IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Introduction to Soft Computing

Multi-Objective Optimization-III

Debasis Samanta

Department of Computer Science and Engineering

IIT Kharagpur

Concept of domination

Notation

- ✓ Suppose, $f_1, f_2 \ldots \ldots . . f_M$ are the objective functions
- ✓ x_i and x_j are any two solutions
- ✓ The operator $\triangleleft$ between two solutions x_i and x_j as $x_i \triangleleft x_j$to denote that solution x_i is better than the solution x_j on a particular objective.
- ✓ Alternatively, $x_i \triangleright x_j$for a particular objective implies that solution x_i is worst than the solution x_j on this objective.

Note :

If an objective function is to be minimized, the operator $\triangleleft$ would mean the " $<$ " (less than operator), whereas if the objective function is to be maximized, the operator $\triangleleft$ would mean the " $>$ " (greater than operator).

Concept of domination

Definition 3: Domination

A solution x_i is said to dominate the other solution x_j if both condition I and II are true.

Condition : I

The solution x_i is no worse than x_j in all objectives. That is $f_k(x_i) \ntriangleright f_k(x_j)$ for all $k = 1, 2, \ldots\ldots, M$

Condition : II

The solution x_i is strictly better than x_j in at least one objective. That is is $f_{\bar{k}}(x_i) \triangleleft f_{\bar{k}}(x_j)$ for at least one $\bar{k} = \{1,2 \ldots \ldots M\}$

Properties of dominance relation

- ✓ Definition 3 defines the dominance relation between any two solutions.
- ✓ This dominance relation satisfies four binary relation properties.

Reflexive :

The dominance relation is **NOT** reflexive.

- ✓ Any solution x does not dominate itself.
- ✓ Condition II of definition 3 does not allow the reflexive property to be satisfied.

Properties of dominance relation

Symmetric :

The dominance relation also **NOT** symmetric

$x \preccurlyeq y$ does not imply $y \preccurlyeq x$

Antisymmetric :

Dominance relation **can not be** antisymmetric

Transitive :

The dominance relation is **TRANSITIVE**

If $x \preccurlyeq y$ and $y \preccurlyeq z$, then $x \preccurlyeq z$

Properties of dominance relation

Note :

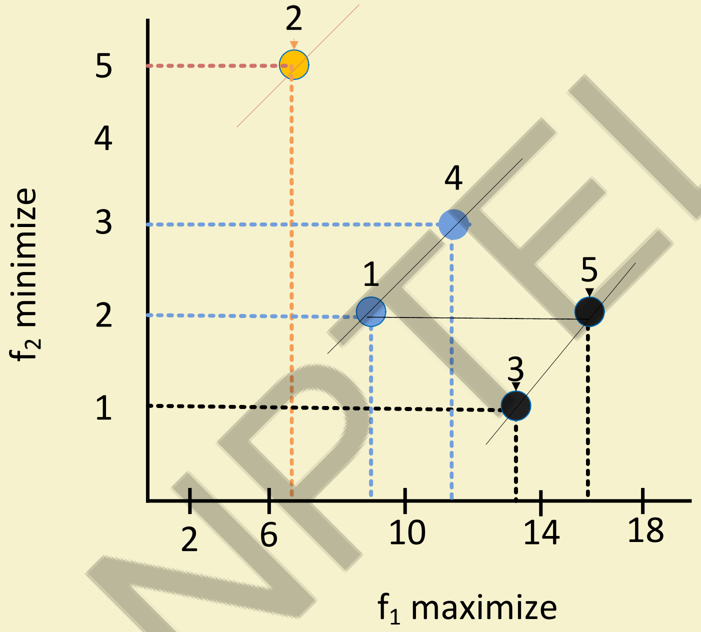
1) An interesting property that dominance relation possesses is : If solution x does not dominate solution y, this does not imply that y dominates x.

2) In order for a binary relation to qualify as an ordering relation, it must be at least transitive. Hence, dominance relation qualifies as an ordering relation.

3) A relation is called partially ordered set, if it is reflexive, antisymmetric and transitive. Since dominance relation is NOT REFLEXIVE, NOT ANTISYMMETRIC, it is NOT a PARTIALLY ORDER RELATION

4) Since, the dominance relation is not reflexive, it is a STRICT PARTIAL ORDER.

Pareto optimality



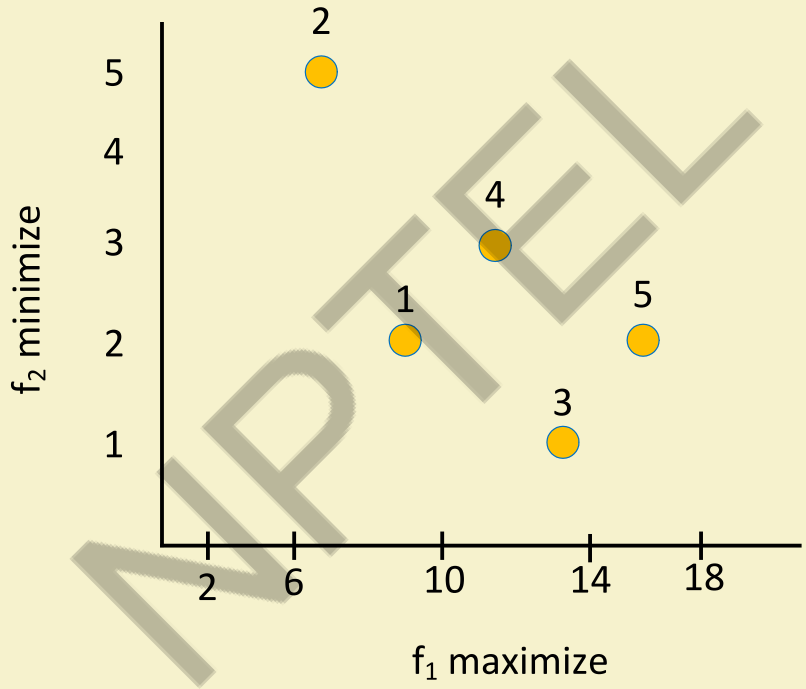
Non-dominated front

Pareto optimality

Consider solution 3 and 5.

- ✓ Solution 5 is better than solution 3 with respect to f_1 while 5 is worse than 3 with respect to f_2.
- ✓ Thus, condition I (of **Definition 3**) is not satisfied for both of these solutions.
- ✓ Hence, we can not conclude that 5 dominates 3 nor 3 dominated 5.
- ✓ In other words, we can not say that two solutions 3 and 5 are better.

Non-dominated set

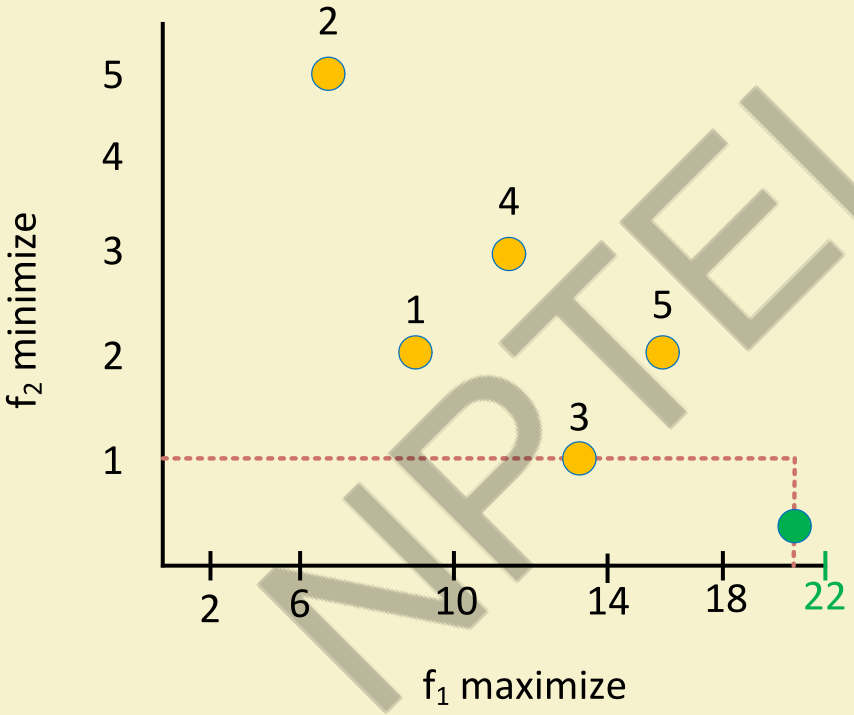


Non-dominated set

From the figure it is evident that

- ✓ There are a set of solutions namely 1, 2, 3, 4 and 5.
- ✓ 1 dominates 2; 5 dominates 1 etc.
- ✓ Neither 3 dominates 5 nor 5 dominates 3. We say that solution 3 and 5 are non-dominated with respect to each other.
- ✓ Similarly, we say that solution 1 and 4 are non-dominated.
- ✓ In this example, there is not a single solution, which dominates all other solution

Non-dominated set: A counter example

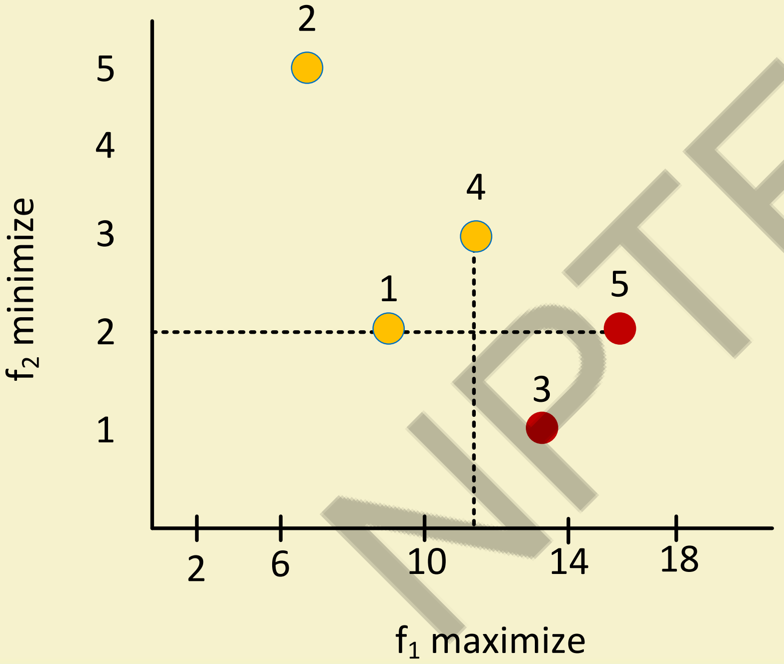


Non-dominated set

Definition 4: Non-dominated set

Among a set of solutions P, the non-dominated set of solutions P' are those which are not dominated by any member of the set P.

