PREFERENCE-BASED MULTI-OBJECTIVE DISTINCT CANDIDATES OPTIMIZATION

Peter Dueholm Justesen
Aarhus University
Department of Computer Science, Aabogade 34, DK-8200 Aarhus N, Denmark
juste@cs.au.dk

Rasmus K. Ursem
Grundfos R & T,
Grundfos Management A/S, Poul Due Jensens Vej 7, DK-8850 Bjerringbro, Denmark
ursem@cs.au.dk

Abstract Traditional evolutionary multi-objective optimization (EMO) algorithms typically return several hundred non-dominated candidate solutions. From a practical point-of-view, a small set of 5-10 distinct candidates is often preferred because post-processing of several hundred solutions may be too costly, too time-consuming, too difficult to compare design differences, or similar solutions may turn out to be statistically equal in prototyping and manufacturing. Interestingly, these limitations apply to most if not all real-world problems.

In this paper we introduce a novel approach to incorporating preferences in order to make an EMO algorithm return a small set of clearly different solutions with respect to performance and design. Here, we distinguish between generalized and domain-specific preferences, where generalized preferences address the aforementioned limitations and the domain-specific preferences cover the whishes of the decisison maker.

We further suggest the General Cluster-Forming Differential Evolution (GCFDE) algorithm complying to the approach of returning a small, diverse result set. The algorithm is tested on five well-known mechanical engineering problems and a real-world many-objective problem from electrical engineering. On all test problems, GCFDE located distinct optimal solutions, which shows that this is a promising approach for handling preferences in both multi- and many-objective settings.

Keywords: Multi-objective optimization, many-objective optimization, decision making, distinct candidates, MCDM, MODCO, diversity management.

1. Introduction

The application of multi-objective evolutionary algorithms (MOEAs) on a real-world problem typically consists of two steps. First, the optimization step where the problem is set up, the MOEA is run and all non-dominated solutions are gathered. Second, the decision making step where a few solutions to further investigate and perhaps implement are chosen among the non-dominated solutions found in the first step. In this process, the decision maker (DM) has to apply his preferences among the objectives to select the final solutions. Three popular approaches of doing this are:

1 **Aggregating** multiple objectives into one.

2 **Incorporating** decision support systems into the MOEA.

3 **Pruning** the result set of the MOEA.

However, there are several challenges connected to each of these approaches. For algorithms using approach 1, aggregating several objectives into one omits the possibility of exploring trade-offs between objectives by collapsing a population to a single point. For algorithms using approach 2, it can be hard for a DM to express preferences as decision support systems, as well as it is near impossible to distinguish between several hundred non-dominated solutions, which is necessary for a DM using algorithms complying to approach 3. Thus, these authors are inspired by approaches for evaluating non-dominated solutions, see [2].

In this paper, we introduce a novel approach for EMO algorithms addressing the aforementioned problems by directly incorporating different preferences into the algorithm. We deal with the challenges just described in the following ways:

1 Return only a *low number* of solutions.

2 Enable easy incorporation of *domain-specific preference functions*.

3 Ensure *performance and design distinctiveness* of solutions.

The small branch of MO termed Multi-Objective Distinct Candidates Optimization (MODCO) incorporates both *generalized preferences* and *domain-specific preferences* into the algorithm with the goal of finding a small set of 5-10 distinct candidates to make step 2 of the optimization process manageable. See [10] for extensive practival motivation, problem formulation and a survey of related research on the MODCO approach.

In MODCO, the concept *generalized preferences* ensures that the algorithm returns a **low-cardinality result set with distinct solutions**, which is desirable for most if not all real-world applications. In addition, *domain-specific preferences* covers indicator functions, directing the search toward *preferred areas* of the objective space.

The MODCO parameters K_{NC}, K_{PD} and K_{DD} constitute the generalized preferences, and must be implemented in MODCO algorithms to control result set cardinality and distinctiveness in objective and design space. Setting K_{PD} or K_{DD} to 0 express no demand for distinctiveness, while setting K_{PD} or K_{DD} to 1 express a wish for maximal distinctiveness in either objective or design space, respectively. For inbetween settings, K_{PD} and K_{DD} may express a demand for an intermediate level of distinctiveness.

