Supplementary Material for ESOEA: Ensemble of Single Objective Evolutionary Algorithms for Many-Objective Optimization

Monalisa Pal[a,*], Sanghamitra Bandyopadhyay[a]

[a] *Machine Intelligence Unit, Indian Statistical Institute, 203, Barrackpore Trunk Road, Kolkata - 700108*

Abstract

An alternative method of distributing weight vectors is discussed in this material which, in turn, will aid the reference direction associated many-objective optimization algorithms. For an uniform scattering of a set of points in space, the proposed approach poses a single-objective optimization (SOO) problem to maximize the minimum pair-wise distance over all pairs of points in the set. This material presents a detailed description of the problem and a discussion on some preliminary results of distributing points through this approach with the help of Genetic Algorithm (GA), Differential Evolution (DE) and cooperative coevolution based evolutionary algorithm with dynamically changing grouping structure (DECC-G).

1. An alternate method of distributing weight vectors

First, the conventional two layered approach of weight vector distribution is considered [1–4]. In this approach, considering P divisions along each objective, the total number of reference directions (K) for an M-objective problem [1, 2] is given by ${}^{M+P-1}C_P$. For $M \geq 8$, if $P \geq M$, no intermediate weight vectors are chosen. Hence, a two layer approach [2] is used. For example, only with 2 divisions in the outside layer and 1 division in the inside layer require $K = 230$ points or weight vectors for $M = 20$ objectives. This increases the computational complexity of this approach of weight vector distribution for large values of M. Moreover, as number of divisions is arbitrarily chosen for every different number of objectives, the number of weight vectors also varies randomly i.e. no rule of increase in weight vectors with increase in number of objectives is observed to keep the computational complexity manageable.

As an alternative approach, instead of using the systematic approach in [1, 2] for defining the direction vectors, the distribution of the weight vectors is posed as a single objective optimization (SOO) problem. Thus, the number of direction vectors could be regulated as desirable and it remains independent of the number of objectives. It should be noted that the proposed algorithm ESOEA depends on the final distribution of the weight vectors and not on the strategy chosen to distribute the weight vectors. Hence, independent of the optimization problem, at hand, for a given number of objective, the SOO routine can be executed offline and its result can be used in the ESOEA framework.

2. Single objective optimization problem for distributing weight vectors

Distribution of a number of weight vectors, in the objective space, constitute the first step of the proposed approach. Optimization follows along each of these weight vectors, which are defined between the origin and certain points in the objective space. Thus, it is essential that the weight vectors are approximately uniformly spread out in the objective space. Due to the disadvantages of the Das and Dennis's approach [1, 2] for declaring the reference vectors (discussed in Section 1), another approach is proposed in this work.

*Corresponding author

Email addresses: monalisap90@gmail.com (Monalisa Pal), sanghami@isical.ac.in (Sanghamitra Bandyopadhyay)

Preprint submitted to Elsevier *July 24, 2018*

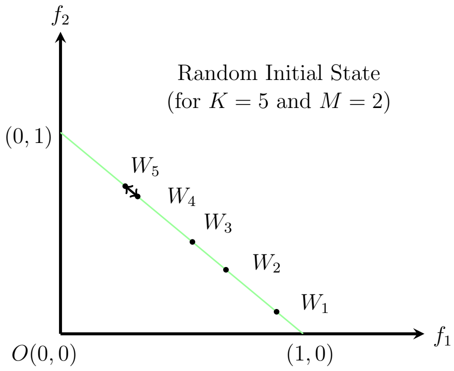

Figure 1: Maximization of minimum pair-wise distance to uniformly distribute weight vectors

The central idea is to choose K points in objective space $(W_1, W_2, \cdots, W_K)$ by maximizing the minimum pair-wise distance among these points, as shown in Fig. 1. This is posed as a single objective optimization problem as given by Eq. (1) with $D(.)$ indicating Euclidean distance [5], subjected to the constraint in Eq. (3). The constraint serves the dual purpose of bounding the weights (the bounds are shown in Eq. (2)) such that they lie on an unit hyperplane and of signifying the contribution of each objective for each instance of the single objective optimizer ensemble, considering there are K such instances.

