COIN Report Number 2024002

Innovation Path: Discovering an Ordered Set of Optimized Intermediate Solutions from an Existing to a Desired Solution

Ahmer Khan
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan, USA
khanahm2@msu.edu

Kalyanmoy Deb
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, Michigan, USA
kdeb@msu.edu

Abstract

In practice, there is often a need to modify an existing implemented solution in order to achieve a better performance or accommodate new demands or technologies. However, a new optimal solution for the updated problem may be quite different from the existing solution, thereby causing an apathy for its implementation involving large cost, changes, and efforts. For such scenarios, we propose a concept of an "innovation path" (IP) containing a sequence of intermediate solutions from the existing to the new target solution with gradual and controlled change from one to the next. To discover intermediate solutions of the IP simultaneously, we propose a bi-objective formulation of the original problem, so that Pareto-optimal solutions of the resulting bi-objective problem become the IP solutions. We demonstrate the working of the proposed approach on a number of single and two-objective test and engineering problems. Results are encouraging and suggest further research and application to make the proposed innovation path approach more efficient and practical.

CCS Concepts

• **Theory of computation** → **Genetic programming**; *Theory of randomized search heuristics*; • **Hardware** → *Hardware description languages and compilation*.

Keywords

Innovation path, Pareto-optimal solutions, Multi-objective optimization, Keyboard layout, combinatorial optimization

ACM Reference Format:

Ahmer Khan and Kalyanmoy Deb. 2024. Innovation Path: Discovering an Ordered Set of Optimized Intermediate Solutions from an Existing to a Desired Solution. In *Genetic and Evolutionary Computation Conference (GECCO '24), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3638529.3654051

1 Introduction

In practice, existing solutions are often questioned for their lack of performance or their viability to meet current need and match new technologies. Despite the need, in practice, users often do not engage in such a complete overhaul of the existing solution. This is, not because suitable computational methods using efficient search and optimization techniques do not exist, but because the users fear that a new optimal solution will be so different from the existing solution that the updates may simply incur too much cost or other related changes that may not be pragmatic to implement. To alleviate this practical implementational issue, we propose a computational approach to find a finite sequence of gradual updates from the existing current solution $\mathbf{x}^{\mathrm{C}}$ to the final target solution $\mathbf{x}^{\mathrm{T}}$, instead of a single direct update from $\mathbf{x}^{\mathrm{C}}$ to $\mathbf{x}^{\mathrm{T}}$. This way, every intermediate update will indicate relatively small but acceptable changes to the previous solution, thereby achieving the desired best solution in several updates. Of course, the user will have a way to specify the allowable changes either in the solution or its cost/effort implications and will cause the complete IP solution set together a long-term viable optimal strategy, rather than every one being an independent optimal solution without any implication to each other. The sequence (or chain) of such interacting optimal solutions from the current to the target (or a near-target) solution is referred here as *innovation path* (IP) solutions.

This paper proposes a multi-objective optimization procedure to find the sequence of intermediate IP solutions, starting from the current solution and ending up close to a target solution. Since multiple solutions are the goal here, the original single or multi-objective problem can be altered to constitute a bi-objective problem of minimizing a scalarized objective and deviation from the current solution. This task can be classified as one of the 'multiobjectivization' studies proposed by EMO researchers in the past [2].

In the remainder of this paper, we present the concept of innovation path in Section 2 through simple example problem. The proposed modifications to an existing EMO algorithm to find IP solutions are described in Section 3. Discussion and results on single and multi-objective problems are then presented in Sections 4 to 5, including a keyboard layout and an engineering design problem. Finally, conclusions and a list of immediate future studies are mentioned in Section 6.

2 Innovation Path (IP)

An innovation path starts from the current solution (C) and must end up on or close to a target solution (T), dictated by the user. Moreover, the path is expected to contain a finite set of intermediate points, satisfying a limit on the difference from previous solution. The above discussion points to standard outcome from an evolutionary multi-objective optimization (EMO) algorithm of producing a set of Pareto-optimal (PO) solutions. However, to achieve this using EMO, we would need to have two or more conflicting objectives to search. For this purpose, we formulate an extended multi-objective optimization problem in which every IP solution including the current and a target solutions become a part of the resulting Pareto-efficient solution set for the extended problem. We

GECCO '24, July 14–18, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0494-9/24/07.
https://doi.org/10.1145/3638529.3654051

describe a specific approach of the extended optimization problem here, although other approaches are possible.

For a given M-objective optimization problem, we introduce a new objective function $f_{M+1}(\mathbf{x})$ as the difference of the variable vector ($\mathbf{x}$) from the prescribed current feasible solution ($\mathbf{x}^{\mathrm{C}}$) to be minimized:

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

By including this objective, we create an $(M+1)$-dimensional extended problem, for which the unique Pareto-optimal solution(s) of the original M-objective problem is guaranteed to be a part of the Pareto-optimal front of the extended problem.

To illustrate, we consider a single-objective ($M = 1$), two-variable ($n = 2$) sphere function $f(\mathbf{x})$ shown in Figure 1a. The extended two-objective space and resulting Pareto-optimal front are shown in Figure 1b with blue and red colored points, respectively. The sphere function can be thought of a hypothetical *cost* function of a two-variable design problem for which the minimum cost lies at $\mathbf{x}^* = (0, 0)$ with $f^* = 0$. Let us also assume that the currently implemented solution is at $\mathbf{x}^{\mathrm{C}} = (4, 3)$ having a cost value of $f = 25$, which is marked with a filled square. Note that the current solution is not particularly a good solution for the sphere function and can be improved to reduce the f-value. A jump of f from its current solution (4,3) to (0,0) (optimal f) may be too large to make in a single practical update. This calls for a few intermediate updates, as shown in Figure 1b with diamonds. Certainly other paths from the current (square) to the target (star) solutions are possible, but here we argue that a path following the minimization of the new objective (Equation 1) and the original objective f_1 would make a set of intermediate solutions having an optimal trade-off between reduction in f_1 and moving away from the current solution. We refer the optimal solutions of the two-objective problem as the *extended PF* or E-PF, in which one of the extreme points corresponds to $f(\mathbf{x}^{\mathrm{C}})$ and other corresponds to $f(\mathbf{x}^*)$ or $f(\mathbf{x}^{\mathrm{T}})$. The sequence of optimal improvements from $f(\mathbf{x}^{\mathrm{C}})$ to $f(\mathbf{x}^{\mathrm{T}})$ is called an innovation path here and the intermediate solutions are called anchor points.

