

# Exploiting Gradient Information in Numerical Multi-Objective Evolutionary Optimization

Peter. A.N. Bosman  
Centre for Mathematics and Computer Science  
P.O. Box 94079  
1090 GB Amsterdam  
The Netherlands  
Peter.Bosman@cwi.nl

Edwin D. de Jong  
Institute of Information and Computing Sciences  
Utrecht University  
P.O. Box 80089  
3508 TB Utrecht  
The Netherlands  
dejong@cs.uu.nl

## ABSTRACT

Various multi-objective evolutionary algorithms (MOEAs) have obtained promising results on various numerical multi-objective optimization problems. The combination with gradient-based local search operators has however been limited to only a few studies. In the single-objective case it is known that the additional use of gradient information can be beneficial. In this paper we provide an analytical parametric description of the set of all non-dominated (i.e. most promising) directions in which a solution can be moved such that its objectives either improve or remain the same. Moreover, the parameters describing this set can be computed efficiently using only the gradients of the individual objectives. We use this result to hybridize an existing MOEA with a local search operator that moves a solution in a randomly chosen non-dominated improving direction. We test the resulting algorithm on a few well-known benchmark problems and compare the results with the same MOEA without local search and the same MOEA with gradient-based techniques that use only one objective at a time. The results indicate that exploiting gradient information based on the non-dominated improving directions is superior to using the gradients of the objectives separately and that it can furthermore improve the result of MOEAs in which no local search is used, given enough evaluations.

## Categories and Subject Descriptors

G.1.6 [Numerical Analysis]: Optimization—*Gradient methods*; I.2 [Artificial Intelligence]: Problem Solving, Control Methods, and Search

## General Terms

Algorithms, Performance, Experimentation, Theory

## Keywords

Evolutionary Algorithms, Memetic Algorithms, Multi-Objective Optimization, Numerical Optimization, Gradients

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'05, June 25–29, 2005, Washington, DC, USA.  
Copyright 2005 ACM 1-59593-010-8/05/0006 ...\$5.00.

## 1. INTRODUCTION

Evolutionary algorithms (EAs) seek to exploit the global structure of the search space. Examples are the use of crossover, the aim of which is to fruitfully exchange good partial solutions, and the use of estimation-of-distribution techniques, which provide a way of discovering and using information about variable linkage. Information about the *local* structure of the search space is often disregarded.

In many problems of interest however, particularly those defined over continuous spaces, local structure can provide a great deal of information about the directions in which improvement can be achieved. Indeed, a substantial part of the methods studied in the field of machine learning are based on the principle of following the gradient of a performance function [14]. Hence, a sensible question is whether EAs for numerical (i.e. real-valued) optimization can be improved by incorporating local optimization techniques.

For single-objective problems, the notion of combining EAs with local (possibly gradient-based) optimization is well known. The resulting hybrid EAs have mainly become known under the name of Memetic Algorithms [15]. Also in multi-objective optimization, the use of such approaches has recently come into focus. The main focus has been on combinatorial optimization problems [9, 12, 20].

In single-objective numerical optimization, the gradient  $\nabla f$  of the function  $f$  to be optimized conveys useful information. For any  $l$ -dimensional point  $\mathbf{y}$ , the gradient at that point  $-(\nabla f(\mathbf{x}))(\mathbf{y})$  is the direction of greatest decrease of  $f$  starting from point  $\mathbf{y}$ . Hence this direction can be used in an algorithm to find local minima of  $f$ . Many such algorithms exist, ranging from straightforward ones such as gradient descent to more advanced ones such as conjugate gradients [8]. These algorithms are efficient at finding local minima. Hybridizing EAs with such gradient-based optimization algorithms has been shown to lead to good results for single-objective optimization [1]. A natural question therefore is how we can expand gradient-based techniques to a multi-objective setting in an attempt to improve upon existing MOEAs for numerical multi-objective optimization.

In multi-objective optimization the situation is however more complicated. For instance, there is typically no single direction to move a solution in so that all objectives are improved simultaneously. Therefore, important first questions for applying gradient techniques in a multi-objective setting are whether we can and should use the gradients of the individual objective functions separately, what the gradient of

a multi-objective function looks like and how we can utilize the information conveyed by this gradient. In this article we will provide an answer to these questions and use these answers to construct new hybrid MOEAs with.

In this work, we compare three approaches to using gradient information. First, we consider a method that randomly chooses an objective and optimizes it using standard single-objective gradient-based optimization techniques. Different individuals will thus be optimized with respect to different objectives. This method has strong similarities with the early multi-objective method VEGA [18]. Next, we consider a method which, for a given individual, optimizes the various objectives in turn, again using standard single-objective gradient-based optimization techniques. A more interesting question however is whether all objectives in a multi-objective problem can be optimized simultaneously by moving a solution in a specific direction. Unless the solution is trapped in a multi-objective local-optimum, using such a direction will always improve the quality of the solution. Our third method achieves this aim by identifying *all* non-dominated directions starting from the current point in which all objectives either improve or remain the same.

## 2. RELATED WORK

Currently, there exist only a few publications regarding real-valued multi-objective memetic algorithms (or hybrid EAs), the best-known of which is the M-PAES [10]. However, the M-PAES does not explicitly make use of gradient information. Lahanas, Baltas and Giannouli do use gradient information explicitly [13]. However, they use weighted aggregation to construct a single objective function which is subsequently optimized. Hence, there is no guarantee of optimizing all objectives simultaneously. Brown and Smith [4] use a hybrid EA based on an analytic description by Fliege and Svaiter [7] of a direction that has the specific property that it is the largest direction of simultaneous improvement. This direction is referred to by the authors as the multi-objective gradient. A multilevel subdivision technique that subdivides the search space to perform local search in each subspace based on a similar derivation of a single direction of descent by Schäffler, Schultz and Weinzierl [19] was proposed by Dellnitz, Schütze and Hestermeyer [6]. However, if the objectives have different ranges, the largest direction of simultaneous descent will be biased towards the objective with the largest range. Even if the objectives are first scaled the same way there are, as we shall show, still multiple (typically infinite) directions of improvement that do not dominate each other (for instance improving objective 0 and leaving objective 1 unchanged versus improving objective 1 and leaving objective 0 unchanged). All of these directions are equally useful in multi-objective optimization.

The difference with our work is that we analytically describe the *complete set* of non-dominated simultaneously improving directions. Hence, we consider the multi-objective gradient to be a set of directions (specifically a  $m - 1$  dimensional manifold in a  $m$ -dimensional space where  $m$  is the number of objectives). Moreover, the hybrid EA used by Brown and Smith is only tested on a 2-dimensional, 2-objective quadratic test problem that has very nice gradient properties, which is not expected to be a good practical test-case. Here we use a well-known set of 5 benchmark tests that have a higher dimensionality and vary in difficulty [21].