$$\begin{aligned} &\text{Maximize: } \min_{i \neq j} D(W_i, W_j) \\ &\text{where, } W_i = \{w_{i1}, w_{i2}, \cdots, w_{iM}\}, \forall i = 1, \cdots, K \end{aligned} \tag{1}$$

$$\text{and, } w_{ij} \in (0, 1], \forall j = 1, \cdots, M \tag{2}$$

$$\text{subjected to } |W_i| = \sum_{j=1}^{M} w_{ij} = 1, \forall i = 1, \cdots, K \tag{3}$$

The solution (X_W) of this maximization problem is required to encode a set of K M-dimensional points. Thus, the solution is expressed as an array of length $K \times M$, which has all the K points horizontally concatenated as shown in Eq. (4).

$$X_W = [W_1, W_2, \cdots, W_K] = [w_{11}, \cdots, w_{1M}, \cdots, w_{K1}, \cdots, w_{KM}] \tag{4}$$

3. Experimental details of the proposed weight vector distribution approach

The proposed work is implemented Matlab R2017a using a 64-bit computer with 8 GB RAM and Intel Core i7 @2.20 GHz processor. For investigating the optimization problem defined in Eq. (1) to (3) using an evolutionary computation algorithm, the following specifications are essential:

1. *Choice of evolutionary algorithm:* Cooperative coevolution [6] based framework of Differential Evolution with dynamic grouping structure (DECC-G) [7] is used to address this problem. This algorithm chooses the different sub-components of the population randomly at every cycle, and thus, it is effective at handling nonseparable problems.
   For addressing the constraint maximization problem (Eq. (1) to (3)) all the steps of DECC-G generating a new candidate (initialization, mutation and crossover) are interspersed with normalization step (defined in Eq. (5)). Thus, constraint handling is avoided during selection step [7–10].

$$X_W = \left[\frac{w_{11}}{\sum_j w_{1j}}, \cdots, \frac{w_{1M}}{\sum_j w_{1j}}, \cdots, \frac{w_{K1}}{\sum_j w_{Kj}}, \cdots, \frac{w_{KM}}{\sum_j w_{Kj}} \right] \tag{5}$$

It is to be noted from Eq. (2) that while values tending to 0 are allowed, 0 is not allowed as a weight. This ensures that normalization by Eq. (5) does not yield undefined weight values. However, with all but one decision variable tending to 0, points are obtained, which are very close to the axes of the objective space.

2. *Representing candidate solutions:* The candidate solutions are encoded as arrays of length $K \times M$ (Eq. (4)), followed by normalization (Eq. (5)), which represent K, linearly appended, M-dimensional points in unit hyperplane.
3. *Avoiding local optima:* If the best objective value deviates by less than 10^{-4} (implying the best candidate is practically stagnant) successively for more than 10 generations, then a candidate is randomly initialized within the bounds (given in Eq. (2)) of decision variables (weights), which after normalization (by Eq. (5)), replaces a random candidate (except the best performing candidate) of the population. If this randomly initialized candidate performs better than the best candidate of the population, it helps in avoiding a local optima. Thus, such random replacement of candidates, when the best candidate has been stagnant, reduces the chances of trapping at a local optima.
4. *Termination criteria:* For this work, either of the following two criteria are used for the stopping the evolutionary algorithm:
   - DECC-G is allowed to run for maximum of 200 cycles unless otherwise terminated.
   - Similar to the strategy of avoiding local optima, if the best objective value deviates by less than 10^{-4} successively for more than 50 cycles (thus, giving several opportunities to escape from the local optima), DECC-G is terminated.

Using these specifications, the optimization problem of distributing the weight vectors is realized and the obtained set of weights, for a particular number of objectives (M), is saved for future use, as this set will be invariant to all M-objective optimization problems.

Thus, using this solution for a set of K weight vectors, the proposed framework of Ensemble of Single Objective Evolutionary Algorithm (ESOEA) is implemented.

4. Results and discussion on the proposed weight vector distribution approach

The parameter specifications and the results of the single objective optimization problem (distributing weight vectors) are discussed in this section. The mean and standard deviation of the performance indicators over 30 runs are noted for performance analysis of the proposed approach.