If an EMO is used to optimize the extended two-objective problem, we are expected to obtain a representative theoretical PO front, as shown in Figure 1b. But, our goal is to find a sequence of a few (a small finite set) intermediate solutions starting from $\mathbf{x}^{\mathrm{C}}$ and having L normalized step-constraints ($\widehat{G}_l$, $l = 1, 2, \ldots, L$) to be satisfied between every pair of anchor solutions ($\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$) on the innovation path, starting with $\mathbf{x}^{(1)} = \mathbf{x}^{\mathrm{C}}$:

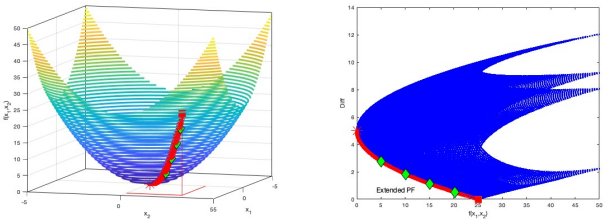
$$\widehat{G}_l(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}) \leq 0, \quad l = 1, 2, \ldots, L. \tag{2}$$

Since the user may not be interested in too small a change between two intermediate anchor solutions, thereby leading to too many implementations to reach the target, the following may be one of the step-constraints:

$$\widehat{G}_1 \equiv 1 - (f_{M+1}(\mathbf{x}^{(j)}) - f_{M+1}(\mathbf{x}^{(i)}))/\Delta_1 \leq 0, \tag{3}$$

where Δ_1 is a minimum difference in f_{M+1} objective value specified for any two consecutive anchor solutions of the innovation path. For $\Delta_1 = 5$ units, Figure 1b marks four anchor solutions with green diamonds, which, along with the current and final points, constitute the resulting innovation path for the sphere example problem. The respective points are also shown in Figure 1a.

Following two theorems help to develop our proposed EMO method.



(a) Innovation path on sphere fn. **(b) Optimization on two-obj space**

Figure 1: Illustration of innovation path on sphere function.

THEOREM 1 (CURRENT SOLUTION AND EXTENDED PE SET:). *The current solution $\mathbf{x}^{\mathrm{C}}$ is always an extreme solution of the extended Pareto-Efficient set.*

PROOF. According to Equation 1, the current solution $\mathbf{x}^{\mathrm{C}}$ is the unique minimum solution of the $(M+1)$-st objective function with $f_{M+1}(\mathbf{x}^{\mathrm{C}}) = 0$. Thus, $\mathbf{x}^{\mathrm{C}}$ is always a member of the extended PE set. □

THEOREM 2 (ORIGINAL AND EXTENDED PO FRONTS:). *The original Pareto-optimal front with non-duplicate objective vectors is always a part of the extended Pareto-optimal front.*

PROOF. We prove this theorem by contradiction. Let us assume that there exists a feasible solution $\mathbf{y}$ (with $f_i(\mathbf{y}) \neq f_i(\mathbf{x}^P)$, $\forall i = 1, \ldots, M$) which dominates a PO solution $\mathbf{x}^P$ (of the original M-dimensional problem) in the extended problem. This means that for every objective ($i = 1, 2, \ldots, M+1$), $f_i(\mathbf{y}) \leq f_i(\mathbf{x}^P)$ and for at least one objective (k) $f_k(\mathbf{y}) < f_k(\mathbf{x}^P)$. Since $\mathbf{x}^P$ is Pareto-optimal to the M-dimensional problem, by definition, there must exist at least one objective ($j = 1, 2, \ldots, M$) for which $f_j(\mathbf{y}) > f_j(\mathbf{x}^P)$. Both of these conditions cannot be satisfied simultaneously. Hence, the theorem is proved. □

3 Proposed EMO Approach to Find IP Solutions

It is clear that the proposed innovation path corresponds to a directed set of solutions starting from the current solution $\mathbf{x}^{\mathrm{C}}$ and terminating close to the target solution $\mathbf{x}^{\mathrm{T}}$. Since it is a directed set of solutions, they can be conceived to lie on a one-dimensional manifold. Since each of the anchor solutions on the innovation path must correspond to certain optimal trade-off for minimal change in solution and progress towards the target solution, the solutions can be construed to lie on a one-dimensional Pareto-optimal front, irrespective of the number of objectives involved in the original problem. We discuss handling multi-objective problems later, but build the IP concept for a single-objective optimization problem first.

We employ the NSGA-II procedure [7] as our baseline EMO algorithm to find solutions on the innovation path, although other two-objective EMO algorithms can also be used in a similar way. Note that the exact number of intermediate solutions from the current to a near-target solution need not be known beforehand. However, the user would have the option to provide one or more *step-constraints*,

Algorithm 1: Finding Anchor Points

Input: Population size N, Population R_t with vectors $\mathbf{x}^{(i)}$, $i = 1, \ldots, 2N$, of which one is the current feasible solution $\mathbf{x}^{\mathrm{C}}$, normalized problem constraint violation function $\mathrm{CV}(\mathbf{x})$, scalarized objective $s(\mathbf{x})$, difference objective $f_{M+1}(\mathbf{x})$, normalized step-constraints $\widehat{G}_l(\mathbf{x}) \leq 0$, $l = 1, 2, \ldots, L$

Output: Next generation anchors A_t, rest of feasible population B_t, infeasible population H_t and there non-dominated rank

ND = ND-Sort(R_t,'Constraint_Domination') in the two-objective (s, f_{M+1}) space;
$H_t = \{\mathbf{x} | CV(\mathbf{x}) > 0\}$; // infeasible members;
$R_t = R_t \backslash H_t$;
S = Sort(R_t,'Ascend') using f_{M+1}-objective;
$A_t = \mathbf{x}^{S(1)}$ // Current solution;
$I = 1$, $B_t = \emptyset$;
for *each* $i = 2, 3, \ldots |R_t|$ **do**
  SCV = $\sum_{l=1}^{L} \langle \widehat{G}_l(\mathbf{x}^{S(i)}, \mathbf{x}^{S(I)}) \rangle$;
  if $\mathbf{x}^{S(i)}$.ND = 1 AND $SCV = 0$ **then**
    $A_t = A_t \cup \mathbf{x}^{S(i)}$;
    $I = I + 1$; // anchor solution counter
  else
    $B_t = B_t \cup \mathbf{x}^{S(i)}$;
    $\mathbf{x}^{S(i)}$.ND = $\mathbf{x}^{S(i)}$.ND + 1;
  end
end

that will ensure a minimum allowable difference between two consecutive anchor solutions in the final innovation path.

