Visualization and Performance Metric in Many-Objective Optimization

Zhenan He and Gary G. Yen

IEEE Transactions on Evolutionary Computation, Vol. 20, No. 3, June 2016

Presented by Drew Buck
10/4/2016

Outline

- Background
  - Multi-objective optimization
  - Multi-objective evolutionary algorithms (MOEAs)
  - Performance metrics
  - Many-objective evolutionary algorithms (MaOEAs)
  - Existing visualization methods
- Proposed Method
  - Visualization technique
  - Performance metric (p-Metric)
- Experimental Results
  - Comparison of 5 MaOEAs on several benchmark functions
- Conclusion

Multi-objective Optimization

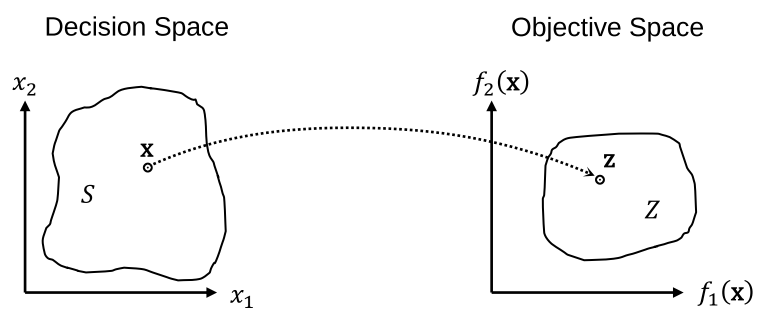
A multi-objective optimization problem (MOP) is of the form

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S \end{array}$$

> A maximization objective $f_i(\mathbf{x})$ can be converted to a minimization objective $f_i'(\mathbf{x})$ by setting $f_i'(\mathbf{x}) = -f_i(\mathbf{x})$.

Where

- $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$ is an *objective function*
- k (≥ 2) is the number of (conflicting) objective functions
- $\mathbf{x} = (x_1, \dots, x_n)$ is a *decision vector*
- $\mathbf{z} = f(\mathbf{x}) = \big(f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\big) = (z_1, \dots, z_k)$ is an *objective vector*
- S is the *feasible region* formed by constraints
- Z $(= f(S))$ is the *feasible objective region* of the objective space

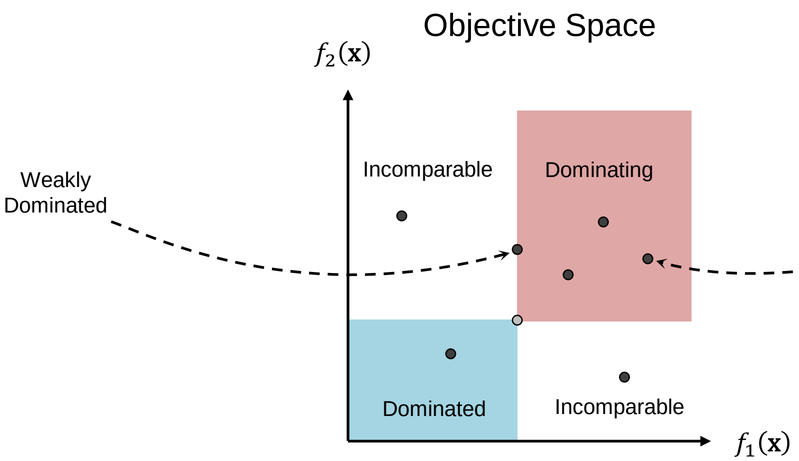


Pareto Dominance

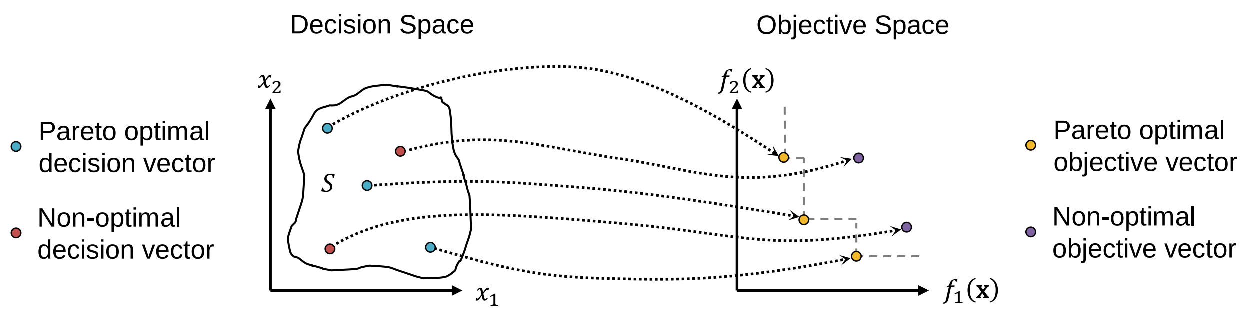
A solution vector $\mathbf{x}$ *dominates* another solution vector $\mathbf{y}$ if and only if:

- $f_i(\mathbf{x})$ is not worse than $f_i(\mathbf{y}), \forall\, i = 1, 2, \ldots, k$
- $f_j(\mathbf{x})$ is better than $f_j(\mathbf{y})$ for at least one $j = 1, 2, \ldots, k$

Obviously, an ideal solution to a MOP should not be dominated by any other solution.



Pareto Optimality



A decision vector $\mathbf{x}^*$ is *Pareto optimal* if and only if there does not exist another decision vector $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i = 1, \dots, k$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one index j.

An objective vector $\mathbf{z}^* = f(\mathbf{x}^*)$ is *Pareto optimal* if and only if there does not exist another objective vector $\mathbf{z} \in Z$ such that $z_i \leq z_i^*$ for all $i = 1, \dots, k$ and $z_j^* < z_j$ for at least one index j.

The Pareto Optimal Set

The set of Pareto optimal solutions forms the Pareto optimal (PO) set.

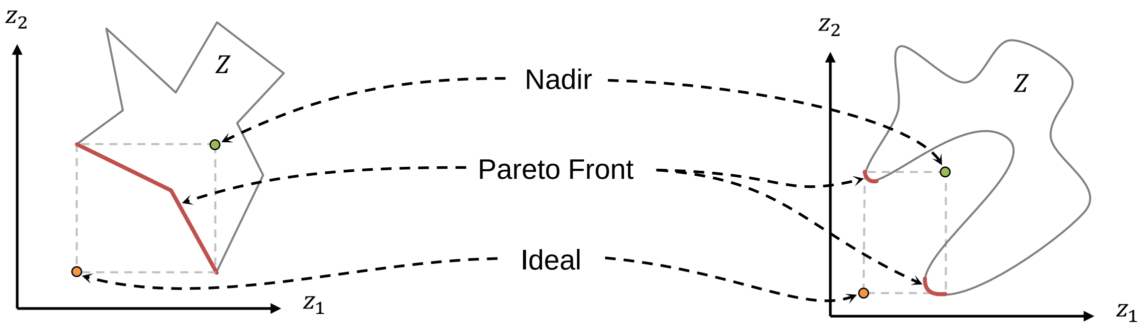
- In objective space, the PO set is called the *Pareto front*.
- There may be many (infinite) solutions within the PO set.
- Computing all solutions within the PO set may be infeasible.

The *ideal objective vector* $\mathbf{z}^{\star} \in \mathbb{R}^k$ represents the best possible solution, obtained by minimizing each objective independently.

- The ideal objective vector is typically not in the feasible objective region.

The *nadir objective vector* $\mathbf{z}^{\text{nad}}$ is formed from the upper bounds of the PO set.

- The nadir objective vector may or may not be in the feasible objective region.

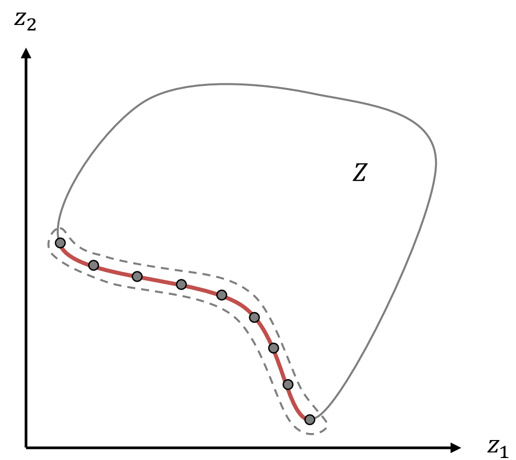


Finding the Pareto Optimal Set

- Multi-objective evolutionary algorithms (MOEAs) are well-suited for solving MOPs.
- The goal of a MOEA is to return a set of solutions that is a good approximation of the true PO set:
  - Close to the theoretical true Pareto front.
  - Well distributed over the entire theoretical true Pareto front.

Considerations:

- No longer a single optimal solution
  - How to assign fitness, perform selection, and do crossover and mutation?
- Need to maintain population diversity
  - Don't let the population converge to a single point.



