Towards an Efficient Innovation Path Seeking Algorithm Using Directed Domination

Ahmer Khan and Kalyanmoy Deb

Michigan State University, East Lansing MI 48823, USA
khanahm2@msu.edu, kdeb@msu.edu
https://www.coin-lab.org

Coin Report Number 2024007

Abstract. In practice, users are often stuck with an existing solution, despite agreeing on the fact that the solution needs a substantial change. The hesitation might stem from the hefty cost or amount of effort required to adopt a new solution or could just be human apathy to large changes. In this regard, a recent preliminary study proposed a step-constrained based bi-objective optimization approach which attempts to discover a set of acceptable intermediate solutions starting from the current solution to the desired target solution leading to an *innovation path* (IP). Intermediate solutions, obtained using a multiobjectivization approach, reduce the amount of change required between two successive steps, thereby facilitating multiple gradual changes more acceptable by the users. In this paper, we propose a directed domination concept to make the IP-seeking algorithm more computationally efficient. Results on a number of test and engineering problems reveal that the proposed new approach reaches closer to the target solution and finds closer to optimally trade-off solutions than the previous IP approach.

Keywords: Innovation Path · Directed Domination · Kaizen Principle.

1 Introduction

Instigated by a need to change the existing solution (a design, process or a tool that defines a current implementation of a problem) for achieving a new and a more appropriate goal, users are always skeptical on the extent of change the new optimized solution will cause from the existing solution. In scenarios where the new solution is quite different from the current in-practice solution, implementing the new solution might become undesirable, as it could incur a heavy cost and laborious effort. In such scenarios, a series of solutions is required that leads from the current to the desired solution with a finite number of small gradual changes. Such a series of solutions are referred to as Innovation Path (IP) in recent studies [10, 11]. These paths are based on the well-known "Kaizen" principle [13, 18] which argues that humans are more amenable to accept a few gradually-changed solutions as a sequence of improvements from the current solution, rather than one large change, following the so-called "Kaikaku" approach [9]. Traditionally, the Kaizen principle has been used for industrial manufacturing processes [1, 2, 4, 8, 15, 16], but no generic algorithmic approach is proposed to find intermediate solutions making a gradual change.

The previous IP-seeking algorithm proposed an evolutionary multi-objective optimization approach minimizing two objectives: (i) extent of change and (ii) desired goal(s) for improvement and satisfying user-specified step-constraints which restrict a limiting change desired between two consecutive solutions on the IP. Although the previous IP was able to find a path of solutions satisfying step-constraints, in some problems, the final solution deviated far from the true target solution and in some problems, ended up prematurely with a few intermediate solutions, due to the simplicity adopted in identifying anchor points (which ultimately leads to IP solutions) in the very first study. Moreover, the preliminary idea of identifying the anchor points in a population ignored dominated solutions, which may find themselves to be valid anchor points when the solutions that dominated them became infeasible based on step-constraints by the new anchor points. The simplicity of the anchor point identification also made the process expensive and not particularly efficient.

In this study, in addition to the amplitude of the step, we also consider the direction of such a step toward the next anchor solution that ultimately leads closer to the desired target and is also computationally more efficient. We test the algorithm on a number of test and engineering problems and compare its results with the previous IP-seeking algorithm serving as the baseline method. We additionally test on a few new many-objective problems not considered in the preliminary study [10].

In the remainder of the paper, Section 2 presents a brief introduction to the previous IP-seeking algorithm and highlights its shortcomings. Section 3 describes the modified IP-seeking algorithm by defining the directed dominance concept. Section 4 presents extensive results on test and engineering problems. Section 5 concludes this study with possible future extensions.

2 Existing Innovation Path Seeking Algorithms

In recent literature, we have devised a simple technique to find IPs [10]. The method defined a bi-objective problem using the "multiobjectivization" principle proposed by EMO researchers in the past [3, 5–7, 12, 14, 17]. The Pareto Front (PF) of this bi-objective problem aims to contain the IP solutions. For a set of M new objectives to be optimized simultaneously as a target, our IP-seeking algorithm always formulates a bi-objective problem, in which one of the objectives is the function $f_{M+1}(\mathbf{x})$ as the difference between the variable vector ($\mathbf{x}$) and the prescribed current feasible solution ($\mathbf{x}^{\mathrm{C}}$), to be minimized:

$$f_{M+1}(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^{\mathrm{C}}\|_2. \tag{1}$$

While the second objective is the new goal to be optimized in the case of single-objective scenario, a scalarized objective $s(x)$, such as the achievement scalarization function (ASF) [19] with objective preference information in terms of weights ($\mathbf{w}$) and an aspiration point ($\mathbf{z}$), was proposed in the case of multi- or many-objective scenarios as new goal. To find the PF of the resulting bi-objective problem, the IP-seeking algorithm used the classical domination principle with an added layer of intermediate solutions (anchors) detection in a non-dominated sorted population [10]. Here, anchors are defined as non-dominated solutions satisfying the step-constraints; hence, the algorithm looked for the anchors only in the first non-dominated feasible set. This technique was tested on various single, and multi-objective problems with promising results.