3.1 Finding Anchor Points in a Population

Algorithms 1 to 4 provide pseudo-codes of the discovery of anchor points, association and revised crowding distance computations, along with survival and mating selection operators.

Irrespective of number of objectives in the original problem, two objectives are considered in NSGA-II: (i) scalarizing function $s(\mathbf{x})$ discussed in Section 5 for multi-objective innovation path problems and (ii) the difference function $f_{M+1}(\mathbf{x})$ defined in Equation 1. For single-objective innovation path problems, $s(\mathbf{x}) = f(\mathbf{x})$. In the presence of constraints, every population member's constraint violation CV is computed as follows:

$$\mathrm{CV}(\mathbf{x}) = \sum_{j=1}^{J} \langle \widehat{g}_j(\mathbf{x}) \rangle, \tag{4}$$

where $\langle \cdot \rangle$ is the bracket operator, which returns the operand, if it is positive, else it returns zero. The constraint $\widehat{g}_j$ is normalized value of g_j [5]. Like in the original NSGA-II, the combined population $R_t = P_t \cup Q_t$ (parent (P_t) and offspring (Q_t)) is sorted according to constraint non-domination levels of two objectives: $f_{M+1}(\mathbf{x})$ and $s(\mathbf{x}) = f(\mathbf{x})$. Note infeasible solutions are excluded for anchor point determination and they cannot be anchor points.

The current point (which is feasible) is always included in the initial population. Since $f_{M+1}(\mathbf{x}^{\mathrm{C}})$ (defined in Equation 1) is zero at the current solution, it is always one of the extreme non-dominated solution at every generation of NSGA-II. First, anchor points are identified in a population using the pseudo-code presented in Algorithm 1. It is clear that there is at least one feasible ND solution ($\mathbf{x}^{\mathrm{C}}$) in any population. Anchor points are feasible and are those that lie on the first ND front and satisfy all step-constraints. Clearly, the anchor points at the final generation become the members of the innovation path.

After R_t is sorted according to different levels of non-domination and infeasible members are excluded from R_t, the population R_t is also sorted in ascending order of the difference function f_{M+1}. The sorted indices are stored in array S. Clearly, the top-most member in the list S is the current solution $\mathbf{x}^{\mathrm{C}}$, having the minimum difference zero from itself. The solution $\mathbf{x}^{\mathrm{C}}$ is included in anchor archive A_t and stays as the first anchor member of the innovation path.

Then, following procedure is repeated by going down the sorted list S until all R_t members are considered. If the next member in the sorted list belongs to the first ND front, its step-constraint violation (SCV) is computed as presented in the algorithm. If SCV is zero, meaning that all step-constraints have negative values and are all satisfied, it is the next anchor point and is included in archive A_t.

If any of the step-constraint get violated, SCV value will be non-zero and the solution cannot be an anchor point. After all the anchors are determined, the remaining population members are added to B_t with an increase in their ND rank.

Thus, at the end of Algorithm 1, the combined population R_t is classified into an anchor set (A_t) with feasible solutions, non-anchor feasible set (B_t), and the infeasible set H_t. Figure 2 illustrates how the above procedure determines the anchor points ($A_t = \{1, 2, 4, 6\}$) in a hypothetical population of 14 points. Horizontal dashed lines mark the step-constraint (Equation 3) for each obtained anchor point. Notice that some of the original non-dominated points (3 and 5) are not anchor points, due to the violation of step-constraints applied to already-found anchors. Note also that a path is created from the current solution with three other anchor points in the illustrative example. At the end of the optimization process, this path will correspond to be the innovation path for the problem.

3.2 Association of a Population Member with an Anchor Point

Next, each member of B_t and H_t is associated with a single anchor point based on their *closeness* in terms of step-constraint violation. Algorithm 2 computes an *association metric* derived as the difference in step-constraint value at a point $\mathbf{x}^{(i)}$ with that at the anchor point $A_t^{(k)}$:

$$\Lambda^{(k,i)} = \sum_{l=1}^{L} \left(\widehat{G}_l \left(A^{(k)}, A^{(k)} \right) - \widehat{G}_l \left(\mathbf{x}^{(i)}, A^{(k)} \right) \right). \tag{5}$$

For the single step-constraint presented in Equation 3, the association metric is given below:

$$\Lambda^{(k,i)} = f_{M+1}(\mathbf{x}^{(i)}) - f_{M+1}(A^{(k)}). \tag{6}$$

For every solution $\mathbf{x}^{(i)}$, the minimum of positive $\Lambda^{(k,i)}$ values derived from each anchor point (k) is saved as the crowding distance

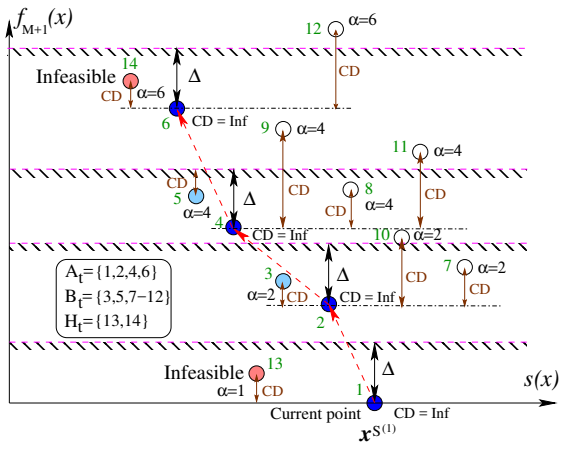


Figure 2: Identification of anchor points in a hypothetical R_t population having 14 members. Here, step-constraint $G_1 \equiv \Delta - \left(f_{M+1}(\mathbf{x}^{S^{(i)}}) - f_{M+1}(\mathbf{x}^{S^{(I)}})\right) \leq 0$ is used for easier illustration. CD and association α of anchor points A_t, feasible (B_t) and infeasible (H_t) non-anchor points are shown.

