Hierarchical Convergence to Multiple Alternate Solutions: Population versus Point-based Algorithms

COIN Report 2025002

Ahmer Khan
khanahm2@msu.edu
Michigan State University
East Lansing, MI, USA

Kalyanmoy Deb
kdeb@msu.edu
Michigan State University
East Lansing, MI, USA

ABSTRACT

In many practical problem-solving tasks, multiple solutions are sought. Multi-modal and multi-objective optimization are two examples. An optimization algorithm needs to find multiple such solutions to solve the associated problem. Due to the population approach, evolutionary computation (EC) algorithms have been suitably modified to find and store multiple optimal solutions in a single application. This is also made possible due to their implicit parallelism property which causes multiple good and diverse solutions to be created and emphasized simultaneously within a population. However, these modified EC algorithms do not follow any restriction in finding multiple optimal solutions in any specific order. However, there exist certain problems, in which a hierarchical discovery of multiple critical solutions is must to constitute the desired set of solutions. A recently introduced *innovation path* (IP) problem is one such problem. In solving such problems, point-based algorithms, which find one targeted solution in a single application, becomes relevant. In this paper, we address this important aspect of convergence behavior of an evolutionary multi-objective optimization (EMO) algorithm vis-a-vis a point-based algorithm. The challenges of both approaches are discussed with examples. Results on a number of problems show that despite the apparent advantage of a point-based approach for finding hierarchical solutions one by one, the proposed population-based EMO approach enables a flexible and global search for finding multiple IP solutions in a single run. This study raises a fundamental question related to the convergence behavior of EC/EMO algorithms in finding an ordered set of solutions.

CCS CONCEPTS

• **Theory of computation** → *Convex optimization*; **Nonconvex optimization**; **Genetic programming**; Stochastic approximation.

KEYWORDS

EVolutionary algorithms, Multi-solution algorithms, Alternate solutions, Multi-objective optimization, Hierarchical convergence

1 INTRODUCTION

Population-based evolutionary algorithms (EAs) have been extended to find multiple optimal solutions for multi-modal problems [9, 15] and multiple Pareto-optimal solutions in multi- and many-objective optimization problems [1, 4] in a single simulation run. These algorithms are successful due to the implicit parallel search property of EAs, which through their selecto-recombination operators are able to create and emphasize multiple good yet diverse solutions within an evolving population. For discovering multiple diverse solutions, this property enables multiple regions to be searched in a parallel manner, thereby making them more computationally efficient than finding each optimal solution at a time by a point-based optimization algorithm [16].

However, one aspect of these multi-optimal problems is that they do not demand that multiple optimal solutions be found in any specific order. For multi-modal problems, any one of the global or local optimal solutions can be found first, followed by another global or local solution in a different part of the search space, without any contiguity or order. Therefore, population-based multi-solution seeking algorithms were also developed in a free-form manner so that evolution can emphasize multiple solutions in any order of appearance, as found convenient by the algorithm. As long as a diverse and well-converged set of optimal solutions are found at the end, the algorithm's performance is judged well. No particular attention is made in forcing the order and sequence of discovery of multiple solutions. Implicit parallelism, as it is argued in the literature [8, 10], works well when such free-form evolution can be tolerated to solve a problem. Clearly, many efficient multi-modal EAs and evolutionary multi- and many-objective optimization (EM(a)O) algorithms have shown successful discovery of multiple solutions enjoying the free and unordered discovery of solutions.

However, there exist certain problems in practice, which demand that multiple solutions must be found in a specific order of sequence, resulting in a *path* or an ordered set of solutions. The optimality of a solution in the path closely depends on the previous solution in the path. Thus, it does not make sense to find an arbitrary member of the path by not finding previous members of the path. We present one such problem, which has been recently introduced.

Imagine a practical scenario, in which there is an existing solution $\mathbf{x}^{\mathrm{C}}$ which needs an update, simply because of the fact that this solution is not quite appropriate to the current circumstances due to availability of more efficient technologies, materials, and resources or due to a change in focus and policy of the organization. When optimized for the new circumstances, a completely different and new solution $\mathbf{x}^{\mathrm{T}}$ emerges, however this target solution may be quite different from $\mathbf{x}^{\mathrm{C}}$ requiring a huge cost or time to make its implementation or adoption. Instead of making a single huge step to update the existing solution, 'Kaizen' principle [18] suggests to take multiple yet gradual small changes to make the transition. Although Such an approach may require multiple adoptions and changes, but demands less cost and time to make each change and more importantly may motivate human users to accept the changes. This problem was referred to as the 'innovation path' (IP) problem [11], constituting a path (an ordered set) of solutions starting from the existing solution $\mathbf{x}^{\mathrm{C}}$ and enabling allowable changes to gradually convert the solution to or near $\mathbf{x}^{\mathrm{T}}$. Clearly, the allowable changes between two consecutive solutions in the IP must be predefined using *step-constraints* and may simply be the difference in variable vectors between two consecutive solutions in the path is at least a threshold value.

It is clear from the above description that a hierarchical process must be followed for the task of finding IP solutions. Since i-th solution in the path is defined accurately only when a stable $(i-1)$-th solution in the path is found, the problem demands a hierarchical convergence of solutions. The behavior of an EA or EM(a)O algorithm in following such a hierarchical convergence process is the matter of discussion of this paper.

A deep-rooted question regarding the working principles of EAs is as follows: 'Are the canonical multi-solution EAs – multimodal and multi-objective EAs, for example – suitable for hierarchical convergence to mutliple solutions?' If not, 'How they may be changed to make them suitable?' Another interesting question is 'Are population-based and multi-solution EAs competitive to point-based optimization algorithms?' The latter question is relevant for this work, since it makes an intuitive sense to find one solution at a time using a point-based optimization algorithm, thereby addressing the hierarchical convergence issue by decomposing the multi-solution problem into a number of independent optimization tasks. In such an approach for the IP problem, the first intermediate solution of the path can be found from the existing solution by satisfying all step-constraints and original constraints. Then, the second intermediate solution can be found from the obtained first intermediate solution and so on.