Non-dominated set



Here P = {1,2,3,4,5}

Non-dominated set

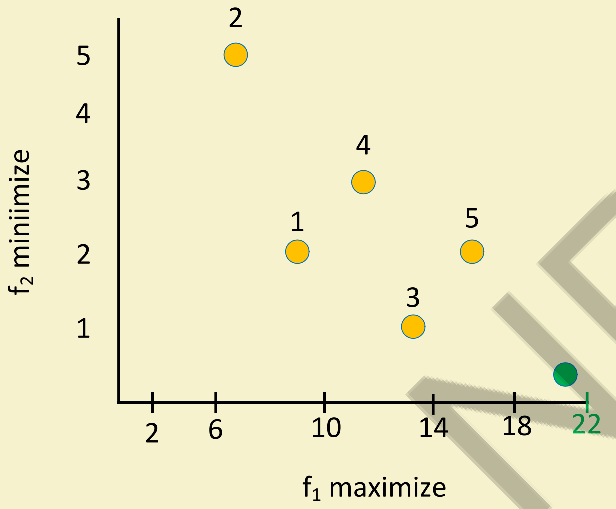
P' = {3, 5}

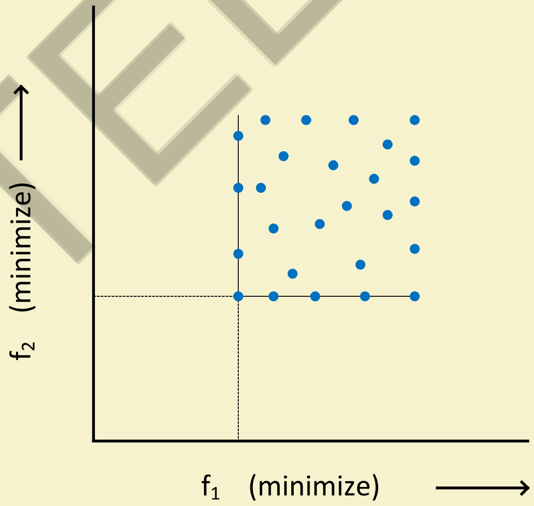
How to find a non-dominated set ?

- ✓ For a given finite set of solutions, we can perform all pair-wise comparisons.
  - Find which solution dominates
  - Find which solutions are non-dominated with respect to each other.
- ✓ Property of solutions in non-dominated set
  - $\exists x_i, x_j \in P'$ Such that $x_i \npreceq x_j$ and $x_j \npreceq x_i$
  - A set of solution where any two of which do not dominate each other if
    - $\exists x_i \in P$ and $x_i \notin P'$ then $x_i \preceq x_j$ where $x_i \notin P'$ for any solution outside of the non-dominated set, we can always find a solution in this set which will dominate each other.

Some important observations

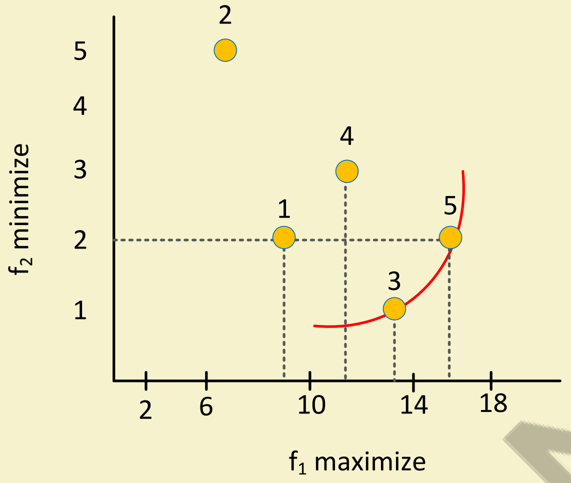
The above definition does not applicable to ideal situation.

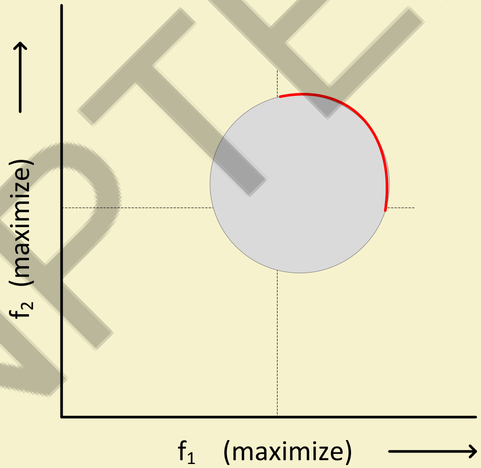


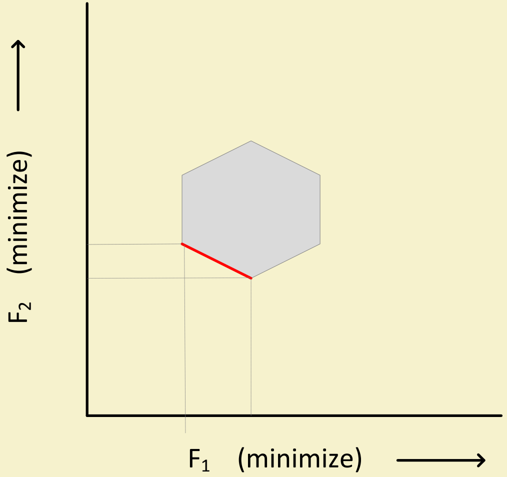


Some important observations

The non-dominated set concept is applicable when there is a trade-off in solutions.



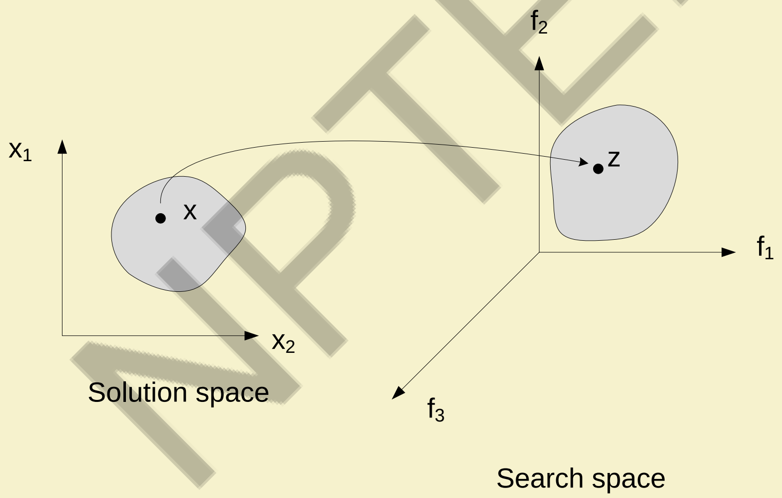




Pareto optimal set

Definition 5: Pareto optimal set

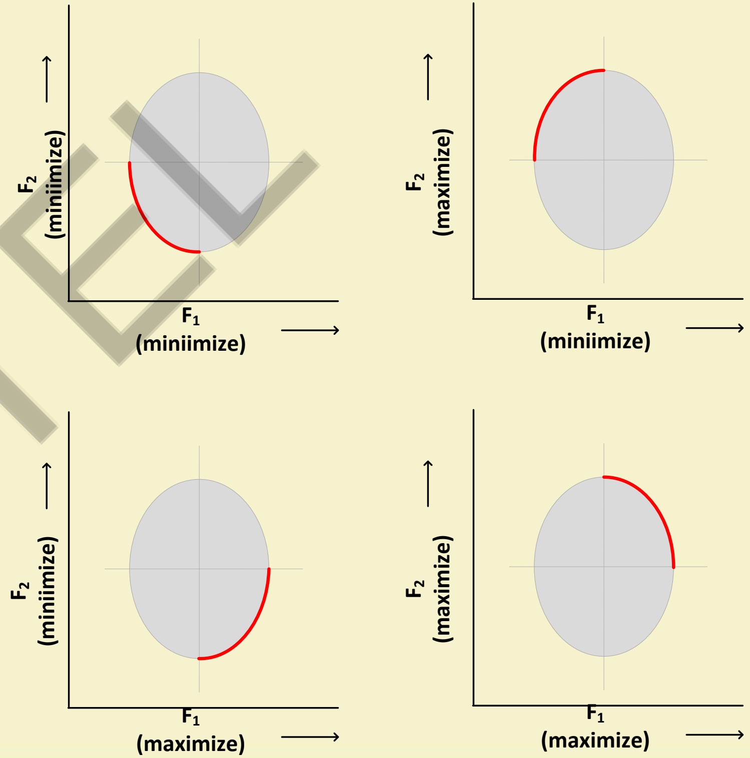
When the set P is the entire search space, that is $P = S$, the resulting non-dominated set P' is called the Pareto-optimal set.