## 3. GRADIENTS & MULTIPLE OBJECTIVES

### 3.1 Single objective

The gradient of  $f$  returns for any point  $\mathbf{y}$  the direction of greatest increase of  $f$  starting from  $\mathbf{y}$  and it is defined as:

$$\nabla f(\mathbf{x}) = (\partial f / \partial x_0, \partial f / \partial x_1, \dots, \partial f / \partial x_{l-1}) \quad (1)$$

The direction of greatest decrease of  $f$  starting from  $\mathbf{y}$  is just the negative gradient at  $\mathbf{y}$ , i.e.  $-(\nabla f(\mathbf{x}))(\mathbf{y})$ . The directional derivative of  $f$  in direction  $\mathbf{u}$  is a function that for any point  $\mathbf{y}$  returns the rate of change of  $f$  in direction  $\mathbf{u}$  and it is defined as:

$$(\nabla_{\mathbf{u}} f(\mathbf{x}))(\mathbf{y}) = ((\nabla f(\mathbf{x}))(\mathbf{y}))^T \hat{\mathbf{u}} \quad (2)$$

Clearly, it can be shown that the directional derivative (in some point) is minimal if and only if  $\mathbf{u}$  points in the same direction as the negative gradient of  $f$ , i.e.  $\hat{\mathbf{u}} = \hat{\mathbf{g}}$  where  $\mathbf{g} = -\nabla f(\mathbf{x})$ . Hence, finding the direction of greatest decrease of  $f$  at any point  $\mathbf{y}$  is actually a single-objective optimization problem defined as:

$$\min_{\hat{\mathbf{u}}} \{ -(\nabla_{\mathbf{u}} f(\mathbf{x}))(\mathbf{y}) \} \quad (3)$$

The answer to this optimization problem of course is the negative gradient of  $f$  at point  $\mathbf{y}$ , i.e.  $-(\nabla f(\mathbf{x}))(\mathbf{y})$ .

### 3.2 Multiple objectives

We assume to have  $m$  real-valued objective functions that, without loss of generality, we seek to minimize simultaneously. We denote the objective functions by  $f_i(\mathbf{x})$  where  $i \in \{0, 1, \dots, m - 1\}$  and  $\mathbf{x} \in \mathbb{R}^l$ .

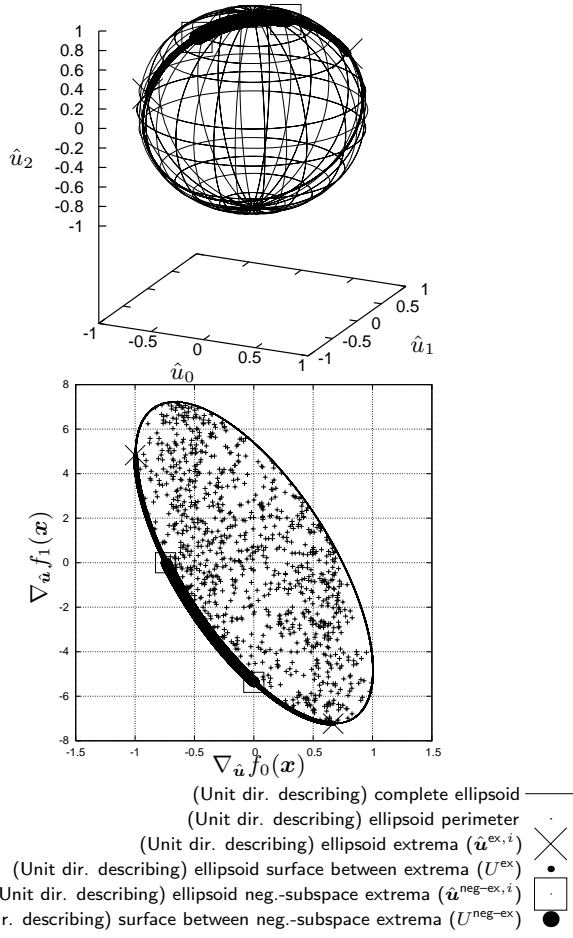
To keep the analogy with the single-objective setting, we would like to define the gradient of  $\mathbf{f} = (f_0, f_1, \dots, f_{m-1})$  for any point  $\mathbf{y}$  as the direction of greatest increase of  $\mathbf{f}$  starting from  $\mathbf{y}$ . However, in a multi-objective setting there is in general no single such direction as we shall see.

Observe the optimization-problem definition of finding the best direction, i.e. find the unit direction  $\hat{\mathbf{u}}$  such that the directional derivative in direction  $\hat{\mathbf{u}}$  indicates the best improvement. In the multi-objective case we define the directional derivative  $\nabla_{\mathbf{u}} \mathbf{f}(\mathbf{x})$  to be a vector such that the  $i^{\text{th}}$  component of that vector is the directional derivative in the  $i^{\text{th}}$  objective, i.e.  $(\nabla_{\mathbf{u}} \mathbf{f}(\mathbf{x}))_i = \nabla_{\mathbf{u}} f_i(\mathbf{x})$ , or:

$$\nabla_{\mathbf{u}} \mathbf{f}(\mathbf{x}) = \mathbf{G} \hat{\mathbf{u}} \quad (4)$$

$$\text{where } \mathbf{G} = (\nabla f_0(\mathbf{x}), \dots, \nabla f_{m-1}(\mathbf{x}))^T = \begin{bmatrix} (\nabla f_0(\mathbf{x}))^T \\ \vdots \\ (\nabla f_{m-1}(\mathbf{x}))^T \end{bmatrix}$$

To find the best direction  $\hat{\mathbf{u}}$  we must solve the optimization problem in Equation 3 with  $f$  replaced by  $\mathbf{f}$ . This optimization problem is now multi-objective. Intuitively, this is because a direction indicates improvement in  $\mathbf{f}$  if and only if the directional derivatives of all individual objective functions are non-positive (i.e. improvement in all objectives or remain the same). Now, similar to the single-objective case we are ultimately interested in the direction that *maximizes* the improvement (i.e. the negative gradient). Hence, in the multi-objective case we prefer the *set* of directions that correspond to the Pareto-optimal front of improving directions. Indeed, in the multi-objective setting there is thus in general no single direction of greatest increase of  $\mathbf{f}$  starting from  $\mathbf{y}$ . An important question now is whether we can find a parametric description of the set of directions that corresponds to the Pareto-optimal front of improving directions. Otherwise the only way to proceed is to run a multi-objective optimization algorithm just to find a good simultaneously improving direction, which clearly is less preferable.



**Figure 1: Top: Unit directions in a 3-dimensional parameter space. Bottom: Mapping of the unit directions to the 2-dimensional multi-objective directional-derivative space.**

Each of the  $m$  objectives in the function to optimize in Equation 4 is just a linearly weighted sum of the components of the direction vector, i.e. the  $j^{\text{th}}$  objective,  $j \in \{0, 1, \dots, m-1\}$ , is  $-(\nabla f_j(\mathbf{x}))^T \hat{\mathbf{u}} = \sum_{i=0}^{m-1} -(\nabla f_j(\mathbf{x}))_i \hat{u}_i$ . Now because the  $\hat{\mathbf{u}}$  to optimize over lie on the surface of a unit hypersphere of dimensionality  $l-1$  the set of all objective vectors form an ellipsoid of dimensionality  $m-1$ . Moreover, if  $m < l$ , which is usually the case, the objective vectors lie both on the surface of the ellipsoid as well as inside the ellipsoid. The points on this surface are given by directions  $\hat{\mathbf{u}}$  on the  $l$ -dimensional hypersphere that are themselves connected through a  $m$ -dimensional hypersphere. For instance if  $l = 3$  and  $m = 2$ , then the  $\hat{\mathbf{u}}$  lie on a unit sphere, the objective vectors  $\nabla_{\hat{\mathbf{u}}} \mathbf{f}(\mathbf{x})$  lie on and inside a 2-dimensional ellipsoid and the  $\hat{\mathbf{u}}$  that are projected onto the perimeter of this ellipsoid together form a unit circle across surface of the unit sphere. This case is illustrated in Figure 1.