1 **Number of candidates:** $K_{NC} \in [1 : \infty] \subseteq \mathbb{N}$
How many candidates is it practically and economically feasible to inspect, analyze, and compare in post-processing?

2 **Performance distinctiveness:** $K_{PD} \in [0.0 : 1.0] \subset \mathbb{R}$
How different should the candidates be in performance space?

3 **Design distinctiveness:** $K_{DD} \in [0.0 : 1.0] \subset \mathbb{R}$
How different should the candidates be in design space?

This paper present a novel algorithm complying to the MODCO goal of returning a few distinct candidates for constrained multi-objective problems. The algorithm is tested on five well known multi-objective constrained benchmark problems from mechanical engineering, as well as one many-objective real world problem from electrical engineering.

This paper is structured as follows. Section 2 introduces the General Cluster-Forming Differential Evolution algorithm (GCFDE), which is an extension of our earlier published CFDE [3]. In section 3, we describe test problems and provide performance comparison. Finally, section 4 concludes the paper.

2. The General Cluster-Forming Differential Evolution algorithm

The General Cluster-Forming Differential Evolution (GCFDE) algorithm is based on evolving K_{NC} subpopulations using Differential Evolution, each defining an objective and a design space **centroid**.

The search is based on a primary selection criterion (PSC) and a secondary selection criterion (SSC), which together defines a total ordering of individuals. The primary fitness is based on discrete Pareto-ranking, while the secondary fitness is applied according to diversity. The GCFDE algorithm variants are named as GCFDE/PSC/SSC, with SSC denoting the domain-specific preference based function.

GCFDE extends the first concrete algorithm complying to the goals of MODCO, CFDE [3], which was tested on unconstrained multi-objective problems with two and three objectives. To address the challenges of constrained many-objective optimization GCFDE differs from CFDE on a few accounts. All calculations now takes place in normalized objective and design space to ensure interval-independence. Further, we have:

1 An alternate centroid definition is introduced.

2 Primary selection now incorporates constraint handling.

3 Secondary selection now also handles design distinctiveness.

Algorithm 1 lists the pseudocode of the GCFDE algorithm, and nomenclature is found in Table 1. In pseudocode, $minObjDist(CO_i)$ denotes the function returning the minimum Euclidean distance from centroid CO_i to the nearest other centroid in normalized objective space, and $minDesDist(CD_i)$ denotes the corresponding function for normalized design space.

2.1 Centroid definition

To enhance convergence in a many-objective setting, an average of the non-dominated individuals with the best secondary fitness in P_i now defines the centroids both wrt. objectives and design parameters. Thus, when solving many-objective problems, we use the single individual with the highest secondary fitness in P_i to define centroids CO_i and CD_i, by setting $CO_i = f(x_{i,1})$ and $CD_i = d(x_{i,1})$ after truncation in the main loop. For multi-objective problems, we use the average placement of all N/K_{NC} individuals in each P_i as objective and design centroids, see [3].

2.2 Primary selection criteria

The primary selection criteria (PSC) assigns a rank to the individuals and it thereby determines the individuals to place in the highest ranked front, i.e., those individuals that are selected wrt. the SSC.

In this paper, we use the global constraint-domination **(GCD)** relation from GDE3 [5], which states that an individual x constraint-dominates individual y iff:

- x is feasible and y is not.
- x and y are infeasible and x dominates y in constraint space.
- x and y are feasible and x dominates y in objective space.