(CD) of $\mathbf{x}^{(i)}$. The solution $\mathbf{x}^{(i)}$ is then associated with the anchor point causing the minimal $\Lambda^{(k,i)}$ value.

Algorithm 2: Association and Crowding Distance of Population Members of B_t and H_t

Input: Subpopulations A_t, B_t, and H_t association metric $G^{(k,i)}$ (Equation 5)
Output: Associated anchor point $\mathbf{x}.\alpha$ and crowding distance $\mathbf{x}$.CD of B_t and H_t members

1 $\mathbf{x}^{(j)}.\alpha = 0$ and $\mathbf{x}^{(j)}.\text{CD} = 0, \forall \mathbf{x}^{(j)} \in A_t$;
2 **for** *each* $i = 1 : |B_t \cup H_t|$ **do**
3 $[\text{CD}, IDj] = \min\{G^{(k,i)} | G^{(k,i)} > 0, k = 1 : |A_t|\}$;
4 $\mathbf{x}^{(i)}.\alpha = IDj$; $\mathbf{x}^{(i)}.\text{CD} = \text{CD}$;
5 **end**

In case of a different step-constraint directly involving variable space separation between two anchor points:

$$G_2(\mathbf{x}) = \Delta_2 - \|A^{(k)} - \mathbf{x}\|_2 \leq 0, \tag{7}$$

the association metric is

$$\Lambda^{(k,i)} = \|A^{(k)} - \mathbf{x}^{(i)}\|_2. \tag{8}$$

What we observe is that Δ_2 does not appear in $G^{k,i}$ metric. However, for multiple step-constraints, the threshold values become important due to normalization procedure. For example, if both G_1 and G_2 step-constraints are used, their normalization will result in

$$\widehat{G}_1(\mathbf{x}, A^{(k)}) \equiv 1 - \left(f_{M+1}(\mathbf{x}) + f_{M+1}(A^{(k)})\right)/\Delta_1 \leq 0,$$
$$\widehat{G}_2(\mathbf{x}, A^{(k)}) \equiv 1 - \|A^{(k)} - \mathbf{x}\|_2/\Delta_2 \leq 0,$$

Between two non-dominated population members belonging to the same anchor point, the one with a higher CD value is considered better. For example, for the anchor point 2, points 7, and 10 are associated and also non-dominated to each other. Since point 10 has a larger CD value, it will be considered better than over point 7, if they are compared in one of the subsequent selection operators.

3.3 Survival Selection Operator

Next, Algorithm 3 presents a procedure of choosing the next generation of survived population P_{t+1} of size N from the combined population R_t of size $2N$. The survival selection operator is similar to original NSGA-II's survival operator. Starting from copying the first ND rank solutions of R_t, entire subsequent ranked solutions are copied in P_{t+1} one by one until the size of P_{t+1} exceeds the population size (N). Then, the last front (F_l) which could not be accepted entirely is processed to pick the remaining $(N - |P_{t+1}|)$ solutions from F_l. Note that A_t members belong to the first ND rank and hence have the highest priority to be selected. If there are more anchor points than N (satisfying $|A_t| > N$), only N anchor points are chosen sequentially starting from the current solution.

A linear probability distribution $w(j) = a(\gamma + (1-\gamma)(j-1)/(K-1))$ for the j-th anchor point is assumed (Line 26), with one user-defined parameter γ indicating the rate of increase of preference of anchor points away from the current solution. The parameter K is the number of *active* anchor points, having at least one associated member remained to be selected. Every time an anchor's associated member is copied to P_{t+1}, its count is reduced by one by turning the member's flag to one. The anchor points are renumbered from current point ($j = 1$) to the furthest anchor point ($j = K$). Note that for $K = 1$, the γ is always reset to one to assign a probability of one to the sole active anchor point. The above probability distribution assigns the current solution a probability of $w(1) = \gamma a$ (where $\gamma \in [0, 1]$ is a parameter) and the furthest anchor point a probability of $w(K) = a$. For a given γ, the parameter a is computed in a way to make $\sum_{j=1}^{K} w(j) = 1$. For a chosen anchor, the associated member with the largest CD is saved in P_{t+1}. The process is continued until N members are chosen in P_{t+1}.

Note that infeasible solutions are treated similarly as the feasible dominated solutions, except that their ND rank is worse than the feasible solutions. Since the selection of ND solutions is performed using their ND rank values, infeasible solutions will have a chance to enter into P_{t+1} only if the total number of feasible solutions in $A_t \cup B_t$ is less than N in R_t. Each infeasible solution is likely to be ranked individually in a separate rank, based on their CV values. In this case, $K = 1$ and the sole infeasible solution of F_l will be picked with probability one.

3.4 Mating Selection Operator

Finally, Algorithm 4 presents the mating selection operator in which one of the parent is always an anchor point j chosen using the probability distribution $w(j)$. The second parent is selected from two randomly picked associated members of the j-th anchor point and choosing the one with better constraint-domination principle.