2.1 Shortcomings of the Existing IP-seeking Algorithm

Though the preliminary algorithms exhibited promising results, it is still limited by its anchor discovery mechanism. Since [10] only looks for anchors in the first non-dominated set, it is unable to find anchors in case where the IP might deviate from the PF given specific step-constraints. Moreover, the anchors are dependent on the solutions available in the front, resulting in incomplete paths or paths with big jumps in between. Figure 1a shows an example of a non-dominated sorted population of 14 solutions with the current solution marked as C and a desired target solution marked as T. Note that the target is not part of the population but is marked to demonstrate the concept. Figure 1b shows the IP found by previous algorithm [10] for the sample population. The IP is found with the following step-constraint:

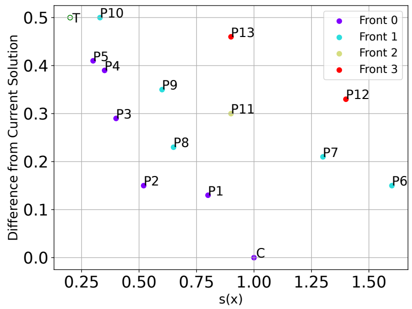
$$G_1(\mathbf{x}, \mathbf{x}^{(k)}) \equiv f_{M+1}(\mathbf{x}) - f_{M+1}(\mathbf{x}^{(k)}) \geq \Delta_1, \quad (2)$$

$$\widehat{G}_1(\mathbf{x}, \mathbf{x}^{(k)}) = 1 - (f_{M+1}(\mathbf{x}) - f_{M+1}(\mathbf{x}^{(k)}))/\Delta_1 \leq 0, \quad (3)$$

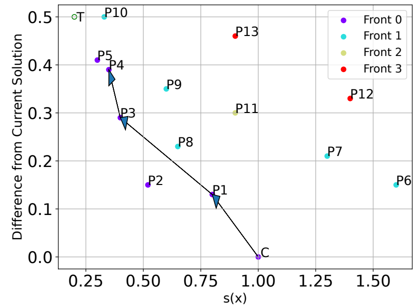
Here $\mathbf{x}$ is the variable vector associated with a population member and $\mathbf{x}^{(k)}$ is the previous anchor discovered by the algorithm thus far. To start, $\mathbf{x}^{(k)} = \mathbf{x}^{\mathrm{C}}$ (supplied current point). In this example, $\Delta_1 = 0.1$ such that the two consecutive IP solutions have at least Δ_1 difference in f_{M+1}. For the IP found by [10], seen in Figure 1b, there is a big jump from solution P1 to P3. as P2 does not satisfy the step-constraint G_1 being at least Δ_1 away from P1. Also, the path stops at P4, whereas solution P10 is closer to the target T. Lastly, since this algorithm moves in increasing value of the diff-obj (f_{M+1}), it selects P1 as the first anchor, where P2 might be a better trade-off choice, as it has a lower $s(x)$ value with a slight loss in diff-obj.

3 Proposed Update for Anchor Selection

In this study, we propose an update to the anchor finding mechanism where the algorithm does not go front by front but considers the ideal direction the path should move from one intermediate solution to the next. The basic characteristics of a new anchor is a solution that is not dominated by previously found anchors and satisfies the step-constraints, but it does not necessarily have to be a solution with minimum diff-obj value. To achieve this we transform all feasible solutions to a step-constraint (SC) space, where each axis represents a normalized step-constraint function value and the origin represents the latest found anchor. For



(a) NSGA-II sorting of a sample population.



(b) IP (C $\rightarrow$ P1 $\rightarrow$ P3 $\rightarrow$ P4) found by previous approach for sample population.

Fig. 1: ND-sorted sample population of 14 solutions and the IP by the previous approach [10].

L step-constraints, the SC-space is L-dimensional. Solutions are transformed to the SC-space using the following relation:

$$\widehat{S}_l(\mathbf{x}, \mathbf{x}^{(k)}) = 1 - \widehat{G}_l(\mathbf{x}, \mathbf{x}^{(k)}), \quad l = 1, 2, \ldots, L. \tag{4}$$

For example, the normalized SC function for the G_1 step-constraint defined in Equation 2 is given below:

$$\widehat{S}_1(\mathbf{x}, \mathbf{x}^{(k)}) = (f_{M+1}(\mathbf{x}) - f_{M+1}(\mathbf{x}^{(k)}))/\Delta_1. \tag{5}$$

We always include a default constraint $\widehat{S}_0(\mathbf{x}, \mathbf{x}^{(k)})$ originating from $G_0(\mathbf{x}, \mathbf{x}^{(k)})$, as follows:

