space, 3) the geometric model of reflection of light is applied to refine the search, and 4) the local search is iterated according to a certain user-defined budget in order to optimize the scalarizing function with a different penalty value. Our results indicate that our proposed approach is able to improve the performance of MOEA/D in a variety of test problems.

As a part of our future work, we plan to design a mechanism that can properly identify some special cases (e.g., when the Pareto front has disconnected regions) in which our proposed approach doesn't perform well. In these cases, the use of a self-adaptive local search mechanism may be more effective and we would like to explore this alternative. We are also interested in exploring the use of other geometrical models. Additionally, we plan to test our approach in many-objective optimization problems, although this may require some additional refinements to our current selection scheme.

## Acknowledgments

The first author acknowledges support from IPN and CONACyT to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN. The second author gratefully acknowledges support from CONACyT project no. 221551.

## References

[1] A. Alhindi and Q. Zhang. MOEA/D with guided local search: Some preliminary experimental results. In *CEEC*, pages 109–114, Sept 2013.

[2] T. Banchoff. *Beyond the Third Dimension: Geometry, Computer Graphics, and Higher Dimensions*. W. H. Freeman & Co., New York, NY, USA, 1990.

[3] P. A. Bosman. On Gradients and Hybrid Evolutionary Algorithms for Real-Valued Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 16(1):51–69, February 2012.

[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, 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.

[6] C.-K. Goh, Y.-S. Ong, and K. C. Tan, editors. *Multi-Objective Memetic Algorithms*. Springer, Berlin, Germany, 2009. ISBN 978-3-540-88050-9.

[7] N. Gonçalves. On the reflection point where light reflects to a known destination on quadratic surfaces. *Opt. Lett.*, 35(2):100–102, Jan 2010.

[8] A. J. Hanson. Geometry for n-dimensional graphics. In P. S. Heckbert, editor, *Graphics Gems IV*, pages 149–170. Academic Press Professional, Inc., San Diego, CA, USA, 1994.

[9] 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, October 2006.

[10] H. Ishibuchi, N. Akedo, and Y. Nojima. Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Transactions on Evolutionary Computation*, 19(2):264–283, April 2015.

[11] L. Ke, Q. Zhang, and R. Battiti. MOEA/D-ACO: A Multiobjective Evolutionary Algorithm Using Decomposition and Ant Colony. *IEEE Transactions on Cybernetics*, 43(6):1845–1859, Dec 2013.

[12] A. Lara, G. Sanchez, C. A. Coello Coello, and O. Schütze. HCS: A New Local Search Strategy for Memetic Multi-Objective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 14(1):112–132, February 2010.

[13] H. Li and D. Landa-Silva. An Adaptive Evolutionary Multi-objective Approach Based on Simulated Annealing. *Evol. Comput.*, 19(4):561–595, Dec. 2011.

[14] A. Mohammadi, M. N. Omidvar, X. Li, and K. Deb. Sensitivity analysis of penalty-based boundary intersection on aggregation-based emo algorithms. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 2891–2898, May 2015.

[15] J. P. Serrano Rubio, A. Hernández, and R. Herrera Guzmán. Function Optimization in Conformal Space by Using Spherical Inversions and Reflections. In A. L. Bazzan and K. Pichara, editors, *Advances in Artificial Intelligence – IBERAMIA 2014, 14th Ibero-American Conference on AI*, pages 418–429. Springer. Lecture Notes in Artificial Intelligence Vol. 8864, Santiago de Chile, Chile, November 24-27 2014.

[16] S. Thakur and A. J. Hanson. A framework for exploring high-dimensional geometry. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, N. Paragios, S.-M. Tanveer, T. Ju, Z. Liu, S. Coquillart, C. Cruz-Neira, T. Müller, and T. Malzbender, editors, *Advances in Visual Computing: Third International Symposium, ISVC 2007, Lake Tahoe, NV, USA, November 26-28, 2007, Proceedings, Part I*, pages 804–815. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[17] S. Yang, S. Jiang, and Y. Jiang. Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes. *Soft Computing*, pages 1–15, 2016.

[18] S. Zapotecas Martínez and C. A. Coello Coello. A Direct Local Search Mechanism for Decomposition-based Multi-Objective Evolutionary Algorithms. In *2012 IEEE Congress on Evolutionary Computation (CEC'2012)*, pages 3431–3438, Brisbane, Australia, June 10-15 2012. IEEE Press.

[19] S. Zapotecas Martínez and C. A. Coello Coello. A Hybridization of MOEA/D with the Nonlinear Simplex Search Algorithm. In *Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM'2013)*, pages 48–55, Singapore, April 16–19 2013. IEEE Press.

[20] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[21] X. Zhong, W. Fan, J. Lin, and Z. Zhao. Hybrid Non-dominated Sorting Differential Evolutionary Algorithm with Nelder-Mead. In X. Huang, L. D. Xu, Z. D. Zhou, Z. Fan, M. Gupta, and P. Wang, editors, *2010 Second WRI Global Congress on Intelligent Systems (GCIS'2010)*, volume 1, pages 306–311, Wuhan, China, 16-17 December 2010. IEEE Computer Society Press.

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

[23] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.