Since we are interested in extrema (i.e. non-dominated directional derivatives), we only want to describe the surface of the ellipsoid. Specifically, we are interested in the part of the surface that intersects with  $(-\infty, 0]^{m-1}$  since we are only interested in negative directional derivatives for our minimization task. This part of the surface of the ellipsoid is described by all unit vectors that are contained between the specific unit vectors  $\hat{\mathbf{u}}^{\text{ex},i}$ ,  $i \in \{0, 1, \dots, m-1\}$  for which

the multi-objective directional derivative  $\nabla_{\hat{\mathbf{u}}^{\text{ex},i}} \mathbf{f}(\mathbf{x})$  is an extremum in the  $i$ -th objective. This set of vectors which we will denote  $U^{\text{ex}}$  is contains all vectors that are obtained by taking a linearly weighted sum of the extrema unit vectors  $\hat{\mathbf{u}}^{\text{ex},i}$ . Note that these vectors have to be normalized to ensure that they are unit vectors:

$$U^{\text{ex}} = \left\{ \frac{\sum_{i=0}^{m-1} \alpha_i \hat{\mathbf{u}}^{\text{ex},i}}{\left\| \sum_{i=0}^{m-1} \alpha_i \hat{\mathbf{u}}^{\text{ex},i} \right\|} \mid \alpha_i \in [0, 1], \sum_{i=0}^{m-1} \alpha_i = 1 \right\} \quad (5)$$

However, set  $U^{\text{ex}}$  isn't tight enough because it can still involve directions for which not all components of the multi-objective directional derivative are negative *simultaneously*. This is illustrated for our example case in Figure 1. Hence we need  $m$  additional constraints to get the desired set  $U^{\text{neg-ex}}$ :

$$U^{\text{neg-ex}} = \left\{ \frac{\sum_{i=0}^{m-1} \alpha_i \hat{\mathbf{u}}^{\text{ex},i}}{\left\| \sum_{i=0}^{m-1} \alpha_i \hat{\mathbf{u}}^{\text{ex},i} \right\|} \mid \alpha_i \in [0, 1], \sum_{i=0}^{m-1} \alpha_i = 1, \nabla_{\sum_{i=0}^{m-1} \alpha_i \hat{\mathbf{u}}^{\text{ex},i}} f_j(\mathbf{x}) \leq 0 \right\} \quad (6)$$

The ultimate goal is to take a random direction from set  $U^{\text{neg-ex}}$ . A straightforward way to do so using Equation 6 is to repeat drawing values for the  $\alpha_i$  until the  $m$  inequality constraints are met. However, the number of unit vectors for which all directional derivatives are at most 0 may be significantly smaller than the number of unit vectors for which at least one directional derivative is smaller than 0, especially if  $m$  becomes larger. Hence it would be convenient to have a sampling method without constraints. To this end we first find all  $m$  unit vectors within set  $U^{\text{ex}}$  that map to corner points of  $U^{\text{neg-ex}}$ , i.e. all unit vectors  $\hat{\mathbf{u}}^{\text{neg-ex},i}$  for which the multi-objective directional derivative is negative in the  $i^{\text{th}}$  direction and zero in all other directions. This means that we must solve  $\nabla_{\hat{\mathbf{u}}^{\text{neg-ex},i}} \mathbf{f}(\mathbf{x}) = -\lambda_i \hat{\mathbf{e}}^i$  where  $\hat{\mathbf{e}}^i$  is a unit column vector with a 1 at position  $i$  and zeros at all other positions and  $\lambda$  is a non-negative real value,  $\lambda \geq 0$ . Moreover, there is a unique solution to this equation because in the objective space solving this equation means finding the intersection of the negative part of an axis with the surface of an ellipsoid that is centered at the origin. Using the definition in Equation 4 we obtain:

$$\mathbf{G} \hat{\mathbf{u}}^{\text{neg-ex},i} = -\lambda_i \hat{\mathbf{e}}^i \Leftrightarrow \mathbf{G} \left( \frac{1}{\lambda_i \|\hat{\mathbf{u}}^{\text{neg-ex},i}\|} \right) \hat{\mathbf{u}}^{\text{neg-ex},i} = -\hat{\mathbf{e}}^i \quad (7)$$

Since  $\frac{1}{\lambda_i \|\hat{\mathbf{u}}^{\text{neg-ex},i}\|}$  is just a scaling factor of  $\frac{1}{\lambda_i \|\hat{\mathbf{u}}^{\text{neg-ex},i}\|}$  in Equation 7 and Equation 7 must have a unique solution, also  $\mathbf{G} \hat{\mathbf{u}}^{\text{neg-ex},i} = -\hat{\mathbf{e}}^i$  must have a unique solution which we can use to find  $\hat{\mathbf{u}}^{\text{neg-ex},i} = \hat{\mathbf{u}}^{\text{neg-ex},i} / \|\hat{\mathbf{u}}^{\text{neg-ex},i}\|$  with. Because matrix  $\mathbf{G}$  is a  $m \times l$  matrix, it is generally speaking not square and cannot be inverted. However we know that  $\hat{\mathbf{u}}^{\text{neg-ex},i} \in U^{\text{ex}}$  and hence we can write  $\hat{\mathbf{u}}^{\text{neg-ex},i} = \sum_{j=0}^{m-1} \alpha_j^i \hat{\mathbf{u}}^{\text{ex},j}$  where we do not know the  $\alpha_j^i$ . We now have:

$$\mathbf{G} \left( \sum_{j=0}^{m-1} \alpha_j^i \hat{\mathbf{u}}^{\text{ex},j} \right) = \begin{bmatrix} \sum_{j=0}^{m-1} \alpha_j^i (\nabla f_0(\mathbf{x}))^T \hat{\mathbf{u}}^{\text{ex},j} \\ \vdots \\ \sum_{j=0}^{m-1} \alpha_j^i (\nabla f_{m-1}(\mathbf{x}))^T \hat{\mathbf{u}}^{\text{ex},j} \end{bmatrix} \quad (8)$$

We define a matrix  $\mathbf{D}^i$  such that  $\mathbf{D}_{kj}^i = (\nabla f_k(\mathbf{x}))^T \hat{\mathbf{u}}^{\text{ex},j}$  and a vector  $\boldsymbol{\alpha}^i$  such that  $\boldsymbol{\alpha}^i_k = \alpha_k^i$ . Then  $\mathbf{G} \hat{\mathbf{u}}^{\text{neg-ex},i}$  can be rewritten as  $\mathbf{D}^i \boldsymbol{\alpha}^i$ . Moreover, because  $\mathbf{D}^i$  is a square matrix, we can find a unique solution for  $\hat{\mathbf{u}}^{\text{neg-ex},i}$  through  $\boldsymbol{\alpha}^i$  if we solve  $\mathbf{D}^i \boldsymbol{\alpha}^i = -\hat{\mathbf{e}}^i$  by inverting  $\mathbf{D}^i$ , i.e.  $\boldsymbol{\alpha}^i = (\mathbf{D}^i)^{-1} \hat{\mathbf{e}}^i$ . Finally, we note that  $\hat{\mathbf{u}}^{\text{ex},j}$ , the extremum in the negative direction of the multi-objective directional derivative in the  $j^{\text{th}}$  dimension is, using the result of Equation 3, of course just  $-(\nabla f_j(\mathbf{x}))$ . We can therefore now write:

$$\hat{\mathbf{u}}^{\text{neg-ex},i} = -\frac{\sum_{i=0}^{m-1} \left( (\mathbf{D}^i)^{-1} \hat{\mathbf{e}}^i \right)_i (\nabla f_i(\mathbf{x}))}{\left\| \sum_{i=0}^{m-1} \left( (\mathbf{D}^i)^{-1} \hat{\mathbf{e}}^i \right)_i (\nabla f_i(\mathbf{x})) \right\|} \quad (9)$$