4.1. Single Objective Optimization Problem of Distributing Weight Vectors

The maximum value of the fitness for the concerned optimization problem is noted as a performance measure. The performance of DECC-G [7] is compared with Differential Evolution (DE) [11] and Genetic Algorithm (GA) [12]. The algorithm DECC-G is setup as per the specifications in [7]. It should be noted that DECC-G uses both classical DE (DE/rand/1/bin) and DE with self-adaptive neighborhood search (SaNSDE) [10]. DE/rand/1/bin [8, 9] indicates random mutation operation with 1 difference vector, followed by binomial crossover operation is applied with DE. The following parameters are used to set up the evolutionary algorithms:

1. Population Size: 100
2. Maximum generations/cycles: 200
3. Scale factor for DE/rand/1/bin: randomly in the range $[0, 2]$
4. Crossover rate for DE/rand/1/bin: 0.8 to favour new solutions
5. SBX Crossover parameters for GA: $\eta_c = 5$ with crossover probability $= 0.9$
6. Polynomial Mutation parameters for GA: $\eta_m = 10$ with mutation probability $= 0.1$
7. Binary tournament selection parameter for GA: 2
8. Number of weight vectors (K): $10 \times M$

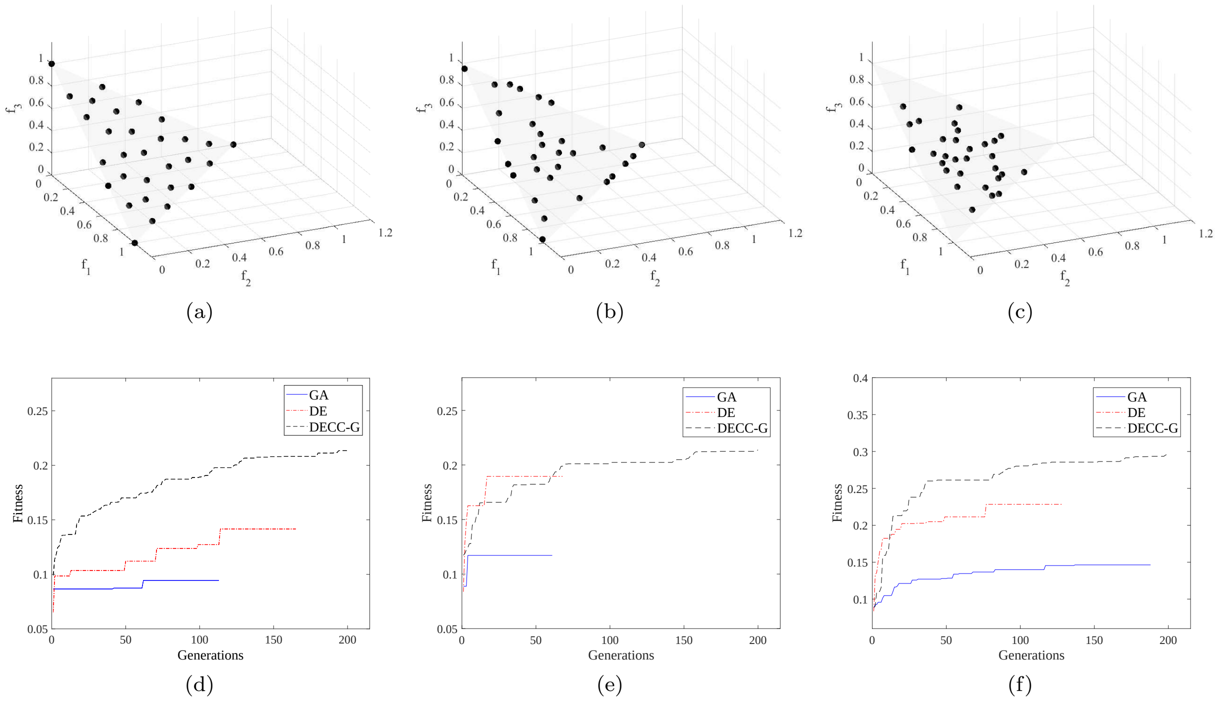

Figure 2: (a) Resulting weight vectors using DECC-G ($M = 3$), (b) Resulting weight vectors using DE ($M = 3$), (c) Resulting weight vectors using GA ($M = 3$), (d) Comparing fitness maximization for $M = 5$, (e) Comparing fitness maximization for $M = 10$, (f) Comparing fitness maximization for $M = 20$