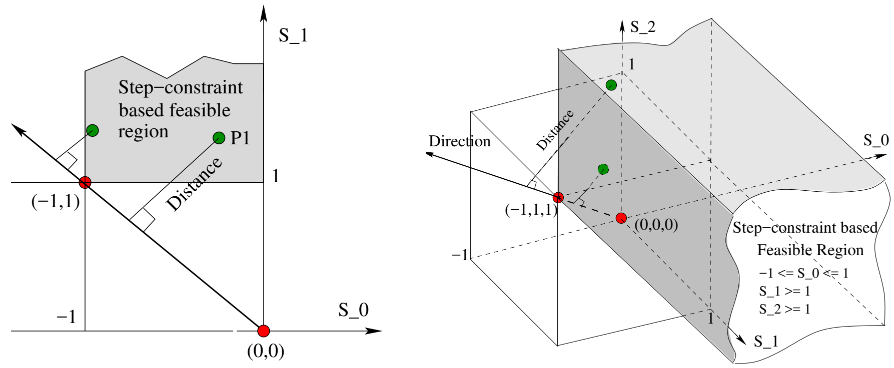
$$G_0(\mathbf{x}, \mathbf{x}^{(k)}) \equiv s(\mathbf{x}) - s(\mathbf{x}^{(k)}) \leq 0, \tag{6}$$

$$\widehat{S}_0(\mathbf{x}, \mathbf{x}^{(k)}) = -1 + s(\mathbf{x})/s(\mathbf{x}^{(k)}). \tag{7}$$

We use this default constraint realizing that $s(\mathbf{x})$ must always be smaller than or equal to the ASF at $\mathbf{x}^{(k)}$ (the previous anchor), as our goal is to continuously improve $s(\mathbf{x})$ from one anchor to the next in the IP.

Thus, in the presence of the G_1 step-constraint alone, the SC-space is two-dimensional and the current point $\mathbf{x}^C$ or the current anchor point $\mathbf{x}^{(k)}$ lies at the origin $(0, 0)$ in the SC-space $(\widehat{S}_0, \widehat{S}_1)$. All step-constraint based feasible solutions will lie on the second quadrant of the SC-space (smaller $\widehat{S}_0$ and larger $\widehat{S}_1$) and above the $\widehat{S}_1 = 1$ line, shown in Figure 2a. For a feasible point P1, we compute the distance D from P1 to a vector joining the origin (location of the current anchor point) to the intersection of two constraints (desired location of the next anchor point): $s(\mathbf{x}) = 0$ (or $\widehat{S}_0 = -1$) and $G_1(\mathbf{x}, \mathbf{x}^{(k)}) = \Delta_1$ (or $\widehat{S}_1 = 1$). The feasible point having the smallest perpendicular distance D is chosen as the next anchor.

The above directed domination principle can be extended for more than one supplied step-constraints. To illustrate, let us assume the following two con-



(a) For one supplied step-constraint. (b) For two supplied step-constraints.

Fig. 2: Proposed anchor finding mechanism in the SC-space.

straints:

$$G_1(\mathbf{x}, \mathbf{x}^{(k)}) \equiv f_{M+1}(\mathbf{x}) - f_{M+1}(\mathbf{x}^{(k)}) \geq \Delta_1, \tag{8}$$

$$G_2(\mathbf{x}, \mathbf{x}^{(k)}) \equiv \|\mathbf{x} - \mathbf{x}^{(k)}\| \geq \Delta_2. \tag{9}$$

With the default constraint $\widehat{S}_0$, we have a three-dimensional SC space

$$\widehat{S}_0(\mathbf{x}, \mathbf{x}^{(k)}) = -1 + s(\mathbf{x})/s(\mathbf{x}^{(k)}),$$
$$\widehat{S}_1(\mathbf{x}, \mathbf{x}^{(k)}) = (f_{M+1}(\mathbf{x}) - f_{M+1}(\mathbf{x}^{(k)}))/\Delta_1,$$
$$\widehat{S}_2(\mathbf{x}, \mathbf{x}^{(k)}) = \|\mathbf{x} - \mathbf{x}^{(k)}\|/\Delta_2.$$

Note that the current anchor point ($\mathbf{x}^{(k)}$) lies at the origin of the SC-space, shown in Figure 2b. The intersection of three planes, the direction vector and the step-constraint based feasible and improvement space are shown in the figure. It is clear that the intersection point, if exists, will make the shortest perpendicular distance to the line, thereby creating the next anchor point which makes all supplied step-constraints almost active. Interestingly, it can be proven using Karush-Kuhn-Tucker optimality conditions that for both-side bounded step-constraints (for example, $G_1 \geq \Delta_1$ and $G_1 \leq \Delta_2$, the shortest distance from the vector happens for a point making the first constraint ($G_1 \geq \Delta_1$) active. Thus, in such a case, the step-constraint of type $G_1 \leq \Delta_2$ can be ignored for the anchor point identification process.