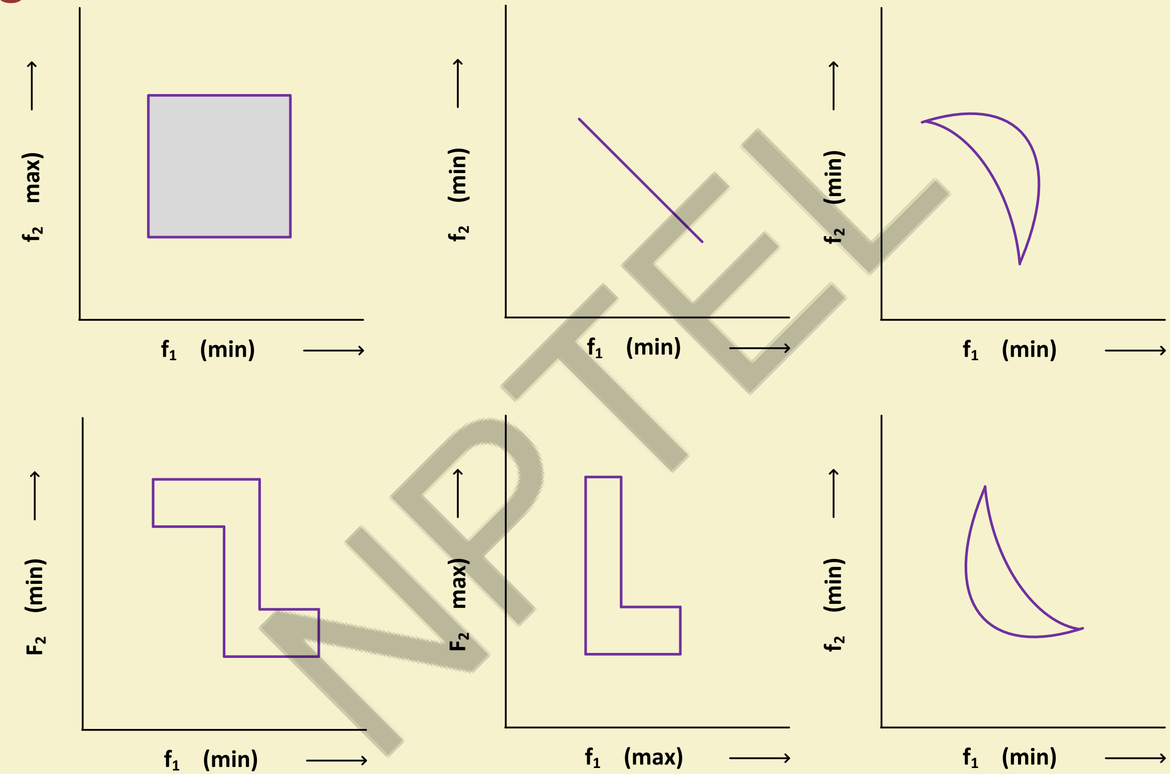
Examples: Pareto optimal sets

Following figures shows the Pareto optimal set for a set of feasible solutions over an entire search space under four different situations with two objective functions F_1 and F_2.

In visual representation, all Pareto optimal solutions lie on a front called Pareto optimal front, or simply, Pareto front.

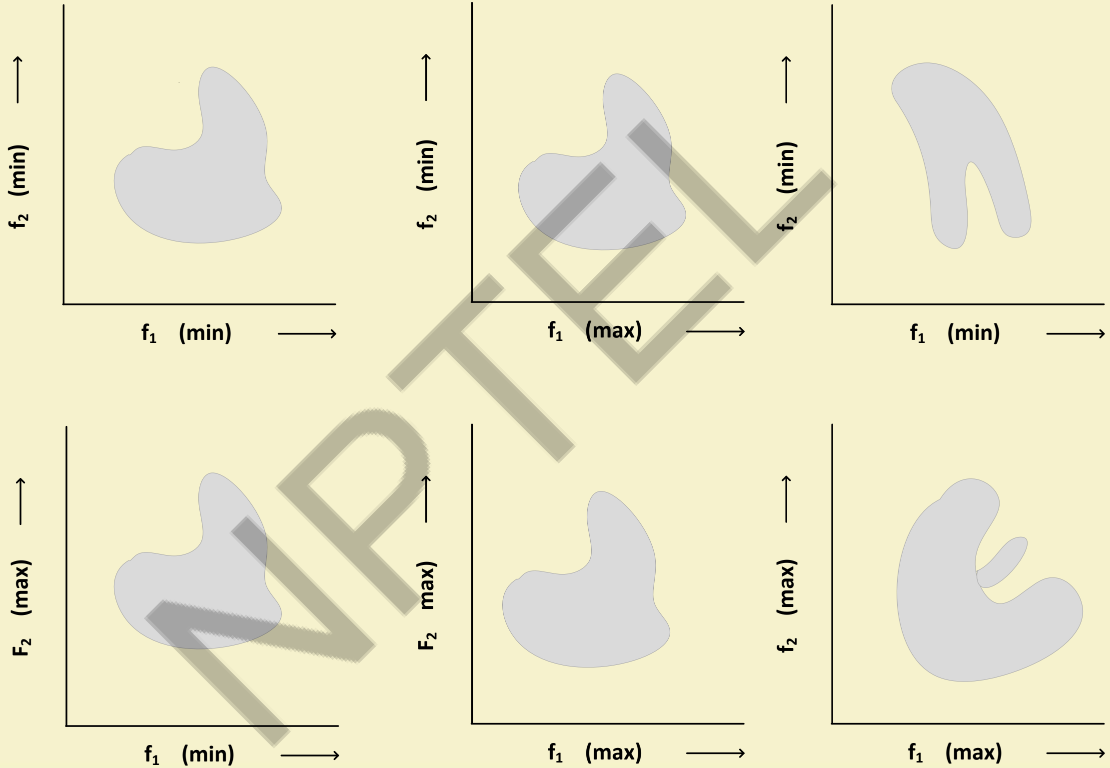


Examples



f_2 max)
f_1 (min)
f_2 (min)
f_1 (min)
f_2 (min)
f_1 (min)
F_2 (min)
f_1 (min)
F_2 max)
f_1 (max)
f_2 (min)
f_1 (min)

Examples



f_2 (min)
f_1 (min)
f_2 (min)
f_1 (max)
f_2 (min)
f_1 (min)
F_2 (max)
f_1 (min)
F_2 max)
f_1 (max)
f_2 (max)
f_1 (max)

Few good articles to read

1) *"An Updated Survey of GA Based Multi-objective Optimization Techniques"* by Carles A Coello Coello, ACM Computing Surveys, No.2,Vol. 32, June 2000

2) *"Comparison of Multi-objective Evolutionary Algorithm : Empirical Result"* by E. Zitzler, K.Deb, Lother Thiele, IEEE Transaction of Evolutionary Computation, No.2, Vol.8, Year 2000.

Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Introduction to Soft Computing

Approaches to Solve MOOPs

Prof. Debasis Samanta

Department of Computer Science & Engineering

IIT Kharagpur

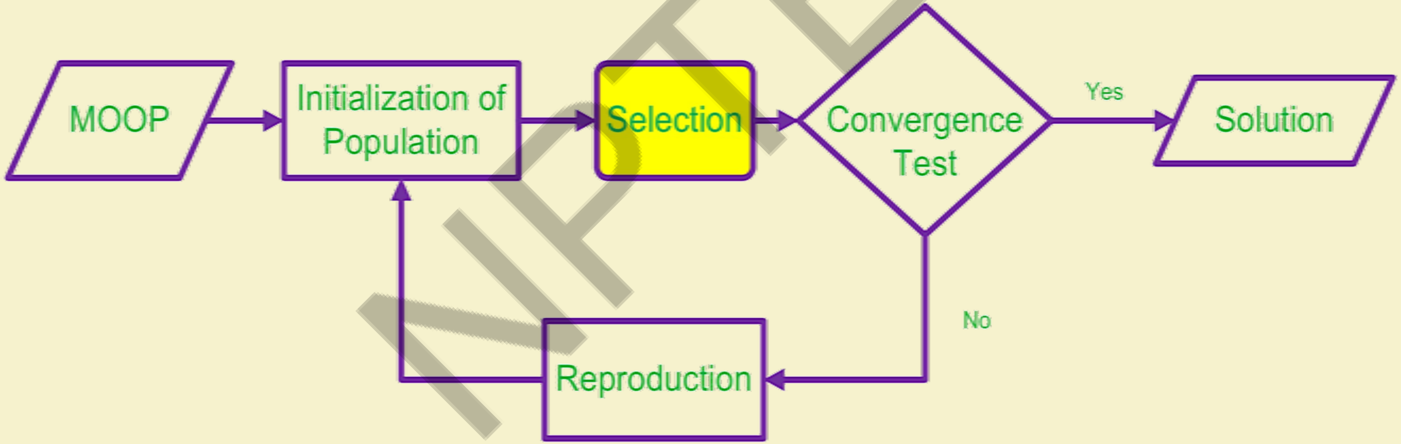
Multi-objective evolutionary algorithm

- To distinguish the GA to solve single objective optimization problems to that of MOOPs, a new terminology called Evolutionary Algorithm (EA) has been coined.
- In many research articles, it is popularly abbreviated as MOEA, the short form of **M**ulti-**O**bjective **E**volutionary **A**lgorithm.

Debasis Samanta
CSE
IIT Kharagpur

Multi-objective evolutionary algorithm

- The following is the MOEA framework, where *Reproduction* is same as in GA but different strategies are followed in *Selection*.



Debasis Samanta
CSE
IIT Kharagpur

Difference between GA and MOEA

- Difference between GA and MOEA are lying in input (single objective vs. multiple objectives) and output (single solution vs. trade-off solutions, also called Pareto-optimal solutions).
- Two major problems are handled in MOEA
    - How to accomplish fitness assignment (evaluation) and selection thereafter in order to guide the search toward the Pareto optimal set.
    - How to maintain a diverse population in order to prevent premature convergence and achieve a well distributed Pareto-optimal front.

Classification of MOEA techniques

- MOEA Techniques
  - A priori approach
    - Aggregation (Ordering)
      - Lexicographic ordering
    - Aggregation (Scalarization)
      - Linear fitness evaluation (SOEA)
      - Non-linear fitness evaluation (SOEA)
      - Goal attainment
      - Weighted Min-max method
      - Game theory
  - A posteriori approach
    - Independent sampling
    - Hybrid Selection
    - Criterion selection (VEGA)
    - Pareto Selection
      - Ranking
      - Ranking and Niching
      - Demes
      - Elitist

Classification of MOEA techniques

Note :

- A priory technique requires a knowledge to define the relative importance of objectives prior to search.
- A posteriori technique searches for Pareto-optimal solutions from a set of feasible solutions.

MOEA techniques to be discussed

1) A priori approaches
    - Lexicographic ordering
    - Simple weighted approach (SOEA)

2) A posteriori approaches
    - Criterion selection (VEGA)
    - Pareto-based approaches
        - ✓ Rank-based approach (MOGA)
        - ✓ Rank + Niche based approach (NPGA)
        - ✓ Non-dominated sorting based approach (NSGA)
        - ✓ Elitist non-dominated sorting based approach (NSGA-II)

MOEA techniques to be discussed

- ✓ **Non-Pareto based approaches**
  - Lexicographic ordering
  - Simple weighted approach (SOEA)
  - Criterion selection (VEGA)
- ✓ **Pareto-based approaches**
  - Rank-based approach (MOGA)
  - Rank + Niche based approach (NPGA)
  - Non-dominated sorting based approach (NSGA)
  - Elitist non-dominated sorting based approach (NSGA-II)

Lexicographic Ordering

Lexicographic ordering method

Reference :

"Compaction of Symbolic Layout using Genetic Algorithms" by M.P Fourman in Proceedings of 1st International Conference on Genetic Algorithms, Pages 141-153, 1985.