This latter result now allows us to rewrite equation 6 and define the set  $U^{\text{neg-ex}}$  of unit directions to choose from for multi-objective gradient search without the inequality constraint, but in the same fashion as set  $U^{\text{neg-ex}}$  instead:

$$U^{\text{neg-ex}} = \left\{ \frac{\sum_{i=0}^{m-1} \beta_i \hat{\mathbf{u}}^{\text{neg-ex},i}}{\left\| \sum_{i=0}^{m-1} \beta_i \hat{\mathbf{u}}^{\text{neg-ex},i} \right\|} \mid \beta_i \in [0, 1], \sum_{i=0}^{m-1} \beta_i = 1 \right\} \quad (10)$$

## 4. GRADIENT-BASED HYBRID MOEAS

### 4.1 Base MOEA

The base MOEA we use is the naive MIDEA [3]. This MOEA is an EDA specifically designed for multi-objective optimization. It has been shown to give good results on a wide variety of problems defined in both discrete and continuous parameter spaces. Moreover, it is fast and easy to understand, making it a good baseline algorithm. The following gives a brief overview of its main features. For specific details the interested reader is referred to the literature [3].

The naive MIDEA maintains a population of size  $n$ . In each generation it selects a subset of this population of size  $\lceil \tau n \rceil$ ,  $\tau \in [\frac{1}{n}, 1)$ , to perform variation with. By means of variation  $n - \lceil \tau n \rceil$  new solutions are generated which replace the solutions in the population that were not selected.

Selection is performed using a diversity-preserving selection operator. Since the goal in multi-objective optimization is both to get close to the Pareto optimal front and to get a good diverse representation of that front, a good selection operator must exert selection pressure with respect to both aspects. The selection operator in the naive MIDEA does this by using truncation selection on the basis of domination count (i.e. the number of times a solution is dominated). If the number of non-dominated solutions exceeds the targeted selection size  $\lceil \tau n \rceil$ , a nearest-neighbour heuristic in the objective space is used to ensure that a well-spread, representative subset of all non-dominated solutions is chosen.

The variation operator is geometrical in nature and is specifically designed to provide an advantage over traditional variation operators. The selected solutions are first clustered in the objective space. Subsequently, the actual variation takes place only between individuals in the same cluster, i.e. a mating restriction is employed. The rationale is that variation inside each cluster can process specific information about the different regions along the Pareto front. Such a parallel exploration automatically gives a better probability of obtaining a well-spread set of offspring solutions. To further stimulate diversity along the Pareto front each new offspring solution is constructed from a randomly chosen cluster. Variation inside a single cluster is done by estimating a one-dimensional normal-distribution for each variable separately.

### 4.2 Hybrid MOEAs

#### 4.2.1 General hybrid MOEA framework

The hybridization scheme that we employ is a generational one. At the end of a generation, i.e. after one step of selection, variation and re-evaluation is finished, a set of candidate solutions is determined. The gradient-based local search operator is then applied with each of these candidate solutions as a starting point. We identify three classes of candidate-solution sets:

1. *The entire population*

This allows for the widest search and the largest probability of discovering of new local optima, but also has the largest chance of finding improvements that are not better than the current non-dominated solutions.

2. *The selected solutions*

Since the set of selected solutions contains all promising solutions from the viewpoint of the MOEA, this set is a logical choice for selecting solutions to attempt to improve. This set represents a rational trade-off choice between the entire population and the set of the non-dominated solutions only.

3. *The non-dominated solutions*

Although it is directly beneficial if solutions from this set can be improved upon, there is also a considerable probability that the solutions in this set lie in a multi-objective local optimum and hence cannot be improved further using local search.

To control the ratio between how much search effort is spent by the MOEA and how much search effort is spent by the gradient-based local search operator, we introduce a ratio parameter  $\rho_e$ . In our scheme we aim to keep the ratio of the number of evaluations required by the gradient-based local search operator and the total number of evaluations required so far equal to  $\rho_e$ . To this end, the gradient-based local search operator is applied only as long as the current actual ratio is smaller than  $\rho_e$ . Moreover, the order in which the candidate solutions are searched is randomized. We do not suggest that this hybridization approach is optimal, but it will point out whether a gradient-based local search algorithm can aid in real-valued multi-objective optimization.

#### 4.2.2 Random-objective conjugate gradients

In this straightforward approach single-objective gradient-based local search is applied to a randomly chosen objective. Because the local search is now focused on a single objective, more advanced single-objective algorithms known from literature can be used. We use the conjugate gradients algorithm and call the resulting strategy ROCG (Random-Objective Conjugate Gradients). It depends completely on the correlation between the objectives whether the best local improvement in a single objective also leads to an improvement in the other objectives. Since this is typically not the case, the percentage of local searches that leads to an improvement (i.e. a solution that dominates the solution from where the local search started) is expected to be small.

#### 4.2.3 Alternating-objective repeated line-search

To reduce the probability of improving a single objective while making the objective value in the other objective worse, the objective that is searched locally can be altered during local search. This still allows the use of well-known single-objective approaches. However, care should be taken in the design because it does not make sense to let the local search in a single objective converge to a minimum. Doing so results in the same approach as ROCG. Hence we propose to perform a line-search (i.e. find a local minimum with respect to a single direction) in a single, alternatingly chosen objective in the direction of the negative gradient of that objective. This process is repeated until a multi-objective local minimum is found. In our multi-objective setting we propose to switch to a randomly chosen different objective once a line-search has terminated. We refer to this strategy as AORL (Alternating-Objective Repeated Line-search).

#### 4.2.4 Combined-objectives repeated line-search

This final strategy uses the results from Section 3.2. The approach is best described as a multi-objective version of gradient descent. A line-search is performed in a promising direction. When the line-search terminates, a new line-search is initiated in a new promising direction found at the new location. It was shown in Section 3.2 that there is a set of non-dominated directions that are all most promising. These directions are described by equation 10. Hence, when a line-search terminates, a random vector of  $\beta$  is drawn such that each  $\beta_i \in [0, 1]$  and  $\sum_{i=0}^{m-1} \beta_i = 1$ . Equation 10 is then used to obtain a new direction. Note that a line-search in this case does not aim to minimize a single objective but aims to find the best non-dominated solution in the given direction, i.e. the line-search in this strategy is a multi-objective search algorithm as well. We refer to this strategy as CORL (Combined-Objectives Repeated Line-search).

## 5. EXPERIMENTS

### 5.1 Setup

#### 5.1.1 Multi-objective optimization problems

The problems we have used have been taken from the literature on designing difficult and interesting multi-objective optimization problems and on comparing various MOEAs [5, 21]. Specifically, we have used the problems known as  $EC_i$ ,  $i \in \{1, 2, 3, 4, 6\}$ . For specific details regarding the difficulty of these problems we refer the interested reader to the indicated literature. Here we only present their definition.