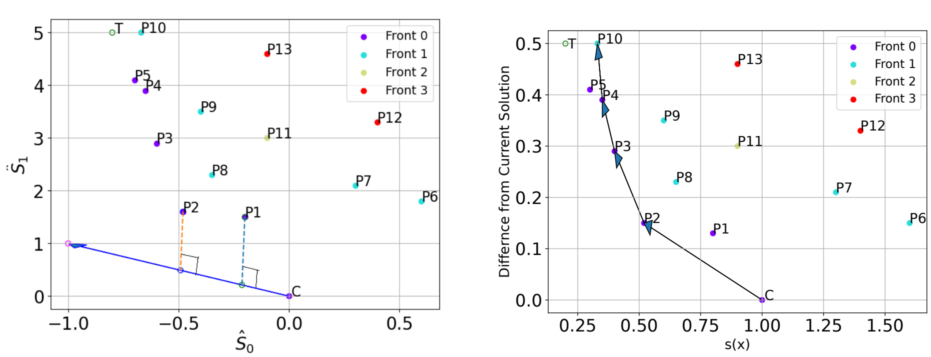
3.1 Finding Anchors and Sorting Population

As shown in Algorithm 1 we start as in the original NSGA-II with combined population $R_t = P_t \cup Q_t$ (parent (P_t) and offspring (Q_t)). Note that infeasible solutions are excluded for anchor point determination. The current point (which is feasible) is always included in the initial population. Since $f_{M+1}(\mathbf{x}^{\mathrm{C}})$ is zero at the current solution, it is always one of the extreme non-dominated solution at every generation of NSGA-II. It is clear that there is always at least one feasible

ND solution ($\mathbf{x}^{\mathrm{C}}$) in any population. Anchor points are solutions that are feasible and are those that satisfy all step-constraints hence, the non-dominated front at the final generation becomes the members of the IP.

After infeasible members are excluded from R_t, the population R_t is sorted in ascending order of the difference function f_{M+1}. The sorted indices are stored in array O. Clearly, the top-most member in the list O is the current solution $\mathbf{x}^{\mathrm{C}}$, having the minimum difference zero from itself. Then, following procedure is repeated in following the sorted list O until all R_t members are considered. If the next member in the list is non-dominated by the already selected anchor its step-constraint violation (SCV) is computed as presented in the algorithm. If SCV is zero, meaning that all step-constraints are satisfied, it is transformed to the SC space and its distance from the ideal direction vector is calculated. The member with minimum distance from the direction vector is selected as the next anchor. If this anchor existed in the previous generation the difference would be calculated as in the algorithm and the count C_{t-1} updated. For our experiments we used $\hat{\delta} = 10^{-4}$. If any of the step-constraints get violated, SCV value will be non-zero and the solution cannot be an anchor point. its normal domination relation with other members of the population is calculated.

Thus, at the end of Algorithm 1, the combined population R_t has defined domination relationships in a matrix where the points not dominated by any other member are the first front and anchors. Similar others fronts for rest of the population can be found. The Algorithm also provides infeasible set H_t. Figure 3a illustrates how the above procedure determines the anchor point (P_2) in a hypothetical population of 14 points. Figure 3b shows the final found IP for the population. As seen in Figure 3b the new approach prefers P2 over P1 reducing both the difference between the first and second anchor and improving the tradeoff on the first anchor. Also it includes P_{10} as the final anchor reaching more closer to the target as compared to the approach used in [10].



(a) First anchor (P2) discovery. Ortho. dist. from P2 to the direction is shortest.

(b) IP (C → P2 → P3 → P4 → P10) determined by the new approach.

Fig. 3: A single step of discovering a new anchor with the proposed approach and the complete IP found using the new approach for the sample population.

Algorithm 1: Identifying Anchor Points

Input: Pop. size N, Pop. R_t with vectors $\mathbf{x}^{(i)}$, $i = 1, \ldots, 2N$, normalized problem constraint violation $\mathrm{CV}(\mathbf{x})$, bi-objective vector $\mathbf{F}(\mathbf{x}) = \{s(\mathbf{x}), f_{M+1}(\mathbf{x})\}$, normalized step-constraints $\widehat{G}_l(\mathbf{x}) \leq 0$, normalized transformations $S_l \equiv G_l(\mathbf{x}, \mathbf{x}^{(k)}) \geq \Delta_l$, anchor set A_{t-1} and stability count of anchors $\mathbf{C}_{t-1}$.

Output: Domination relation matrix dom_mat, C_t, infeasible pop. H_t

```
H_t = {x | CV(x) > 0}; // infeasible members;
R_t = R_t \ H_t;
O = Sort(R_t, 'Ascend') using f_{M+1}-objective;
dom_mat = zeros(|O|, |O|);
next_can, I = 1, golden_step = (-1, ones(|step|));
for i = 1, 2, ... |O| do
    if i = next_can then
        step_check = True;
        minimum = inf, dom_mat(i, :) = 0;
    else
        step_check = False;
    end
    for j = i+1, i+2, ... |O| do
        if x^{O(i)} ≺ x^{O(j)} then
            dom_mat(i, j) = 1;
        else if x^{O(j)} ≺ x^{O(i)} then
            dom_mat(j, i) = 1;
        else if step_check then
            if SCV = \sum_{l=1}^{L} ⟨\widehat{G}_l(x^{O(j)}, x^{O(i)})⟩ > 0 then
                dom_mat(j, i) = 1
            else
                point = point ∪ \widehat{S}_l(x^{O(j)}, x^{O(i)}),   l = 0, 1, 2, ... L;
                distance = perpendicular_dist(point, golden_step);
                if distance < minimum then
                    if minimum ≢ inf then
                        dom_mat(next_can : j, i) = 1;
                    end
                    minimum = dist, next_can = j;
                    if C_{t-1}(I) then
                        if ||F(A_{t-1}(I)) - F(x^{O(j)})||_2 <= \hat{δ} then
                            C_t(I) = C_{t-1}(I) + 1;
                        else
                            C_t(I) = 0;
                        end
                    else
                        C_t(I).append(0);
                    end
                    I += 1; // update current anchor
                end
            end
        end
    end
end
```