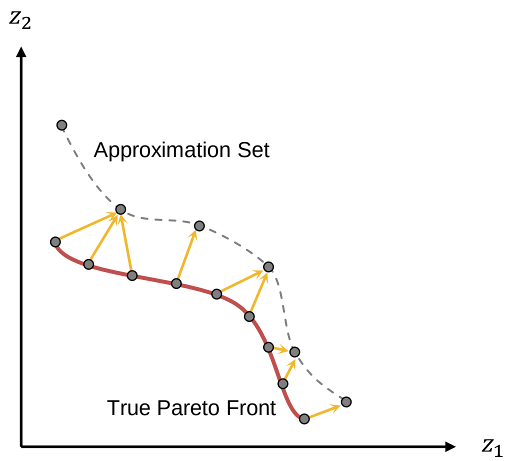
Performance Metrics

- Given a set of solutions, how well does it approximate the true Pareto optimal set?

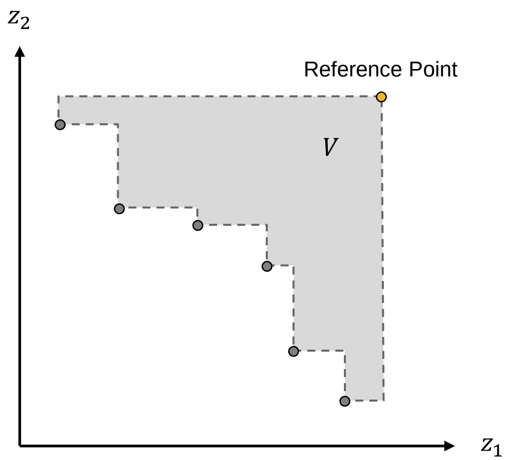
- 5 broad categories of performance metrics:
  - Methods that assess the number of Pareto optimal solutions in the set
    - Example: Ratio of Nondominated Individuals (RNI)
      - Measures the proportion of nondominated solutions to population size
  - Methods that measure how close solutions are to the theoretical true Pareto front
    - Example: Inverted Generational Distance (IGD)
      - Measures the distance between solutions on the true Pareto front and their closest neighbors on the approximate Pareto front
  - Methods that quantify the distribution of the set
    - Example: How evenly are the solutions distributed?
  - Methods that are concerned with the spread of the set
    - Example: Maximum Spread (MS)
      - Measures how well the true Pareto front is covered by the approximation set
  - Methods that consider both closeness to the theoretical true Pareto front and solution diversity simultaneously
    - Example: Hypervolume (S-metric)
      - Calculates the volume or area of the region covered by the approximation set with respect to a given reference point

Performance Metrics

- Two common metrics are compared in this paper:
  - Inverted Generational Distance (IGD) measures the average distance between points on the true Pareto front and the closest point in the approximation set.
  - Hypervolume (S-metric) measures the volume covered by the set with respect to a given reference point.



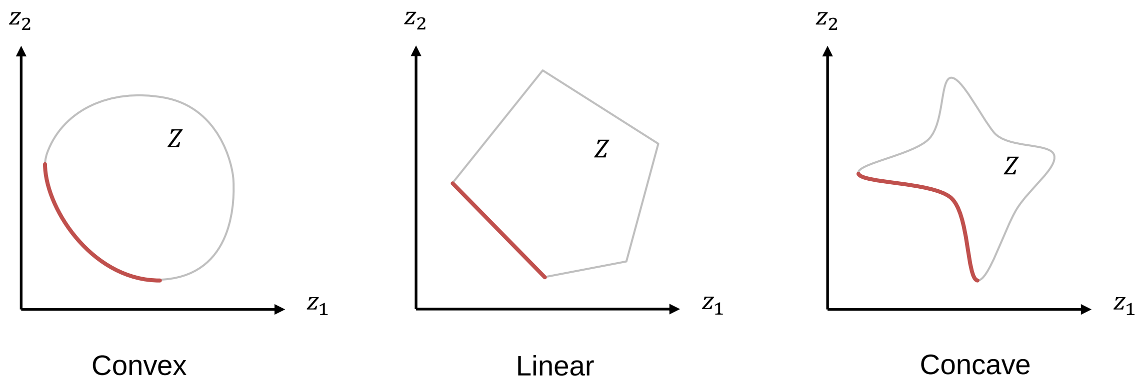
Inverted Generational Distance (IGD)



Hypervolume (S-metric)

Shape of the Pareto Front

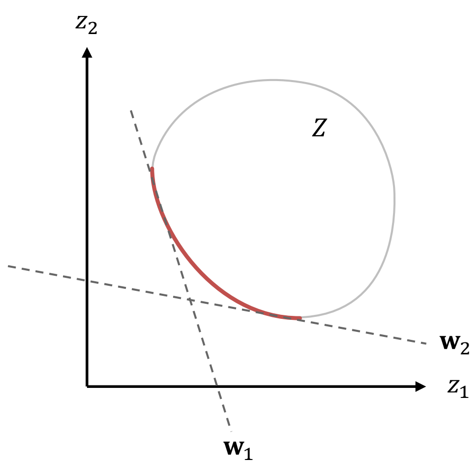
There are three basic shapes of Pareto fronts: convex, linear, and concave. The shape is determined by the feasible objective region of the problem.

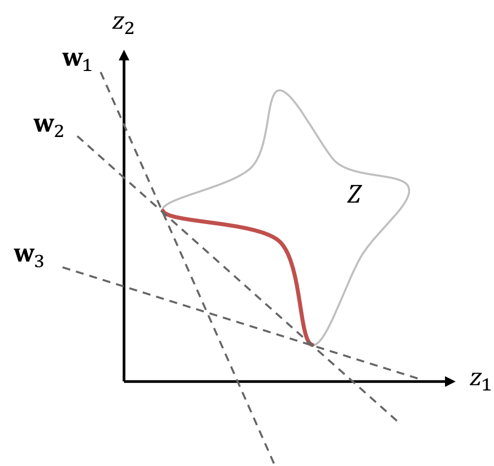


The Pareto front can also consist of a mixed front, which is a combination of the three basic types, or be discontinuous.

Why Does Shape Matter?

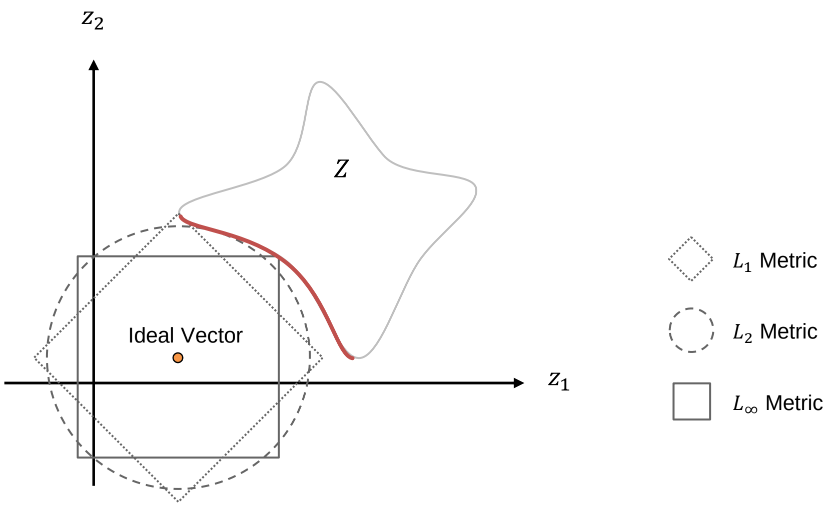
- The shape of the Pareto front can affect how well an optimization algorithm performs.
- Consider the common strategy of choosing a weight vector to reduce the multi-objective problem to a single objective:
  - Given a weight vector $\mathbf{w} = (w_1, \dots, w_k)$, pick the solution $\mathbf{z} \in Z$ that minimizes $\sum_{i=1}^{k} w_i z_i$
- If the Pareto front is concave, linear weighting will only produce solutions at the edges.





Generalized Weighted Metric

- Different solutions can be obtained by using a weighted L_p metric:
  - Pick the solution $\mathbf{z} \in Z$ that minimizes $\left(\sum_{i=1}^{k} w_i \, |z_i - z_i^\star|^p\right)^{\frac{1}{p}}$
- The L_∞ metric is also called the Tchebycheff metric:
  - Pick the solution $\mathbf{z} \in Z$ that minimizes $\max_{i=1,\ldots,k} w_i |z_i - z_i^\star|$

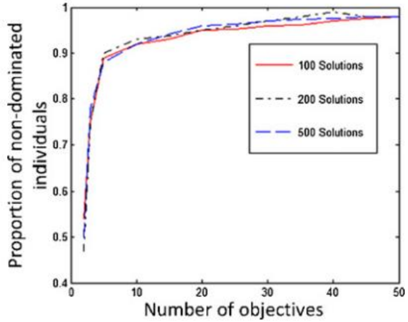


Many-Objective Optimization

- If $k \geq 4$, the problem is considered a many-objective optimization problem (MaOP).

- Difficulties in handling many objectives:
  - A large fraction of the population is nondominated
    - It becomes difficult for a solution to be the best in all objectives
  - Evaluation of diversity measure becomes computationally expensive
    - Need to find neighbors in k-dimensional space
  - Recombination operation may be inefficient
    - Children may be very far from parents
  - Representation of trade-off surface is difficult
    - Exponentially more points are required
  - Performance metrics are computationally expensive to compute
    - Example: calculating hypervolume has exponential complexity with respect to number of dimensions
  - Visualization is difficult
    - Hard to display >3 dimensional space

What proportion of randomly distributed individuals are nondominated in high-dimensional spaces?



