Subspace-based Archived Multi-Objective Simulated Annealing
Project Report II

Md Sahil
BCSEIV, Roll:001710501029
Jadavpur University

Abstract

Unlike other population-based approaches, Simulated Annealing is known for its point-based approach towards addressing an optimization problem. Hence, it has not been widely explored for Many-Objective Optimization (MaOO) problems. This article proposes a Subspace-based Archived Many-Objective Simulated Annealing (SAMOSA), which improves on Archived Multi-Objective Simulated Annealing (AMOSA). It also incorporates the concept of maintaining an archive to provide a set of trade-off solutions for the problem under consideration. While AMOSA fails to provide adequate results for MaOO problems, SAMOSA incorporates the concept of decomposition of objective space in order to provide better spread and diversity of solutions while estimating the Pareto-Front for MaOO problems. This article establishes the superior performance of SAMOSA over AMOSA on 3 to 10-objective DTLZ problems. Future work can be conducted on validating its efficacy on real-world applications.

1 Introduction

Multi-objective optimization deals with optimizing M conflicting objectives. Various challenges (scalable population maintenance, decreasing selection pressure by Pareto-dominance, lack of visualization, properties of quality solutions - convergence and diversity, etc.) become more prominent as M increases. The problems dealing with $M > 3$ objectives are termed as Many-Objective Optimization (MaOO) [16, 3]. A box-constrained M-objective minimization problem is mathematically formulated by Eq. (1) where the N-dimensional decision vector $\mathbf{X}$ maps to a M-dimensional objective vector $\mathbf{F}$ and the decision space is bounded by lower (x_i^L) and upper limits (x_i^U) of decision variables.

$$\begin{aligned} \text{Minimize: } \mathbf{F}(\mathbf{X}) = [f_1(\mathbf{X}), f_2(\mathbf{X}), \cdots, f_M(\mathbf{X})] \text{ where, } \mathbf{X} \in \mathcal{D} \left(\subseteq \mathbb{R}^N\right), \\ \mathbf{F}(\mathbf{X}) : \mathcal{D} \mapsto \mathbb{R}^M \text{ and } \mathcal{D} : x_i^L \leq x_i \leq x_i^U, \text{ for } i = 1, 2, \cdots, N. \end{aligned} \tag{1}$$

Fitness-based comparison of two solutions in MaOO problems are done on the basis of Pareto-dominance [6, 17]. If $\mathbf{X}_1$ and $\mathbf{X}_2$ be two feasible solutions, then $\mathbf{X}_1 \prec \mathbf{X}_2$ or $\mathbf{X}_1$ Pareto-dominates $\mathbf{X}_2$ when Eq. (2) is satisfied.

$$\forall i \in \{1, \cdots, M\},\ f_i(\mathbf{X}_1) \leq f_i(\mathbf{X}_2),\ \exists j \in \{1, \cdots, M\},\ f_j(\mathbf{X}_1) < f_j(\mathbf{X}_2) \tag{2}$$

If no solution could be found to Pareto-dominate a set of mutually non-dominated solutions, then non-dominated set indicates an approximation of Pareto-optimality. The corresponding set of objective vectors estimate the Pareto-Front (PF). The closeness and the uniform spread of solutions in the estimated PF with respect to the true PF are respectively known as convergence and diversity. These are the two most essential quality indicators of a MaOO algorithm.

Majority of MaOO algorithms in the literature are Pareto-dominance based, indicator-based or decomposition-based [22]. A decomposition-based approach

partitions the objective space into multiple subspaces and the MaOO problem into associated sub-problems. These sub-problems collaborate with each other to get optimized [13]. The decomposition is guided by Das and Dennis's two-layered approach [4, 28]. Such decomposition-based approaches are very popular [27] as they neither reduce the selection pressure nor require the extreme computational effort of indicator evaluation. However, most of the state-of-the-art approaches deal with population-based approaches.