Table 1: Optimizing the Distribution of Weight Vectors

#Objs (M)	**Maximum fitness by DECC-G**	**Maximum fitness by DE**	**Maximum fitness by GA**
3	$\mathbf{0.1661 \pm 0.0031}$	$0.0771 \pm 0.0093(+)$	$0.0520 \pm 0.0023(+)$
5	$\mathbf{0.2135 \pm 0.0064}$	$0.1416 \pm 0.0082(+)$	$0.0945 \pm 0.0087(+)$
10	$\mathbf{0.2138 \pm 0.0041}$	$0.1895 \pm 0.0045(+)$	$0.1171 \pm 0.0087(+)$
20	$\mathbf{0.2965 \pm 0.0077}$	$0.2284 \pm 0.0080(+)$	$0.1463 \pm 0.0092(+)$

Except for the number of weight vectors, all the other parameters have been set to the commonly used specifications [5, 7, 9, 10, 12–14]. For the weight vectors, as an heuristic $K = 10 \times M$ is chosen, keeping in mind $K > M$ is required for intermediate weight vectors in the objective space [2].

The mean and standard deviation of the maximum fitness value of the optimization problem (Eq. (1) to (3)) are noted in Table 1 with the best values in boldface. Also, when the results of DECC-G are compared with DE and GA for all the different test cases in Table 1, the p-values are less than 10^{-5} for a 95% confidence interval, thereby rejecting the null hypothesis that the performance of DECC-G is equivalent to DE and GA for the given problem. This significance of superior performance by DECC-G over DE and GA is indicated by the + sign beside the fitness values observed in Table 1. A few random runs of the optimization problem are illustrated in Fig. 2 for comparison.

Thus, DECC-G clearly outperforms DE and GA for this problem, and at the end of optimization, a decent approximately uniform distribution of weight vectors are generated which can be used as direction vectors for ESOEA. The underlying cause of this superior performance of DECC-G could be attributed to the fact that with increase in number of objectives (M), the number of decision variables ($K \times M = 10 \times M^2$) increases exponentially and the optimization problem then becomes a large-scale optimization problem. Hence, the cooperative coevolutionary approach performs better.

5. Conclusion

This material describes an alternative approach for distributing weight vectors where the task is posed as a constrained single objective optimization problem. Only preliminary experiments have been performed to integrate its result with ESOEA. More experiments on several many-objective optimization problem are required to establish the efficacy of the weight vector distribution generated through the proposed approach. Nonetheless, this method provides flexibility to regulate the number of points/weight vectors with increase in number of objectives. Hence, it helps to keep a check on the computational complexity of reference direction associated many-objective optimization evolutionary algorithms for large number of objectives.

References

[1] I. Das, J. E. Dennis, Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multi-criteria optimization problems, SIAM Journal on Optimization 8 (3) (1998) 631–657.

[2] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints., IEEE Trans. Evolutionary Computation 18 (4) (2014) 577–601.

[3] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Transactions on Evolutionary Computation 19 (5) (2015) 694–716.

[4] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility, IEEE Transactions on Evolutionary Computation.

[5] M. Pal, S. Saha, S. Bandyopadhyay, DECOR: Differential evolution using clustering based objective reduction for many-objective optimization, Information Sciences 423 (2018) 200 – 218. doi:https://doi.org/10.1016/j.ins.2017.09.051.
URL `http://www.sciencedirect.com/science/article/pii/S0020025517309696`

[6] M. A. Potter, K. A. De Jong, A cooperative coevolutionary approach to function optimization, in: Y. Davidor, H.-P. Schwefel, R. Männer (Eds.), Parallel Problem Solving from Nature — PPSN III, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994, pp. 249–257.

[7] Z. Yang, K. Tang, X. Yao, Large scale evolutionary optimization using cooperative coevolution, Information Sciences 178 (15) (2008) 2985–2999.

[8] S. Dasgupta, S. Das, A. Biswas, A. Abraham, On stability and convergence of the population-dynamics in differential evolution, AI Communications 22 (1) (2009) 1–20.

[9] A. K. Qin, P. N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: Evolutionary Computation, 2005. The 2005 IEEE Congress on, Vol. 2, IEEE, 2005, pp. 1785–1791.

[10] Z. Yang, K. Tang, X. Yao, Self-adaptive differential evolution with neighborhood search, in: Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, IEEE, 2008, pp. 1110–1116.

[11] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Transactions on Evolutionary Computation 10 (6) (2006) 646–657. doi:10.1109/TEVC.2006.872133.

[12] K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, Complex systems 9 (2) (1995) 115–148.

[13] M. Pal, S. Saha, S. Bandyopadhyay, Clustering based online automatic objective reduction to aid many-objective optimization, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 1131–1138. doi:10.1109/CEC.2016.7743915.

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