- ✓ It is an a priori technique based on the principle of "aggregation with ordering".

Lexicographic ordering method

Suppose, a MOOP with k objectives and n constraints over a decision space x and is denoted as.

Minimize

$$f = [f_1, f_2, \cdots, fk]$$

Subject to

$$g_j(x) \leq c_j, \text{ where } j = 1,2,\cdots,n$$

1) Objectives are ranked in the order of their importance (done by the programmer). Suppose, the objectives are arranged in the following order.

$$f = [f_1 < f_2 < f_3 < \cdots < f_k]$$

Here, $f_i < f_j$ implies f_i is of higher importance than f_j

Lexicographic ordering method

2) The optimum solution $\bar{x}^*$ is then obtained by minimizing each objective function at a time, which is as follows.

(a) **Minimize** $f_1(x)$

Subject to $g_j(x) \le c_j, \quad j = 1,2,\cdots,n$

Let its solution be $\bar{x}_1^*$, that is $f_1^* = f_1(\bar{x}_1^*)$

(b) **Minimize** $f_2(x)$

Subject to $g_j(x) \le c_j, \quad j = 1,2,\cdots,n$

$f_1(x) = f_1^*$

Let its solution be $\bar{x}_2^*$, that is $f_2^* = f_2(\bar{x}_2^*)$

....

....

Lexicographic ordering method

....

....

(c) At the i-th step, we have

Minimize $f_i(x)$

Subject to $g_j(x) \le c_j, \quad j = 1,2,\cdots,n$

$$f_l(x) = f_l^*, \quad l = 1,2,\ldots,i-1$$

Lexicographic ordering method

This procedure is repeated until all k objectives have been considered in the order of their importance.

The solution obtained at the end is $\bar{x}_k^*$, that is $f_k^* = f_k(\bar{x}_k^*)$. This is taken as the desired solution $\bar{x}^*$ of the given multi-objective optimization problem.

Remarks on Lexicographic ordering method

Remarks :

- Deciding priorities (i.e. ranks) of objective functions is an issue. Solution may vary if a different ordering is taken.
- Different strategies can be followed to address the above issues.
  1) Random selection of an objective function at each run.
  2) Naive approach to try with $k!$ number of orderings of k objective functions and then selecting the best observed result.

Note :

It produces a single solution rather than a set of Pareto-optimal solutions.

Single Objective Evolutionary Algorithm

SOEA: Single-Objective Evolutionary Algorithm

- This is an a priori technique based on the principle of ”linear aggregation of functions”.
- It is also alternatively termed as (SOEA) ”**S**ingle **O**bjective **E**volutionary **A**lgorithm”.
- In many literature, this is also termed as Weighted sum approach.
- In fact, it is a naive approach to solve a MOOP.

SOEA approach to solve MOOPs

- This method consists of adding all the objective functions together using different weighting coefficients for each objective.
- This means that our multi-objective optimization problem is transformed into a scalar optimization problem.
  - In other words, in order to optimize say n objective functions $f_1, f_2, \cdots, f_n$. It compute fitness using

$$fitness = \sum_{i=1}^{n} w_i \times f_i(x)$$

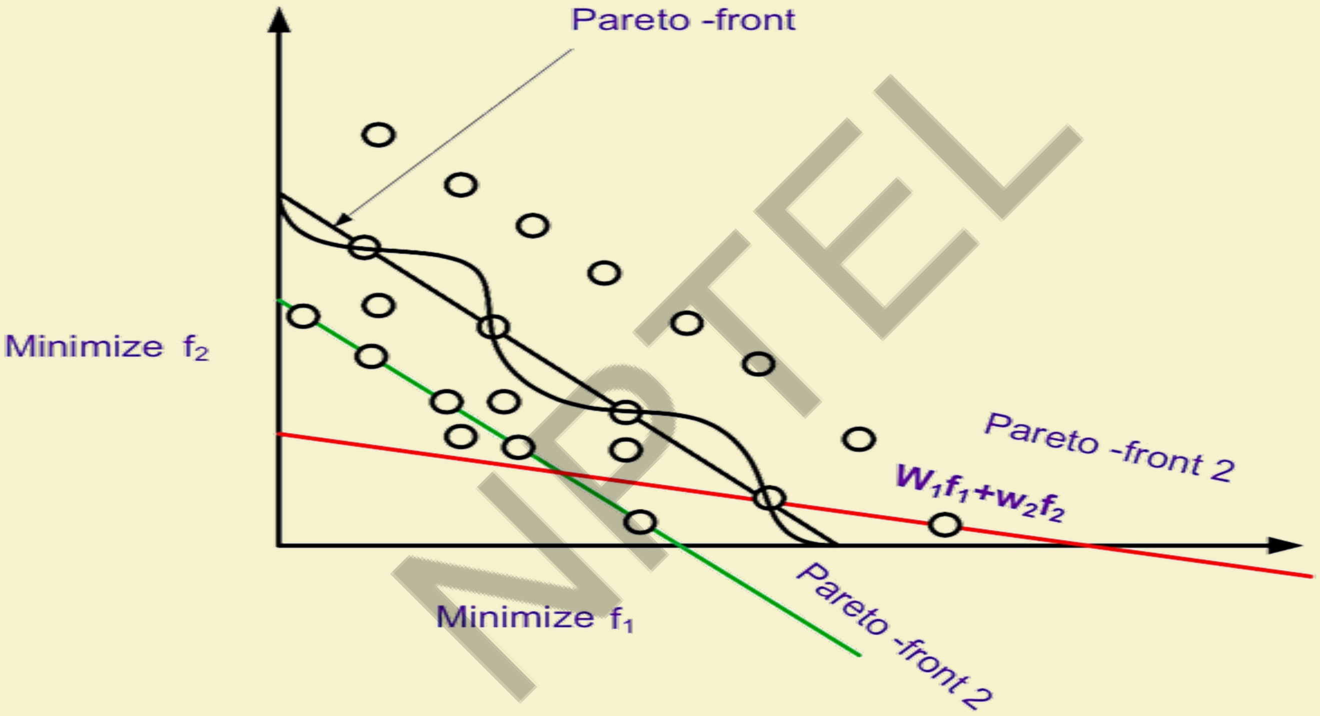
where $w_i \geq 0$ for each $i = 1, 2, \ldots n$ are the weighting coefficients representing the relative importance of the objectives. It is usually assume that

$$\sum_{i=1}^{n} w_i = 1$$

Comments on SOEA

- This is the simplest approach and works in the same framework of Simple GA.
- The results of solving an optimization problem can vary significantly as the weighting coefficient changes.
- In other words, it produces different solutions with different values of w_i 's.
- Since very little is usually known about how to choose these coefficients, it may result into a local optima.

Local optimum solution in SOEA



Comments on SOEA

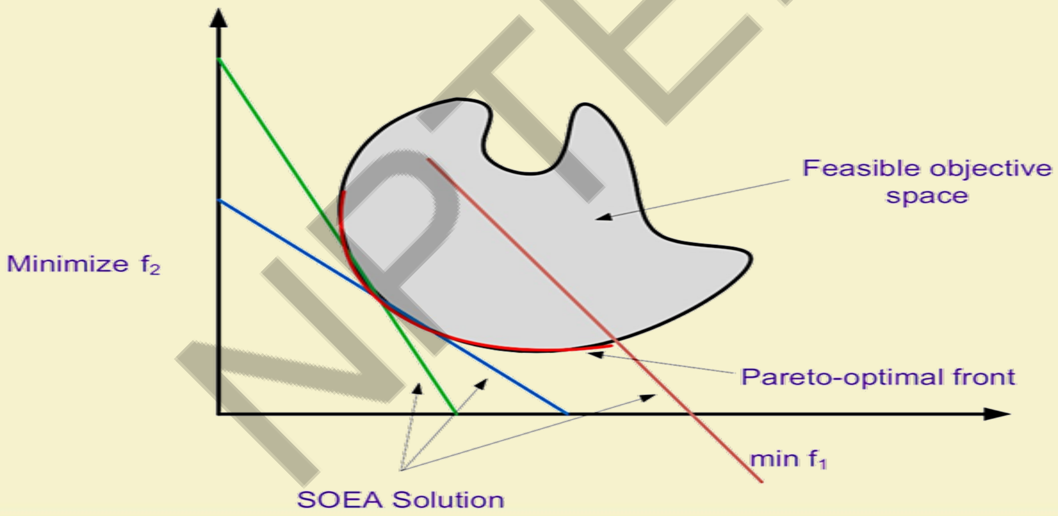
- As a way out of this, it is necessary to solve the same problem for many different values of w_i's.
- The weighting coefficients do not proportionally reflects the relative importance of the objectives, but are only factors, which, when varied, locate points in the Pareto set.
- This method depends on not only w_i's values but also on the units in which functions are expressed.
- In that case, we have to scale the objective values. that is

$$fitness = \sum_{i=1}^{n} w_i \times f_i(x) \times c_i$$

where c_i's are constant multipliers that scales the objectives properly.

Naive Approach : Weighted sum approach

- The technique cannot be used to find Pareto-optimal solutions which lie on the convex portion of the Pareto optimal front. In that case, it gives only one solution, which might be on the Pareto front.



Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Introduction to Soft Computing

Non-Pareto based approaches to solve MOOPs

Prof. Debases Samanta

Department of Computer Science & Engineeering

IIT Kharagpur

MOEA techniques to be discussed

- ✓ **Non-Pareto based approaches**
  - Lexicographic ordering
  - Simple weighted approach (SOEA)
  - **Criterion selection (VEGA)**
- **Pareto-based approaches**
  - Rank-based approach (MOGA)
  - Rank + Niche based approach (NPGA)
  - Non-dominated sorting based approach (NSGA)
  - Elitist non-dominated sorting based approach (NSGA-II)

Debasis Samanta
CSE
IIT Kharagpur

Vector Evaluated Genetic Algorithm

Vector Evaluated Genetic Algorithm (VEGA)

- Proposed by David Schaffer (1985) in

”Multiple objective optimization with vector evaluated genetic algorithm - Genetic algorithm and their application”: Proceeding of the first international conference on Genetic algorithm, 93-100, 1985.