Table 1. Nomenclature for GCFDE

Symbol	**Meaning**	**Symbol**	**Meaning**
P	Population	P_i	Subpopulation i
CO_i	Objective space centroid of P_i	CD_i	Design space centroid of P_i
$f(x)$	Objective vector of x	$d(x)$	Design vector of x
x	Individual	$x_{i,j}$	x at the j'th index in P_i

Algorithm 1 General Cluster-Forming Differential Evolution

Require: Population size N, K_{NC}, K_{PD}, K_{DD}
Ensure: K_{NC} distinct, feasible, non-dominated individuals.
Initialize K_{NC} subpopulations with N/K_{NC} random individuals.
Assign to all individuals their rank as primary fitness.
Assign to all individuals $SF(x)$ given by preference based function.
while Halting criterion has not been met **do**
  Calculate all subpopulation centroids CO_i, CD_i
  Perform global $DE/rand/1/bin$ mating with replacement - store incomparable offspring.
  Migrate incomparable offspring to nearest subpopulation wrt. Euclidean distance to objective space centroids.
  Assign to all individuals their rank as primary fitness.
  for All $P_i \in P$ **do**
    if $minObjDist(CO_i) < K_{PD}/K_{NC}$ **then**
      $\forall\, x_{i,j} \in P_i$ assign $SF(x)$ according to Equation 1.
    else if $minDesDist(CD_i) < K_{DD}/K_{NC}$ **then**
      $\forall\, x_{i,j} \in P_i$ assign $SF(x)$ according to Equation 2.
    else
      $\forall\, x_{i,j} \in P_i$ assign $SF(x)$ given by preference based function.
    end if
  end for
  Truncate subpopulations to a size of N/K_{NC} by sorting wrt. rank first, then secondary fitness.
end while
Return K_{NC} distinct solutions, by making a final sorting of each P_i wrt. rank, then preference based secondary fitness, and returning $x_{i,1}$ for $i = 1...K_{NC}$.

2.3 Secondary selection criteria

The secondary selection criteria (SSC) determines which individuals are chosen from the highest ranked front to be included in the next generation. In GCFDE, the SSC firsts ensures performance distinctiveness, and then design distinctiveness. However, if both performance and design distinctiveness is achieved, GCFDE performs domain-specific preference based search. This allows concurrent application of both generalized and domain-specific preferences with MODCO parameters controlling the balance between the two. It is important to realize that the preference based search may be defined by **any** domain-specific preference criterion without violating the desire for returning distinct candidates.

As seen in Algorithm 1, the secondary fitness assignment may differ from subpopulation to subpopulation. Subpopulations that are partly overlapping in objective space selects next generation based on performance distinctiveness, while subpopulations not violating neither performance nor design distinctiveness performs preference based search.

GCFDE uses the secondary fitness assignment to fulfill distinctiveness requirements using the centroid distance measure introduced in [3]:

$$SF(x) = min(\{dist(f(x), CO_j), j = 1..K_{NC}, j \neq i\}) \tag{1}$$

That is, we assign the minimal Euclidean distance in normalized objective space to another centroid to each individual in P_i. As this measure is to be maximized, individuals close to another subpopulation centroid will be penalized, guiding subpopulations away from each other if they are too close wrt. K_{PD}. This also enhances clustering as individuals far away from their own centroid are more likely to be penalized.

If performance distinctiveness is achieved, but design distinctiveness is not, the procedure is performed in design space, assigning secondary fitness to each $x \in P_i$ according to Equation 2. Along with ensuring design distinctiveness, K_{DD} also controls in which extent the DM wishes a 1:1 correspondence between result set design and performance, by guiding individuals towards clusters in both objective and design space, if both K_{PD} and K_{DD} are high. Note, that these divergent SSCs are analogous, working in objective space and decision space, respectively.

$$SF(x) = min(\{dist(d(x), CD_j), j = 1..K_{NC}, j \neq i\}) \tag{2}$$

If both performance and design distinctiveness wrt. K_{PD} and K_{DD} is achieved, GCFDE performs preference based search. In this paper, we experiment with two well known domain-specific preference functions. The first such is the **weighted sum (WS)** function, aggregating multiple objectives into one fitness, by summing the weighted contribution of each using a weight vector supplied by the DM. The **WS** function guides the search towards the areas of objective space which are the most biased wrt. the weight vector supplied by the DM.

Another preference based utility function is the **knee utility (KNEE)** function proposed by Branke et al. [4], which is designed to discover knee regions by calculating an average fitness value for a large number of randomly sampled weight vectors, 100 vectors in our experiments. If the average fitness is good, the individual is more likely to reside in a knee-region. The knee utility function is generally applicable, and guide the search towards regions with good trade-offs between objectives.