Proportion of non-dominated individuals within a population.

MaOEAs

Many-Objective Evolutionary Algorithms (MaOEAs) are optimized for solving problems with many objectives.

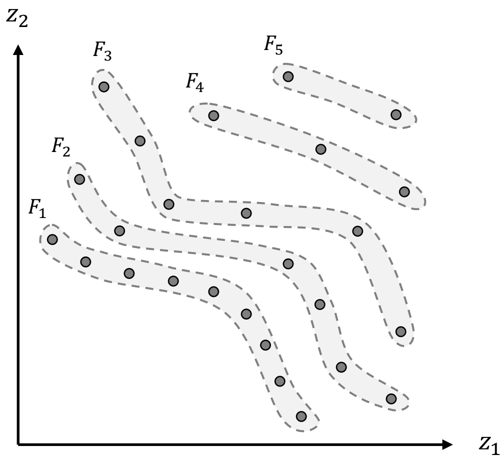
- This paper compares five state-of-the art MaOEAs:
  - MOEA/D
    - MOEA based on decomposition
  - NSGA-III
    - Reference-point based many-objective nondominated sorting genetic algorithm (NSGA)-II
  - ε-MOEA
    - ε-domination-based MOEA
  - HypE
    - Hypervolume estimation algorithm for multi-objective optimization
  - GrEA
    - Grid-based evolutionary algorithm

MOEA/D

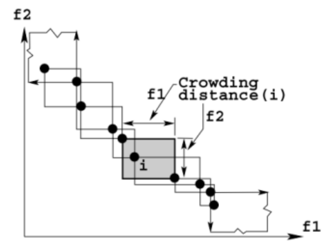
- MOEA/D is a multiobjective evolutionary algorithm based on decomposition.
  - Simplify the problem into several different single-objective problems using different scalarization functions.
  - Solve these problems using traditional optimization techniques.
- The main algorithm is:
  1. Create a uniformly distributed set of weight vectors
  2. Define the neighborhood region around each weight vector (e.g. 10 nearest neighbors)
  3. Create an initial population by solving the single-objective problem defined by each weight vector
  4. Each iteration, for each weight vector
     a. Select two solutions from the neighborhood of the weight vector and generate a new solution using genetic operators
     b. Perform a problem specific repair/improvement heuristic
     c. If the new solution dominates its neighbors, use it as the representative solution for this weight vector
     d. Update the archive population with the set of nondominated solutions

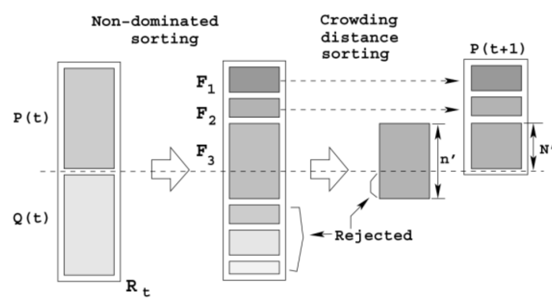
NSGA-II

- Generate a new population using binary tournament selection.
- Sort individuals based on dominance depth.
- Partitions that fit into the new population are copied directly.
- The last partition is further sorted based on crowding distance.
  - In high-dimensional spaces, this is often the first front!
- Individuals with the largest distance are added until the new population is filled.



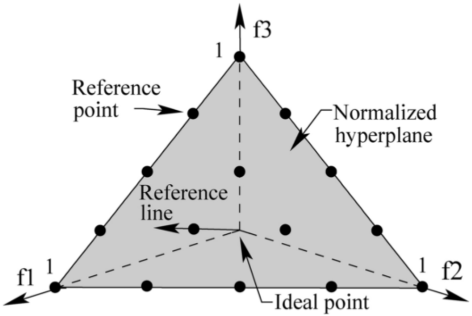
Dominance Depth Partitioning

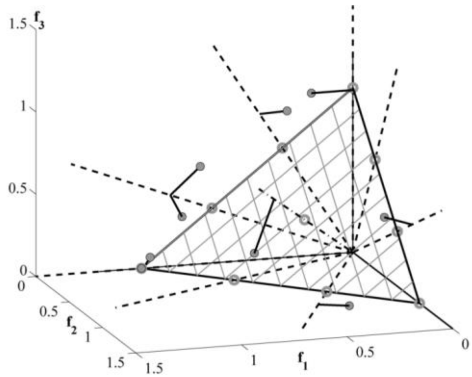




NSGA-III

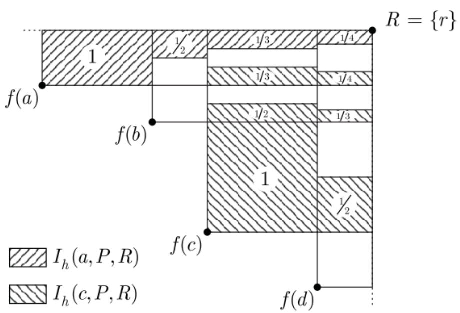
- Uses the same dominance based partitioning as NSGA-II.
- Instead of using crowding distance to accept solutions from the last front, NSGA-III uses uniformly distributed reference vectors, similar to MOEA/D.
- Each individual is associated with the nearest reference point, and niching is used to ensure that the subsequent generation contains a relatively uniform distribution of individuals.
- This also ensures that the population has a good distribution and spread over the entire Pareto front.

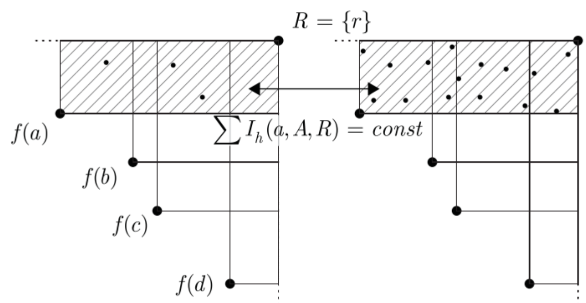




HypE

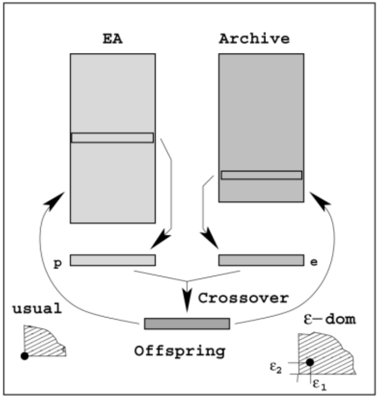
- Same approach as NSGA-II for generating a new population and partitioning into nondominated fronts.
- For the last front, compute the fitness of each individual using the hypervolume indicator, using Monte Carlo sampling to estimate the value in high-dimensional space, and move only the individuals with the best fitness to the new population. (Ex. 10,000 sample points)
  - Fitness is based on how much the hypervolume would change if this individual were removed from the front.
  - Approximate values are okay since only the rank of the individuals is important.

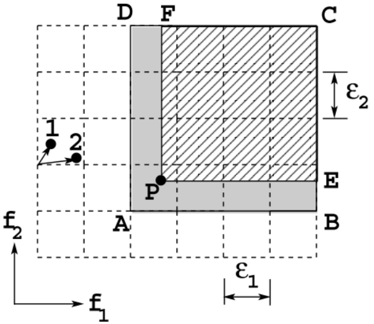




ε-MOEA

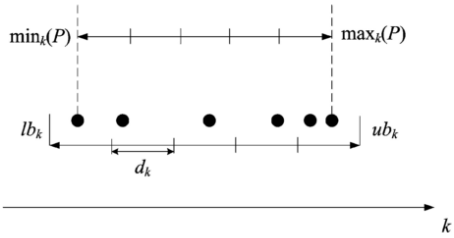
- ε-MOEA is a steady-state algorithm that uses two co-evolving populations: an EA population P and an archive population A containing the best ε-nondominated solutions.
- A random solution p is picked from P using binary dominance selection and a solution e is picked from A randomly.
- The child of p and e is accepted into the population P if it dominates an existing individual, which it replaces.
- The child is accepted into the archive population only if it is ε-nondominated.
  - Only one solution is allowed in each ε-sized grid cell, ensuring population diversity.