Simulated Annealing (SA), on the other hand, is a popular meta-heuristic employing point-based approach for optimization [1]. There has not been much attention in exploring SA for MaOO problems. Archived Multi-Objective Simulated Annealing (AMOSA) has discussed this aspect at length. To combine the strengths of subspace-based algorithms, AMOSA is extended to Subspace-based AMOSA (SAMOSA). Its key contributions are as follows:

- To avoid unchecked growth of archive due to inclusion of newly generated solutions, AMOSA performs Single Linkage Clustering [1]. Instead of clustering the archive, SAMOSA adopts subspace partitioning which enhances solution diversity, combines scalarization of objective vectors (using Penalty-based Indicator) and Pareto-dominance [15], and assists in monotonic diversity improvement.

- Although AMOSA compares the solutions to be perturbed with respect to the archive, there is no guarantee that solutions from every subspace will eventually get perturbed. However, it is essential to improve the archive along all the subspaces. SAMOSA periodically checks and perturbs solutions even in subspaces which are unvisited in immediate past. This periodic lookup assists in better exploration.

- AMOSA always perturbs a solution in a random direction [1]. However, utilizing the neighborhood property of sub-spaces [21], SAMOSA considers solutions from neighboring sub-spaces to set the perturbation direction. Hence, SAMOSA can generate solutions in the empty (unexplored) subspaces.

With the above motivation in mind, the algorithmic framework of SAMOSA, its performance analysis results and a summary of the article are provided in the subsequent sections.

2 Algorithmic Framework

The overall framework of SAMOSA is presented in Algorithm 1. It begins with initialization (line 1) of an archive $\mathcal{A}$ with l_{hard} number of randomly generated points within the bounds of the decision variables and the set of reference vectors ($\mathcal{W}$) by Das and Dennis's approach [22, 15]. In line 2, these points are then associated with the subspaces ($\mathcal{S}_i$) based on their $d2$ values [8] using Eq. (3).

$$\text{Associate } \mathbf{F} \text{ with } \mathcal{S}_i \ni d2(\mathbf{W}_i, \mathbf{F}) = \min_{j=1}^{n_{dir}}(d2(\mathbf{W}_j, \mathbf{F})) \tag{3}$$

The main loop of the algorithm (lines 6 to 28) starts with temperature $T = T_{max}$ and ends at $T = T_{min}$, where at each iteration the temperature is reduced as $T = T \times \alpha$ (line 27). The parameter α (cooling rate) depends on a fitness evaluation budget ($MaxFES$). For each candidate, $\mathbf{F}_{new}$ is evaluated once. Thus, maintaining $MaxFES$, α is calculated by Eq. (4).

$$\alpha = \left(\frac{T_{min}}{T_{max}}\right)^{\frac{n_{iter}}{MaxFES}} \tag{4}$$

In Algorithm 1, α is set during initialization on line 1.

At each temperature, the internal loop (lines 7 to 26) runs n_{iter} times. During each iteration, a point is assigned as $\mathbf{X}_{cur}$ (as done in AMOSA [1]), $\mathbf{X}_{cur}$ is perturbed (as explained in Section 2.1) to generate $\mathbf{X}_{new}$ and corresponding $\mathbf{F}_{new}$ (line 14). Depending on the value of $\mathbf{F}_{new}$, the current point $\mathbf{X}_{cur}$ and the archives ($\mathcal{A}$, $\mathcal{A}_{\mathbf{F}}$) are updated.

At each iteration, it is checked whether the current point belongs to a sub-space that is marked as unvisited in the list $\mathbf{V}$ (line 4). If the sub-space is visited in recent iterations, a random point from a random unvisited sub-space is assigned as $\mathbf{X}_{cur}$ (line 22). Initially when all the sub-spaces are unvisited $\mathbf{V}$ contains all the sub-spaces that have points associated with it (lines 4 and 9).

After selecting $\mathbf{X}_{cur}$, it is perturbed as described in Section 2.1 to yield $\mathbf{X}_{new}$ (line 14). Based on the non-domination status of $\mathbf{X}_{new}$, it is added to $\mathcal{A}$ and the subsequent $\mathbf{X}_{cur}$ is chosen as done in AMOSA [1]. AMOSA's selection scheme is slightly modified to retain dominated points in the archive (line 15). If $\mathcal{A}$ grows beyond a limit l_{soft}, it is resized to l_{hard} (line 17) as described in Section 2.2. SAMOSA finally terminates when T_{min} is reached with the estimated PF in $\mathcal{A}_{\mathbf{F}}$.