- It is normally considered as the first implementation of a MOEA.
- VEGA is a posteriori technique based on the principle of Criterion selection strategy.

Vector Evaluated Genetic Algorithm (VEGA)

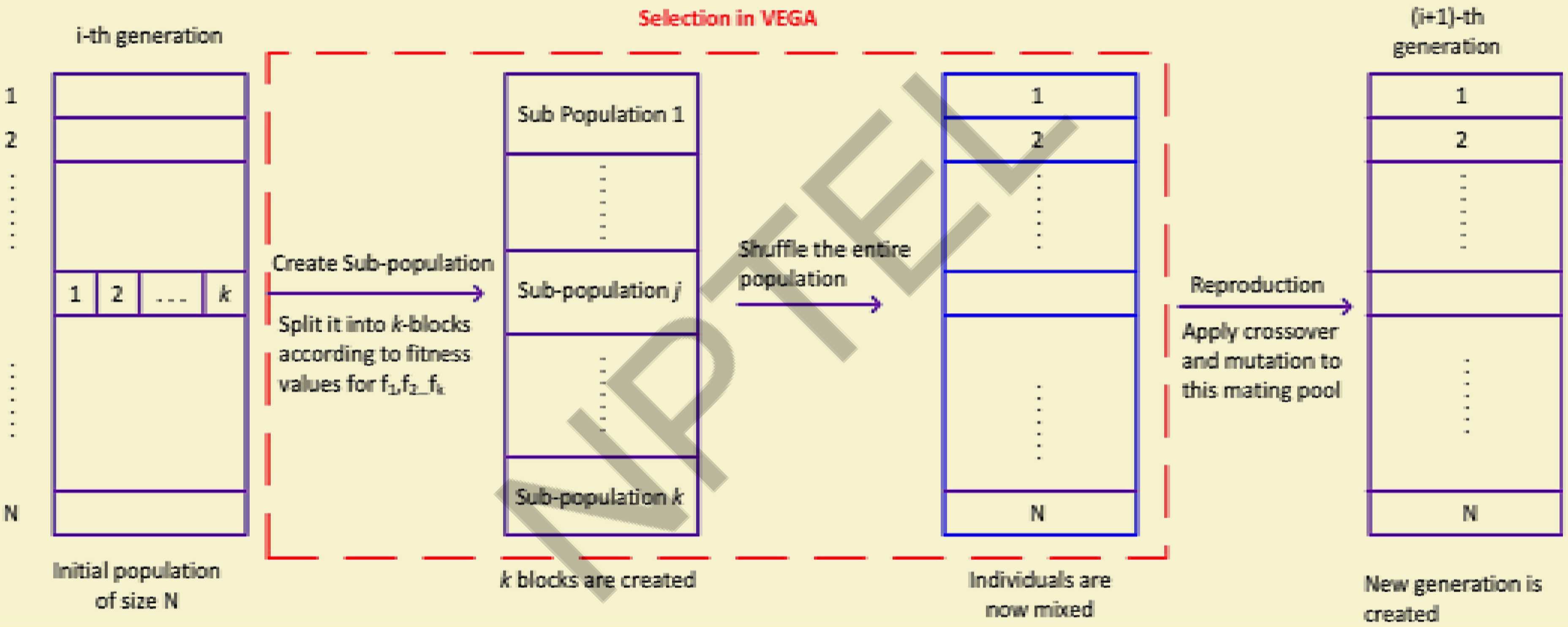
About VEGA :

- It is an extension of Simple Genetic Algorithm (SGA).
- It is an example of a criterion (or objective) selection technique where a fraction of each succeeding population is selected based on separate objective performance. The specific objective for each fraction are randomly selected at each generation.
- VEGA differs SGA in the way in which the selection operation is performed.

Basic steps in VEGA

1) Suppose, given a MOOP is to optimize k objective functions $f_1, f_2, \cdots, f_k$.
2) A number of sub-population is selected according to each objective function in turn.
3) Thus, k-sub-populations each of size $\frac{M}{k}$ are selected, where M is the size of the mating pool ($M \leq N$), and N is the size of the input population.
4) These sub-population are shuffled together to obtain a new ordering of individuals.
5) Apply standard GA operations related to reproduction.
6) This produced next generation and Steps 2-5 continue until the termination condition is reached.

Overview of the VEGA



VEGA selection strategy

VEGA consists of the following three major steps:

1) Creating k sub-populations each of size $\frac{M}{k}$
2) Shuffle the sub-populations
3) Reproduction of offspring for next generation (same as in SGA)

We explain the above steps with the following consideration:

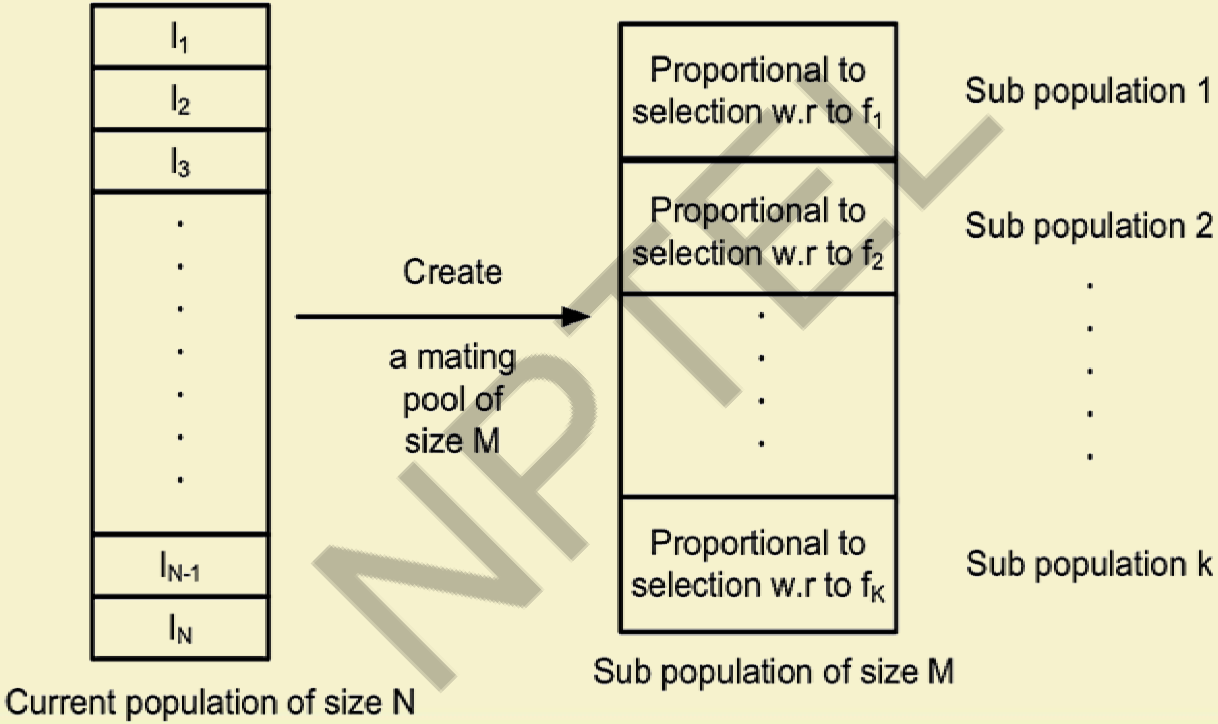
- ✓ Suppose, given a MOOP, where we are to optimize k number of objective functions $f = f_1, f_2, \cdots, f_k$.
- ✓ Given the population size as N with individual $I_1, I_2, \cdots, I_N$.
- ✓ We are to create a mating pool of size M, where $(M \leq N)$.

VEGA: Creation of sub-populations

- **Create a mating pool of size M ($M \le N$)**

  Generate i-th subpopulation of size $\frac{M}{k}$ where $i = 1,2,\cdots,k$. To do this follow the proportional selection strategy (such as Roulette-wheel selection) according to the i-th objective function only at a time.