At the "lexmin" operator in Step 11 of the algorithm, if both s1 and s2 are infeasible, the one with a smaller CV is chosen. In case of a tie in CV values (including zero), the one with a smaller ND rank is chosen. In case of a tie in ND values, the one with the larger CD is chosen. As a result, if one of s1 and s2 is feasible and the other is not, the feasible solution is always chosen. When both are feasible with a tie of CV value being zero, the one with a smaller ND rank is

Algorithm 3: Survival Selection Operator

Input: Anchor A_t, Subpopulations B_t, H_t, and selection prob parameter γ
Output: Next gen. population P_{t+1} of size N

$P_{t+1} = A_t$; $i = 1$; $sub = B_t$;
repeat
  $i = i + 1$;
  $P_{t+1} = P_{t+1} \cup \{\mathbf{x} | \mathbf{x}.\text{ND} = i, \mathbf{x} \in B_t\}$;
until $|P_{t+1}| \geq N$ *or* $|P_{t+1}| = |B_t| + |A_t|$;
if $|P_{t+1}| < N$ **then**
  repeat
    from steps 4 to 6 with $\mathbf{x} \in H_t$;
  until $|P_{t+1}| \geq N$;
  $sub = H_t$;
end
if $|P_{t+1}| > N$ **then**
  $F_l = \{\mathbf{x} | \mathbf{x}.\text{ND} = i, \mathbf{x} \in sub\}$;
else
  $F_l = \emptyset$;
end
$P_{t+1} = P_{t+1} \backslash F_l$;
$\mathbf{x}.\text{flag} = 0, \forall \mathbf{x} \in F_l$;
$X = \text{List}(1, 2, \ldots, |A_t|)$;
for *each* $k = 1 : |X|$ **do**
  $n_k = \sum_{i=1}^{|F_l|} \{1 | \mathbf{x}^{(i)}.\alpha = k, \mathbf{x}^{(i)}.\text{flag} = 0\}$;
  if $n_k = 0$ **then**
    Drop k from X and reorder sequence in X
  end
end
$w = \text{AnchorSelectProb}(\gamma, X)$;
while $|P_{t+1}| < N$ **do**
  $r = \text{random}(0,1)$;
  $j = \text{argmin}\{\sum_{i=1}^{j} w(j) \geq r | j = 1, \ldots, |w|\}$;
  $p = \text{argmax}\{\mathbf{x}.\text{CD} | \mathbf{x}.\alpha = X(j), \mathbf{x}.\text{flag} = 0, \forall \mathbf{x} \in F_l\}$;
  $P_{t+1} = P_{t+1} \cup \mathbf{x}^{(p)}$;
  $\mathbf{x}^{(p)}.\text{flag} = 1$;
end

Algorithm 4: Mating Selection Operator

Input: Current anchor A_{t-1}, and population P_t and γ
Output: Two selected parents p1 and p2 for mating

Select j using line 29 in Algorithm 3;
p1 = $A_{t-1}(j)$; // first parent;
s1 = random$\{\mathbf{x} | \mathbf{x}.\alpha = j, \mathbf{x} \in P_t \backslash A_{t-1}\}$;
if *s1*$= \emptyset$ **then**
  Drop $A_{t-1}(j)$ and recalculate j;
else
  s2 = random$\{\mathbf{x} | \mathbf{x}.\alpha = j, \mathbf{x} \in P_t \backslash A_{t-1}\}$;
  if *s2*$= \emptyset$ **then**
    p2 = s1
  else
    p2 = lexmin$\{(\mathbf{x}.\text{CV}, \mathbf{x}.\text{ND}, -\mathbf{x}.\text{CD}) | \mathbf{x} \in (s1, s2)\}$;
  end
end

chosen and in the case of a tie in ND values, the one with a larger CD is chosen.

Selected members p1 and p2 are recombined to create two new offspring solutions, each of which is then mutated and saved in the offspring population Q_t. When the size of Q_t is N, it is combined with the parent population P_t to create R_t, and new anchor points are found using Algorithm 1 and the whole process is repeated again. This procedure is continued until the termination condition is met.

The proposed update of NSGA-II is significant in many ways. First, anchor points from the first ND ranked population members are identified by using supplied step-constraints. Second, newly found anchor points are given more emphasis in both survival and mating selection operators so that the search can be more effective in finding increasingly more anchor points away from the supplied current solution. Third, non-anchor population members are associated with an anchor point using the violation of step-constraints and a modified crowding distance is assigned based on the extent of step-constraint violation. Finally, the recombination operator confined between an anchor point and one of its associated non-anchor point from the population. This ensures that meaningful offspring solutions are created near the step-constraint boundary of an anchor point.

Notice that if at any generation, the current feasible solution $\mathbf{x}^{\text{C}}$ is the only anchor point, all other feasible population members are selected based on step-constraint value. The violated solutions are given more preference closer to the step-constraint boundary and then non-violated ones are given preference based on the extent of their violation.

3.5 Similar Past Studies

In the recent past, EMO researchers have addressed problem-solving tasks that are not multi-objective in nature by converting the task into a multi-objective problem, named multiobjectiziation tasks [10]. For example, the constrained single-objective optimization problem-solving is not a multi-objective optimization task, but when a constraint violation (CV) function is added as a second objective and the resulting problem is solved using a bi-objective optimization algorithm to focus near the smallest CV value, it causes an efficient approach [3, 4]. Reduction of bloating in genetic programming is achieved by minimizing program size as a second objective [1]. Some work has been done in Multi-Criteria Decision Making (MCDM) for selecting a desirable solution post-optimization run [12].

4 Results on Single-obj. Problems

We have tested our proposed IP finding algorithm on a number of well-known single-objective test problems. The parameter values for each problem are presented in Table 1. We are interested in solving the following single-objective optimization problem, in which there are n variables ($\mathbf{x} \in \mathbb{R}^n$):

$$\begin{array}{ll} \text{Minimize} & f(\mathbf{x}), \\ \text{subject to} & g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \ldots, J, \end{array} \tag{9}$$

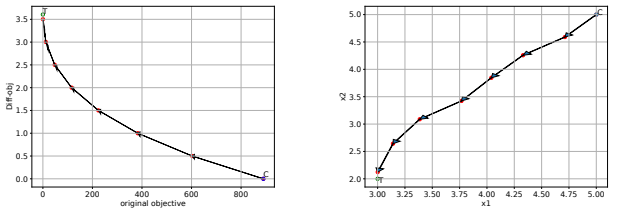
having at least one local or global feasible minimum $\mathbf{x}^*$ with function value f^*. For this problem, there exists a current solution $\mathbf{x}^{\text{C}}$ with

objective value f^C, which is supposedly worse than f^* or $f^C \gg f^*$. Since there is a single minimum solution, our goal is to arrive at the minimum (target) solution $\mathbf{x}^T = \mathbf{x}^*$ having a better objective value than f^C. The value for N and the maximum number of generations T_{max} used for each problem can be found in Table 1. We use SBX crossover [6] with $\eta = 15$ and $prob = 0.9$ and polynomial mutation [8] with $\eta = 20$ for all problems.