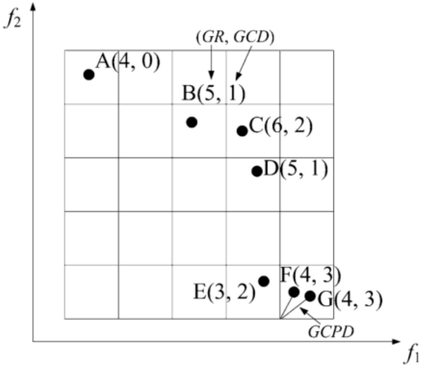
GrEA

Grid-based Evolutionary Algorithm



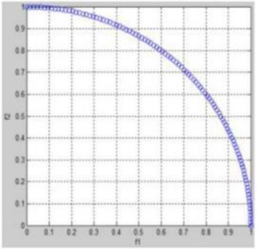
Each iteration:

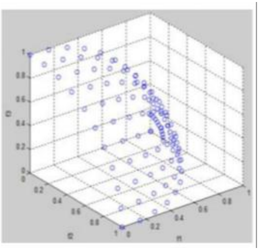
- Divide each objective range into a uniform number of grid cells
- Compute 3 fitness values for each individual:
  - Grid rank: $GR(\mathbf{x}) = \sum_{k=1}^{M} G_k(\mathbf{x})$, where $G_k(\mathbf{x})$ is the grid coordinate of $\mathbf{x}$ in objective k.
  - Density: $GCD(\mathbf{x}) = \sum_{\mathbf{y}\in N(\mathbf{x})}\big(M - GD(\mathbf{x},\mathbf{y})\big)$, where $GD(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{M}|G_k(\mathbf{x}) - G_k(\mathbf{y})|$ and $\mathbf{y} \in N(\mathbf{x}) \Leftrightarrow GD(\mathbf{x},\mathbf{y}) < M$.
  - Hyperbox distance: $GCPD(\mathbf{x}) = \sqrt{\sum_{k=1}^{M}\left(\frac{(F_k(\mathbf{x})-(lb_k+G_k(\mathbf{x})\times d_k))}{d_k}\right)^2}$, where $G_k(\mathbf{x})$ and $F_k(\mathbf{x})$ denote the grid coordinate and actual objective value of $\mathbf{x}$ for objective k. lb_k and d_k are the lower boundary and grid width for objective k.
- Generate new individuals using binary tournament selection and a hierarchy of the fitness values.
- Partition the new solutions into nondominated fronts as in NSGA-II and for the last front, use the computed fitness values to decide which solutions to add.
  - The fitness of neighbor solutions is adjusted as solutions are selected and removed from the previous population.

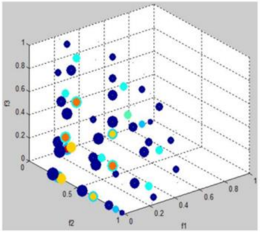


Visualizing MaOPs

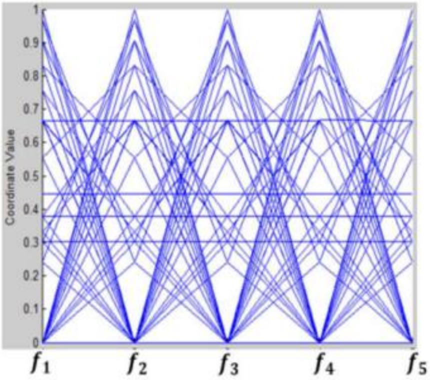
- In low-dimensional space, a scatter plot can show the location, distribution, and shape of the approximated front.
  - Each axis represents one objective
  - Limited to 2 or 3 objectives
  - A bundle chart also plots size and color, extending the number of possible objectives to 5.
- Most existing approaches can be categorized as…
  - Methods based on a parallel coordinate system:
    - Parallel coordinates
    - Heatmap
  - Methods based on mapping:
    - Sammon mapping
    - Neuroscale
    - RadViz
    - Self-Organizing Map (SOM)
    - Isomap

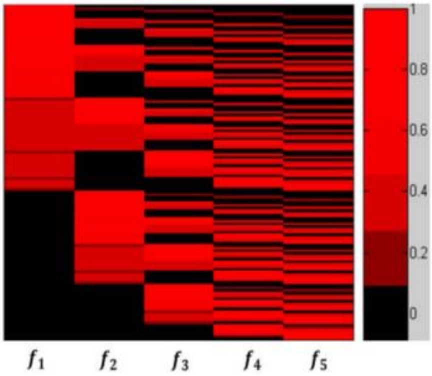






Parallel Coordinate Methods





<u>Parallel Coordinates</u>

Each M-dimensional vector is represented as a polyline that connects points on parallel axes.

- Shows dependencies between objectives
- Many individuals leads to over-crowded lines

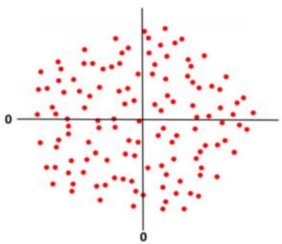
<u>Heatmap</u>

Each individual is a row in the image.
Color indicates objective value.
Rows are clustered based on similarity.

- Used to display microarray data
- High information density
- Difficult to see tradeoff between objectives

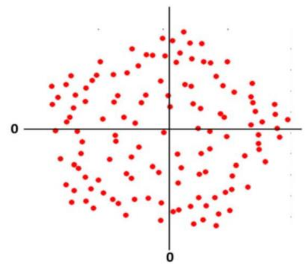
Stress Minimization Methods

The points are mapped from a high-dimensional space to a low-dimensional space in a way that preserves the distances between points.



<u>Sammon Mapping</u>

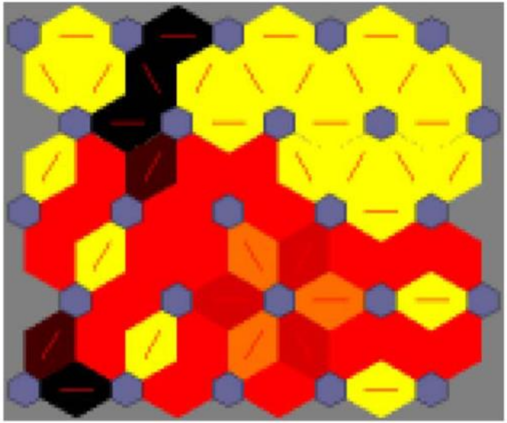
Uses gradient descent or other iterative methods to minimize a stress function of the form $E = \frac{1}{\sum_{i<j} d_{ij}^*} \sum_{i<j} \frac{\left(d_{ij}^* - d_{ij}\right)^2}{d_{ij}^*}$, where d_{ij} and d_{ij}^* are distances between the i^{th} and j^{th} points in the original and projected spaces respectively.

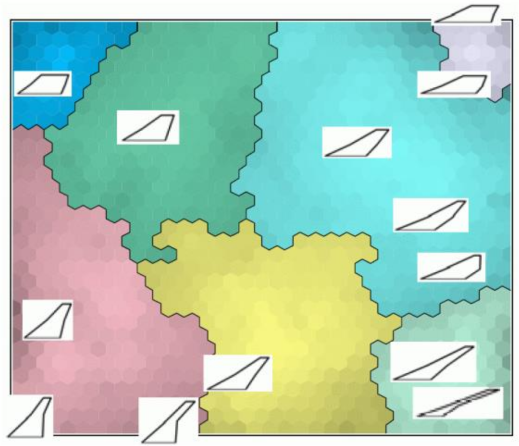


<u>Neuroscale</u>

Similarly, minimizes a stress function using an RBF neural network to generalize the projection transformation to unseen data points.

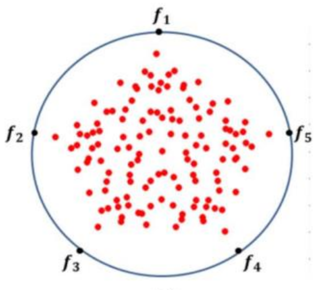
Self-Organizing Maps



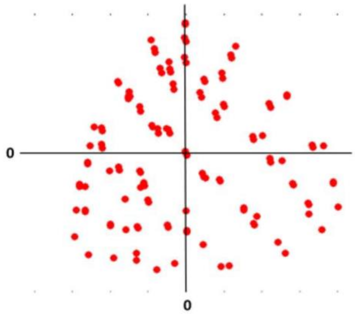


- Individuals in a high-dimensional space are mapped onto a grid of neurons arranged in a (usually 2D) topology.
- When trained, nearby vectors in the high-dimensional space are mapped onto nearby neurons in the SOM.
- Regional clusters in the SOM represent similar feature vectors.

Other Mapping Methods



(g)



<u>RadViz</u>

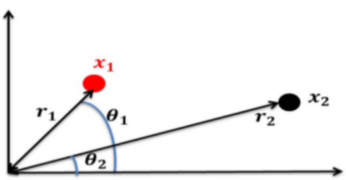
- Objectives are represented as anchors on a circle.
- Individuals are connected to each anchor with “springs” that are weighted according to the relative objective values.
- Preserves the distribution of vectors, but does not show the shape of the Pareto front.

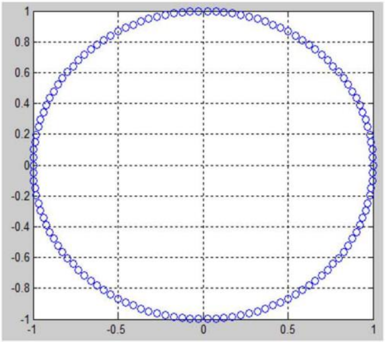
<u>Isomap</u>

- Topological geometry of the high-dimensional space is preserved by linking points only to their nearest neighbors.
- Distance between points is computed as the shortest path through the topology.
- Multidimensional scaling is applied to map points to 2D while preserving pairwise distances.

Proposed Visualization Method

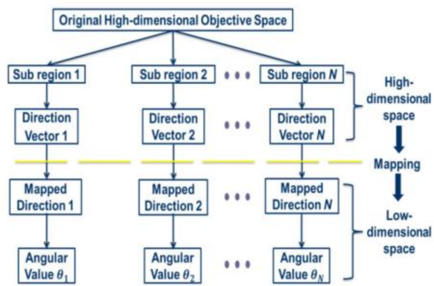
- The proposed visualization method maps individuals from a high-dimensional Cartesian space into a 2D polar coordinate system.
  - Angular coordinates show the distribution of individuals on the approximated Pareto front and the crowdedness in each subregion of high-dimensional space.
  - Radial coordinates show the convergence status toward the theoretical true Pareto front.