Name	Objectives	Domain
$EC_1$	$f_0 = x_0, \quad f_1 = \gamma \left(1 - \sqrt{f_0/\gamma}\right)$ $\gamma = 1 + 9 \left(\sum_{i=1}^{l-1} x_i / (l-1)\right)$	$[0, 1]^{30}$ ( $l = 30$ )
$EC_2$	$f_0 = x_0, \quad f_1 = \gamma \left(1 - (f_0/\gamma)^2\right)$ $\gamma = 1 + 9 \left(\sum_{i=1}^{l-1} x_i / (l-1)\right)$	$[0, 1]^{30}$ ( $l = 30$ )
$EC_3$	$f_0 = x_0$ $f_1 = \gamma \left(1 - \sqrt{f_0/\gamma} - (f_0/\gamma)\sin(10\pi f_0)\right)$ $\gamma = 1 + 9 \left(\sum_{i=1}^{l-1} x_i / (l-1)\right)$	$[0, 1]^{30}$ ( $l = 30$ )
$EC_4$	$f_0 = x_0, \quad f_1 = \gamma \left(1 - \sqrt{f_0/\gamma}\right)$ $\gamma = 1 + 10(l-1) + \sum_{i=1}^{l-1} (x_i^2 - 10\cos(4\pi x_i))$	$[-1, 1] \times [-5, 5]^9$ ( $l = 10$ )
$EC_6$	$f_0 = 1 - e^{-4x_0} \sin^6(6\pi x_0), \quad f_1 = \gamma \left(1 - (f_0/\gamma)^2\right)$ $\gamma = 1 + 9 \left(\sum_{i=1}^{l-1} x_i / (l-1)\right)$ <sup>0.25</sup>	$[0, 1]^{10}$ ( $l = 10$ )

#### 5.1.2 Performance indicator

To measure performance we only consider the subset of all non-dominated solutions in the population upon termination. We call such a subset an approximation set and denote it by  $\mathcal{S}$ . A performance indicator is a function of approximation sets  $\mathcal{S}$  and returns a real value that indicates how good  $\mathcal{S}$  is in some aspect. More detailed information regarding the importance of using good performance indicators for evaluation may be found in literature [2, 11, 22].

Here we use a single performance indicator that is based on knowledge of the optimum, i.e. the Pareto-optimal front. We define the distance  $d(\mathbf{x}^0, \mathbf{x}^1)$  between two multi-objective solutions  $\mathbf{x}^0$  and  $\mathbf{x}^1$  to be the Euclidean distance between their objective values  $f(\mathbf{x}^0)$  and  $f(\mathbf{x}^1)$ . The performance indicator we use computes the average of the distance to the closest solution in an approximation set  $\mathcal{S}$  over all solutions in the Pareto-optimal set  $\mathcal{P}_S$ . We denote this indicator by

$D_{\mathcal{P}_F \rightarrow \mathcal{S}}$  and refer to it as the distance from the Pareto-optimal front to an approximation set. A smaller value for this performance indicator is preferable and a value of 0 is obtained if and only if the approximation set and the Pareto-optimal front are identical. This indicator is ideal for evaluating performance if the optimum is known because it describes how well the Pareto-optimal front is covered and thereby represents an intuitive trade-off between the diversity of the approximation set and its proximity (i.e. closeness to the Pareto-optimal front). Even if all points in the approximation set are on the Pareto-optimal front the indicator is not minimized unless the solutions in the approximation set are spread out perfectly.

Because the Pareto-optimal front may be continuous a line integration over the entire Pareto front is required in the definition of the performance indicator. In a practical setting, it is easier to compute a uniformly sampled set of many solutions along the Pareto optimal front and to use this discretized representation of  $\mathcal{P}_F$  instead. We have used this approach using 5000 uniformly sampled points. The performance indicator now is defined as follows:

$$D_{\mathcal{P}_F \rightarrow \mathcal{S}}(\mathcal{S}) = \frac{1}{|\mathcal{P}_S|} \sum_{\mathbf{x}^1 \in \mathcal{P}_S} \min_{\mathbf{x}^0 \in \mathcal{S}} \{d(\mathbf{x}^0, \mathbf{x}^1)\} \quad (11)$$

#### 5.1.3 General algorithmic setup

For selection we set  $\tau$  to 0.3, conforming to earlier work [3] and the rule-of-thumb for FDA [16]. We allowed gradient-based local search operators 10 iterations each time they were called. We set  $\rho_e \in \{0, 0.1, 0.25, 0.5\}$ . Note that for  $\rho_e = 0$  the pure naive MIDEA is obtained. Gradient information was approximated when required using  $\Delta x_i = 10^{-13}$ . Furthermore, we have used the Polak-Ribiere variant of the conjugate gradient algorithm [17]. All reported results were averaged over 30 runs, except for the convergence plots, which were averaged over 100 runs.

It is important to note that all variables have a bounded range. If the variables move outside of this range, some objective values can become non-existent. It is therefore important to keep the variables within their ranges. However, a simple repair mechanism that changes a variable to its boundary value if it has exceeded this boundary value gives artifacts that may lead us to draw false conclusions about the performance of the tested MOEAs. If for instance the search probes a solution that has all negative values for each of the variables  $x_i$  with  $i \geq 1$ , then the repair mechanism in the case of all problems except problem  $EC_6$  sets all these variables to 0. This is especially well possible during a gradient-search procedure because the gradient with respect to the second objective points in the direction of all negative values for variables  $x_i$  with  $i \geq 1$ . It is not hard to see that the solution resulting after boundary repair lies on the Pareto front. We have therefore adapted the local search operators such that local search never changes a solution into one that lies out of the problem range. Similarly, the sampling procedure of the naive MIDEA is changed to prevent the generation of solutions that are out of bounds.

## 5.2 Results

In Table 1 the average  $D_{\mathcal{P}_F \rightarrow \mathcal{S}}$  value is shown for all tested MOEAs with a maximum of either 20000 or 200000 evaluations. All population sizes between 0 and 1000 with a stepsize of 25 were tested and ultimately the one that lead to the best average indicator value was chosen.