4.1 Innovation Path on Extended Pareto Front

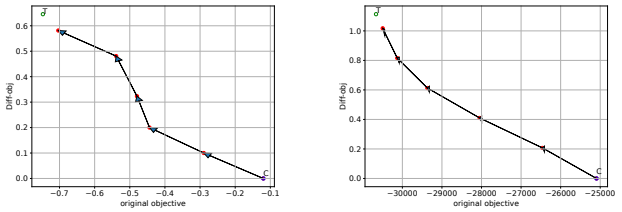
First, the paths are found using the G_1 step-constraint. Figures 3a and 3b show IP solutions for the Himmelblau problem in the Diff-obj and in the variable spaces, respectively. Starting from the current solution, both plots present a few intermediate solutions satisfying the G_1 step-constraint between two consecutive solutions to reach close to the target solution. It is important to highlight that all eight solutions together make a chain of solutions with an optimal trade-off between two consecutive solutions. An improvement in f in one IP solution comes from a minimal difference in the solution (in terms of step-constraints) from the previous IP solution. Satisfaction of the step-constraints bind the intermediate IP solutions and provide them the optimal trade-off properties.

Figures 4a and 4b show the IP solutions for the g-series problems g2 and g4, respectively. The blue point shows the current solution (C) while the green point on the plot represents the target solution (T). The path is represented by red points joined by arrows.



(a) Diff-Obj space **(b) variable space**

Figure 3: IP with G_1 step-constraint for the Himmelblau problem.

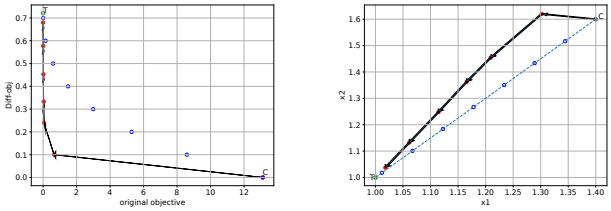


(a) g2 **(b) g4**

Figure 4: Innovation paths with G_1 step-constraint for constrained test problems g2 and g4.

4.2 Comparison with a Simple Approach

If the current and target solutions are known a priori, it is interesting to compare IP solutions with a series of step-constraint satisfied solutions falling on a straight line joining current and target solutions. Clearly, latter solutions are not likely to result in optimal solutions on the extended problem. Figure 5 shows IP solutions and the straight-line solutions for a step-size of 0.1 on Rosenbrock problem. As can be seen in Figure 5a the straight-line solutions in blue get dominated by the IP solutions in the objective space.

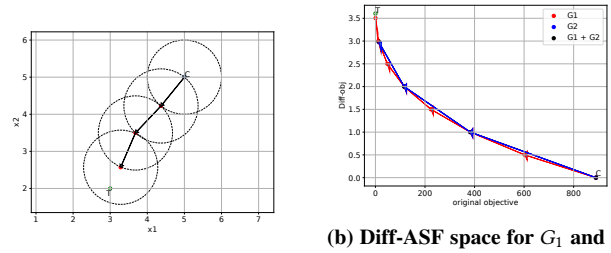


(a) Diff-Obj space **(b) variable space**

Figure 5: IPs with G_2 step-constraint for Rosenbrock problem, compared to a straight line approach.

4.3 Effect of Step-constraints on IP

To show the effect of different step-constraints on the IP solutions, we choose the Himmelblau problem. Figure 6a shows the IP solutions using the G_2 step-constraint in the variable space. We additionally show boundary of the constraint on each obtained point to demonstrate that two consecutive IP solutions satisfy the G_2 step-constraint. The last IP solution is too close to the target to have any further feasible point. Figure 6b compares IP solutions found by G_1 and G_2 step-constraints, showing not much difference in the outcome from these two step-constraints.



(a) var. space for G_2 step-constr. **(b) Diff-ASF space for G_1 and G_2 step-constr**

Figure 6: IPs with G_1 and G_2 step-constraints for Himmelblau.

4.4 Keyboard Layout Optimization Problem

As found in an earlier study [9], the most optimal keyboard for a minimum finger movement objective (tested on a large volume of text) is completely different than the standard QWERTY configuration. Hence in this study, we model the problem to find set of IP solutions from QWERTY to more optimal finger movement solution. Due to its combinatorial nature, we use a genetic algorithm proposed in [9]. We use a population size of 100 and run for 100 generations with the step size of 3 and γ of 0.1. Figure 7 shows the IP solutions starting from the QWERTY layout at the bottom right corner of the plot. Figure 8 shows all four keyboard configurations found by our proposed method. It can be seen how the number of gradual changes

at each step is made to arrive at the final configuration, thereby reducing the finger movement from 180,000 to about 100,000. At most four key changes are made from one IP solution to the next to make the transition in achieving maximum typing efficiency improvement in all consecutive pairs of IP solutions in one application of the proposed EMO algorithm.

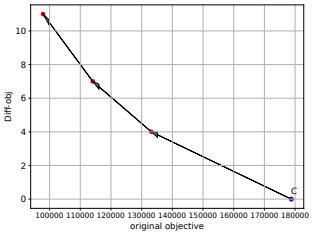


Figure 7: IP in Diff-Obj. space for keyboard problem.

(a) QWERTY (Q)

(b) Config. 1: 4 diff. from Q

(c) Config. 2: 7 diff. from Q, 3 from (b)

(d) Final: 11 diff. from Q, 4 from (c)

Figure 8: Keyboards on IP starting from QWERTY.