VEGA: Creation of sub-populations

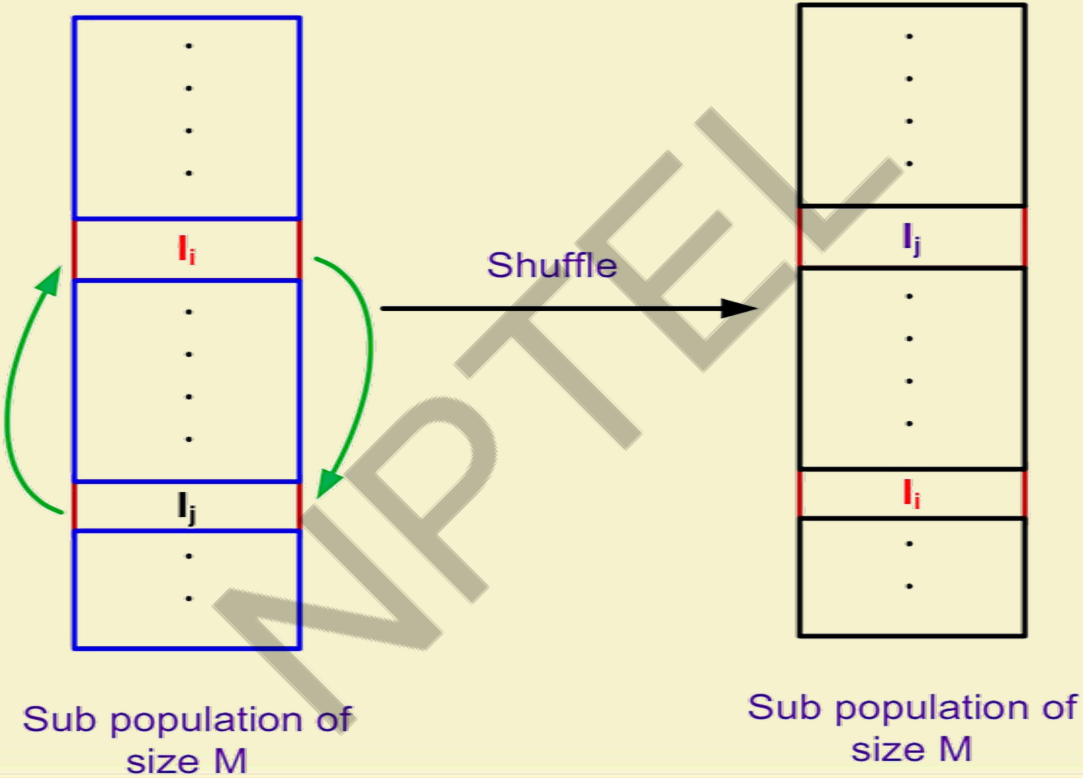


VEGA: Shuffle the sub-populations

- **Shuffle the sub-populations**

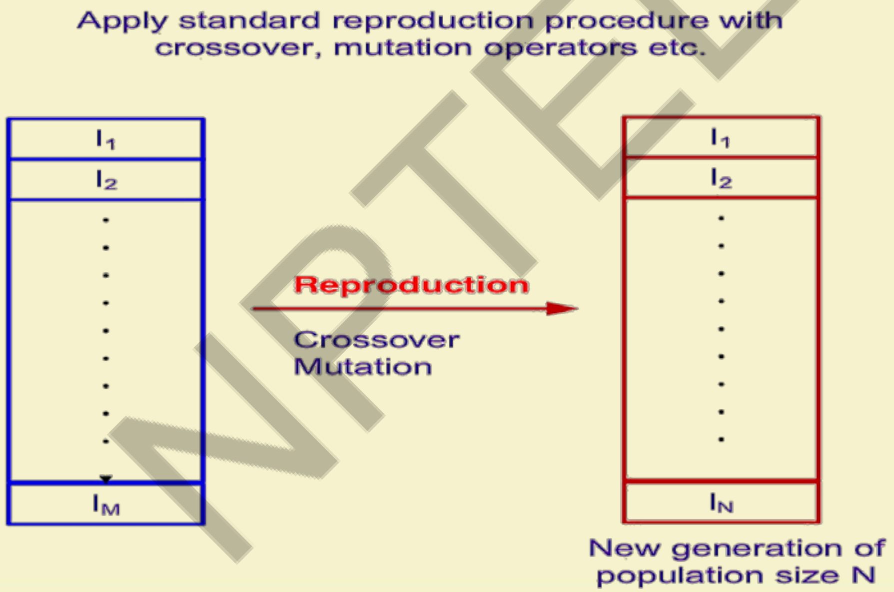
  Using some shuffling operation (e.g., generate two random numbers i and j) between 1 and M both inclusive and then swap I_i and I_j which are in the i and j sub-populations.

VEGA: Shuffle the sub-populations



VEGA: Reproduction

- **Reproduction:** Perform reproduction to produce new generation of population size N.



Debasis Samanta
CSE
IIT Kharagpur

Comments on VEGA

Advantages:

- VEGA can be implemented in the same framework as SGA (only with a modification of selection operation).
- VEGA can be viewed as optimizing $f_1, f_2, \cdots, f_k$ simultaneously. That is, $f(x) = \hat{e}_1 f_1(x) + \hat{e}_2 f_2(x) + \cdots + \hat{e}_k f_k(x)$ where e_i is the i-th vector.
  - Thus, VEGA is a generalization from scalar genetic algorithm to vector evaluated genetic algorithm (and hence its name!).
- VEGA leads to a solution close to local optima with regard to each individual objective.

Comments on VEGA

Disadvantages:

- The solutions generated by VEGA are locally non-dominated but not necessarily globally dominated. This is because their non-dominance are limited to the current population only.
- “Speciation” problem in VEGA : It involves the evolution of ”Species” within the population (which excel on different objectives).
- This is so because VEGA selects individuals who excel in one objective, without looking at the others.
- This leads to “middling” performance (i.e., an individual with acceptable performance, perhaps above average, but not outstanding for any of the objective function.

MOEA strategies

- MOEA Solution Techniques
  - **A priori approach**
    - Lexicographic ordering
    - Game theory approach
    - Non-linear fitness evaluation
    - SOEA
    - Min-Max method
  - **A posteriori approach**
    - Independent sampling
    - Aggregate Selection
    - Criterion selection (VEGA)
    - Pareto selection
      - Ranking (MOGA)
      - Ranking and Niching
      - Demes
      - Elitist

Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Introduction to Soft Computing

Pareto-based approaches to solve MOOPs

Prof. Debases Samanta

Department of Computer Science & Engineeering

IIT Kharagpur

MOEA strategies

- MOEA Solution Techniques
  - A priori approach
    - Lexicographic ordering
    - Game theory approach
    - Non-linear fitness evaluation
    - SOEA
    - Min-Max method
  - A posteriori approach
    - Independent sampling
    - Aggregate Selection
    - Criterion selection (VEGA)
    - Pareto selection
      - Ranking (MOGA)
      - Ranking and Niching
      - Demes
      - Elitist

MOGA : Multi-Objective Genetic Algorithm

MOGA : Multi-Objective Genetic Algorithm

- It is Pareto-based approach based on the principle of ranking mechanism proposed by Carlos M. Fonseca and Peter J. Fleming (1993).

Reference :

C. M. Fonseca and P. J. Fleming, ”Genetic Algorithm for multi-objective Optimization : Formulation, Discussion and Generalization” in Proceeding of the 5*th* International Conference on Genetic Algorithm, Page 416-423, 1993.

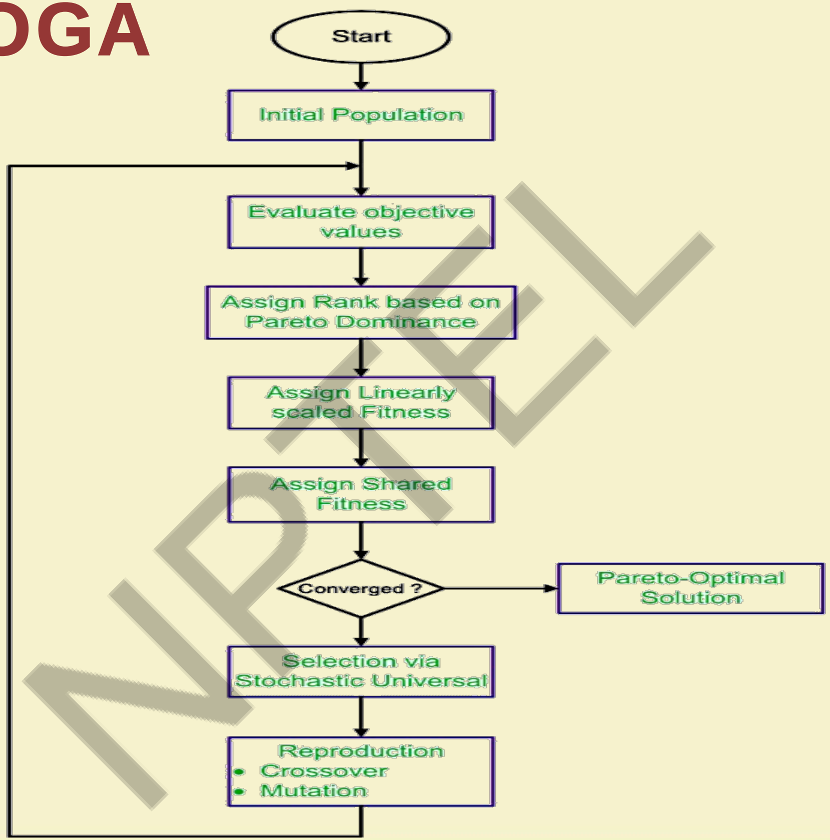
MOGA : Multi-Objective Genetic Algorithm

- Regarding the "generation" and "selection" of the Pareto-optimal set, ordering and scaling techniques are required.

- MOGA follows the following methodologies:

  For ordering: Dominance-based ranking,
  For scaling: Linearized fitness assignment and fitness averaging.

Flowchart of MOGA



Dominance-based ranking

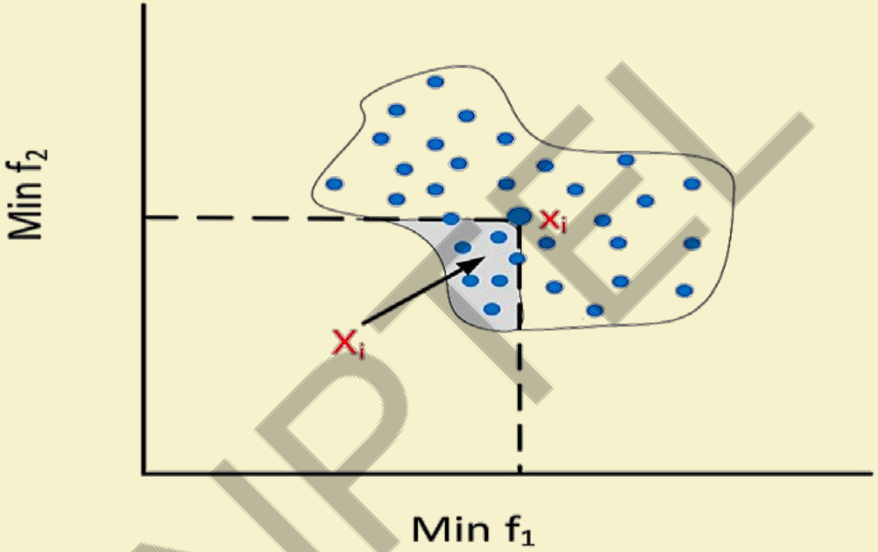
Definition : Rank of a solution

The rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated.