Grad.	$C$	Max. eval. = 20000					Max. eval. = 200000				
		EC <sub>1</sub> (10 <sup>1</sup> )	EC <sub>2</sub> (10 <sup>1</sup> )	EC <sub>3</sub> (10 <sup>1</sup> )	EC <sub>4</sub>	EC <sub>6</sub>	EC <sub>1</sub> (10 <sup>3</sup> )	EC <sub>2</sub> (10 <sup>3</sup> )	EC <sub>3</sub> (10 <sup>2</sup> )	EC <sub>4</sub>	EC <sub>6</sub> (10 <sup>2</sup> )
None	-	<b>0.43</b>	<b>0.83</b>	<b>1.17</b>	<b>3.28</b>	<b>1.21</b>	<b>1.88</b>	<b>2.10</b>	<b>0.56</b>	1.58	<b>1.17</b>
$\rho_e = 0.10$											
ROCG	P	0.53	1.33	1.31	4.31	1.44	2.07	2.28	0.61	1.55	1.40
ROCG	S	0.54	1.11	1.29	3.89	1.43	2.11	2.26	0.64	1.49	1.32
ROCG	N	0.52	1.08	1.28	4.07	1.38	2.13	2.28	0.63	1.71	1.37
AORL	P	0.49	1.25	1.30	4.03	1.39	2.12	2.28	0.62	1.60	1.42
AORL	S	0.54	1.29	1.28	3.75	1.41	2.08	2.32	0.62	1.36	1.55
AORL	N	0.51	1.28	1.31	3.97	1.41	2.05	2.35	0.63	1.38	1.55
CORL	P	0.50	1.14	1.29	4.47	1.42	2.04	2.26	0.61	1.07	1.40
CORL	S	0.51	1.15	1.28	4.22	1.46	2.03	2.26	0.58	1.10	1.41
CORL	N	0.50	1.06	1.27	4.10	1.42	2.02	2.27	0.59	1.25	1.40
$\rho_e = 0.25$											
ROCG	P	0.73	1.54	1.49	5.22	1.62	2.50	2.75	0.72	1.78	1.88
ROCG	S	0.72	1.77	1.58	4.86	1.62	2.51	2.73	0.74	1.69	1.74
ROCG	N	0.76	1.89	1.57	4.35	1.69	2.53	2.73	0.74	1.62	1.86
AORL	P	0.74	1.50	1.56	3.50	1.54	2.50	2.83	0.72	1.57	2.08
AORL	S	0.73	1.86	1.55	3.49	1.64	2.53	2.89	0.78	1.64	2.35
AORL	N	0.74	1.69	1.61	4.64	1.65	2.45	2.84	0.77	1.44	2.14
CORL	P	0.67	1.59	1.50	3.62	1.80	2.35	2.59	0.71	1.02	1.65
CORL	S	0.68	1.76	1.43	3.37	1.67	2.28	2.53	0.64	1.20	2.16
CORL	N	0.68	1.68	1.44	3.39	1.55	2.33	2.53	0.66	1.31	1.86
$\rho_e = 0.50$											
ROCG	P	1.44	3.37	2.04	6.34	2.37	3.94	4.34	1.05	1.82	4.72
ROCG	S	1.43	3.20	2.22	6.01	2.44	3.91	4.35	1.19	2.00	6.40
ROCG	N	1.39	3.22	2.20	6.60	2.34	3.95	4.54	1.20	1.60	7.01
AORL	P	1.40	3.30	2.31	<b>3.25</b>	2.23	3.88	4.57	1.23	1.36	3.45
AORL	S	1.43	3.19	2.28	3.80	2.19	3.94	4.52	1.26	1.92	5.93
AORL	N	1.39	3.32	2.28	5.30	2.31	3.86	4.50	1.22	2.12	4.70
CORL	P	1.19	3.08	1.92	3.15	2.16	3.27	3.77	0.99	<b>0.82</b>	4.79
CORL	S	1.14	2.97	1.81	3.54	2.29	3.15	3.55	0.82	1.17	5.19
CORL	N	1.05	3.11	1.74	4.76	2.20	3.28	3.70	0.80	1.48	5.57

Table 1: Average  $D_{P_F \rightarrow S}$  metric for all tested MOEAs and all benchmark problems for two different maximum number of evaluations.  $C$  indicates the set of candidates, P = population, S = selected solutions, N = non-dominated solutions.

The first question to answer is which local gradient-based search technique is the most promising. The hybridization based on CORL almost always gives better results than the other approaches. The superiority of CORL is emphasized further in Table 2 where it is clear that CORL has by far the largest probability of improving a solution. Hence, if local gradient-based search is to be used, the CORL approach proposed in this paper is the most promising one.

The next question to answer is whether the additional use of local gradient-based search in MOEAs is a useful approach at all. The results in Table 1 show a dominating performance of the base MOEA alone for all problems except EC<sub>4</sub>. Moreover, the the performance generally deteriorates with an increased number of evaluations allowed to be spent by the local search operator (i.e. a larger  $\rho_e$ ). This would at first sight indicate that the added use of local gradient-based search generally speaking only hampers MOEAs. However, this general statement can be differentiated when observing the results more thoroughly.

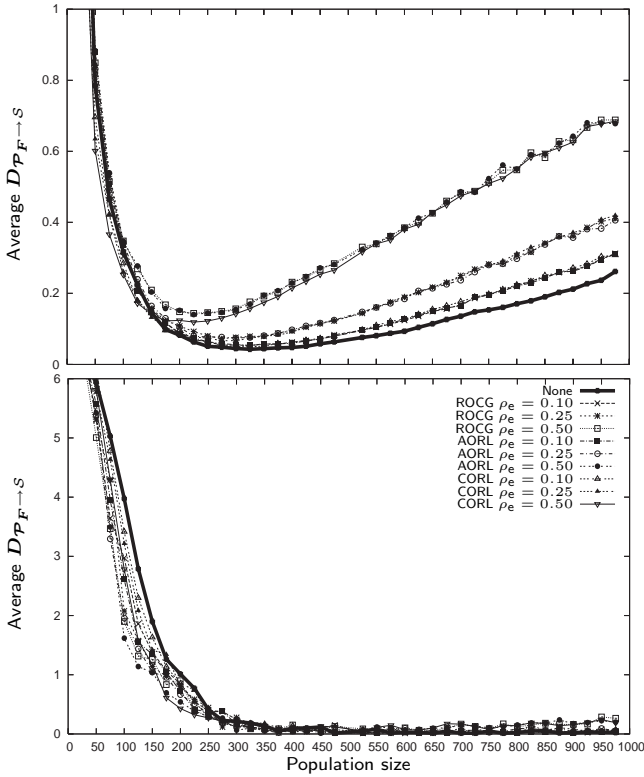
First of all, local search is expensive. Finding a local minimum requires a relatively large amount of resources compared to a global search across the entire landscape. Moreover, in the multi-objective case local search is relatively speaking even more expensive because there are typically many (Pareto-) optimal solutions and a single local search only improves a single solution. An EA on the other hand improves multiple solutions simultaneously (if variation is successful). In multi-objective optimization this may involve the improvement of large parts of the Pareto-front. Hence, local search benefits are better visible if more evaluations are allowed, especially in the multi-objective case. Indeed, Table 1 shows that for EC<sub>4</sub> the improvement of the