2.1 Perturbation Scheme

The perturbation scheme consists of probabilistic switching [26] between Differential Evolution (DE) [5, 31] and Simulated Binary Crossover (SBX) [24, 30] reproduction schemes followed by laplacian mutation strategy. The choice between DE and SBX is decided based on the probability $P_{perturb}$. DE requires three more parent points other than $\mathbf{X}_{cur}$. These three points are sampled from the non-empty sub-space nearest to the sub-space that $\mathbf{X}_{cur}$ belongs to. Similarly, SBX requires one more parent point other than $\mathbf{X}_{cur}$ which is also sampled in the same manner.

2.2 Resize strategy

Once the size of the archive exceeds l_{soft}, the resizing is triggered. In the first step, the dominated solutions in the archive are removed with a probability of $P_{remove\_dominated} = 2 - \frac{2}{1+\exp(-T)}$, i.e., the probability decreases with T as shown in Fig. 1a. After removal of dominated solutions (Fig. 1b), if the archive size is still greater than l_{hard}, the sub-space with maximum number of associated

Algorithm 1 Basic framework of SAMOSA

Require: $prob(M, N)$: problem specifications; l_{hard}: hard limit; l_{soft}: soft limit; n_{dir}: number of sub-spaces; T_{max}, T_{min}: maximum and minimum temperatures; $MaxFES$: function evaluation budget

Ensure: $\mathcal{A}$: Estimated Pareto-optimal Set

1: $\mathcal{A}, \alpha, n_{iter}, \mathcal{W} \leftarrow$ INITIALIZE($prob, M, N, l_{soft}, n_{dir}, MaxFES$)
2: ASSOCIATE-ARCHIVE($\mathcal{A}$, $\mathcal{W}$)
3: $T = T_{max}$; $i = 0$ (Initialize temperature and iteration counter)
4: $\mathbf{V}$ = GET_ASSOCIATED_SUBSPACES($\mathcal{S}$) $\leftarrow$ list of non-empty unvisited sub-spaces
5: $\mathbf{X}_{cur} \leftarrow$ random point from $\mathcal{A}$
6: **while** $T > T_{min}$ **do**
7: **while** $i < n_{iter}$ **do**
8: **if** mod $(i, n_{dir} + 1) = 0$ or $\mathbf{V}$ is empty **then**
9: $\mathbf{V}$ = GET_ASSOCIATED_SUBSPACES($\mathcal{S}$)
10: **end if**
11: $\mathcal{S}_r \leftarrow$ Sub-space with which $\mathbf{X}_{cur}$ is associated
12: **if** $\mathcal{S}_r$ in $\mathbf{V}$ ($\mathcal{S}_r$ is unvisited) **then**
13: remove $\mathcal{S}_r$ from $\mathbf{V}$ (Set $\mathcal{S}_r$ as visited)
14: $\mathbf{X}_{new}$ = PERTURB_ASSOCIATE($\mathbf{X}_{cur}, \mathcal{A}, \mathbf{W}, \mathcal{S}$) (explained in section 2.1)
15: $\mathbf{X}_{cur}, \mathcal{A} \leftarrow$ AMOSA_SELECT($\mathcal{A}, \mathbf{X}_{cur}, \mathbf{X}_{new}, T$) (Revise $\mathcal{A}, \mathcal{A}_\mathbf{F}$ and $\mathbf{X}_{cur}$)
16: **if** $|\mathcal{A}| > l_{soft}$ **then**
17: $\mathcal{A}, \mathcal{A}_\mathbf{F} \leftarrow$RESIZE($\mathcal{A}, \mathcal{A}_\mathbf{F}, \mathcal{S}, l_{hard}, T$) (To avoid unchecked growth of $\mathcal{A}$ using Algorithm 2)
18: **end if**
19: **else**
20: Randomly select another sub-space $\mathcal{S}_r$ from $\mathbf{V}$
21: Remove $\mathcal{S}_r$ from $\mathbf{V}$
22: $\mathbf{X}_{cur} \leftarrow$ a random point from $\mathcal{A}$ associated with $\mathcal{S}_r$
23: Same as steps 14 to 18
24: **end if**
25: $i = i + 1$ (Update iteration counter)
26: **end while**
27: $T = T \times \alpha$ (Update temperature)
28: **end while**