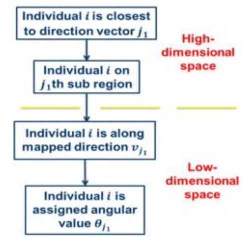




Angular Coordinate

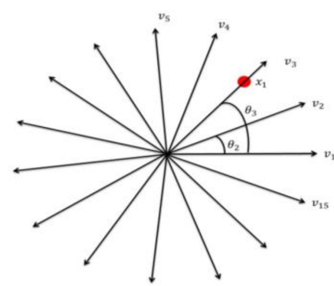
- The high-dimensional objective space is evenly divided into subregions.
- Each subregion is represented by one direction vector and assigned an angular coordinate.
- Individuals in the original objective space are mapped onto the closest direction vector and assigned the same angular coordinate.





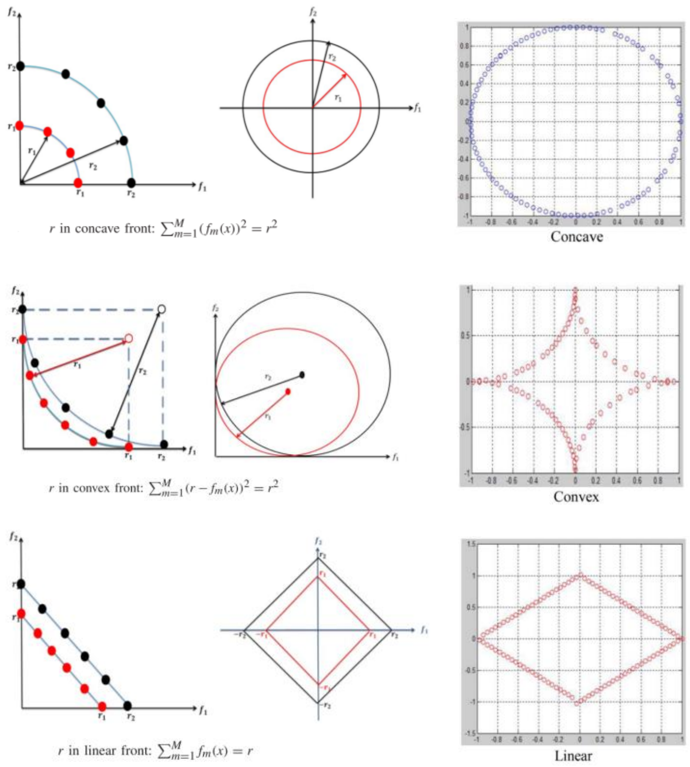
MAPPING FROM 3-D TO 2-D SPACES

Sub Region (3D)	Direction Vector (3D)			Mapped Direction (2D)	θ_i (2D)
1	0	0	1	v_1	0°
2	0	0.25	0.75	v_2	24°
3	0	0.5	0.5	v_3	48°
4	0	0.75	0.25	v_4	72°
5	0	1	0	v_5	96°
6	0.25	0	0.75	v_6	120°
7	0.25	0.25	0.5	v_7	144°
8	0.25	0.5	0.25	v_8	168°
9	0.25	0.75	0	v_9	192°
10	0.5	0	0.5	v_{10}	216°
11	0.5	0.25	0.25	v_{11}	240°
12	0.5	0.5	0	v_{12}	264°
13	0.75	0	0.25	v_{13}	288°
14	0.75	0.25	0	v_{14}	312°
15	1	0	0	v_{15}	336°



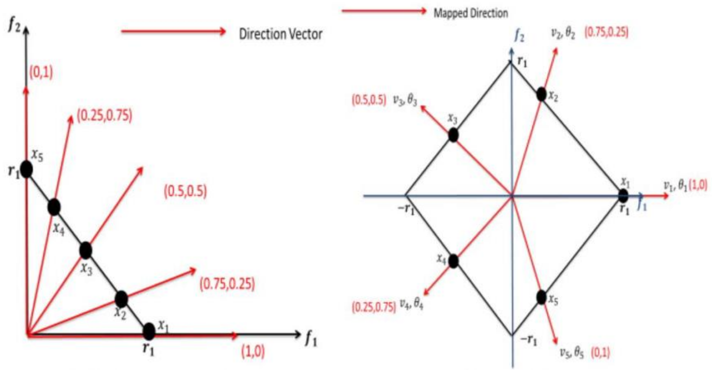
Radial Coordinate

- The radial coordinate indicates how close an individual is to the theoretical true Pareto front.
- For each of the three basic front shapes, there is a constant r that can be used to characterize the shape of the front:
  - Concave: $\sum_{m=1}^{M} f_m(\mathbf{x})^2 = r^2$
  - Convex: $\sum_{m=1}^{M} \left(r - f_m(\mathbf{x})\right)^2 = r^2$
  - Linear: $\sum_{m=1}^{M} f_m(\mathbf{x}) = r$
- Smaller values of r indicate better convergence performance.
- Individuals are assigned a radial coordinate based on closeness to the true Pareto front.
- Each quadrant of the mapped space represents a subfront of the objective space.



Visualization Process

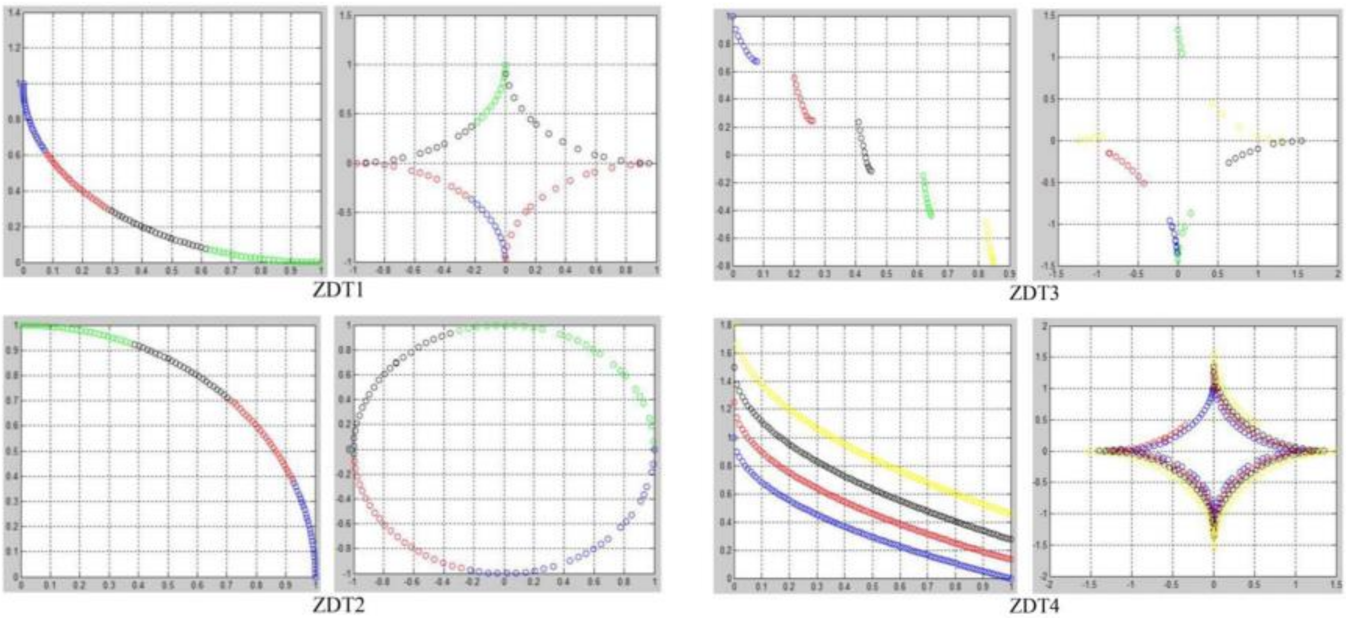
1. Precompute a set of equally distributed direction vectors in high-dimensional space and assign a fixed angular-coordinate to each.
2. Map each individual to the nearest direction vector in objective space and assign it the corresponding angular coordinate.
3. Determine the shape of the approximate front by solving for r using the three possible shapes.
4. If most individuals achieve the same value of r under one shape, use this shape as the approximate front. Otherwise consider a mixed front and treat each subpart independently.
5. Assign the radial coordinate for each individual based on the closeness to the approximate front.