3. Experiments and results

In our introducing article [3], the CFDE algorithm was tested on a large suite of standard benchmark problems, including ZDT [1], DTLZ [6] and knee problems [4]. On these problems, CFDE showed superior convergence compared to DEMO versions [7]. Further, parameter usage, diversity of results, and knee search were demonstrated.

In this paper, we compare GCFDE with GDE3 on a set of well-known benchmark problems from mechanical engineering and on a many objective real world problem from electrical engineering. This problem models a part of the control circuit for the Grundfos Alpha Pro pump, which is a small circulation pump for heating in private houses. This circuit is also used in the Alpha2 pump, see more at `www.grundfos.com/alpha2`.

3.1 Mechanical engineering problems

To investigate the **convergence capability** of GCFDE on constrained multi-objective problems, we use the Two Member Truss Design (TMTD), the Gear Train Design (GTD), the Multiple Disk Clutch Design (MDCD), the Spring Design (SD) and the Welded Beam Design (WBD) problem from [8]. These are all constrained and bi-objective.

To compare convergence performance, we investigate to which extent the GCFDE/GCD/KNEE result sets dominate the most similar solutions from the returned populations of GDE3. 20 runs have been performed for both GDE3 and for GCFDE on each test problem. For each generated result set of GCFDE, we compare each of the K_{NC} GCFDE individuals to their most similar counterpart from each of the GDE3 populations, i.e. the GDE3 individual being closest in normalized Euclidean objective space. This yields $K_{NC} \cdot 20 \cdot 20$ comparisons per problem, e.g. 2000 for $K_{NC} = 5$. This gives a percentage of the amount of dominating, dominated, incomparable and equal individuals produced by GCFDE, with relations defined as in [1]. Results are shown in Table 2.

For all problems and on both algorithms, we used a population size $N = 100$ along with DE parameters $F = 0.5$ and $CF = 0.3$. On all problems except the GTD problem, we have performed only 100 generations of both algorithms, whereas for the GTD problem 200 generations were performed to ensure convergence. For the GCFDE algorithm, we have used $K_{NC} = 5$, $K_{PD} = 0.5$ and $K_{DD} = 0.0$ on all runs.

Table 2. GCFDE/GCD/KNEE versus GDE3.

GCFDE vs. GDE3	TMTD	GTD	MDCD	SD	WBD
Dominates (%)	27.5	0.0	2.0	55.5	14.5
Variance (%)	± 5.0	± 0.0	± 5.0	± 4.8	± 8.3
Dominated (%)	3.0	5.7	0.0	32.4	18.0
Variance (%)	± 2.1	± 5.5	± 0.0	± 6.4	± 3.3
Incomparable (%)	69.5	1.3	8.4	12.1	67.5
Variance (%)	± 4.4	± 3.8	± 5.4	± 4.4	± 8.0
Equal (%)	0.0	93.0	91.4	0.0	0.0
Variance (%)	± 0.0	± 8.0	± 5.5	± 0.0	± 0.0

As seen in Table 2 the GCFDE algorithm outperforms the GDE3 algorithm on the TMTD and SD problems with the highest percentage of solutions dominating the GDE3 counterparts. Due to the continous objectives of these problems, there is also a high percentage of incomparable solutions on the TMTD problem.

For the problems with some discrete objectives, GTD and MDCD, the two algorithms find roughly identical solutions, and only few are incomparable. Here, the two algorithms seems to have equal performance. Taking variance into account, this also goes for the WBD problem with a similar percentage dominating/dominated solutions produced by GCFDE, along with a high percentage of incomparable solutions.

Overall, the GCFDE algorithm appears to perform equal to or better than the GDE3 algorithm on the problem set, giving confidence in the ability of GFCDE to converge to the true Pareto-front of constrained multi-objective problems.