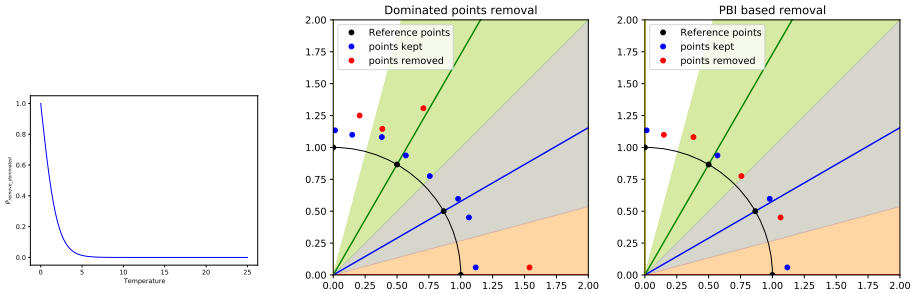
points is selected and the point with the worst PBI value in that sub-space is removed from the archive as shown in Fig. 1c. This step performed repeatedly till the size of the archive is less then or equal to l_{hard}.

Algorithm 2 Algorithm to reduce size of archive to l_{hard}

Require: $\mathcal{A}$: current set of solution points; $\mathcal{S}$: set of sub-spaces; l_{hard}: hard limit of archive size; T: current temperature

Ensure: $\mathcal{A}$, $\mathcal{A}_{\mathbf{F}}$: archives with size reduced to l_{hard}

procedure RESIZE($\mathcal{A}, \mathcal{A}_{\mathbf{F}}, \mathcal{S}, l_{hard}, T$)
  $P_{remove\_dominated} = 2 - \frac{2}{1+\exp(-T)}$
  if RAND( )$< P_{remove\_dominated}$ **then**
    $\mathcal{A}, \mathcal{A}_{\mathbf{F}}$ = REMOVE_DOMINATED($\mathcal{A}, \mathcal{A}_{\mathbf{F}}$)
  end if
  while size of $\mathcal{A} > l_{hard}$ **do**
    $\mathcal{S}_r \leftarrow$ sub-space with highest number of points
    $\mathbf{X}_r \leftarrow$ point in $\mathcal{S}_r$ with worst PBI value
    $\mathbf{F}_r \leftarrow$ point in objective space corresponding to $\mathbf{X}_r$
    if number of points in $\mathcal{S} > 1$ **then**
      remove $\mathbf{X}_r$ from $\mathcal{A}$
      remove $\mathbf{F}_r$ from $\mathcal{A}_{\mathbf{F}}$
    end if
  end while
  return $\mathcal{A}$, $\mathcal{A}_{\mathbf{F}}$
end procedure



(a) (b) (c)

Figure 1: (a) Variation in $P_{remove\_dominated}$, (b) Dominated solutions for removal, (c) Diversity preserving selection.

3 Results and Discussion

The proposed algorithm (SAMOSA) is implemented in Python 3.4 on a computer having 64-bit Intel Core i3 processor (2.3 GHz) and 4 GB of RAM. An implementation of AMOSA is obtained from https://www.isical.ac.in/~sriparna_r/software.html and the performance of AMOSA and SAMOSA are compared on benchmark problems (DTLZ1 to DTZL4 [7]) in terms of the standard performance metrics - hypervolume (HV) [12] and Inverted Generational Distance (IGD) [2]. Both HV and IGD indicate convergence and diversity of trade-off

solutions over to true PF. Alongside the population dynamics is also established in terms of D_metric [27].

For evaluating HV (Eq. (5)), a set of points (denoted by $\mathcal{H}_{HV}$) is sampled (by Monte-Carlo simulations) inside a hyper-rectangle (defined between a reference point, r_{HV}, and the origin of the objective space) [19, 11]. The proportion of $\mathcal{H}_{HV}$ which are Pareto-dominated by the points estimating PF (denoted by $\mathcal{A}_{\mathbf{F}}$) denotes HV. A higher HV indicates better performance.