In the remainder of this paper, we provide a foundation of multi-solution algorithm by briefly describing multi-modal and multi-objective EAs in Section 2. Section 3 revisits the innovation path problem and briefly describes a previously proposed EA for finding a set of hierarchical solutions. Section 4 proposes a sketch of point-based optimization procedure for finding a set of hierarchical solutions. Thereafter, in Section 5, three different challenges with the point-based approach are highlighted in the context of a few test problems. Section 6 presents comparative results of IP-seeking EA with the point-based optimization approach on a number of problems. A discussion of the results is made in Section 7. Finally, conclusions are drawn in Section 8.

2 MULTI-SOLUTION EAS

Most EA studies dealing with a single objective function aim to find a single optimal solution. However, there have been some studies in which EAs are extended to find multiple solutions simultaneously at the end of simulation run. Since, EAs work with a population of solutions in each generation, this task is theoretically possible. Here, we briefly describe two such multi-solution EAs – multi-modal and multi-objective EAs.

Multi-modal EAs aim to find multiple global (additionally, in some instances, local) solutions in a single simulation run. Just having a population in a search algorithm is not enough to find and capture multiple diverse solutions. The search process must be more exploitative so that each potential population member representing a different optimal region must be allowed to evolve independently and in competitively with other potential members, but also be simultaneously explorative so that an implicit parallel search can partake in multiple optimal regions. Goldberg and Richardson [9] devised a sharing function based niche-preserving genetic algorithm which updates a GA's selection algorithm so that every population member is compared against its neighboring members and two distant population members in the variable space are discouraged. This allowed multiple and distant good solutions to evolve simultaneously, thereby finding multiple optima at the end of the simulation run. Other niched EAs with a similar concept are also proposed [7, 13, 17].

Multi-objective optimization problems give rise to multiple Pareto-optimal (PO) solutions, which includes, in addition to each individual optimal solution for each objective, a number of other compromise trade-off optimal solutions. Evolutionary multi-objective optimization (EMO) algorithms modified the selection operator of single-objective EAs by emphasizing *non-dominated* solutions which are not inferior to any other solution in the population, and by establishing a niching operation so that multiple non-dominated solutions are simultaneously emphasized. These basic principles are implemented in many different ways in the EMO literature to devise different EMO algorithms [2, 5, 21]. Later algorithms solved more than three-objective optimization problems – referred to as many-objective problems – by supplying a set of diverse reference vectors (RVs) in the objective space and expecting population members to follow close to each RV in parallel fashion and approach the respective PO surface [6, 20].

2.1 Properties of Multi-Solution EAs

It is clear from the above discussion that the niched EAs and EMO algorithms are not particularly designed to find multiple global/local or PO solutions in any particular order. They can find any arbitrary global/local optimum or any specific PO solution first and then focus on any other optima as the algorithm finds convenient based on the supplied objective and constraint function landscapes. These algorithms have enjoyed the free-form nature of the problem description. In this paper, we investigate the potential of multi-solution EAs in finding multiple solutions, specifically when they are required to be found in a hierarchical manner.

3 INNOVATION PATH (IP) PROBLEM: AN EXAMPLE REQUIRING HIERARCHICAL CONVERGENCE

Here, we revisit a multi-solution problem – innovation path (IP) problem – which requires its multiple solutions in a hierarchical manner.

3.1 IP Problem Description

Consider a practical scenario in which users are unwillingly forced to use an existing solution ($\mathbf{x}^{\mathrm{C}}$), despite knowing that it is not efficient in the context of new goals, new available resources, or newly available technologies. This is because when they find the new optimal (target) solution ($\mathbf{x}^{\mathrm{T}}$), it is quite different from $\mathbf{x}^{\mathrm{C}}$ in terms of the cost, time and extent of changes needed to make the transition. This calls for a few gradual changes from $\mathbf{x}^{\mathrm{C}}$ to transition to $\mathbf{x}^{\mathrm{T}}$ meeting certain maximal affordable and viable changes between two consecutive solutions. This set of ordered solutions from $\mathbf{x}^{\mathrm{C}}$ to $\mathbf{x}^{\mathrm{T}}$ are referred to as an innovation path [11]. Clearly, this problem is a multi-solution problem, but unlike other problems, two consecutive solutions are closely related to each other with a number of step-constraints. The i-th intermediate solution can not be defined exactly without a stably converged $(i-1)$-th intermediate solution.

To solve the IP problem, a previous study [11] proposed a bi-objective optimization approach to find multiple IP solutions. For a M-objective goal set, the IP-seeking algorithm (IP-EMO) minimized two objectives: (i) the achievement scalarization function (ASF) [19] at a solution $\mathbf{x}$ in the hope of improving all M objectives simultaneously and (ii) a difference metric between $\mathbf{x}^{\mathrm{C}}$ and solution $\mathbf{x}$.

$$\begin{aligned}
\text{Minimize} \quad & \left(\mathrm{ASF}(\mathbf{x}),\ \|\mathbf{x}-\mathbf{x}^{\mathrm{C}}\|\right), \\
\text{subject to} \quad & G_l(\mathbf{x},\mathbf{x}^{(k)}) \leq 0, \quad \mathbf{x}^{(k)} \in \mathbf{x}^*, \\
& \qquad \forall k = 1,\ldots,|\mathbf{x}^*|, \quad l = 1,2,\ldots,L, \\
& g_j(\mathbf{x}) \leq 0, \quad j = 1,2,\ldots,J, \\
& x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1,2,\ldots,n.
\end{aligned} \tag{1}$$

The step-constraints G_l ($l = 1,\ldots,L$) ensure that two consecutive intermediate IP solutions ($\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$) have certain minimum difference in them. One step-constraint can be:

$$G_l\left(\mathbf{x}^{(i)},\mathbf{x}^{(j)}\right) \equiv \Delta - \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\| \leq 0. \tag{2}$$

Problem constraints g_j and variable bounds are included as well. The set $\mathbf{x}^*$ contains the intermediate anchor solutions found so far. Due to the evolving nature of this set, which in turn defines the constraint set, the problem is difficult to optimize.