Table 1: Parameter values for single-objective problems.

	N	$T_{\max}$	γ	Step	Current soln.
Himmelblau	100	100	0.05	(0.5, 1)	[5,5]
Rosenbrock(2)	100	200	0.1	0.1	[1.4,1.6]
g2	120	250	0.1	0.1	[4,6,8,0.8,0.6,1,1,1,1,1]
g4	100	100	0.1	0.2	[100.32, 34.24, 40.47, 42.39, 37.94]

5 IP for Multi-objective Problems

Here, we are interested in solving the following M-objective optimization problem:

$$\begin{array}{ll} \text{Minimize} & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \\ \text{subject to} & g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \ldots, J, \end{array} \tag{10}$$

The original problem, when minimized, will find a Pareto-Efficient (PE) solution set $\mathbf{x}^*$ with respective objective vectors $\mathbf{f}^*$ (Pareto front (PF) points). Unlike in the single-objective case (where there was mostly a single target close to the objective's optimum), here there exist a number of Pareto-Efficient (PE) solutions. To have a single PE solution as a target, we choose a parameterized scalarization function $s(\mathbf{x})$ [11] to define a target solution. For example, the achievement scalarization function (ASF) [13] with a desired reference point $\mathbf{z}$ and a weight vector $\mathbf{w}$ can be used as $s(\mathbf{x})$:

$$s(\mathbf{x}) = \max_{i=1}^{M} \left(f_i(\mathbf{x}) - z_i\right) / w_i, \tag{11}$$

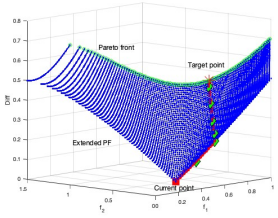
Thereafter, the task of finding the IP can be defined as a two-objective minimization of $s(\mathbf{x})$ and the difference function $f_{M+1}(\mathbf{x})$ (Equation 1).

To illustrate, we consider the two-objective version of a simplified ZDT1 test problem:

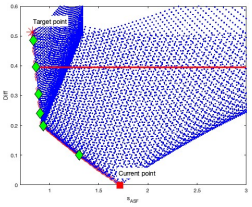
$$\begin{array}{lll} \text{Minimize} & f_1(x_1, x_2) = x_1, & \\ \text{Minimize} & f_1(x_1, x_2) = (1 - \sqrt{x_1})(1 + x_2), & \\ \text{subject to} & 0 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 1. & \end{array} \tag{12}$$

PE solutions satisfy: $x_2^* = 0$ and $x_1^* \in [0, 1]$ and PF points satisfy $f_2 = 1 - \sqrt{f_1}$ for $f_1 \in [0, 1]$ for this problem. Let us also assume that the current solution is at $\mathbf{x}^C = (0.5, 0.5)$ (not a Pareto-Efficient solution) having $\mathbf{f} = (0.500, 0.439)$. Figure 9a shows the three-dimensional extended Pareto-optimal front (E-PF) (shown by blue dots), for which the current point is the extreme (minimum 'Diff' f_3) point. The original PF of the ZDT1 problem is shown with green circles. Clearly, the E-PF is one dimension larger than the original PF for ZDT1, the E-PF is three-dimensional.

The ASF with a weight vector $\mathbf{w} = (0.75, 0.25)$ and a reference point $\mathbf{z} = (0, 0)$ sets the target point shown in a red star. The complete theoretically-optimal IP is marked using red circles, making a trade-off between $s_{\text{ASF}}(\mathbf{x})$ and $f_3(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^C\|_2$ (shown in Figure 9b). We choose a step-constraint $G_1 \equiv 0.1 - \|\mathbf{x}^{(i)} - \mathbf{x}^{(i+1)}\|_2 \leq 0$ for every two consecutive points on the IP. The path, shown with green diamonds in the plot, presents a finite set of intermediate points with gradual changes needed to change the current solution to the target solution in an optimal sense.



(a) IP on 3-obj. extended problem



(b) IP on 2-obj. Diff-ASF space

Figure 9: Illustration of IP for multi-objective problems on simplified ZDT1 problem.

Thus, it is clear from the above description that the task of finding a finite set of IP solutions for two or more objectives can be achieved by solving a two-objective optimization problem of minimizing a scalarized function $s(\mathbf{x})$ and $f_{M+1}(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^C\|_2$. This two-objective trade-off for ZDT1 problem and the respective IP points are shown in Figure 9b. When the current point happens to be one of the PF points, the extended PF becomes a part of the original PF, as will be demonstrated in Figure 11b in the next subsection.

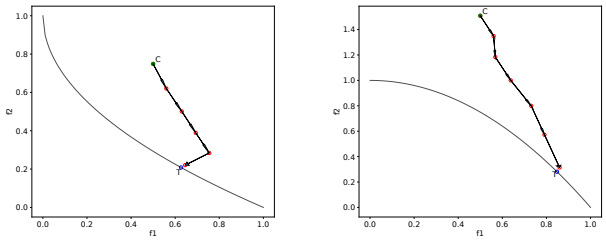
5.1 Results on Multi-objective Problems

We test the IP finding algorithm on unconstrained (ZDT1 to ZDT4 and ZDT6) and constrained (OSY and welded-beam) problems. The parameter settings are shown in Table 2 like the value for N and the maximum number of generations T_{max}. We use SBX crossover [6] with $\eta = 15$ and $prob = 0.9$ and polynomial mutation [8] with $\eta = 20$ for all problems.

Table 2: Parameter values for multi-objective problems.