To illustrate the **diversity** of the result sets of GCFDE wrt. K_{PD}, the GCFDE algorithm was run with different K_{PD} settings on the TMTD problem, see figure 1. As expected, the distance between candidates decreases when K_{PD} is lowered, while extreme solutions are found and maintained when K_{PD} is maximal, confirming our earlier observations from thorough diversity tests on artificial benchmark problems, see [3].

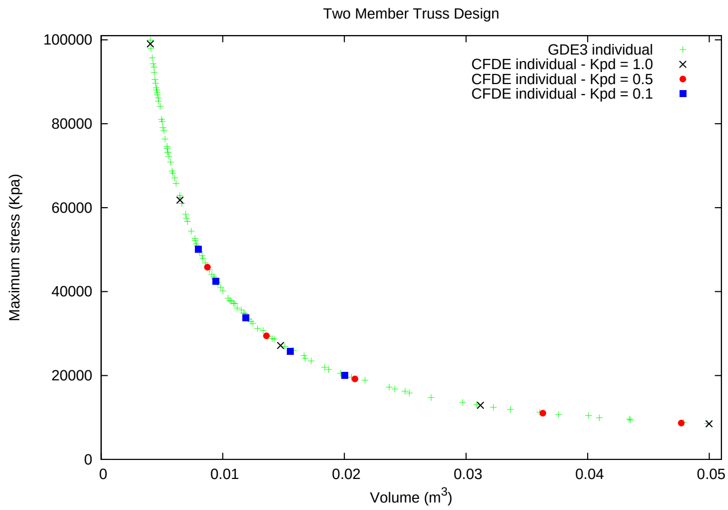

Figure 1. GDE3 and GCFDE/GCD/KNEE result sets on the TMTD problem.

3.2 Circuit design for the Alpha Pro pump

The objective in the circuit design problem is to find component values for a number of resistors and capacitors resulting in a circuit matching the desired functionality. The subcircuit being optimized is a low-pass filter with DC rescaling functionality. It may be possible to address this problem analytically. However, this will most likely produce a solution having component values not available in the standard rows for resistors and capacitors. Rounding such an analytical solution to standard row values will typically decrease circuit performance making this approach less attractive. Instead, optimization may directly find the component values from the standard rows.

The circuit is illustrated in figure 2. The components R1, R3, R4, R5, R6, R7, R8, R9, C1, and C6 are subject to optimization. In this, the available resistor and capacitor values are limited to those in the Grundfos stock, i.e., components used in other Grundfos products.

The part of the circuit subject to optimization has both AC and DC functionality. The AC-functionality of the sub-circuit is to provide low-pass filtering of Idc=0.3A with at least -40dB dampening at 125Hz. The DC-functionalities of the circuit are to attenuate V_{DC} as much as possible and to amplify simultaneously the DC-component, i.e., the average value of the signal as much as possible.

It is out of the scope of this paper to describe the problem in full details. However, a complete description with figures is available in a

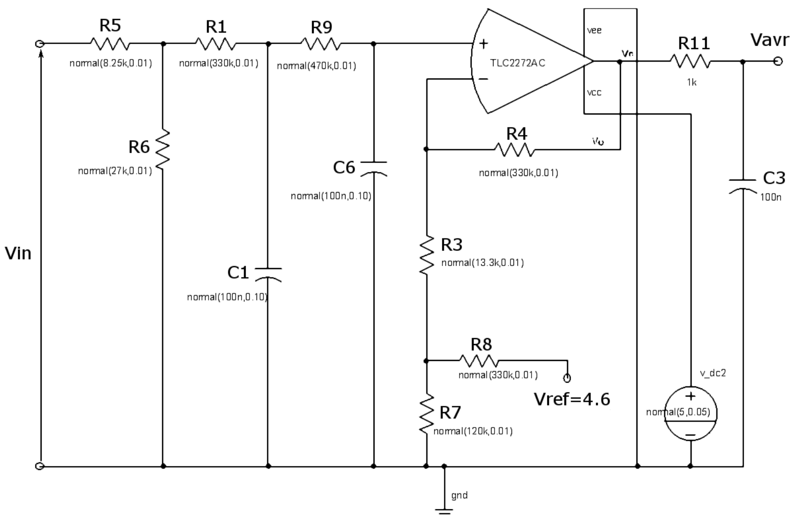

Figure 2. Circuit layout for the low-pass anti-aliasing filter.

technical report [9]. The circuit is simulated using the Saber simulator from Synopsys. In the following, the $M(f_i)$ is the Saber magnitude function (in dB) at frequency f_i. The optimization problem has five objectives and five constraints, which are mainly for ensuring that valid component values are selected.