If thought carefully, these two objectives are conflicting to each other and one of the extreme PO solutions (minimal second objective) for this bi-objective problem is $\mathbf{x}^{\mathrm{C}}$. Theoretically, the other extreme of the PO set is the best desired M-objective solution ($\mathbf{x}^{\mathrm{T}}$). The intermediate PO solutions for this bi-objective problem are potential IP solutions with a path starting from $\mathbf{x}^{\mathrm{C}}$ to $\mathbf{x}^{\mathrm{T}}$. Associated step-constraints discretize the entire PO set into a few finite anchor solutions for the IP problem.

3.2 Population-based IP-seeking EA

Before we discuss the IP-seeking NSGA-II, we apply the original NSGA-II [5] with a slight modification as follows. The existing solution $\mathbf{x}^{\mathrm{C}}$ is included in the initial population and all non-dominated (ND) solutions which satisfied step-constraints starting with $\mathbf{x}^{\mathrm{C}}$ are chosen as first ND front. The remaining ND solutions are called as members of second front. Rest of the NSGA-II procedures are kept as they are. Figure 1 shows IPs found for the two-objective ZDT6 problem. It is clear that despite having the same current solution ($\mathbf{x}^{\mathrm{C}}$) on the IP, the original NSGA-II is not able to produce a better IP compared to IP-NSGA-II. Some changes [12] in NSGA-II were made to develop IP-NSGA-II procedure for finding a much desired ordered set of IP solutions. Next, we briefly highlight the changes made to the original NSGA-II.

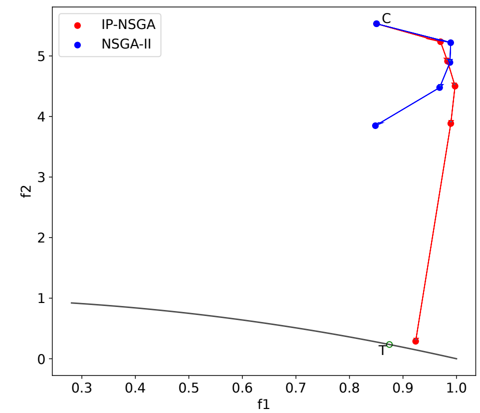


Figure 1: IPs found by original NSGA-II and IP-seeking NSGA-II for ZDT6 in the original problem space. Original NSGA-II fails to find adequate IP solutions.

To enable IP-seeking NSGA-II to identify an ordered set of solutions, the ranking, survival, and selection operators of the original NSGA-II are updated. The crowding distance operator is also replaced with a new measure called ‘Association’ (α), and a new definition of *directed domination* is introduced.

To rank solutions across fronts at each generation, the first front comprises of solutions that satisfy step constraints and are non-dominated under the directed domination definition [12]. These solutions are referred to as *anchors*. The remaining solutions are associated with specific anchors [11]. Once the population is ranked and associated, anchors are given priority during the acceptance phase for the next generation. All other solutions are subsequently accepted, front by front, based on their association with anchors and the degree of step-constraint violation defined in [12].

The updated mating selection process is detailed in Algorithm 1. In this approach, one parent is always chosen as an anchor point, j_a, while the second parent is selected from two randomly chosen members of the remaining population. The individual with the better ‘lexmin’ constraint-domination operator (Step 8) is selected as the second parent. Under this operator, if both candidates (s_1 and s_2) are infeasible, the one with the smaller problem constraint violation (CV) is chosen. If their CV values (including zero) are identical, the solution with the smaller non-dominated (ND) rank is preferred. If ND ranks are also tied, the solution closest to $X(j_a)$ in terms of $\mathbf{x}.\alpha$ is selected. Finally, in the event of a further tie with $\mathbf{x}.\alpha$, the solution with the larger overall step-constraint violation (CD) is chosen.

4 POINT-BASED APPROACH

Point-based optimization algorithms [3, 14] work with a single solution at each iteration. Starting with an initial guess solution, the algorithm iteratively progresses towards an optimal solution. For solving multi-solution problems, point-based algorithms must find one optimal solution at a time. For this purpose, the original optimization problem is first parameterized and is solved to find the respective optimal solution. Thereafter, parameters are changed and the revised problem is solved again to hopefully find a different

---

Algorithm 1: Mating Selection Operator of IP-NSGA-II

Input: Current anchor A_{t-1}, population P_t, γ, $\mathbf{C}_{t-1}$ and ζ
Output: Two selected parents p1 and p2 for mating

X = Ordered index list of A_{t-1} using $\mathbf{C}_{t-1}$ and ζ;
$\boldsymbol{\rho}$ = AnchorSelectProb(γ, X);
r = random(0,1);
$j_a = \text{argmin}\{\sum_{i=1}^{j} \rho_j \geq r | j = 1, \ldots, |\boldsymbol{\rho}|\}$;
$p1 = A_{t-1}(X(j_a))$; // first parent;
$s1 = \text{random}\{\mathbf{x} | \mathbf{x} \in P_t \backslash p1\}$;
$s2 = \text{random}\{\mathbf{x} | \mathbf{x} \in P_t \backslash p1\}$;
$p2 = \text{lexmin}\{(\mathbf{x}.\text{CV}, \mathbf{x}.\text{ND}, \mathbf{x}.\alpha, -\mathbf{x}.\text{CD}) | \mathbf{x} \in (s1, s2)\}$; // second parent, see text for details

---

optimal solution. Since the optimal solutions can be *controlled* by the choice of the parameters, the method provides a hope to find an ordered set of optimal solutions systematically. We investigate the performance of one such point-based algorithm (MATLAB's `fmincon()` routine) for this purpose.