More formally,

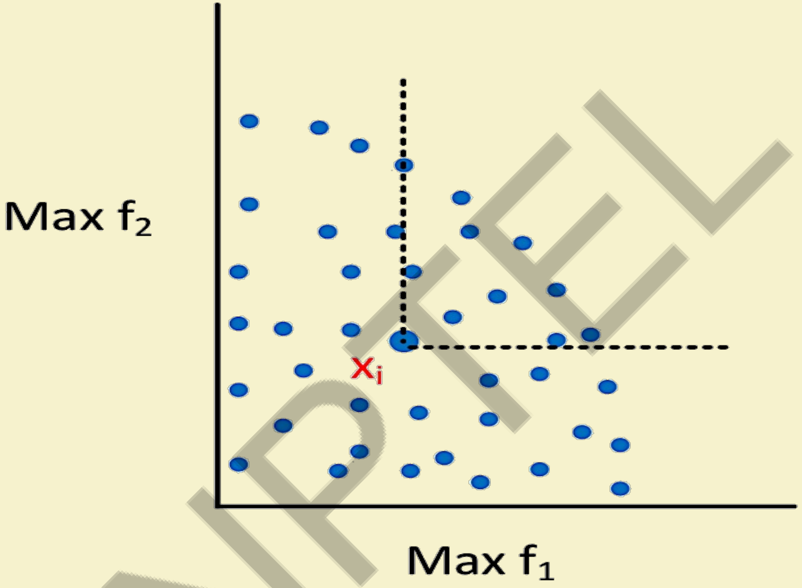
If an individual x_i is dominated by p_i individuals in the current generation, then $rank(x_i) = 1 + p_i$

Example 1: Dominance-based ranking



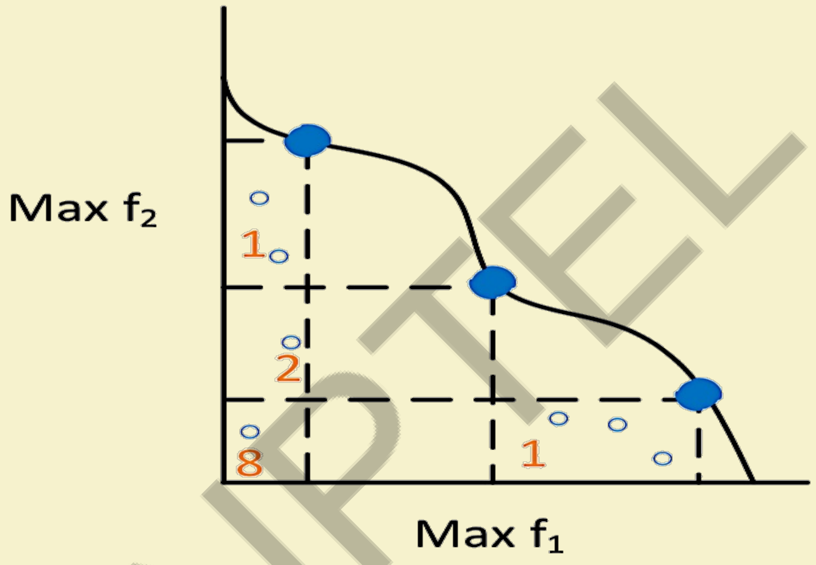
Rank(x_1)=1+|x_i|
Where |x_i| = number of solutions in the shaded region

Example 2: Dominance-based ranking



Rank(x_1)=1+11=12

Example 3: Dominance-based ranking



Number of dominated points
with their domination count

Interpretation : Dominance-based ranking

Note :

1) Domination count = How many individual does an individual dominates.
2) All non-dominated individuals are assigned rank 1.
3) All dominated individuals are penalized according to the population density of the corresponding region of the trade-off surface.

Fitness Assignment in MOGA

Steps :

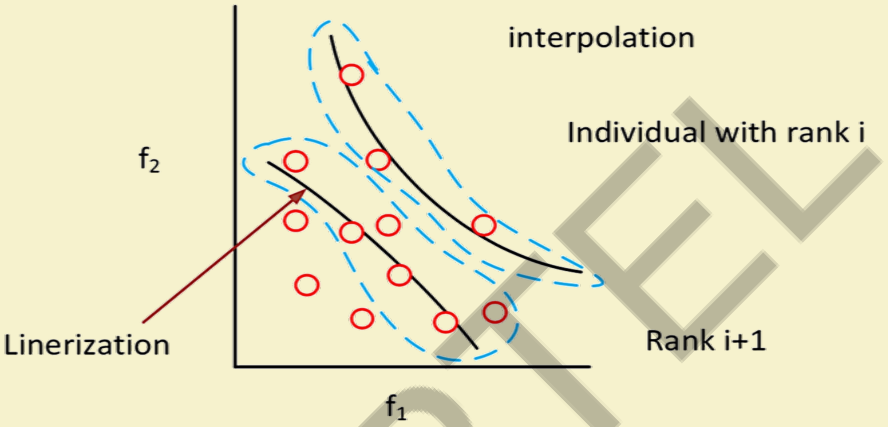
1) Sort the population in ascending order according to their ranks.

2) Assign fitness to individuals by interpolating the best (rank 1) to the worst ($rank \leq N$, N being the population size) according to some linear function.

3) Average the fitness of individual with the same rank, so that all of them are sampled at the same rate.

This procedure keeps the global population fitness constant while maintaining appropriate selective pressure, as defined by the function used.

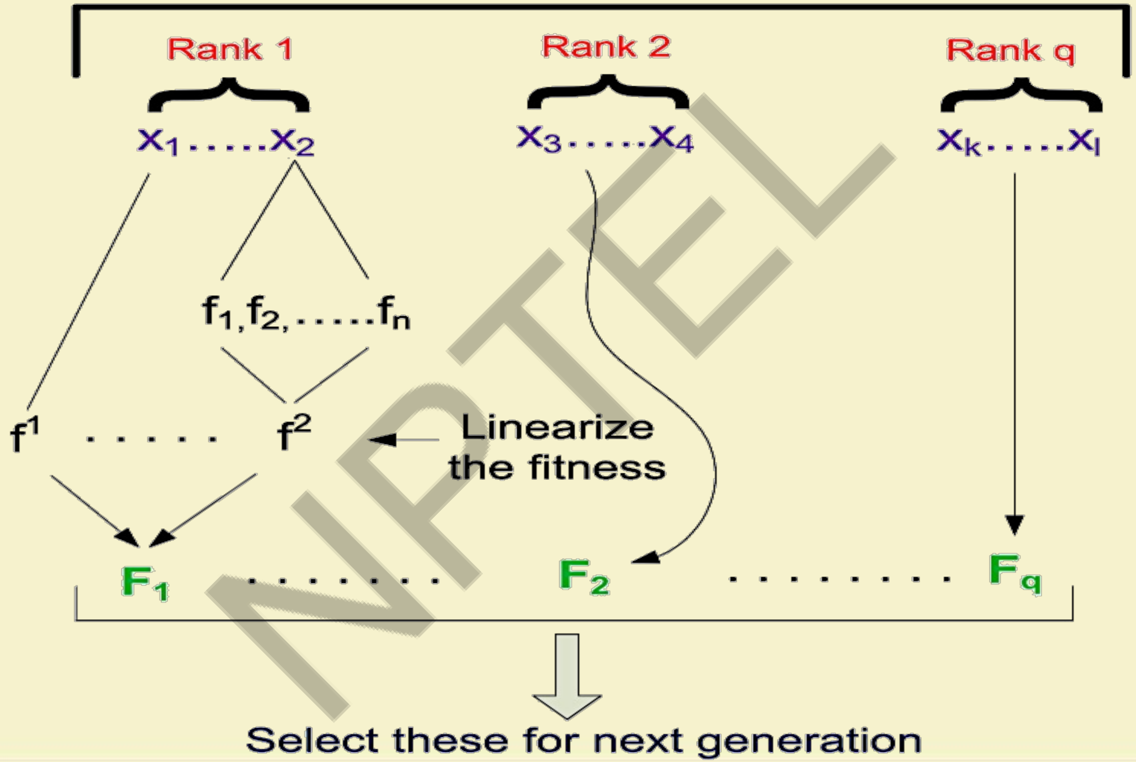
Fitness Assignment in MOGA



Example : $Linearization = \bar{f_i} = \sum_{j=1}^{k} \frac{f_j^i}{\bar{f_j^i}}$

where f_j^i denotes the j-th objective function of a solution in the i-th rank and $\bar{f_j^i}$ denotes the average value of the j-th objectives of all the solutions in the i-th rank.

Illustration of MOGA



Remarks on MOGA

- The fitness assignment (Step 3) in MOGA attempts to keep global population fitness constant while maintaining appropriate selection pressure.
- MOGA follows blocked fitness assignment which is likely to produce a large selection pressure that might lead to premature convergence.
- MOGA founds to produce better result (near optimal) in majority of MOOPs.

Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Introduction to Soft Computing

Pareto-based approaches to solve MOOPs

Prof. Debasis Samanta

Department of Computer Science & Engineering

IIT Kharagpur

MOEA techniques to be discussed

- ✓ **Non-Pareto based approaches**
  - Lexicographic ordering
  - Simple weighted approach (SOEA)
  - Criterion selection (VEGA)
- ✓ **Pareto-based approaches**
  - Rank-based approach (MOGA)
  - **Rank + Niche based approach (NPGA)**
  - **Non-dominated sorting based approach (NSGA)**
  - **Elitist non-dominated sorting based approach (NSGA-II)**

Niched Pareto Genetic Algorithm (NPGA)

Niched Pareto Genetic Algorithm (NPGA)

- J. Horn and N. Nafploitis, 1993 **Reference :** *Multiobjective Optimization using the Niched Pareto Genetic Algorithm* by J. Horn and N. Nafpliotis, Technical Report, University of Illionis at Urbans-Champaign, Urbana, Illionis, USA, 1993

Niched Pareto Genetic Algorithm (NPGA)

- NPGA is based on the concept of tournament selection scheme (based on Pareto dominance principle).
- In this techniques, first two individuals are randomly selected for tournament.
- To find the winner solution, a comparison set that contains a number of other individuals in the population is randomly selected.

Niched Pareto Genetic Algorithm (NPGA)

- Then the dominance of both candidates with respect to the comparison set is tested.
- If one candidate only dominates the comparison set, then the candidate is selected as the winner.
- Otherwise, niched sharing is followed to decide the winner candidate.

Niched Pareto Genetic Algorithm (NPGA)

Pareto-domination tournament

Let N = size of the population, K is the no of objective functions.

Steps :

1) $i = 1$ (The first iteration)
2) Randomly select any two candidates C_1 and C_2
3) Randomly select a "Comparison Set (CS)" of individuals from the current population.

   Let its size be N^* (Where $N^* = P\%N$; P decided by the programmer)

4) Check the dominance of C_1 and C_2 against each individual in CS

Niched Pareto Genetic Algorithm (NPGA)

4) If C_1 is dominated by CS but not by C_2 than select C_2 as the winner

   Else if C_2 is dominated by CS but not C_1 than select C_1 as the winner

   Otherwise Neither C_1 nor C_2 dominated by CS **do_sharing (C_1, C_2)** and choose the winner.