3.2 Association, Survival, and Selection Operations

The Association operator is exactly the same as [10], while survival and selection have a slight change in the AnchorSelectProb mechanism used. The anchors from A_t are chosen based a decreasing selection probability $\rho_j = a(1+(\gamma-1)(j-1)/(K-1))$ is assumed, where j ($j = 1, 2, \ldots, K$) is the ordering index of anchors, K is the total number of anchor points, and $\gamma < 1$ is a user-defined parameter. Anchor points are classified into two classes: (i) unstable anchors and (ii) stable anchors, measured by C_t in Lines 33-40 in Algorithm 1. If C_t for an anchor equals or exceeds a threshold ζ ($= 7$ used here), meaning that the anchor does not change its location in the past ζ generations, it is considered stable. Unstable anchors are ordered in increasing f_{M+1} value, so that unstable anchors closer to the current point are assigned more selection probability for stabilizing them to build the foundation for creating more anchor points towards the target point. Stable anchors are ordered in decreasing f_{M+1} value of an anchor, so that a further stable anchor from the current point gets a larger probability value.

4 Results and Discussions

We test the proposed algorithm on various well-known single, multi- and many-objective problems and compare it with the results obtained in [10]. The results obtained in [10] are treated as a baseline and are referred as such in all the plots. Furthermore, for each problem we use the same step-constraints $\widehat{G_l}$ and Δ_l as used in the baseline. The current solution is marked as 'C' in all the plots while the desired target is marked as 'T'. The source code is available at[1].

4.1 Single-objective Goal

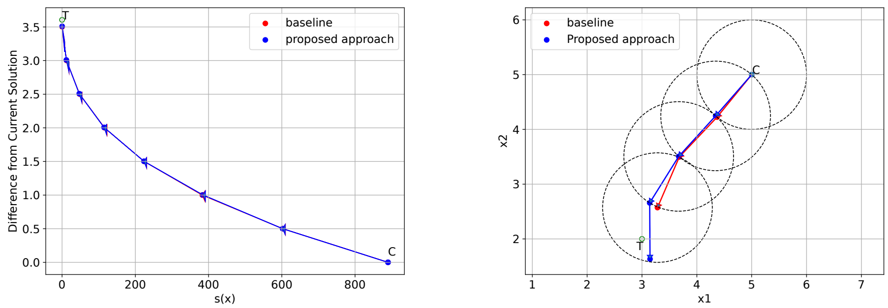
Here we show results on multiple single objective problems and compare with the results achieved in [10]. The different parameter settings for these problems are shown in Table 1. We start with the Himmelblau problem using the G_1 step-constraint, the IP found in the objective space is shown in Figure 4a, while the IP for G_2 step-constraint in the variable space is shown in Figure 4b. As seen the results of the proposed approach are comparable to the baseline for G_1 step-constraint and closer to the target for the G_2 step-constraint. As these results are obtained with half the population size compared to the baseline, it shows the computational efficiency of the proposed approach.

Table 1: Parameter values for problems having a single-objective goal.

	N	$T_{\max}$	γ	Step	Current soln.
Himmelblau	50	100	0.2	(0.5, 1)	[5,5]
Rosenbrock(10)	200	150	0.2	0.3	$w_i = 1.5, \forall i$
G2	100	200	0.25	0.1	[4,6,8,0.8,0.6,1,1,1,1,1]
G4	50	100	0.15	0.2	[100.32, 34.24, 40.47, 42.39, 37.94]

Similarly, Figure 5 shows the IPs found for both the g1 and g2 constrained problem using the G_1 step-constraint. The proposed approach obtains compa-

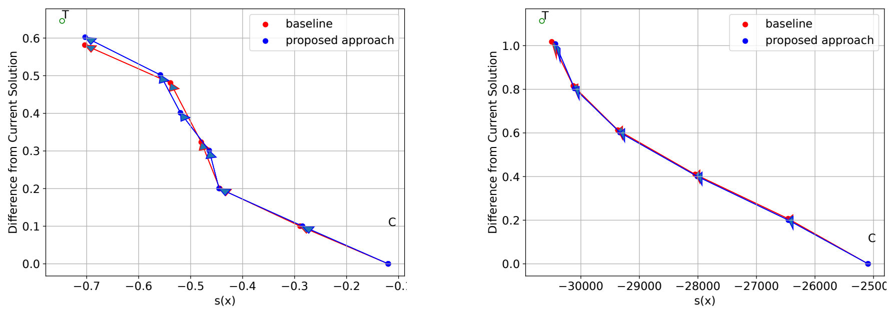
[1] https://github.com/Ahmer-khan/Innovation_path/tree/Directed-Domination



(a) IP with G_1 on objective space.