$$\begin{aligned} HV\left(\mathcal{A}_{\mathbf{F}}, \mathcal{H}_{HV}\right) &= \frac{1}{|\mathcal{H}_{HV}|} \sum_{j=1}^{|\mathcal{H}_{HV}|} \alpha_{HV}\left(\mathbf{H}_j, \mathcal{A}_{\mathbf{F}}\right), \text{ where } \mathbf{H}_j \in \mathcal{H}_{HV} \text{ and} \\ \alpha_{HV}\left(\mathbf{H}_j, \mathcal{A}_{\mathbf{F}}\right) &= \begin{cases} 1, & \text{if } \exists \mathbf{F}\left(\mathbf{X}_i\right) \in \mathcal{A}_{\mathbf{F}} \text{ with } \mathbf{F}\left(\mathbf{X}_i\right) \prec \mathbf{H}_j \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

For evaluating IGD (Eq. (6)), a set of points ($\mathcal{H}_{IGD}$) is sampled on the true PF and the average minimum Euclidean distance $D_E(.)$ of the points ($\mathbf{H}_j$) in $\mathcal{H}_{IGD}$ from the points estimating PF ($\mathbf{F}(\mathbf{X}_i) \in \mathcal{A}_{\mathbf{F}}$) is obtained over the number of points in $\mathcal{H}_{IGD}$ [29, 28]. A higher IGD indicates better performance.

$$\begin{gathered} IGD(\mathcal{A}_{\mathbf{F}}, \mathcal{H}_{IGD}) = \frac{1}{|\mathcal{H}_{IGD}|} \sum_{j=1}^{|\mathcal{H}_{IGD}|} \left(\min_{i=1}^{|\mathcal{A}_{\mathbf{F}}|} \left(D_E\left(\mathbf{F}\left(\mathbf{X}_i\right), \mathbf{H}_j\right)\right)\right), \\ \text{where } \mathbf{F}\left(\mathbf{X}_i\right) \in \mathcal{A}_{\mathbf{F}} \text{ and } \mathbf{H}_j \in \mathcal{H}_{IGD} \end{gathered} \tag{6}$$

For evaluating D_metric (Eq. (7)) at generation G, the distance between current sub-archive spread ($S_G^k = \left|\mathcal{A}_{k,G}^{sub}\right|$) from ideal sub-archive spread (S_{ideal}^k) is obtained over n_{dir} subspaces in the objective space [27, 23]. A lower IGD indicates better performance.

$$D\_metric^G = \frac{n_{dir}}{n_{arch}} \sqrt{\sum_{k=1}^{n_{dir}} (S_G^k - S_{ideal}^k)^2}, \text{ where } S_{ideal}^k = \frac{n_{arch}}{n_{dir}}. \tag{7}$$

The sensitivity of various parameters have been studied and the settings used for conducting the experiments reported in this article are outlined in Table 1. The two-layered reference vector distribution is obtained from https://worksupplements.droppages.com/refvecgen. For IGD and HV parameters, [26] is consulted. For AMOSA paramaters, [1] is consulted. The $MaxFES$ is considered as product of n_{dir} and tot_itr from [26] for various DTLZ problems.

The number of objectives were varied from 3 to 10 for DTLZ problems and the superiority of SAMOSA in terms of IGD and HV values is noted in Fig. 2. While IGD for AMOSA exceed 0.5 as number of objective increases to 8 and beyond, SAMOSA is able to maintain IGD below 0.5 in 15 out of 16 cases. The zero hypervolumes by AMOSA for DTLZ1 and DTLZ3 (multimodal problems [7, 9]) indicate that none of the estimated points by AMOSA are not inside the concerned hyper-rectangles. This drawback is tackled by SAMOSA due to its subspace-based nature.

Table 1: Parameter specifications for performance analysis of SAMOSA.

Parameters	M	**Values**	**Parameters**	**Values**
n_{dir}	3-obj	91	$\lvert\mathcal{H}_{IGD}\rvert$	5000
	5-obj	210	$\lvert\mathcal{H}_{HV}\rvert$	10000
	8-obj	156	r_{HV}	$[0.6, ..., 0.6]$ (DTLZ1)
	10-obj	275		$[1.2, ..., 1.2]$ (DTLZ2-4)
T_{min}	-	0.000001	T_{max}	200
n_{iter}	-	500	$P_{perturb}$	0.5
l_{hard}	-	n_{dir}	l_{soft}	$2 * n_{dir}$