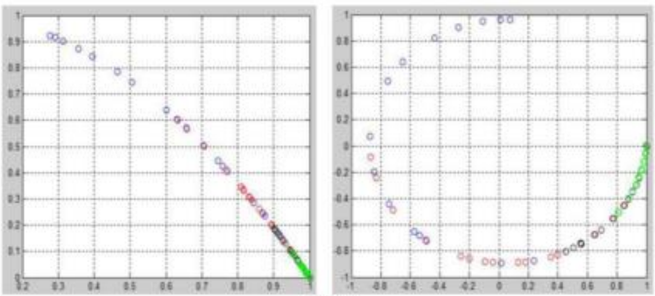
Original Pareto front

Mapped Pareto front

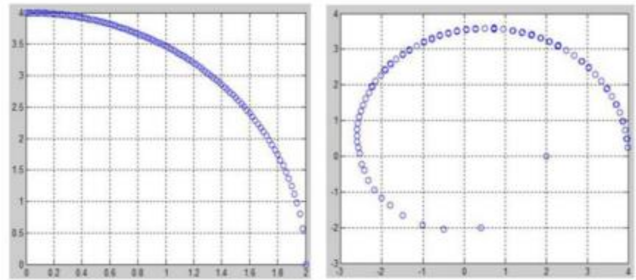
Visualization Examples



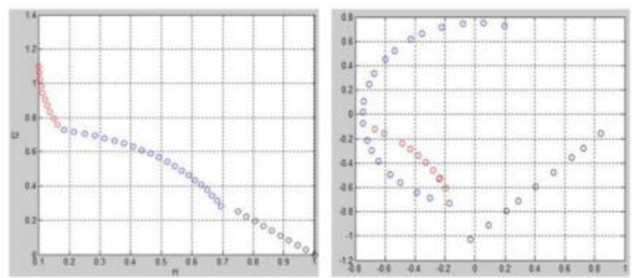
More Examples



ZDT6



Visualization of WFG4 test instance.



Visualizing mixed Pareto front.

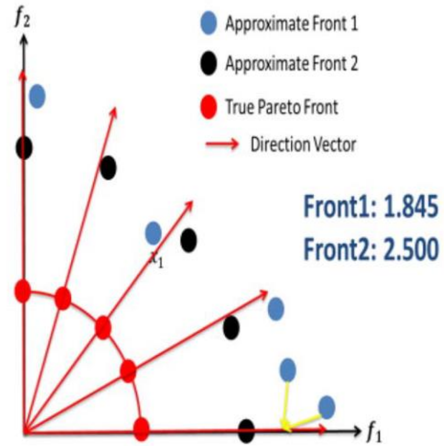
Visualization Summary

- Individuals are mapped from a high-dimensional objective space into a 2D polar coordinate system.
  - Radial coordinate reflects convergence performance
  - Angular coordinate reflects distribution of individuals
- Main contributions:
  - Mapping is consistent
    - Pareto dominance relationship, front shape and location, and the distribution of solutions is maintained
  - Allows for observation of the evolution process
    - Improvement of the approximation set can be tracked in location, range, and distribution as the population evolves
  - Decision-making is easy and effective
    - Solution quality and trade-offs can be observed from the plots
  - Scalable to any number of dimensions
    - High-dimensional objective spaces can be visualized, even with a large number of individuals on the front

p-Metric

- A new performance metric is proposed called p-metric that is based on the proposed visualization approach.

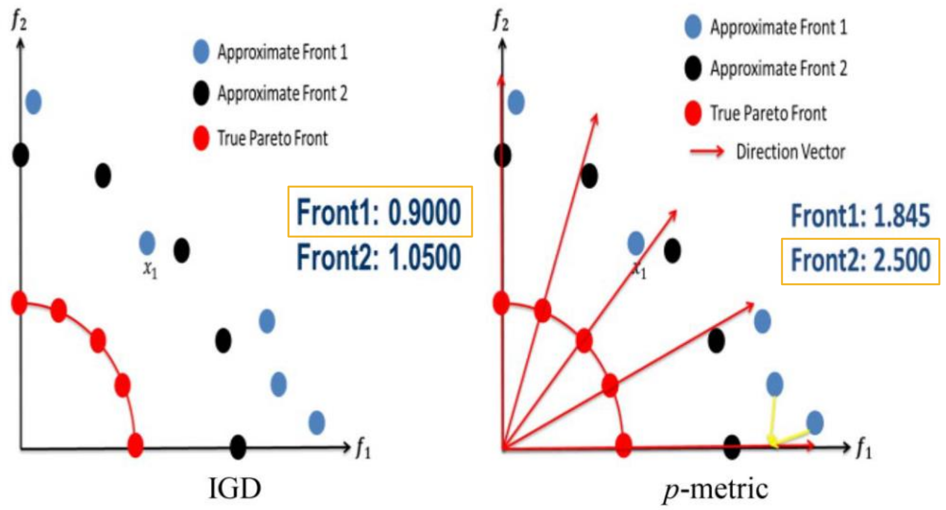
- Method:
  - For each direction vector j:
    - $r_{\min} = \min_{i=1:N_j} r_i$, where N_j is the number of solutions associated with direction vector j and r_i is the radius value of solution i.
    - $d_j = \begin{cases} \frac{1}{r_{\min}}, & N_j > 0 \\ 0, & N_j = 0 \end{cases}$
  - Compute performance score:
    - $S = \sum_{j=1:N} d_j$, where N is the number of direction vectors.



p-metric

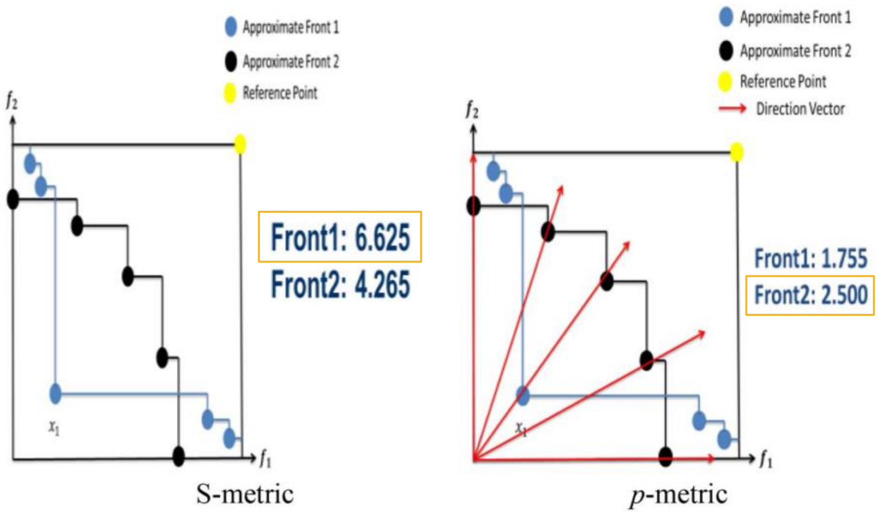
p-Metric Comparison

- All solutions except x_1 in the approximate front 1 (blue) are dominated by at least one solution on the approximate front 2 (black).
  - Based on this property, the authors prefer front 2.
  - However, because x_1 is so close to the true Pareto front, the IGD metric always assigns it as the closest point.
  - The p-Metric favors front 2 because it is well distributed with a solution for each direction vector and dominates most points on front 1.



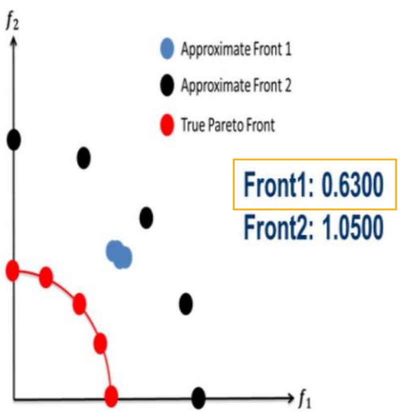
p-Metric Comparison

- All solutions except x_1 in the approximate front 1 (blue) are dominated by a solution on the approximate front 2 (black).
  - Based on this property, the authors prefer front 2 (although x_1 is close to the ideal point).
  - Front 1 encloses a larger area than front 2 so the hypervolume (S-metric) is larger for front 1.
  - The p-Metric favors front 2 because it is well distributed with a solution for each direction vector and dominates most points on front 1.

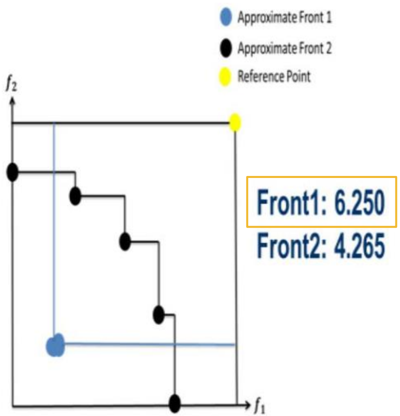


p-Metric Comparison

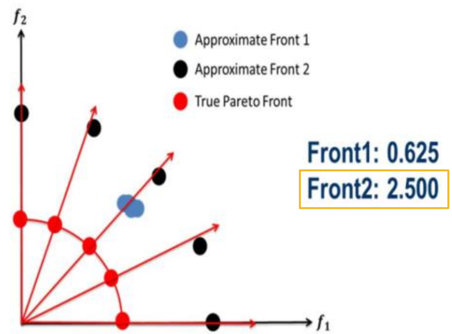
- All solutions from approximate front 1 (blue) are focused in a single neighborhood, as tends to happen in high-dimensional spaces.
- Approximate front 2 is well distributed, but farther from the ideal point.
  - Based on these properties, the authors prefer front 2.
  - As with the previous examples, IGD and S-metric prefer front 1, even with extremely poor diversity.
  - The p-Metric favors front 2 because it is well-distributed, although somewhat farther from the ideal point than front 1.