(b) IP with G_2 on variable space.

Fig. 4: IP with G_1 and G_2 step-constraints for the Himmelblau problem.



(a) Constrained problem g2.

(b) Constrained problem g4.

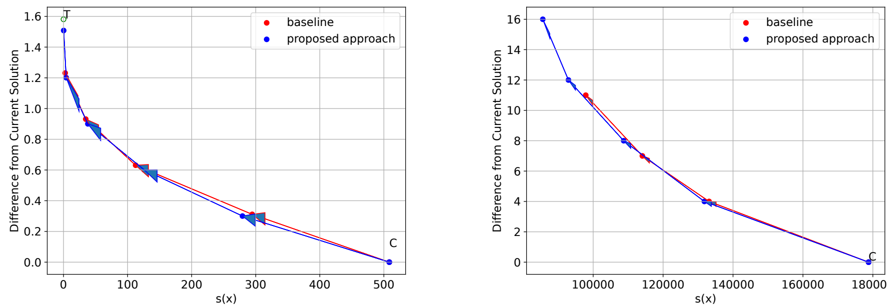
Fig. 5: IPs for the constrained g2 and g4 problem using step-constraint G_1.

rable results to the baseline with a smaller population and fewer generations. Whereas for the 10-var Rosenbrock problem the proposed approach achieve better results, reaching closer to the target as shown in Figure 6a, in a fewer number of generations. Lastly, our approach achieved better IP for the keyboard configuration (KC) problem with the same setting as in [10]. The goal in the KC problem was to reconfigure keys of the QWERTY keyboard in order to achieve fewer finger movements (a new goal) in typing a large volume of text [11]. As seen in Figure 6b, although both the new and previous approaches follow a similar trade-off IP, the new approach minimizes the finger movement objective better than that by the previous algorithm.

4.2 Multi-objective Goals

We test and compare our proposed method with the baseline on various well-known multi-objective test and engineering problems. The experimental setup for each problem is shown in Table 2.

We use the G_1 step-constraint to generate paths for all problems. Figures 7a and 7b shows IP solutions for ZDT4 and ZDT6 problems. The blue point represents the current solution and the green point represents the target. Clearly, the new approach gets closer to the target point T. Figure 8a presents IP solutions



(a) 10-var Rosenbrock problem. (b) Keyboard configuration problem.

Fig. 6: IPs with step-constraint G_1 for Rosenbrock and keyboard problem.

Table 2: Parameter values for problems with multi-objective goals.

	N	$T_{\max}$	γ	Step	$\mathbf{w}$	$\mathbf{z}$	Current soln.
ZDT1	100	100	0.2	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.002,0.001,0,0,0,0]
ZDT2	100	100	0.2	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.002,0.001,0,0,0,0]
ZDT4	100	100	0.2	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.02,0.01,0,0,0,0]
ZDT6	100	200	0.2	0.1	[0.75,0.25]	[0,0]	[0.4,0.5,0.002,0.001,0,0,0,0]
OSY	120	100	0.25	0.1	[0.2,0.8]	[-260,0]	[2,4,4,2]
Weld	100	100	0.3	0.1	[0.8,0.2]	[0,0]	[2,4,4,2]

for constrained real-world welded beam problem. The results are comparable to the baseline, although the new approach gets closer to the target point.

Innovation Path Deviates from Goal Pareto Front: For certain step-constraints, even with a starting solution on the goal PO front, IP solutions may not lie on the goal PO front. This phenomenon is illustrated in Figure 8b using a combination of G_1 step-constraint (Equation 2) and the G_3 step-constraint on ASF values shown in Equation 11.

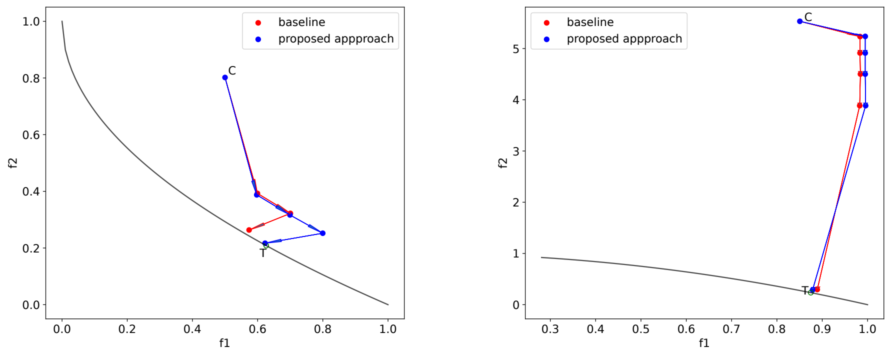
$$G_3(\mathbf{x}, A^{(k)}) \equiv s(A^{(k)}) - s(\mathbf{x}) \leq \Delta_3, \tag{10}$$

$$\widehat{G}_3(\mathbf{x}, A^{(k)}) = (s(A^{(k)}) - s(\mathbf{x}))/\Delta_3 - 1 \leq 0. \tag{11}$$