Grad.	$C$	Max. eval. = 20000					Max. eval. = 200000				
		EC <sub>1</sub>	EC <sub>2</sub>	EC <sub>3</sub>	EC <sub>4</sub>	EC <sub>6</sub>	EC <sub>1</sub>	EC <sub>2</sub>	EC <sub>3</sub>	EC <sub>4</sub>	EC <sub>6</sub>
None	-	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\rho_e = 0.10$											
ROCG	P	0.0	10.6	9.0	0.6	17.6	0.0	8.4	5.9	0.8	9.0
ROCG	S	0.0	11.9	2.7	0.8	20.1	0.0	3.8	1.3	0.4	4.9
ROCG	N	0.0	10.6	3.1	0.2	17.1	0.0	3.3	1.3	0.3	4.6
AORL	P	0.0	58.1	2.1	16.7	39.2	0.0	19.2	5.2	15.6	21.8
AORL	S	0.0	49.7	0.5	6.5	40.4	0.0	14.7	0.0	3.2	14.4
AORL	N	0.0	53.0	0.0	6.0	41.3	0.0	11.2	0.0	1.4	10.2
CORL	P	99.2	95.6	80.8	91.2	88.2	96.9	93.1	78.1	76.2	24.7
CORL	S	97.6	93.0	93.7	77.6	85.2	97.9	94.9	95.7	55.7	17.5
CORL	N	97.2	90.6	91.6	74.4	82.3	98.7	96.3	97.0	58.4	15.5
$\rho_e = 0.25$											
ROCG	P	0.0	10.5	8.8	0.9	14.4	0.0	8.1	6.4	1.7	9.8
ROCG	S	0.0	10.1	2.9	0.5	14.7	0.0	3.7	1.2	0.2	4.7
ROCG	N	0.0	9.3	2.7	0.6	13.1	0.0	2.6	0.9	0.3	3.4
AORL	P	0.0	51.6	2.9	18.0	31.3	0.0	19.2	5.0	17.3	21.2
AORL	S	0.0	46.6	0.1	9.1	32.2	0.0	14.0	0.0	3.6	13.5
AORL	N	0.0	45.1	0.2	4.9	27.5	0.0	10.0	0.0	0.9	8.9
CORL	P	97.3	93.7	78.9	87.1	82.3	93.9	89.6	73.9	69.3	22.7
CORL	S	96.1	94.6	90.7	87.5	75.8	95.2	93.3	92.9	24.7	23.6
CORL	N	96.6	91.3	91.2	53.3	75.4	97.6	94.9	94.5	42.3	20.6
$\rho_e = 0.50$											
ROCG	P	0.0	11.2	9.0	1.7	15.3	0.0	7.7	6.2	2.4	11.2
ROCG	S	0.0	9.5	2.8	1.0	20.4	0.0	3.8	1.5	0.4	5.4
ROCG	N	0.0	8.1	3.3	0.9	7.5	0.0	2.5	1.1	0.3	2.5
AORL	P	0.0	44.1	1.9	13.1	19.7	0.0	19.7	4.8	12.7	19.8
AORL	S	0.0	51.8	0.2	11.5	25.8	0.0	15.0	0.0	3.6	12.4
AORL	N	0.0	35.1	0.1	2.8	20.3	0.0	9.6	0.0	0.5	8.1
CORL	P	94.2	86.2	77.3	80.9	80.0	89.4	83.6	72.8	63.8	42.3
CORL	S	92.7	85.8	88.5	68.0	68.3	91.3	88.4	82.8	41.6	25.3
CORL	N	94.6	76.7	90.1	45.7	58.6	94.7	88.8	88.7	37.5	21.5

Table 2: Average percentage of calls to the local-search operator that resulted in an improvement (i.e. a dominating solution) for all tested MOEAs and all benchmark problems for two different maximum number of evaluations.  $C$  indicates the same as in Figure 1.

base MOEA combined with CORL over base MOEA alone increases if 200000 evaluations are allowed instead of 20000. But still, for all other problems, the base MOEA combined with CORL does not outperform the base MOEA alone.

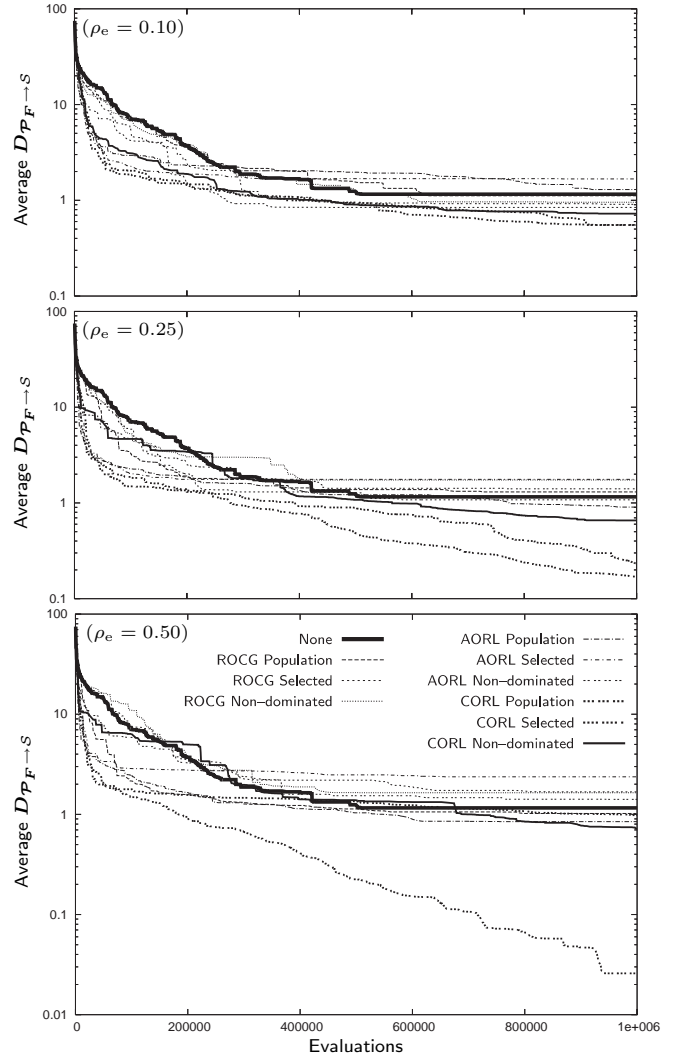
Interestingly enough, although the final outcome is worse than using the base MOEA alone, for problems EC<sub>1</sub>, EC<sub>2</sub> and EC<sub>3</sub> the CORL approach does reach very high probabilities of improving a solution. So, local search doesn't fail in this case. The reason why the base MOEA alone in the end is still better for a fixed maximum number of evaluations is that these three problems are relatively speaking "too easy" for the base MOEA. For these problems, the base MOEA is already able to move the Pareto front by improving many solutions simultaneously. Gradient-search can then not provide additional help fast enough because the number of evaluations required before a solution is improved is relatively large, making local search much more expensive. The ROCG approach requires on average 72 evaluations per local search call, but has a very low improvement-ratio. The AORL approach has a slightly better improvement-ratio but requires 124 evaluations per call on average. Although the the CORL approach has a very good improvement-ratio, it does require 316 evaluations per call on average. Thus, the MOEA may actually be hampered if local search is used because it could have moved many solutions simultaneously using the same number of evaluations. The fundamental difference with single-objective optimization should be noted here. The added use of gradient-search applied to only a single solution can only help if the number of non-dominated solutions moved simultaneously by the base MOEA is not too large. The number of solutions for which this balance is tipped towards the hybrid side or the non-hybrid side is of



**Figure 2:** Average  $D_{\mathcal{P}_F \rightarrow S}$  indicator values when using the complete population as the candidate set for local search. *Top: EC<sub>1</sub>, 20000 evaluations. Bottom: EC<sub>6</sub>, 200000 evaluations.*

course problem specific. The idea of less solutions to move corresponds to smaller population sizes in base MOEA. Indeed, for smaller population sizes, the results of the hybridized MOEA are better. Only when the population size increases can the base MOEA perform better. Although this behavior was observed across the board, two illustrative examples are provided in Figure 2.

Generally speaking, the additional use of gradient-based local search at rate  $\rho_e$  is only useful if the part of the overall improvement contributed by the local search is also at least  $\rho_e$ . The main reason why this ratio was not obtained on problems EC<sub>1</sub>, EC<sub>2</sub> and EC<sub>3</sub> is because the base MOEA could itself reach the optimal Pareto-front with relatively little evaluations and could hence improve many non-dominated solutions at the same time. On the EC<sub>4</sub> problem however, reaching the optimal Pareto-front is hard for the base MOEA alone. Moreover, the CORL approach reaches high probabilities of improving a solution for this problem. Because it is hard for the base MOEA to improve the solutions, the relative contribution to the improvement of the CORL local-search operator is large enough for the hybrid MOEA to lead to better results. CORL can help to move some solutions closer to optimal Pareto-front after which the MOEA is able to find solutions that lie on a similar front as that solution and so advance the entire front. The CORL approach now already helps quickly as can be seen from Table 1. But it helps even more in the longer run as can be seen from the same Figure as well as from Figure 3 in which convergence graphs are shown for a population size of 500 and a maximum number of evaluations of  $1 \cdot 10^6$ . Note again that only the added use of CORL really makes a difference.