IGD



S-metric



p-metric

Experiments

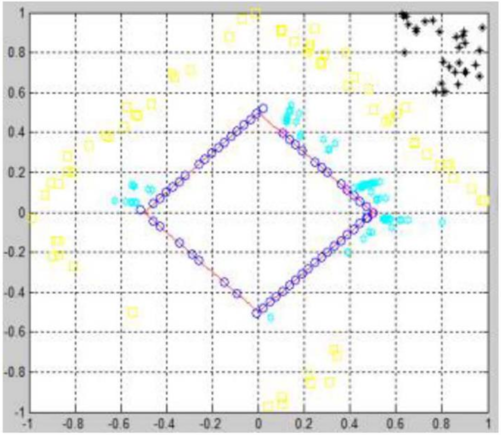
- 5 MaOEAs are tested on 5-D and 10-D benchmark functions DTLZ1-DTLZ7.
- For each problem, the following parameters are used:
  - Population size: 100
  - Stopping criteria: 10,000 generations
  - Initial population: uniform random sampling
  - Crossover: simulated binary crossover (SBX) $p_c = 1$
  - Mutation: polynomial mutation $p_m = \frac{1}{m}$
  - Number of decision variables (m):
    - 5-D DTLZ1: 9
    - 10-D DTLZ1 & 5-D DTLZ2-DTLZ7: 14
    - 10-D DTLZ2-DTLZ7: 19
  - Number of direction vectors:
    - 126 for 5-D
    - 55 for 10-D
- Each algorithm is run 30 times to compute the average performance metric.

DTLZ1

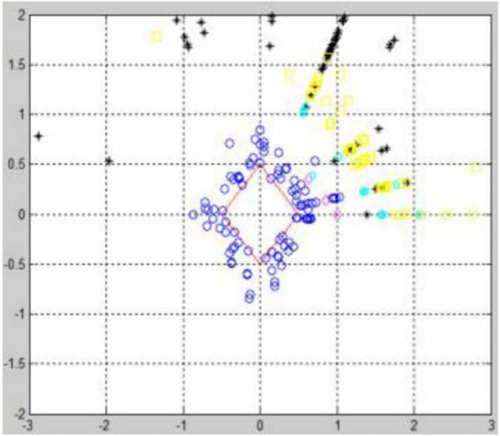
DTLZ1 has a linear Pareto front with a large number of local fronts.

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ1	1	GrEA	GrEA	GrEA
	2	NSGA-III	MOEA/D	MOEA/D
	3	HypE	HypE	HypE
	4	MOEA/D	NSGA-III	NSGA-III
	5	ε-MOEA	ε-MOEA	ε-MOEA

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ1	1	GrEA	GrEA	GrEA
	2	MOEA/D	HypE	HypE
	3	HypE	NSGA-III	MOEA/D
	4	NSGA-III	MOEA/D	NSGA-III
	5	ε-MOEA	ε-MOEA	ε-MOEA



5-D DTLZ1



10-D DTLZ1

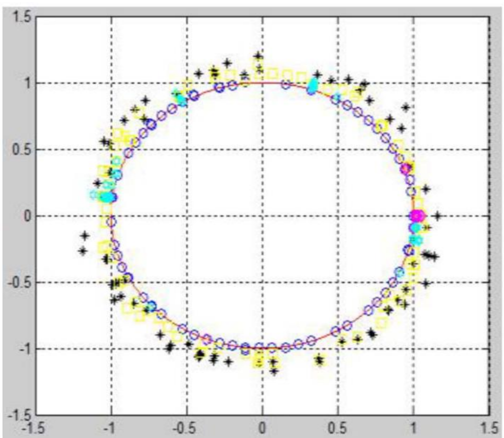
- GrEA (blue) performs best and ε-MOEA (black) is worst.
- In 5-D DTLZ1, the MOEA/D front (magenta) is localized to the top-right, behind the GrEA front (blue).
  - Despite poor diversity, MOEA/D still scores well with IGD and S-metrics.

- GrEA
- ε-MOEA
- NSGA-III
- MOEA/D
- HypE

DTLZ2

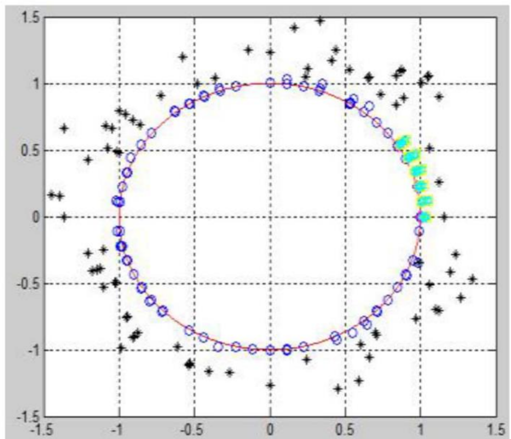
DTLZ2 has a single concave Pareto front.

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ2	1	GrEA	GrEA	MOEA/D
	2	NSGA-III	ε-MOEA	HypE
	3	ε-MOEA	NSGA-III	NSGA-III
	4	HypE	MOEA/D	GrEA
	5	MOEA/D	HypE	ε-MOEA



5-D DTLZ2

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ2	1	GrEA	GrEA	MOEA/D
	2	ε-MOEA	NSGA-III	HypE
	3	HypE	ε-MOEA	NSGA-III
	4	NSGA-III	MOEA/D	GrEA
	5	MOEA/D	HypE	ε-MOEA



10-D DTLZ2

- In 5-D DTLZ2, IGD shows that:
  - ε-MOEA (black) performs better than NSGA-III (yellow), but appears to have worse convergence.
  - MOEA/D (magenta) performs better than HypE (cyan), but has worse diversity.
- GrEA (blue) performs best, which is shown by the p-metric, but not by the S-metric.

○ GrEA
✦ ε-MOEA
□ NSGA-III
◊ MOEA/D
✡ HypE

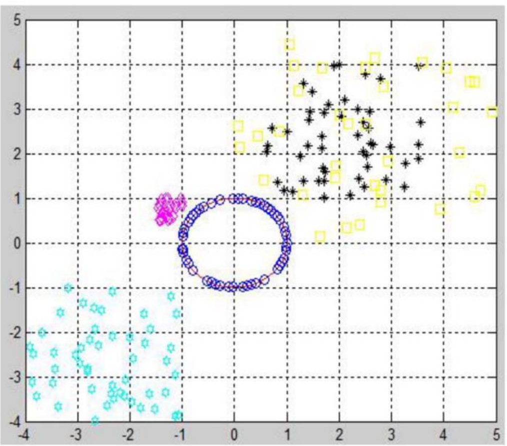
DTLZ3

DTLZ3 has a concave Pareto front and a large number of local Pareto fronts.

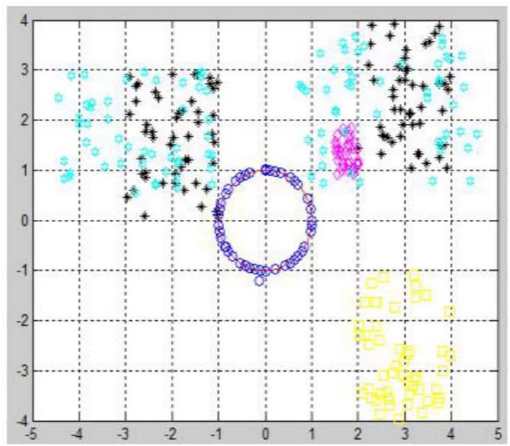
MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ3	1 2 3 4 5	GrEA HypE NSGA-III MOEA/D ε-MOEA	GrEA HypE MOEA/D NSGA-III ε-MOEA	GrEA HypE MOEA/D NSGA-III ε-MOEA

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ3	1 2 3 4 5	GrEA MOEA/D HypE ε-MOEA NSGA-III	GrEA HypE MOEA/D ε-MOEA NSGA-III	GrEA HypE MOEA/D ε-MOEA NSGA-III

- GrEA (blue) performs best.
- Other algorithms can only converge to several different local Pareto fronts.



5-D DTLZ3



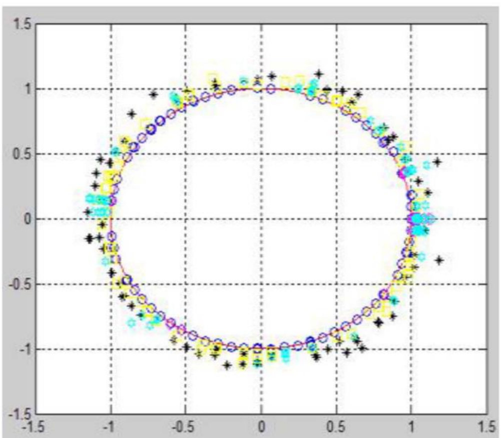
10-D DTLZ3

DTLZ4

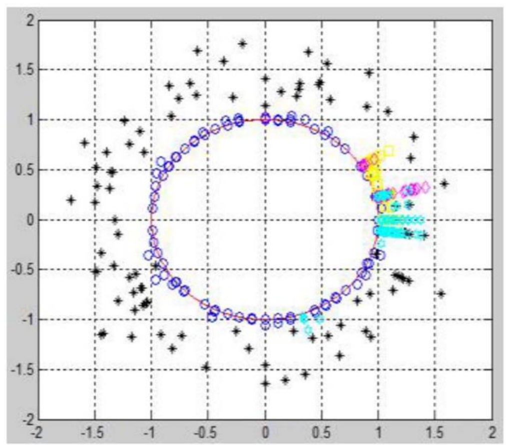
DTLZ4 has a single concave Pareto front with a non-uniform mapping from decision space to objective space to challenge solution diversity.