With G_1 alone, IP lies on the goal PF, as shown by the purple path in the Figure 8b. However, for certain combinations of Δ_1 and Δ_3, all PO solutions of the goal problem become infeasible. Since the baseline only considers anchors in the Non-dominated front it is unable to find a path for this particular problem, whereas the proposed approach can find a valid optimal path diverging away from the PF shown in blue in Figure 8b.

4.3 Many-objective Goals

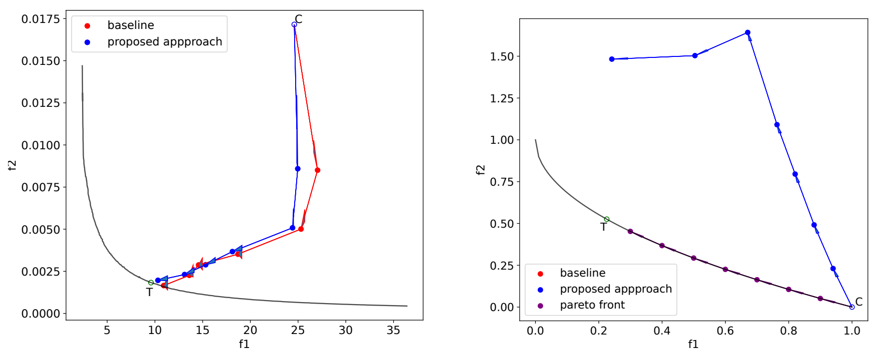
Here we test and compare the proposed approach on three- and five-objective DTLZ2 problems plus the three-objective C2DTLZ2 constrained problem. Though



(a) ZDT4 objective space.

(b) ZDT6 objective space.

Fig. 7: IPs for ZDT4 and ZDT6 problems with G_1 step-constraint originating from a non-optimal solution.



(a) Welded beam with G_1.

(b) ZDT1 with G_1 and G_3.

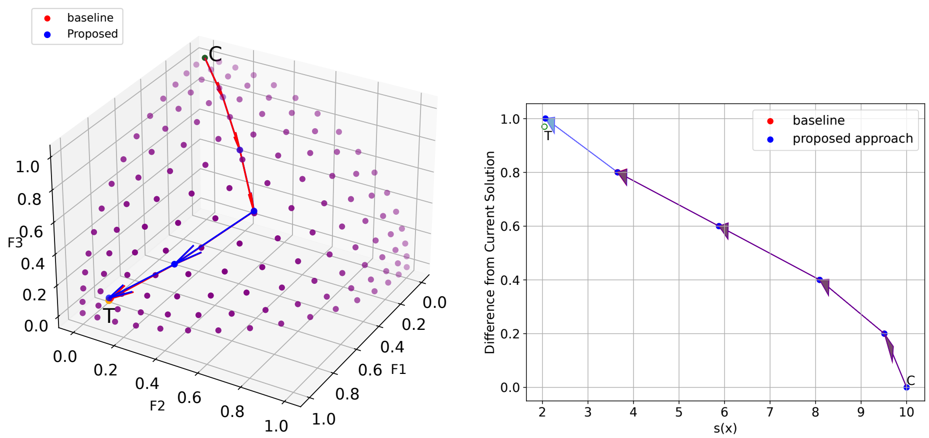
Fig. 8: IPs with step-constraint(s) for welded beam and PF-deviating problem.

originally the baseline approach has not been tested on many objective problems, we additionally ran the baseline on such problems with the same parameter setting as the proposed approach shown in Table 3. Figure 9a shows the IP found by

Table 3: Parameter values for problems with many-objective goals.

	N	$T_{\max}$	γ	Step	$\mathbf{w}$	$\mathbf{z}$	Current solution
DTLZ2-3obj	100	100	0.15	0.2	[0.8,0.1,0.1]	$z_i = 0, \forall i$	$x_1 = 1$ and $x_i = 0.5$ for $i = 2, 3, 4 \ldots 12$
DTLZ2-5obj	100	200	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 $i = 2, 3, 4 \ldots 12$
C2DTLZ2-3obj	200	200	0.15	0.2	[0.1,0.8,0.1]	$z_i = 0, \forall i$	$x_1 = 1$ and $x_i = 0.5$ for $i = 2, 3, 4 \ldots 12$

both approaches for the three-objective DTLZ2 problem. Figure 9b shows the IP found on the five-objective DTLZ2 problem in bi-objective problem space, while Figure 10 shows the IP found for three-objective C2DTLZ2 constrained problem in the original objective space. Where the proposed approach is comparable to the baseline for three-obj DTLZ2 it does better in the other two problems by reaching closer to the target.