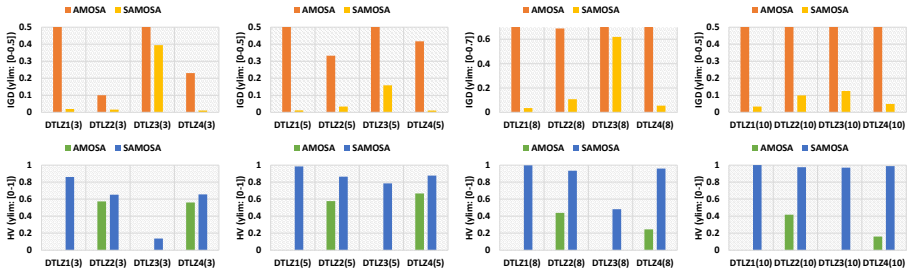
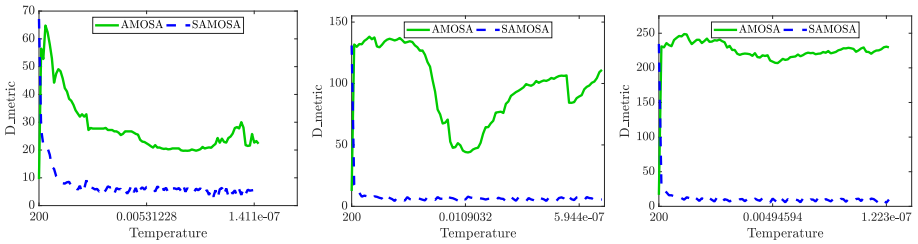


Figure 2: Mean HV and IGD over 31 independent runs of AMOSA variants.



(a) DTLZ4 ($M = 3$) (b) DTLZ4 ($M = 8$) (c) DTLZ4 ($M = 10$)

Figure 3: Population dynamics with decrease in temperature for AMOSA variants.

In problems with biased solution density (like DTLZ4), AMOSA [1] terminates with a very high D_metric (Fig. 3). Unlike AMOSA's Laplacian mutate [1] where a new vector is generated from a single parent vector, SAMOSA gathers information from multiple neighboring parents for producing a new vector. This helps D_metric to converge towards ideal scenario [26].

The above experiments establishes the superiority of SAMOSA over AMOSA for problems with properties like unimodality, multimodality, linear and non-convex PF, meta-variable mapping and biased solution density [10, 18].

4 Conclusion

The meta-heuristic approach of SA in approximating global Pareto-optimal solutions for MaOO problems is relatively less explored due to its point-based approach [1], which motivates this work. This article presents SAMOSA which adopts a subspace-based approach while maintaining an archive of non-dominated solutions for addressing MaOO problems. The decomposition of objective space provides better exploration and results in better diversity and spread of solutions over the estimated Pareto-Front. The efficacy of SAMOSA is established on DTLZ problems. In future, the authors aim to analyze its performance on other new test problems [14] and various real-world application problems [25, 20] while studying its competitiveness against other population-based approaches.

References

[1] Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K.: A simulated annealing-based multiobjective optimization algorithm: AMOSA. IEEE transactions on evolutionary computation **12**(3), 269–283 (2008)

[2] Bosman, P.A.N., Thierens, D.: The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation **7**(2), 174–188 (2003)

[3] Coello, C.A.C., Brambila, S.G., Gamboa, J.F., Tapia, M.G.C., Gómez, R.H.: Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. Complex & Intelligent Systems **6**(2), 221–236 (2020)

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

[5] Das, S., Mullick, S.S., Suganthan, P.N.: Recent advances in differential evolution – an updated survey. Swarm and Evolutionary Computation **27**, 1–30 (2016)

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

[7] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA). pp. 825–830. Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA) (2002)

[8] Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving

problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014)

[9] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization: Theoretical Advances and Applications. pp. 105–145. Springer London, London (2005)

[10] Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation **10**(5), 477–506 (2006)

[11] Ishibuchi, H., Imada, R., Masuyama, N., Nojima, Y.: Comparison of hypervolume, IGD and IGD+ from the viewpoint of optimal distributions of solutions. In: Deb, K., Goodman, E., Coello Coello, C.A., Klamroth, K., Miettinen, K., Mostaghim, S., Reed, P. (eds.) Evolutionary Multi-Criterion Optimization. pp. 332–345. Springer International Publishing, Cham (2019)