The following parameterized single-objective problem $P(\mathbf{z}^{(t)})$ (where $\mathbf{z}^{(t)}$ indicates the reference point for finding the t-th anchor in the ordered set and is set as the $(t-1)$-th anchor $(\mathbf{x}^{*,(t-1)})$) of the bi-objective problem presented in Equation 1 is formulated for the `fmincon()` routine:

$$
\begin{aligned}
\text{Minimize} \quad & \text{AASF}(\mathbf{x}) = \max_{i=1}^{2} \left(f_i(\mathbf{x}) - z_i^{(t)}\right) \\
& \qquad + \rho \sum_{j=1}^{2} \left(f_i(\mathbf{x}) - z_i^{(t)}\right), \\
\text{subject to} \quad & G_l(\mathbf{x}, \mathbf{x}^{*,(t-1)}) \leq 0, \quad l = 1, 2, \ldots, L, \\
& g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \ldots, J, \\
& x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \ldots, n.
\end{aligned} \tag{3}
$$

Here, $f_1(\mathbf{x}) = \text{ASF}(\mathbf{x})$ and $f_2(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^{\text{C}}\|$. A small value $\rho = 10^{-3}$ is used. Here we use each anchor as the reference point and unit weight vectors with a $\rho = 10^{-3}$. Additionally, we use the smoother version of the AASF function with an additional variable and two more constraints to get rid of the min-max formulation in the problem definition. An initial choice of $\mathbf{z}^{(1)} = \mathbf{f}(\mathbf{x}^{\text{C}})$ should produce the first intermediate anchor point (second element of the ordered set). Then, $\mathbf{z}^{(t)} = (\text{ASF}(\mathbf{x}^{*,(t-1)}), \text{Diff}(\mathbf{x}^{*,(t-1)}))$ (for $t \geq 2$) can be set. Figure 2 illustrates the working of the above formulation to find each IP solution one at a time. AASF ensures that a non-weak PO solution will always be found.

5 CHALLENGES WITH POINT-BASED APPROACHES

Here we identify some of the challenges discovered when finding IPs for different problems using the point-based approach.

5.1 Challenge 1: Cumulative Error

The point-based method is set to serially find IP solutions one after the other creating an ordered path. Thus, any error in converging to an anchor solution would carry forward to all proceeding anchors. The phenomenon is shown in Figure 3. Here we use the ZDT1 problem with the $\hat{G}_1$ and $\hat{G}_3$ step constraints shown in Equations 4 and 5 to find an IP which would diverge from the original PO front of ZDT1 given the current solution is one extreme of the ZDT1 PO front.

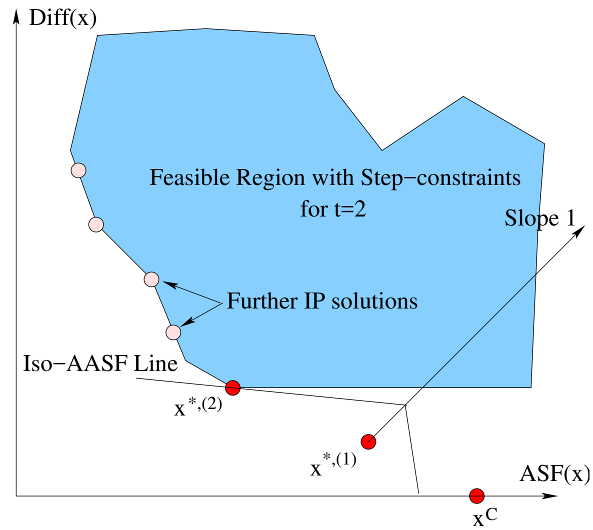


Figure 2: Minimum of AASF corresponds to t-th IP solution when $\mathbf{z} = \left(\text{ASF}(\mathbf{x}^{*,(t-1)}), \text{Diff}(\mathbf{x}^{*,(t-1)})\right)$. $\mathbf{x}^{\text{C}}$ and $\mathbf{x}^{*,(1)}$ are already found IP solutions.

$$\widehat{G}_1(\mathbf{x}, \mathbf{x}^{(k)}) = 1 - (f_{M+1}(\mathbf{x}) - f_{M+1}(\mathbf{x}^{(k)}))/\Delta_1 \leq 0 \tag{4}$$

$$\widehat{G}_3(\mathbf{x}, \mathbf{x}^{(k)}) = (s(\mathbf{x}^{(k)}) - s(\mathbf{x}))/\Delta_3 - 1 \leq 0 \tag{5}$$

As can be seen from the figure, starting with the current implementation (C) both approaches find different first anchors, where our proposed IP-NSGA-II approach gets an anchor at the intersection of both constraints, the point-based method finds an anchor with the same ASF value as our found anchor but with much larger difference objective value. As can be seen, the next anchor found by the point-based method, though at the intersection for both constraints still has an inherent offset in the difference objective making it sub-optimal in comparison to our found anchor.

5.2 Challenge 2: Multiple ASF Optimal Solutions

As highlighted in Section 5.1, the point-based method did indeed find a feasible solution satisfying step constraints minimizing the ASF objective, but it seems minimizing the step-constraint is not a priority for the algorithm as observed in Figure 4. Here we show the intermediate points (fc1-fc8) found by the point-based method before converging to the first anchor P1. We also plot the anchors found by our method (O1-O8), and the purple ASF line of the first anchor in the original objective space of ZDT1. As can be seen in the figure, the point-based search converges to the minimum allowable ASF solution and does not have further motivation to minimize the $\hat{G}_1$ step constraint as such it has the same ASF value found by our method but with a larger 'DIfference from CI' as shown in Figure 3.

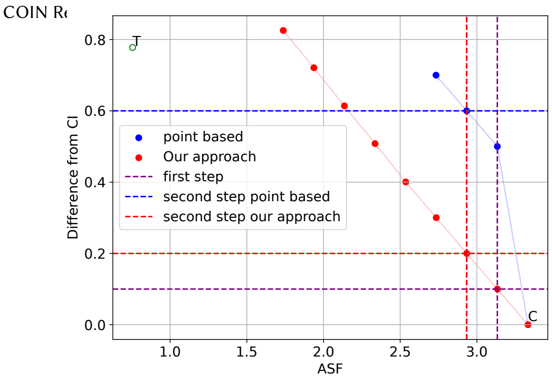


Figure 3: Error carried forward phenomenon of the point-based method on the ZDT1 problem.