(a) Three-objective version. (b) Five-objective version.

Fig. 9: IP with G_1 step-constraint for DTLZ2 with three and five objectives.

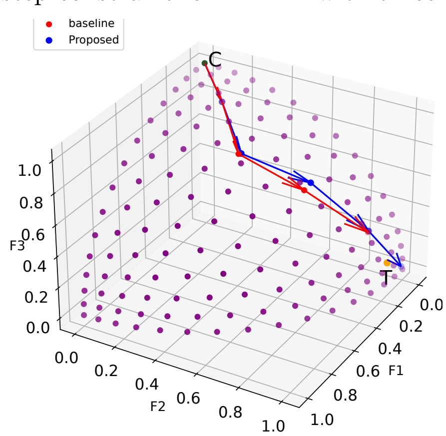


Fig. 10: IP with G_1 step-constraint for C2DTLZ2 problem with three objectives.

5 Conclusions

In this paper, we have proposed a new anchor identification approach based on a concept of directed domination which makes the IP-seeking bi-objective algorithm more computationally efficient compared to the baseline approach proposed earlier. The new approach has been shown to reach closer to the target or achieve comparable results with half the population size and/or using a less number of generations. We have shown these findings on a number of single, multi- and many-objective scenarios. We have also tested the proposed algorithm on new scenarios, where the IP path diverges away for the original PO front due to the peculiarities of the chosen step-constraints. These problems were not possible to be solved by the previous approach. The description of the new directed domination approach and accompanying results show the efficiency and accuracy of the proposed approach.

The IP-seeking procedure requires intermediate solutions to be found one after the other in a serial manner. The efficacy of a population-based EMO approach vis-a-via a point-based optimization approach is worth investigating further in the future. Nevertheless, the algorithmic developments for finding IP solutions is a challenging and useful practical task, which must be pursued and applied further.

References

1. Abdulmouti, H.: The role of kaizen (continuous improvement) in improving companies' performance: A case study. In: 2015 International Conference on Industrial Engineering and Operations Management (IEOM). pp. 1–6. IEEE (2015)
2. Al-Rifai, M.H.: Redesigning and optimizing an electronic device assembly cell through lean manufacturing tools and kaizen philosophy: an application case study. International Journal of Productivity and Performance Management **73**(4), 1273–1301 (2024)
3. Branke, J., Deb, K., Miettinen, K., Slowinski, R.: Multiobjective optimization: Interactive and evolutionary approaches. Springer-Verlag, Berlin, Germany (2008)
4. Carnerud, D., Jaca, C., Bäckström, I.: Kaizen and continuous improvement–trends and patterns over 30 years. The TQM Journal **30**(4), 371–390 (2018)
5. Coello, C.A.C.: Treating objectives as constraints for single objective optimization. Engineering Optimization **32**(3), 275–308 (2000)
6. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester, UK (2001)
7. Deb, K., Datta, R.: A bi-objective constrained optimization algorithm using a hybrid evolutionary and penalty function approach. Engineering Optimization **45**(5), 503–527 (2012)
8. Ferreira, D.M.C., Saurin, T.A.: A complexity theory perspective of kaizen: a study in healthcare. Production Planning & Control **30**(16), 1337–1353 (2019)
9. Gåsvaer, D., von Axelson, J.: Kaikaku - Radical improvement in production. In: International Conference on Operations and Maintenance, Singapore. pp. 37–47 (2012)
10. Khan, A., Deb, K.: 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. p. 529–537 (2024)
11. Khan, A., Deb, K.: Optimizing keyboard configuration using single and multi-objective evolutionary algorithms. In: Proceedings of the Companion Conference on Genetic and Evolutionary Computation. pp. 219–222 (2023)
12. Knowles, J.D., Corne, D.W., Deb, K.: Multiobjective problem solving from nature. Springer Natural Computing Series, Springer-Verlag (2008)
13. Maurer, R., Hirschman, L.A.: Spirit of Kaizen: Creating Lasting Excellence One Small Step at a Time. McGraw-Hill Professional (2012)
14. Mezura-Montes, E., Coello, C.A.C.: Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm and Evolutionary Computation **1**(4), 173–194 (2011)
15. Moi, W.A., Sing, S.H.: Application of toyota way incorporating kaizen, kaikaku and 5s in agricultural sector. International Journal for Research in Applied Science and Engineering Technology **9**(10), 1565–1579 (2021)
16. Qu, Y., Ming, X., Qiu, S., Liu, Z., Zhang, X., Hou, Z.: Process optimization through closed-loop kaizen with discrete event simulation: A case study in china. Proceed-

ings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture **235**(3), 568–579 (2021)
17. Reklaitis, G.V., Ravindran, A., Ragsdell, K.M.: Engineering Optimization Methods and Applications. New York : Wiley (1983)
18. Tozawa, B., Bodek, N.: How to Do Kaizen: A New Path to Innovation: Empowering Everyone to be a Problem Solver. PCS Press (2009)
19. Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making Theory and Applications, pp. 468–486. Berlin: Springer-Verlag (1980)