The first objective $F1$ captures the deviation from the desired AC functionality on three frequencies, $f_i = (0.01, 2.0, 5.0)$.

$$F1 = \sum_{f_i} \begin{cases} 20\log\left(M(f_i)\right) - v_{i,max} & 20\log\left(M(f_i)\right) > v_{i,max} \\ v_{i,min} - 20\log\left(M(f_i)\right) & 20\log\left(M(f_i)\right) < v_{i,min} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The second objective $F2$ models the dampening at 125Hz and it is achieved by minimizing the slope around 40-50Hz.

$$F2 = \frac{20(\log(M(50Hz)) - \log(M(40Hz)))}{50Hz - 40Hz} \tag{4}$$

Third, fourth and fifth objectives models the DC-functionality.

$$F3 = \left|1.0 - \frac{\text{R6}}{\text{R5}} \cdot \frac{\text{R}_{\text{TOT}}}{\text{R4}}\right| \quad \text{R}_{\text{TOT}} = \text{R3} + \frac{\text{R7} \cdot \text{R8}}{\text{R7} + \text{R8}} \tag{5}$$

$$F4 = |4.6 - \Delta V1 - G \cdot 2.603| \quad G = \frac{1.0 + \text{R4}/\text{R}_{\text{TOT}}}{1.0 + \text{R5}/\text{R6}} \tag{6}$$

$$F5 = |0.15 + \Delta V2 - G \cdot 1.301| \tag{7}$$

The variables $\Delta V1 = 0.12$ and $\Delta V2 = 0.10$ in the above equations provide the necessary six-sigma margins for component variations.

The a priori analysis of the circuit design problem resulted in the following generalized preference values:

- $K_{NC} = 4$ – the project manager told us how many prototypes could be built.
- $K_{PD} = 0.0$ – the domain expert (electrical engineer) told us that all objectives could obtain a very low value simultaneously, but this was close to impossible with traditional design methods.
- $K_{DD} = 0.5$ – the domain expert told us that similar performance could be obtained with highly different solutions.

To compare the **convergence capability** of GCFDE and GDE3 on this problem, we use the same procedure as for the benchmark problems. However, the original GDE3 fails due to the absent selection pressure given only by Pareto-classification and a divergent secondary fitness. To enable comparison, we have exchanged the Crowding-distance measure of NSGA-II [1] used as secondary fitness in GDE3 with the convergent, preference based secondary fitness functions presented in section 2.3.

In simulation, we used population size $N = 200$, $F = 0.35$, $CF = 0.2$, and the number of generations was 2000 for both algorithms. In scaling, $[0.0, -1.0, 0.0, 0.0, 0.0]$ was used as best point and $[1.0, 0.0, 1.0, 1.0, 1.0]$ was nadir point. $[1.0, 1.0, 3.0, 1.0, 1.0]$ was used as weight vector in WS.

Table 3. GCFDE versus GDE3 on the Circuit Design for the Alpha Pro problem.

GCFDE GDE3	KNEE KNEE	KNEE WS	WS KNEE	WS WS
Dominates (%) Variance (%)	22.75 ± 20.64	17.25 ± 17.92	23.25 ± 20.08	19.25 ± 18.65
Dominated (%) Variance (%)	0.50 ± 3.50	0.50 ± 3.50	0.25 ± 2.49	0.50 ± 3.50
Incomparable (%) Variance (%)	76.75 ± 20.69	82.25 ± 17.78	76.50 ± 20.25	80.25 ± 18.47

In Table 3, we present a comparison between GCFDE variants and GDE3 variants, both using GCD as PSC. In the top row we denote the GCFDE SCC in the top and the GDE3 SSC below, while the left-most column denotes relations. As no equal solutions were produced in this many-objective setting, this relation is omitted. As simulating the circuit in question takes much more time than calculating the fitnesses of benchmark problems, only 10 runs were performed for the two algorithms, however on two different versions of both. With $K_{NC} = 4$, this amounts to 400 comparisons per column in Table 3. The high variance in this is due to the low cardinality of the GCFDE result sets.