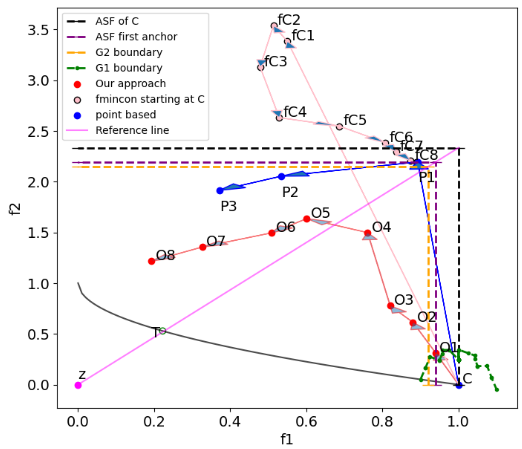


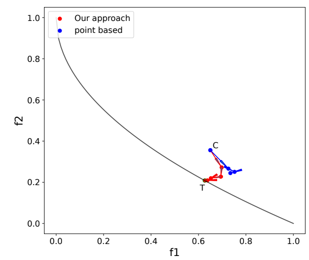
Figure 4: Identification of the ASF minimization focus of the point-based method on the ZDT1 problem.

5.3 Challenge 3: Local Optimality Issue

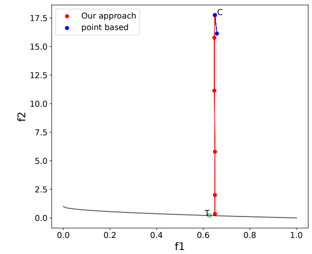
For problems with multiple local PO fronts, point-based methods may get stuck closer to the starting point and do not find more anchors closer to the target. This phenomenon is shown in Figure 5, where Figure 5a shows the IP found by both methods on a simplified ZDT4 problem with no local front present in the search space and Figure 5c shows the IP found by both methods for the regular ZDT4 problem. It can be seen as the complexity of the problem increases in terms of more and more local fronts the IP found by the point-based method gets stuck near the start of the path and does not discover any further anchors.

5.4 Alleviation of Above Challenges with Population-based Approach

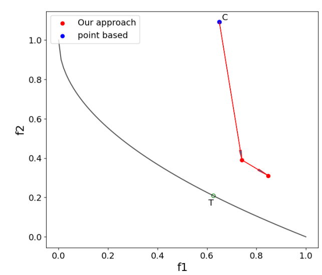
With the population-based approach, we do not face such problems because of the careful and intuitive updates of the NSGA-II algorithm. As we use a probabilistic parent selection of anchors described in Section 3.2, it serves as a repair operator where anchors found at the start still have a chance to mate and find better solutions till the end of the optimization run. Hence, even if there is an error in the initial anchors found it has a high chance of correcting itself later in the run, furthermore since we have a population



(a) Simplfied ZDT4 with no local optimal



(b) Less complex ZDT4 with few local fronts



(c) Normal ZDT4 problem

Figure 5: Converging to local fronts showcasing the difficulty with the point-based approach.

we do not consider an anchor stable till its present for the last γ generations preventing any immature convergence.

To ensure that our proposed algorithm converges to the intersection of different step constraints we use a directed domination principle [12] when selecting anchors for each generation. This principle prefers non-dominated solutions closer to the intersection of constraints ensuring better anchors for the path.

Lastly, the local optimality issue is not a real issue with EAs. As the population can parallely investigate different parts of the search space, it is less likely to get stuck in local optima. This particular phenomenon has been widely discussed and demonstrated in the EA literature.

6 COMPARATIVE RESULTS

Next, we compare the point-based and our proposed population-based IP-NSGA-II approaches on a number of well-studied multi- and many-objective problems where we restrict the number of evaluations of IP-NSGA-II to the same number of evaluations needed by the point-based method as a baseline. Furthermore, since IP-NSGA-II is stochastic in nature, we run it 12 times with different initial populations and compare each path found with the point-based method. We then present results of the median path found in terms of the comparison metric defined below.

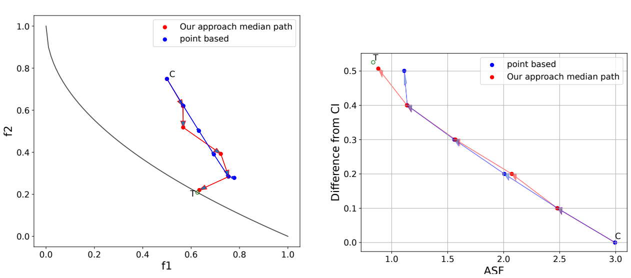
To compare two ordered sets found by two different algorithms, we can use the well-known hypervolume metric, applied to the ASF-Diff objective space, as our goal is to find a set of PO solutions in this space. But there is a caveat which we need to discuss. Due to the pair-wise step-constraints, the exact minimum-ASF solution may not be achievable. In this case, one extreme of the resulting PO set may not be easy to determine. Hence, despite the non-domination nature of the rest of the IP solutions, the extreme IP solution may introduce a difference in hypervolume. Nevertheless, we use the following metric $\Delta_{\text{HV}} = \text{HV}_A - \text{HV}_B$. If $\Delta_{\text{HV}} < 0$ then path B is better than path A. While if $\Delta_{\text{HV}} > 0$ Path A is better than Path B. Both paths are considered equal if $\Delta_{\text{HV}} = 0$.

6.1 Multi-objective Problems

Table 1 shows the chosen parameter values and the $\Delta_{\text{HV}} = \text{HV}_{\text{pop}} - \text{HV}_{\text{point}}$ obtained by comparing the median path found by our approach to that of the point-based method. Figure 6 shows the IPs found by both population and point-based methods in the original and the extended problem spaces. As can be seen in Figure 6b IP-NSGA-II performs better in terms of getting closer to the target (T) and hence has a positive Δ_{HV} as seen in Table 1. Figure 7 shows the IP comparison for ZDT2 problem. Here both approaches perform equally well, although the point-based approach has a slight edge when compared against the median-performing IP-NSGA-II run. Figure 8 shows the IP comparison for ZDT6. IP-NSGA-II performs better than point-based approach, as prominent in both plots of Figure 8. After the fourth IP point, the point-based approach cannot find a near-target solution, although it manages to satisfy the step-constraint but converges to a far-away point from the target, as shown clearly in Figure 8a.