**Figure 3:** Convergence graphs for various settings of  $\rho_e$  for all tested hybrid MOEAs on the EC<sub>4</sub> problem.

For problem EC<sub>6</sub> the base MOEA alone comes close to the optimal Pareto-front only after 200000 evaluations. This is a lot more than for problems EC<sub>1</sub>, EC<sub>2</sub> and EC<sub>3</sub>, where only 20000 evaluations are required to come close to the optimal Pareto-front. Hence, the base MOEA isn't very good on the EC<sub>6</sub> problem either. However, for this problem gradient information is simply less helpful upon approaching the optimal front as can be seen from Table 2. The probability of success for the CORL approach heavily decreases on problem EC<sub>6</sub> as more evaluations are allowed.

## 6. DISCUSSION

The investigation of the results in the previous Section has provided us with explanations for the observed behavior. On problems EC<sub>1</sub>, EC<sub>2</sub> and EC<sub>3</sub>, both the MOEA and the CORL approach work well. However, because the MOEA is capable of improving multiple solutions at the same time, which leads to less required evaluations for an overall bigger improvement, hybridization stands in the way of improvement. On the EC<sub>6</sub> problem the opposite happens. Both the base MOEA and the CORL approach do not work well. Still, the MOEA approach is relatively more useful for the same reasons and hence hybridization again stands in the

way of improvement, albeit less efficient in this case. Finally, on the EC<sub>4</sub> problem, the base MOEA performs even worse but the CORL approach performs reasonably well. For this reason we have observed superiority of the hybrid MOEA on the EC<sub>4</sub> problem.

Combined with the use of EAs, local gradient-based optimization is less fruitful in multi-objective optimization. A good hybridization scheme should thus be carefully constructed. It would be interesting to attempt to construct such a scheme that takes the observations made in this paper into account and to test it on the same benchmark as well as on additional problems of varying difficulty and perhaps a larger dimensionality, both in the sense of the number of objectives as the number of problem variables. Such a scheme would be interesting to subsequently apply to a real-world application.

## 7. CONCLUSIONS

We have presented a parameterized, easy-to-use description of the set of all non-dominated improving directions for any point in the parameter space of a multi-objective optimization problem. We have furthermore investigated the added use of gradient-based local search operators for numerical multi-objective optimization with MOEAs.

Our experiments show that the best way to exploit gradient information in multi-objective optimization is to use the set of all non-dominated maximally-improving directions described in this paper. However, the added use of gradient-based local search in a MOEA is only useful if the relative contribution it makes to the overall improvement is at least as big as the relative amount of resources it is allowed to spend. We have indicated that for multi-objective optimization this criterion is harder to achieve because MOEAs have the ability to advance multiple solutions simultaneously towards different regions of the optimal Pareto-front, giving it a bigger relative advantage than in the single-objective case. Hence, more often than in the single-objective case, the added use of local gradient-based search is not always efficient. Thus, although we have provided a solid basis for exploiting gradient information in numerical multi-objective evolutionary optimization, it should not be disregarded that our results also indicate that EAs really have an advantage over non-population-based approaches if the goal is to obtain a good approximation set (i.e. instead of only a single solution on the optimal Pareto-front).

## 8. REFERENCES

- [1] P. A. N. Bosman and D. Thierens. Exploiting gradient information in continuous iterated density estimation evolutionary algorithms. In B. Kröse et al., editors, *Proceedings of the 13th Belgium-Netherlands Artificial Intelligence Conference BNAIC'01*, pages 69–76, 2001.
- [2] P. A. N. Bosman and D. Thierens. The balance between proximity and diversity in multi-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7:174–188, 2003.
- [3] P. A. N. Bosman and D. Thierens. The naive MIDEA: a baseline multi-objective EA. In C. A. Coello Coello et al., editors, *Evolutionary Multi-Criterion Optimization – EMO'05*, pages 428–442, Berlin, 2005. Springer-Verlag.
- [4] M. Brown and R. E. Smith. Effective use of directional information in multi-objective evolutionary computation. In E. Cantú-Paz et al., editors, *Proceedings of the GECCO-2003 Genetic and Evolutionary Computation Conference*, pages 778–789, Berlin, 2003. Springer-Verlag.
- [5] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.
- [6] M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering pareto sets by multilevel subdivision techniques. *J. of Optimization Theory and Applications*, 124(1):113–136, 2005.
- [7] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.
- [8] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards*, 6:409–436, 1952.
- [9] H. Ishibuchi, T. Yoshida, and T. Murata. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans. on Evolutionary Computation*, 7:204–223, 2003.
- [10] J. D. Knowles and D. Corne. M-PAES: a memetic algorithm for multiobjective optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation – CEC-2000*, pages 325–332, Piscataway, New Jersey, 2000. IEEE Press.
- [11] J. D. Knowles and D. Corne. On metrics for comparing non-dominated sets. In *Proceedings of the 2002 Congress on Evol. Comp. CEC 2002*, pages 666–674, Piscataway, New Jersey, 2002. IEEE Press.
- [12] J. D. Knowles and D. W. Corne. A comparison of diverse approaches to memetic multiobjective combinatorial optimization. In W. Hart et al., editors, *Proceedings of the Workshop on Memetic Algorithms WOMA at the Genetic and Evolutionary Computation Conference – GECCO-2000*, pages 103–108, 2000.
- [13] M. Lahanas, D. Baltas, and S. Giannouli. Global convergence analysis of fast multiobjective gradient based dose optimization algorithms for high-dose-rate brachytherapy. *Phys. Med. Biol.*, 48:599–617, 2003.
- [14] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, New York, 1997.
- [15] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, 1989.
- [16] H. Mühlenbein and T. Mahnig. FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7:353–376, 1999.
- [17] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes In C: The Art Of Scientific Computing*. Cambridge University Press, Cambridge, 1992.
- [18] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette, editor, *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 93–100, Mahwah, New Jersey, 1985. Lawrence Erlbaum Associates, Inc.
- [19] S. Schäffler, R. Schultz, and K. Weinzierl. Stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization Theory and Applications*, 114(1):209–222, 2002.
- [20] E.-G. Talbi, M. Rahoual, M.-H. Mabed, and C. Dhaenens. New genetic approach for multicriteria optimization problems : Application to the flow-shop. In E. Zitzler et al., editors, *Evolutionary Multi-Criterion Optimization – EMO'01*, Berlin, 2001. Springer-Verlag.
- [21] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Computation*, 8(2):173–195, 2000.
- [22] E. Zitzler, M. Laumanns, L. Thiele, C. M. Fonseca, and V. Grunert da Fonseca. Why quality assessment of multiobjective optimizers is difficult. In W. B. Langdon et al., editors, *Proceedings of the GECCO-2002 Genetic and Evolutionary Computation Conference*, pages 666–674, San Francisco, California, 2002. Morgan Kaufmann.