5) If $i = N'$ than exit (Selection is done)

   Else $i = i + 1$, go to step 2

Niched Pareto Genetic Algorithm (NPGA)

- A sharing is followed, when there is no preference in the candidates.
- This maintains the genetic diversity allows to develop a reasonable representation of Pareto-optimal front.
- The basic idea behind sharing is that the more individuals are located in the neighborhood of a certain individual, the more its fitness value is degraded.
- The sharing procedure for any candidate is as follows.

DEBASIS SAMANTA
DEPARTMENT OF CSE
IIT KHARAGPUR

Niched Pareto Genetic Algorithm (NPGA)

Procedure do_sharing(C_1, C_2)

1) $j = 1$. Let $x = C_1$
2) Compute a normalized (Euclidean distance) measure with the individual x_j in the current population as follows,

$$d_{x_j} = \sqrt{\sum_{i=1}^{k} \left(\frac{f_i^x - f_i^j}{f_i^U - f_i^L} \right)^2}$$

where f_i^j denotes the i-th objective function of the j-th individual.
f_i^U and f_i^L denote the upper and lower values of the i-th objective function.

Niched Pareto Genetic Algorithm (NPGA)

3) Let $\sigma_{share} = Niched\ Radius$
Compute the following sharing value

$$sh\left(d_{x_j}\right) = \begin{cases} 1 - \left(\dfrac{d_{x_j}}{\sigma_{share}}\right)^2, & \text{if } d_{x_j} < \sigma_{share} \\ 0 & , otherwise \end{cases}$$

4) Set $j = j + 1$, if $j < N$, go to step 2 else calculate "Niched Count" for the candidate as follows

$$n_1 = \sum_{j=1}^{N} sh(d_{ij})$$

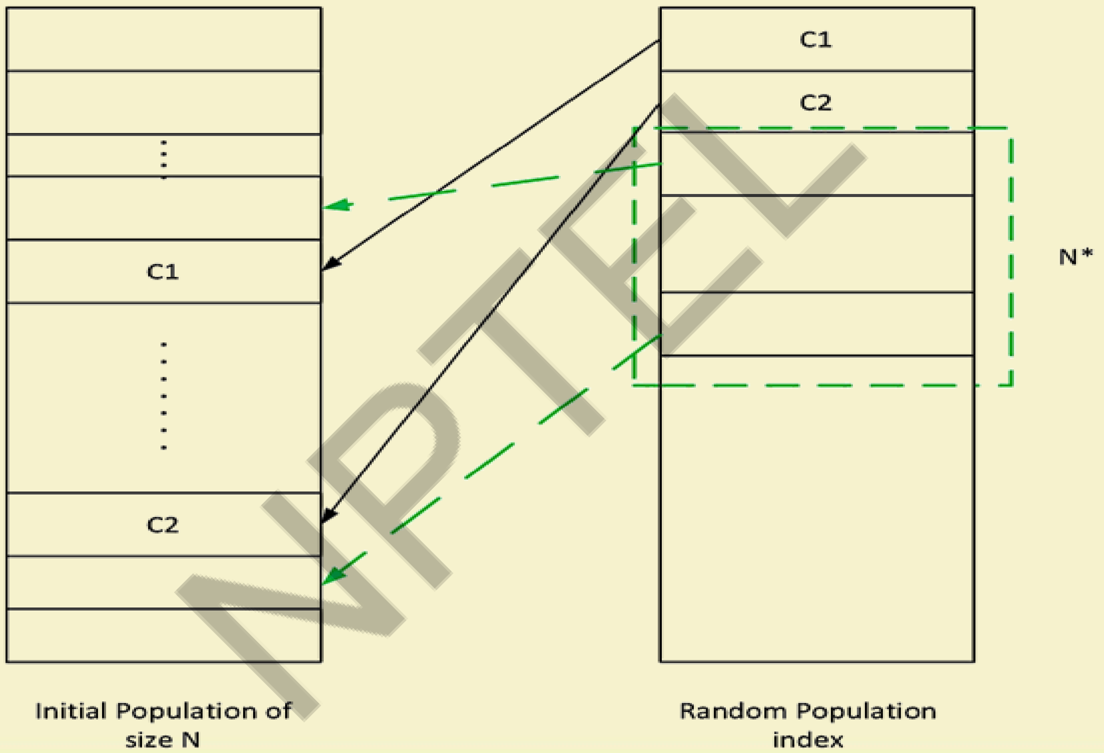
Niched Pareto Genetic Algorithm (NPGA)

5) Repeat step 1-4 for C_2.

   Let the niched count for C_2 be n_2

6) if $n_1 < n_2$ then choose C_2 as the winner else C_1 as the winner.

Niched Pareto Genetic Algorithm (NPGA)



Niched Pareto Genetic Algorithm (NPGA)

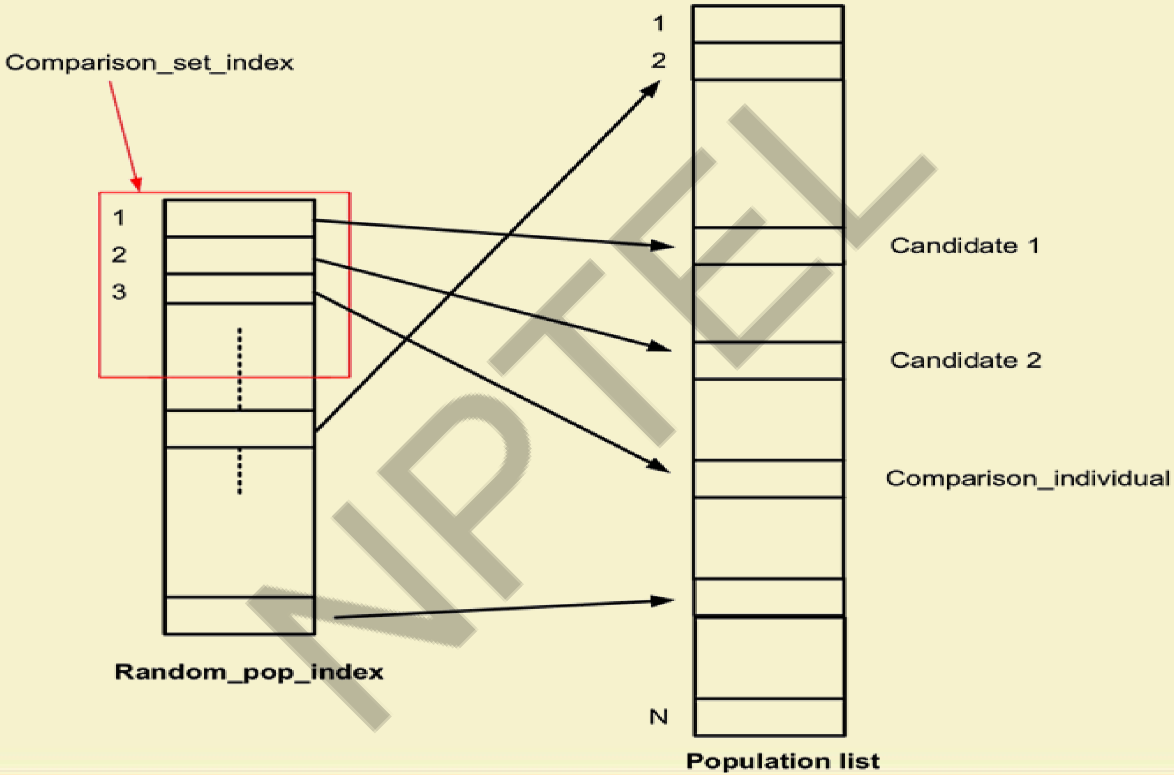
- This approach proposed by Horn and Nafploitis [1993]. The approach is based on tournament scheme and Pareto dominance. In this approach, a comparison was made among a number of individuals (typically 10%) to determine the dominance. When both competitors are dominated or non-dominated (that is, there is a tie) the result of the tournament is decided through fitness sharing (also called equivalent class sharing).

- The pseudo code for Pareto domination tournament assuming that all of the objectives are to be maximized is presented below. Let us consider the following.

Pareto domination Tournament

- S = an array of N individuals in the current population.
- random_pop_index = an array holding N individuals of S, in a random order.
- t_{dom} = the size of the comparison set.

Algorithm Selection



Algorithm Selection

This algorithm returns an individual from the current population S.

Begin

shuffle(*random_pop_index*)

*candidate*_1 = *random_pop_index*[1];

*candidate*_2 = *random_pop_index*[2];

*candidate*_1_*dominated* = *F*;

*candidate*_2_*dominated* = *F*;

for *comparison_set_index* = 3 to t_{dom} + 3 **do**

comparison_individual = *random_pop_index*[*comparison_set_index*];

if *s*[*comparison_set_index*]*dominates*[*candidate*_1] **then**

*candidate*_1_*dominated* = *TRUE*;

end if

Algorithm Selection

if *s*[*comparison_set_index*]*dominates*[*candidate*_2] **then**
*candidate*_2_*dominated* = *TRUE*;
end if
end for
if *candidate*_1_*dominated and* ~*candidate*_2_*dominated* **then**
*return candidate*_1
else if *candidate*_2_*dominated and* ~*candidate*_1_*dominated* **then**
*return candidate*_2

Algorithm Selection

```
else
    if nichecount(candidate_1) > nichecount(candidate_2) then
        return candidate_2
    else
        return candidate_1
    end if
end if
END
```

Algorithm Selection

- ✓ This approach does not apply Pareto selection to the entire population, but only to a segment of it at each num, the technique is very first and produces good non-dominated num that can be kept for a large number of generation.
- ✓ However, besides requiring a sharing factor, this approach also requires a good choice of the value of t_{dom} to perform well, complicating its appropriate use in practice.

Points to Ponder an Multi-objective Evolutionary Algorithm

- ✓ How you solve two optimization problem
  - Strategy 1 : Solve individually
  - Strategy 2 : Solve 1 as main as other as constraint
  - Strategy 3 : $C = C_1 + C_2, \; X = X_1 \cup X_2$

Justify three strategies

- ✓ What are the issues with one minimize and another maximization problem?
- ✓ Explain weighted-sum approach.
- ✓ What are the issues ?
- ✓ How pareto-based approach address this issues ?

Thank You!!