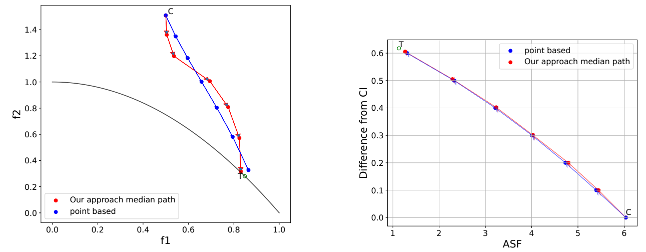
Next, we compare both algorithms on ZDT1 with one extreme of the original PO front as the current solution $\mathbf{x}^{\text{C}}$ making the problem of finding a path a little simpler. Though contrary to the expectations of almost identical results, IP-NSGA-II performs better than the point-based approach by finding IP points on the original PO front, as shown in Figure 9a. Interestingly, the IP points on extended problem space (Figure 6b) are almost identical for both approaches.

Next, we compare IPs found by both algorithms on BNH and OSY (constrained) problems. Figures 10 and 11 show the IPs for BNH and OSY, respectively. As can be seen that IP-NSGA-II performs much better than the point-based approach which seems to struggle with constrained multi-objective problems. The point-based approach deviates from the true IP right from the beginning and accumulation of error from previous to the next IP point continues until the end of the path.



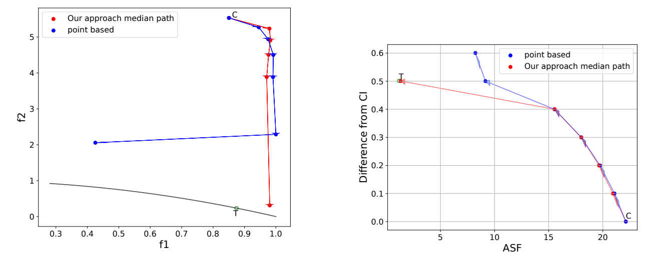
(a) Original problem space **(b) Extended problem space**

Figure 6: IP comparison for the ZDT1 problem in the original and extended problem spaces. Population-based approach performs better.



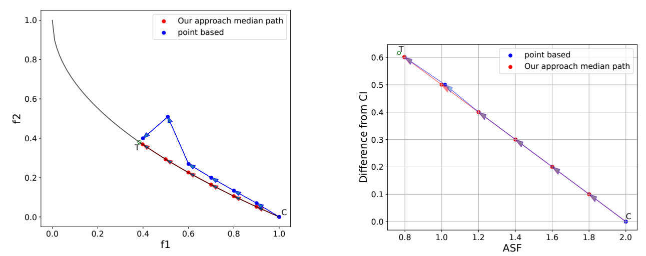
(a) Original problem space **(b) Extended problem space**

Figure 7: IP comparison for the ZDT2 problem in the original and extended problem spaces.



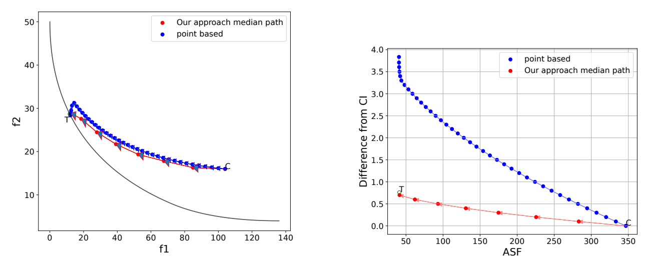
(a) Original problem space **(b) Extended problem space**

Figure 8: IP comparison for the ZDT6 problem in the original and extended problem spaces. Population-based approach performs better.



(a) Original problem space **(b) Extended problem space**

Figure 9: IP comparison for the ZDT1 problem in the original and extended problem spaces with $\mathbf{x}^{\text{C}}$ being one of the extremes of the original PF. Population-based approach performs better.

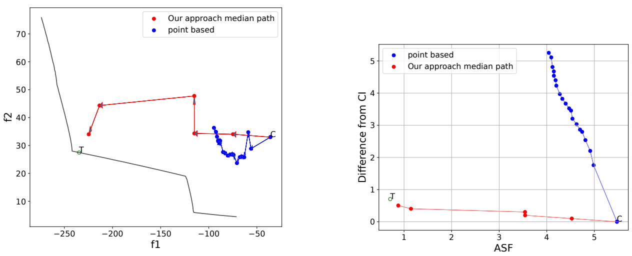


(a) Original problem space **(b) Extended problem space**

Figure 10: Population-based approach performs much better for BNH constrained problem, as shown in original and extended problem spaces.

Table 1: Chosen parameter values for problems with two-objective goals. Δ_{HV} for median IP-NSGA-II approach is shown.

	N	$T_{\max}$	γ	Step	**w**	**z**	Current solution $\mathbf{x}^C$	Δ_{HV}
ZDT1	100	38	0.2	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.002,0.001,0,0,0,0]	0.0025
ZDT2	100	39	0.25	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.002,0.001,0,0,0,0]	−0.021
ZDT6	100	47	0.3	0.1	[0.75,0.25]	[0,0]	[0.4,0.5,0.002,0.001,0,0,0,0]	0.022
ZDT1	100	25	0.15	0.1	[0.5,0.5]	[0,0]	[1,0,0,0,0,0,0,0]	0.0022
BNH	100	61	0.15	0.1	[0.3,0.7]	[0,0]	[5,1]	96.64
OSY	105	101	0.2	0.1	[0.2,0.8]	[−260,0]	[1,1,2,1,1,5]	1.87



(a) Original problem space **(b) Extended problem space**

Figure 11: Population-based approach performs much better for OSY constrained problem, as shown in original and extended problem spaces.

6.2 Many-objective Problems

As our algorithm is not restricted to multi-objective problems, we compare the results for many different many-objective problems. Table 2 shows the parameter settings used for different problems as well as the Δ_{HV} between the median path of our algorithm and the point-based method.