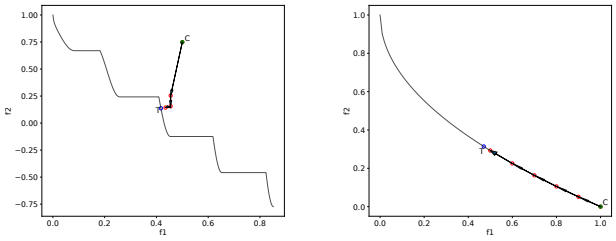
	N	$T_{\max}$	γ	Step	w	z	Current soln.
ZDT1	100	150	0.2	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.002,0.001,0,0,0,0]
ZDT2	100	200	0.3	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.002,0.001,0,0,0,0]
ZDT3	120	120	0.3	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.002,0.001,0,0,0,0]
ZDT4	100	150	0.1	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.02,0.01,0,0,0,0]
ZDT6	100	200	0.2	0.1	[0.75,0.25]	[0,0]	[0.5,0.51,0.002,0.001,0,0,0,0]
OSY	120	100	0.1	0.1	[0.2,0.8]	[-260,0]	[2,4,4,2]
Welded beam	100	100	0.05	0.1	[0.8,0.2]	[0,0]	[2,4,4,2]

We use G_1 step-constraint to generate paths for all problems. Figure 10a shows the generated path on the objective space of ZDT1. The green point represents the current solution and the blue point represents the target. Similarly Figures 10b and 11a show IP solutions for ZDT2 and ZDT3, while Figures 12a and 12b shows IP solutions for ZDT4 and ZDT6 problems. Figures 13a and 13b present IP solutions for constrained problem (OSY and welded beam). Results on other commonly-used test problems are also obtained, but are not shown here due to space restriction.



(a) ZDT1 **(b) ZDT2**

Figure 10: IPs for ZDT1 and ZDT2 problems with G_1 step-constraint originating from a non-optimal solution.



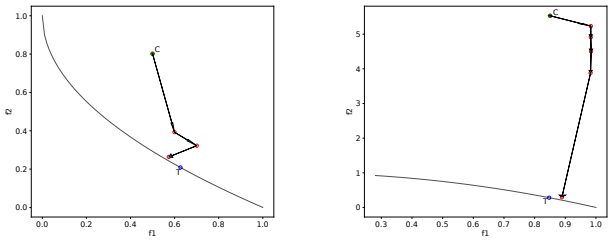
(a) ZDT3 **(b) ZDT1**

Figure 11: IPs for ZDT3 originating from a non-optimal solution and for ZDT1 starting from a Pareto-optimal solution.

Figure 11b shows IP solutions when the current solution is on the original Pareto front. Interestingly, in such a scenario, all IP solutions fall on the Pareto front to reach the close to the target.

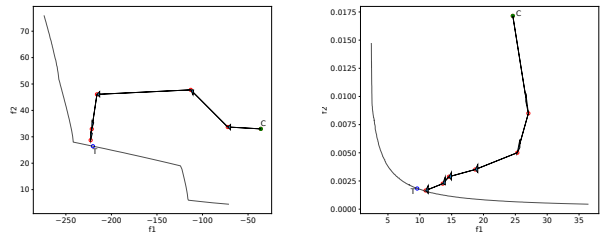
6 Conclusions and Future Studies

This paper has considered a number of single and multi-objective problems to demonstrate the concept of 'innovation path' for optimally migrating from a current solution to a better target solution in a finite number of steps. Interestingly, the task of finding multiple and directed IP solutions has been found by posing and solving a two-objective optimization problem of minimizing a scalarized and preferred function of original objectives (to arrive at a target optimal solution) and a difference metric from current solution. For single-objective problems, the first objective can simply be the objective function itself.



(a) ZDT4 **(b) ZDT6**

Figure 12: IPs for ZDT4 and ZDT6 problems with G_1 step-constraint originating from a non-optimal solution.



(a) OSY **(b) Welded beam**

Figure 13: IP with G_1 step-constraint for OSY and welded beam problems.

Based on this proof-of-principle study, we recommend the following future studies:

- Extend to three or more objective problems: The IP seeking method proposed here is always bi-objective irrespective of number of objectives in the original problem. The proposed method should extend easily.
- Define a new domination principle using supplied step-constraints: The requirement of non-dominated solutions to appear as a sequence has never been addressed before. This will require ideas from queuing theory to define a domination principle to achieve an IP.
- Achieve IP solutions using other EMO algorithms: A substantial change in the original NSGA-II procedure was needed to find IP solutions. An extension of the procedures with other bi-objective EMO would be interesting.
- Arrive at a non-optimal target solution: The user may be interested in arriving at one or more non-optimal target solutions from the existing solution.

References

[1] Bleuler, S., Brack, M., and Zitzler, E. Multiobjective genetic programming: Reducing bloat using SPEA2. In *Proceedings of the 2001 Congress on Evolutionary Computation* (2001), pp. 536–543.

[2] Branke, J., Deb, K., Miettinen, K., and Slowinski, R. *Multiobjective optimization: Interactive and evolutionary approaches*. Springer-Verlag, Berlin, Germany, 2008.

[3] Coello, C. A. C., and Montes, E. M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics 16* (2002), 2002.

[4] DATTA, R., AND DEB, K., Eds. *Evolutionary Constrained Optimization*. Infosys Science Foundation Series, Springer, 2015.

[5] DEB, K. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering 186*, 2–4 (2000), 311–338.

[6] DEB, K., AND AGRAWAL, R. B. Simulated binary crossover for continuous search space. *Complex Systems 9*, 2 (1995), 115–148.

[7] DEB, K., AGRAWAL, S., PRATAP, A., AND MEYARIVAN, T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation 6*, 2 (2002), 182–197.

[8] DEB, K., AND TIWARI, S. Omni-optimizer: A generic evolutionary algorithm for global optimization. *European Journal of Operations Research (EJOR) 185*, 3 (2008), 1062–1087.

[9] KHAN, A., AND DEB, K. Optimizing keyboard configuration using single and multi-objective evolutionary algorithms. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation* (2023), pp. 219–222.

[10] KNOWLES, J., CORNE, D., AND DEB, K. *Multiobjective problem solving from nature: from concepts to applications*. Springer, 2007.

[11] MIETTINEN, K. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, 1999.

[12] SHIR, O. M., CHEN, S., AMID, D., MARGALIT, O., MASIN, M., ANABY-TAVOR, A., AND BOAZ, D. Pareto landscapes analyses via graph-based modeling for interactive decision-making. In *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V* (2014), Springer, pp. 97–113.

[13] WIERZBICKI, A. P. 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, 1980, pp. 468–486.