[12] Ishibuchi, H., Pang, L.M., Shang, K.: Population size specification for fair comparison of multi-objective evolutionary algorithms. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 1095–1102. IEEE (2020)

[13] Kang, Q., Song, X., Zhou, M., Li, L.: A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms. IEEE Transactions on Systems, Man, and Cybernetics: Systems **49**(12), 2416–2423 (Dec 2019)

[14] Li, H., Deb, K., Zhang, Q., Suganthan, P.N., Chen, L.: Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties. Swarm and Evolutionary Computation **46**, 104–117 (2019)

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

[16] Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., Coello, C.A.C.: A survey of multiobjective evolutionary algorithms for data mining: Part I. IEEE Transactions on Evolutionary Computation **18**(1), 4–19 (Feb 2014)

[17] Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., Coello, C.A.C.: Survey of multiobjective evolutionary algorithms for data mining: Part II. IEEE Transactions on Evolutionary Computation **18**(1), 20–35 (Feb 2014)

[18] Pal, M., Saha, S., Bandyopadhyay, S.: Clustering based online automatic objective reduction to aid many-objective optimization. In: 2016 IEEE Congress on Evolutionary Computation (CEC). pp. 1131–1138. IEEE (July 2016)

[19] Pal, M., Saha, S., Bandyopadhyay, S.: DECOR: Differential evolution using clustering based objective reduction for many-objective optimization. Information Sciences **423**, 200 – 218 (2018)

[20] Pal, M., Alyafi, A.A., Ploix, S., Reignier, P., Bandyopadhyay, S.: Unmasking the causal relationships latent in the interplay between occupant's actions and indoor ambience: A building energy management outlook. Applied Energy **238**, 1452 – 1470 (2019)

[21] Pal, M., Bandyopadhyay, S.: Differential evolution for multi-modal multi-objective problems. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 1399–1406. GECCO '19, ACM, New York, NY, USA (2019)

[22] Pal, M., Bandyopadhyay, S.: ESOEA: Ensemble of single objective evolutionary algorithms for many-objective optimization. Swarm and Evolutionary Computation **50**, 100511 (2019)

[23] Pal, M., Bandyopadhyay, S.: Decomposition in decision and objective space for multi-modal multi-objective optimization. Swarm and Evolutionary Computation **62**, 100842 (2021)

[24] Raghuwanshi, M.M., Kakde, O.G.: Survey on multiobjective evolutionary and real coded genetic algorithms. In: Proceedings of the 8th Asia Pacific symposium on intelligent and evolutionary systems. pp. 150–161. Citeseer (2004)

[25] Ray, S., Maulik, U.: Identifying differentially coexpressed module during HIV disease progression: A multiobjective approach. Scientific reports **7**(1), 86 (2017)

[26] Sengupta, R., Pal, M., Saha, S., Bandyopadhyay, S.: NAEMO: Neighborhood-sensitive archived evolutionary many-objective optimization algorithm. Swarm and Evolutionary Computation **46**, 201 – 218 (2019)

[27] Sengupta, R., Pal, M., Saha, S., Bandyopadhyay, S.: Population dynamics indicators for evolutionary many-objective optimization. In: Panigrahi, C.R., Pujari, A.K., Misra, S., Pati, B., Li, K.C. (eds.) Progress in Advanced Computing and Intelligent Engineering. pp. 261–271. Springer Singapore, Singapore (2019)

[28] Tian, Y., Cheng, R., Zhang, X., Cheng, F., Jin, Y.: An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility. IEEE Transactions on Evolutionary Computation **22**(4), 609–622 (2017)

[29] Tian, Y., Zhang, X., Cheng, R., Jin, Y.: A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In: 2016 IEEE Congress on Evolutionary Computation (CEC). pp. 5222–5229 (July 2016)

[30] Tušar, T., Filipič, B.: Differential evolution versus genetic algorithms in multiobjective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) Evolutionary Multi-Criterion Optimization. pp. 257–271. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)

[31] Wang, X., Dong, Z., Tang, L.: Multiobjective differential evolution with personal archive and biased self-adaptive mutation selection. IEEE Transactions on Systems, Man, and Cybernetics: Systems pp. 1–13 (2018)