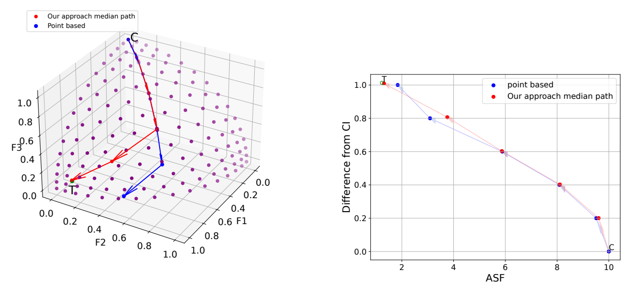
Figure 12 shows the IPs found for the DTLZ2-3obj problem. By Δ_{HV} metric, point-based method does slightly better compared to our approach. However, if we look at the actual paths in the original problem space (Figure 12a), it is clear that our method has found a better path to reach the target displaying the working of the algorithm in the original space, clearly indicating the need for a better performance metric for such studies.

Figure 13a shows the IPs found for the DTLZ2-5obj problem where our method does slightly better in terms of Δ_{HV}. Figure 13b shows the IPs found for the DTLZ2-10obj problem, and by the Δ_{HV} comparison, the point-based method seems to perform slightly better. However, on a closer look, it can be seen that the last few anchors found by the point-based method diverge away from the target point raising a question of their importance.

Figure 14 shows the IPs found for the C4DTLZ2-3obj problem where the point-based method does slightly better by finding the last anchor. Whereas for C4DTLZ2-5obj our method does a lot better compared to the point-based method as shown in Figure 15a. Our method also finds a better IP for the C4DTLZ2-10obj problem as shown in Figure 15b. With more solution evaluations than required by deterministic point-based algorithm, better performance may be achieved.

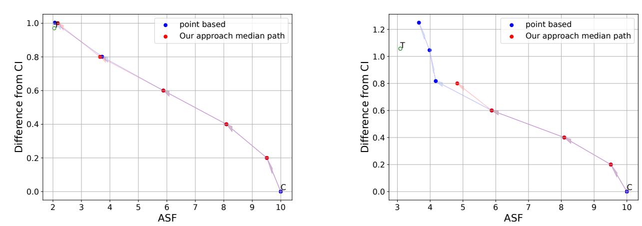
7 DISCUSSION

Point-based optimization methods are primarily designed to identify a single optimal solution, making them the preferred choice for single-objective problems. In contrast, population-based methods



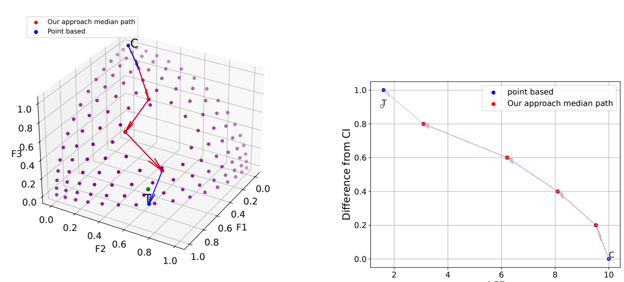
(a) Original problem space **(b) Extended problem space**

Figure 12: IP comparison for the DTLZ2-3obj problem in the original and extended problem spaces. Population approach performs better (12a), but extended space plot (12b) does not reflect that.



(a) DTLZ2-5obj **(b) DTLZ2-10obj**

Figure 13: IP comparison for the DTLZ2-5obj and DTLZ2-10obj problems in the extended problem space. Point-based approach gets closer to target point on 10-objective problem.

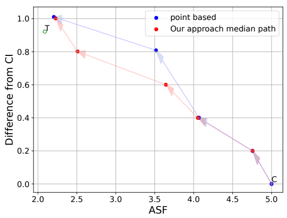


(a) Original problem space **(b) Extended problem space**

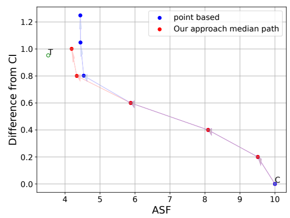
Figure 14: IP comparison for the C4DTLZ2-3obj problem in the original and extend problem spaces. Point-based approach gets closer to target point.

Table 2: Chosen parameter values for three and many-objective problems.

	N	$T_{\max}$	γ	Step	$\mathbf{w}$	$\mathbf{z}$	Current solution $\mathbf{x}^C$	Δ_{HV}
DTLZ2-3obj	50	40	0.15	0.2	[0.8,0.1,0.1]	$z_i = 0, \forall i$	$x_1 = 1$ and $x_i = 0.5$ for $i = 2, 3, \ldots, 12$	−0.148
DTLZ2-5obj	70	65	0.2	0.2	[0.3,0.3,0.2,0.1,0.1]	$z_i = 0, \forall i$	$x_1 = 1$ and $x_i = 0.5$ for $i = 2, 3, \ldots, 12$	0.013
DTLZ2-10obj	100	62	0.15	0.2	[0.15,0.05,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1]	$z_i = 0, \forall i$	$x_1 = 1$ and $x_i = 0.5$ for $i = 2, 3, \ldots, 12$	−0.143
C4DTLZ2-3obj	50	75	0.2	0.2	[0.5,0.4,0.1]	$z_i = 0, \forall i$	$x_1 = 1$ and $x_i = 0.5$ for $i = 2, 3, \ldots, 12$	−0.008
C4DTLZ2-5obj	70	51	0.2	0.2	[0.3,0.3,0.1,0.1,0.2]	$z_i = 0, \forall i$	$x_1 = 1$ and $x_i = 0.5$ for $i = 2, 3, \ldots, 12$	0.213
C4DTLZ2-10obj	77	105	0.2	0.2	[0.15,0.05,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1]	$z_i = 0, \forall i$	$x_1 = 1$ and $x_i = 0.5$ for $i = 2, 3, \ldots, 12$	0.032



(a) C4DTLZ2-5obj



(b) C4DTLZ2-10obj

Figure 15: IP comparison for the C4DTLZ2-5obj and C4DTLZ2-10obj problems in the extended problem space. The population-based IP-NSGA-II performs better.

are better suited for discovering multiple optimal solutions, making them ideal for multi-objective optimization problems resulting in multiple Pareto-optimal solutions. This difference arises from their underlying mechanisms: point-based methods typically employ local search strategies, iteratively converging to the optimal solution point by point. On the other hand, population-based methods explore different parts of the search space in parallel, enabling them to identify diverse optimal solutions without being confined to a single trajectory.