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ4	1	GrEA	NSGA-III	MOEA/D
	2	NSGA-III	ε-MOEA	HypE
	3	ε-MOEA	GrEA	NSGA-III
	4	HypE	HypE	GrEA
	5	MOEA/D	MOEA/D	ε-MOEA

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ4	1	GrEA	ε-MOEA	MOEA/D
	2	ε-MOEA	GrEA	HypE
	3	NSGA-III	NSGA-III	NSGA-III
	4	HypE	HypE	GrEA
	5	MOEA/D	MOEA/D	ε-MOEA



5-D DTLZ4



10-D DTLZ4

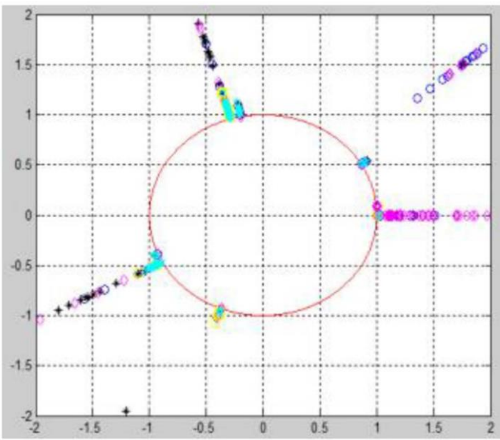
- GrEA (blue) continues to perform best.
- S-metric claims MOEA/D (magenta) and HypE (cyan) are best, despite poor diversity.
- In 10-D DTLZ4, IGD ranks ε-MOEA (black) above GrEA (blue) despite better convergence.
- In 10-D DTLZ4, S-metric ranks MOEA/D (magenta), HypE (cyan), and NSGA-III (yellow) above GrEA (blue) despite poor diversity.

DTLZ5

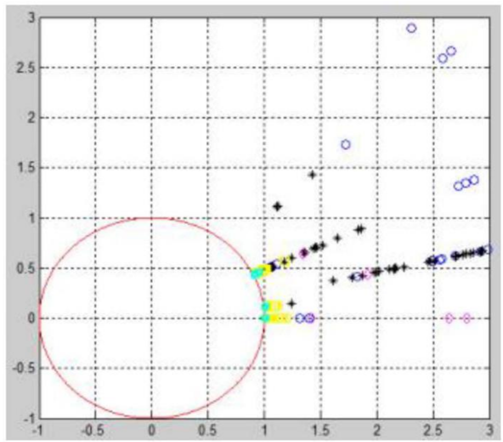
DTLZ5 has a degenerated hypersurface as the Pareto front.

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ5	1 2 3 4 5	MOEA/D NSGA-III HypE ε-MOEA GrEA	HypE MOEA/D NSGA-III ε-MOEA GrEA	MOEA/D HypE NSGA-III ε-MOEA GrEA

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ5	1 2 3 4 5	HypE NSGA-III GrEA ε-MOEA MOEA/D	HypE ε-MOEA NSGA-III GrEA MOEA/D	NSGA-III HypE ε-MOEA MOEA/D GrEA



5-D DTLZ5



10-D DTLZ5

- In 10-D-DTLZ5, IGD ranks ε-MOEA (black) above NSGA-III (yellow), despite worse convergence.
- NSGA-III (yellow) and HypE (cyan) perform best on this problem.
- GrEA (blue) performs poorly on DTLZ5 with a degenerated hypersurface.

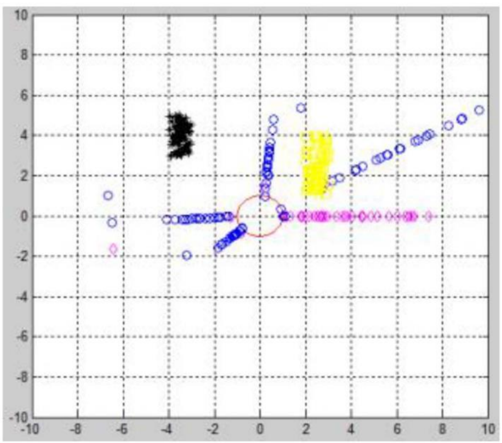
○ GrEA
✦ ε-MOEA
□ NSGA-III
◊ MOEA/D
✡ HypE

DTLZ6

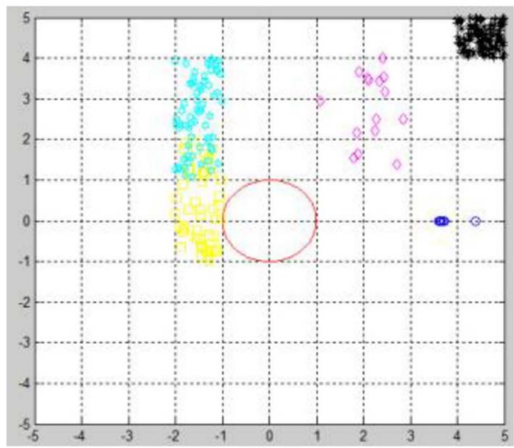
DTLZ6 has a large number of local Pareto fronts and disconnected Pareto-optimal regions.

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ6	1	MOEA/D	GrEA	MOEA/D
	2	GrEA	MOEA/D	GrEA
	3	NSGA-III	HypE	HypE
	4	HypE	ε-MOEA	NSGA-III
	5	ε-MOEA	NSGA-III	ε-MOEA

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ6	1	NSGA-III	NSGA-III	HypE
	2	HypE	MOEA/D	MOEA/D
	3	MOEA/D	HypE	NSGA-III
	4	ε-MOEA	GrEA	GrEA
	5	GrEA	ε-MOEA	ε-MOEA



5-D DTLZ6



10-D DTLZ6

- ε-MOEA (black) shows poor performance in DTLZ6.
- GrEA (blue) performs poorly in 10-D DTLZ6, but does well in 5-D DTLZ6.
- NSGA-III (yellow), HypE (cyan), and MOEA/D (magenta) perform best on high-dimensional disconnected problems.

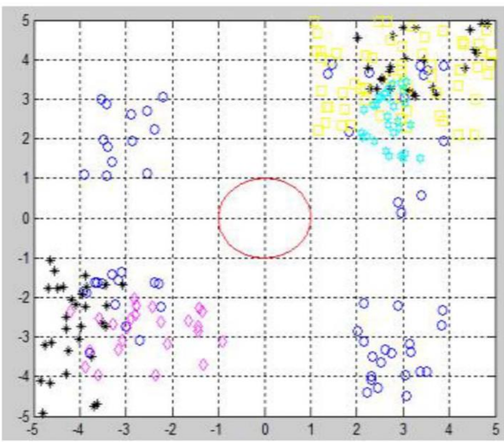
○ GrEA
✦ ε-MOEA
□ NSGA-III
◊ MOEA/D
✡ HypE

DTLZ7

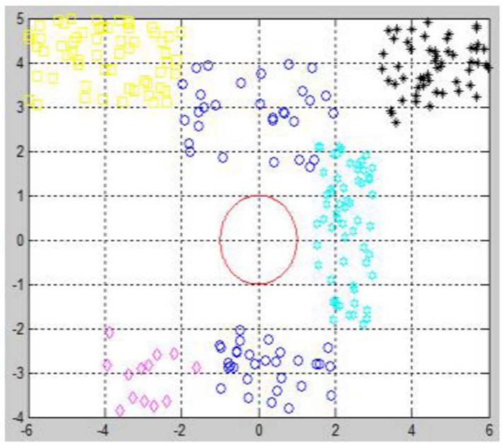
DTLZ7 has a Pareto front at the intersection of a straight line and a hyperplane.

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ7	1	GrEA	GrEA	HypE
	2	HypE	HypE	GrEA
	3	NSGA-III	NSGA-III	MOEA/D
	4	MOEA/D	MOEA/D	ε-MOEA
	5	ε-MOEA	ε-MOEA	NSGA-III

MaOP	Rank	*p*-metric	IGD	S-metric
DTLZ7	1	HypE	GrEA	HypE
	2	GrEA	HypE	GrEA
	3	ε-MOEA	ε-MOEA	ε-MOEA
	4	MOEA/D	MOEA/D	MOEA/D
	5	NSGA-III	NSGA-III	NSGA-III



5-D DTLZ7



10-D DTLZ7

GrEA
ε-MOEA
NSGA-III
MOEA/D
HypE

- None of the tested MaOEAs are effective at converging to the true Pareto front.
- A single metric alone cannot show this and some form of visualization is required to observe both convergence and diversity performance.

Conclusion

- Visualization is an important tool for evaluating MaOEAs and MaOPs.
  - The proposed visualization approach maps a high-dimensional objective space into a 2D polar coordinate plot while preserving Pareto dominance, shape and location of the Pareto front, and population diversity.
  - The approach is scalable to a large number of dimensions and can display many individuals and fronts simultaneously.
- The proposed performance metric, p-Metric is well-suited for high-dimensional MaOPs.
  - Convergence is measured by radial value.
  - Distribution is shown with angular coordinates.
  - Provides a comprehensive and consistent comparison among MaOEAs.