Table 3 demonstrates that even using the same preference based utility functions as guide, the GCFDE algorithm clearly outperforms the GDE3 algorithm on this many-objective problem, with around 20 % GCFDE solutions dominating GDE3 solutions and below 1 % of GDE3 solutions dominating GCFDE solutions, for all algorithm variants. Thus, GCFDE performs well independently of the convergent SSC used.

The main reasons for this performance are most likely the more focused search using subpopulations with migration, as well as the application of both divergent and convergent SSCs in the GCFDE algorithm. The GDE3 algorithm using GCD ranking and a divergent SSC fails in a many-objective setting, because it relies to heavily on Pareto-classification, whereas it seems to converge prematurely using GCD ranking and a convergent SSC due to lack of solution diversity.

Overall, this shows that the approach of returning few, distinct solutions from the true Pareto-front of a many-objective problem is more feasible than trying to cover the full front, but also that maintaining diversity is necessary in achieving optimality. Thus, both diversity and optimality should be considered in many-objective optimization.

4. Conclusions

In this paper, we have introduced the Cluster-Forming Differential Evolution (GCFDE) algorithm, which is able to handle both multi- and many-objective constrained problems. Convergence was demonstrated on five well known constrained problems from mechanical engineering and a real-world many-objective problem from electrical engineering. GCFDE was found to perform equal or slightly better than GDE3 on multi-objective problems, whereas for the many-objective problem, the GCFDE algorithm clearly outperformed the GDE3 algorithm.

In summary, the solutions produced by GCFDE are both distinct and more than competitive to solutions found with GDE3, even when using the same preference functions to guide the optimization, which shows the ability of GCFDE to converge to a low number of distinct optimal solutions – even when solving a many-objective problem. This further shows that diversity and optimality can be maintained in synergy.

References

[1] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms.* Chichester, UK: John Wiley & Sons, 2001, iSBN 0-471-87339-X.

[2] D. W. Corne and J. D. Knowles, Techniques for highly multiobjective optimisation: Some nondominated points are better than others, in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007, pp. 773–780.

[3] P. D. Justesen and R. K. Ursem, "Multiobjective distinct candidates optimization (modco): A cluster-forming differential evolution algorithm," in *Proceedings of the fifth international conference on Evolutionary Multi-Criterion Optimization (EMO-2009)*, vol. 1, 2009, pp. 525–539.

[4] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Parallel Problem Solving from Nature - PPSN VIII*, 2004, pp. 720–729.

[5] S. Kukkonen and J. Lampinen, "GDE3: the third evolution step of generalized differential evolution", in *Proceedings of the 2005 IEEE congress on Evolutionary Computation (EC05)*, vol. 1, pp. 443–450.

[6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimization Test Problems", in *Congress on Evolutionary Computation (CEC'2002)*, vol. 1, 2002, pp. 825–830.

[7] T. Robič and B. Filipič, "DEMO: Differential Evolution for Multiobjective Optimization", in *Evolutionary Multi-Criterion Optimization. Third International Conference*, LNCS Vol. 3410, 2005, pp. 520–533.

[8] K. Deb and A. Srinivasan, "Innovization: innovating design principles through optimization," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 1629–1636.

[9] R. K. Ursem, "A many-objective circuit design problem: The anti-alliasing filter". Grundfos technical report, 2009-01. Download: `www.daimi.au.dk/~ursem/publications/TR-09-01.pdf`

[10] R. K. Ursem, "Multi-Objective Distinct Candidate Optimization (MODCO)". Grundfos technical report, 2009-02. Submitted to BIOMA 2010. Download: `www.daimi.au.dk/~ursem/publications/TR-09-02.pdf`