However, certain problems, such as the IP problem addressed in this study, fall outside the typical use-cases of both point-based and population-based methods. The IP problem requires the discovery of a chronological sequence of intermediate solutions, forming a solution path from a current to a near-target solution. This sequential nature demands that each solution in the path is meaningful only if the preceding solution has already converged to its rightful place. As a result, standard population-based methods, which are designed to find multiple non-dominated (ND) solutions in parallel and not in any specific order, may not perform well in such problems. Similarly, while point-based methods could theoretically be applied iteratively to build the path one by one starting from the current solution, they would require decomposing the problem into multiple sub-problems. Each sub-problem need to be formulated in a way so that its optimal solution corresponds to the desired intermediate solution in the path, adding significant complexity.

Despite the apparent advantage of point-based approaches in solving the IP problem, it has been observed that they have their own challenges – error accumulation from inaccurate convergence of previous IP solutions, presence of multi-modal solutions of the chosen scalarized function, and convergence to local optimal solutions.

However, major changes in standard EMO algorithms are needed to make them suitable for finding hierarchical solutions. One such implementation (IP-NSGA-II) has updated its ranking, survival, and mating selection procedures with a directed domination principle that dynamically emphasizes the search region near the latest IP solution. Adaptive methods for determining adequate stability of an IP solution are enforced before forwarding the focus to the next IP solution. The population approach allows future corrective measures for an inaccurately converged IP solution at any generation during the optimization process. Results with IP-NSGA-II are, in general, much better than the point-based approach. This highlights the fact that despite apparent advantage of point-based optimization approaches in solving hierarchically convergent problems, their idiosyncratic properties make them vulnerable, whereas despite apparent issues with evolutionary population-based approaches in making ordered convergence, a few changes in them allow them to find a set of ordered solutions more reliably.

The extent of changes needed in a standard EMO algorithm for making it suitable for problems requiring an ordered convergence of solutions raises several interesting issues. Besides the IP problem, what are other problem types which would require a hierarchical convergence to multiple optimal solutions? Has there been any evidence of a hierarchical convergence in natural evolutionary process? Does there exist a better dynamically focused EMO approach to find an ordered set of PO solutions. These questions are fundamental in the way we understand the working of a standard EC or EMO. Further attention must be made to investigate what changes are needed to make EC and EMO algorithms more suitable for solving such problems.

8 CONCLUSIONS

The study addresses key questions regarding the applicability and performance of Evolutionary Algorithms (EA) and evolutionary multi- and many-objective (EM(a)O) algorithms for solving problems requiring a hierarchical convergence of solutions. Specifically, it has explored whether these algorithms are suitable for achieving hierarchical convergence to multiple solutions and whether they are competitive with point-based methods for such tasks. To tackle the unique challenges posed by hierarchical convergence problems, the study has proposed modified versions of both point-based and population-based techniques.

The findings indicate that population-based methods, when appropriately adapted, are more effective than point-based methods for tasks requiring hierarchical convergence. By testing the proposed methods across a range of well-known optimization problems, the study has demonstrated that population-based methods are not only competitive but often superior in scenarios demanding sequential and hierarchical solution discovery.

REFERENCES

[1] C. A. C. Coello, D. A. VanVeldhuizen, and G. Lamont. 2002. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Boston, MA: Kluwer.

[2] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. 2001. PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*. San Mateo, CA: Morgan Kaufmann Publishers, 283–290.

[3] K. Deb. 1995. *Optimization for Engineering Design: Algorithms and Examples*. New Delhi: Prentice-Hall.

[4] K. Deb. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK.

[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. 2002. A fast and Elitist multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[6] K. Deb and H. Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (2014), 577–601.

[7] K. A. DeJong. 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph. D. Dissertation. Ann Arbor, MI: University of Michigan. Dissertation Abstracts International 36(10), 5140B (University Microfilms No. 76-9381).

[8] D. E. Goldberg. 1989. *Genetic Algorithms for Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.

[9] D. E. Goldberg and J. Richardson. 1987. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*. 41–49.

[10] K. A. De Jong. 2006. *Evolutionary Computation: A Unified Approach*. MIT Press.

[11] A. Khan and K. Deb. 2024. Innovation Path: Discovering an Ordered Set of Optimized Intermediate Solutions from an Existing to a Desired Solution. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 529–537.

[12] Ahmer Khan and Kalyanmoy Deb. 2024. *Towards an Efficient Innovation Path Seeking Algorithm Using Directed Domination*. Technical Report COIN Report Number 2024007. Michigan State University, East Lansing, USA.

[13] A. Pétrowski. 1996. A Clearing Procedure as a Niching Method for Genetic Algorithms. In *IEEE 3rd International Conference on Evolutionary Computation (ICEC'96)*. 798–803.

[14] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell. 1983. *Engineering Optimization Methods and Applications*. New York : Wiley.

[15] Mahesh Shankar, Palaniappan Ramu, and Kalyanmoy Deb. in press. A Directed Batch Growing Self-Organizing Map based Niching Differential Evolution for Multimodal Optimization Problems. *Swarm and Evolutionary Computation* (in press).

[16] P. Shukla and K. Deb. 2005. Comparing classical generating methods with an evolutionary multi-objective optimization method. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO-2005)*. 311–325. Lecture Notes on Computer Science 3410.

[17] F. Streichert, G. Stein, H. Ulmer, and A. Zell. 2003. A clustering based niching EA for multimodal search spaces. In *Proceedings of the International Conference Evolution Artificielle*. 293–304. LNCS 2936.

[18] Bunji Tozawa and Norman Bodek. 2009. *How to Do Kaizen: A New Path to Innovation: Empowering Everyone to be a Problem Solver*. PCS Press.

[19] A. P. Wierzbicki. 1980. The use of reference objectives in multiobjective optimization. In *Multiple Criteria Decision Making Theory and Applications*, G. Fandel and T. Gal (Eds.). Berlin: Springer-Verlag, 468–486.

[20] Q. Zhang and H. Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on* 11, 6 (2007), 712–731.

[21] E. Zitzler, M. Laumanns, and L. Thiele. 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou et al. (Ed.). International Center for Numerical Methods in Engineering (CIMNE), 95–100.