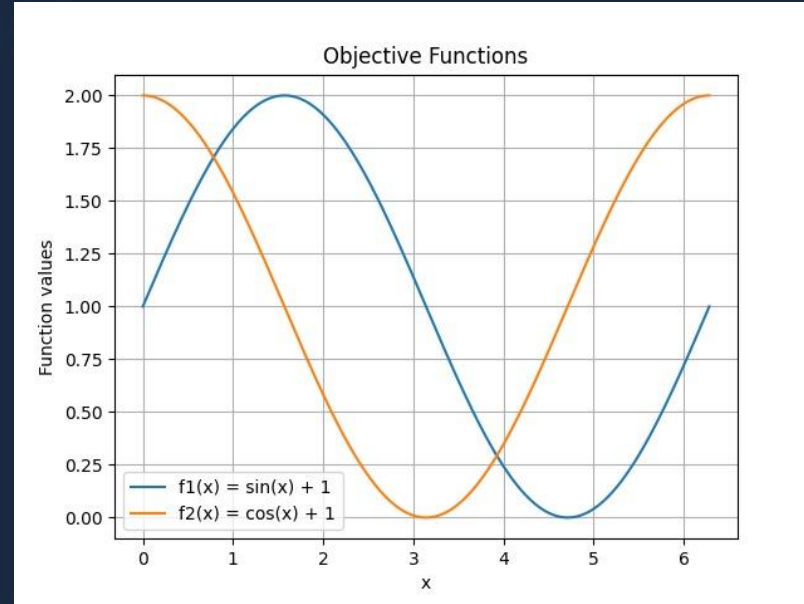


# Problem 1



$$\begin{aligned} \min f1(x) &= \sin(x) + 1 \\ \min f2(x) &= \cos(x) + 1 \\ x &\in [0, 2\pi] \end{aligned}$$

*Utilize the Pareto ranking method for resolution and illustrate with graphs.*



# SPEA2 (Strength Pareto Evolutionary Algorithm 2)

**Steps:**

a) **Initialization** :Generate an initial population  $P_t$  and create the empty archive  $\bar{P}_t$  (external set) . Set  $t = 0$  .

$N$  (population size)

$\bar{N}$  (archive size)

$T$  (maximum number of generations)

**fitness assignment strategy**

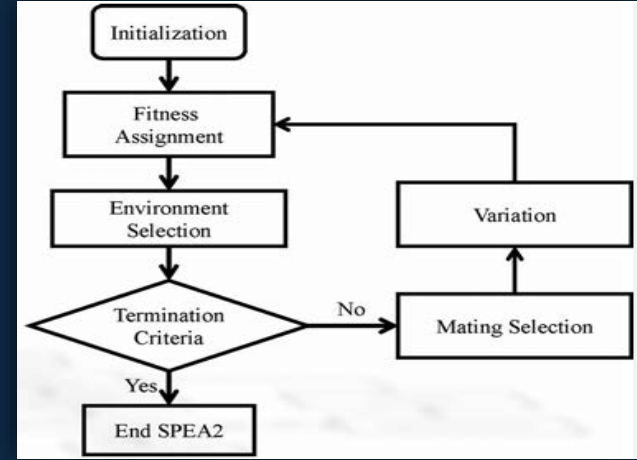
b) For each individual “ $i$ ” in  $P_t$  and  $\bar{P}_t$  is assigned a strength value  $S(i)$

$$S(i) = \frac{n}{N + 1}$$

**Where :** “ $N$ ” is the population size.

“ $n$ ” is the number of individuals that are dominated by or equal to individual “ $i$ ”

**strength pareto** : it has an important role in SPEA because this shows how solutions close to the first rank.



c) Calculates Raw fitness value based on sum of the strength of each “ $i$ ” individual dominated by or equal to individual “ $j$ ”

$$R(i) = \sum_{j \in P_t + \bar{P}_t, j \phi i} S(j)$$

$\phi$  corresponds to the Pareto dominance relation

- d) To introduce variability among the solutions within the same Pareto front we calculate the density “D” corresponding to “i”.

$$D(i) = \frac{1}{\sigma_i^k + 2}$$

Where :  $\sigma_i^k$  is the “k –th” nearest neighbor distance for each “i” individual.

- e) Calculates Fitness value based on the sum of raw fitness + density of each individual “i”.

$$F(i) = R(i) + D(i)$$

### Environmental Selection

- f) Copy all nondominated individuals, i.e., those which have a fitness lower than “1” (**Best individuals**), from archive and population to the archive of the next generation:

$$\bar{P}_{t+1} = \{i \mid i \in P_t + \bar{P}_t \wedge F(i) < 1\}$$

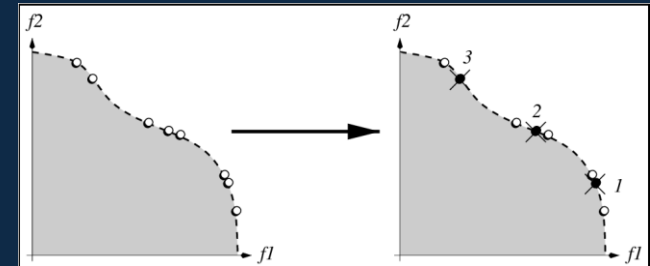
- g) If the non-dominated front fits exactly into the archive environmental selection step is completed.

$$|\bar{P}_{t+1}| = \bar{N}$$

- h) If the archive is too small  $(|\bar{P}_{t+1}| < \bar{N})$  the best  $\bar{N} - |\bar{P}_{t+1}|$  Dominated individuals in the previous archive and population are copied to the new archive.

- i) If the archive is too large  $(|\bar{P}_{t+1}| > \bar{N})$ , an archive truncation procedure is invoked which iteratively removes individuals from  $|\bar{P}_{t+1}|$  until  $|\bar{P}_{t+1}| = \bar{N}$  that is, the individual which has the minimum distance to another individual is chosen to be eliminated.

### Density truncation



# NSGA-II (Non-dominated Sorting Genetic Algorithm II)

## Fast non-dominated Sorting

### Objective:

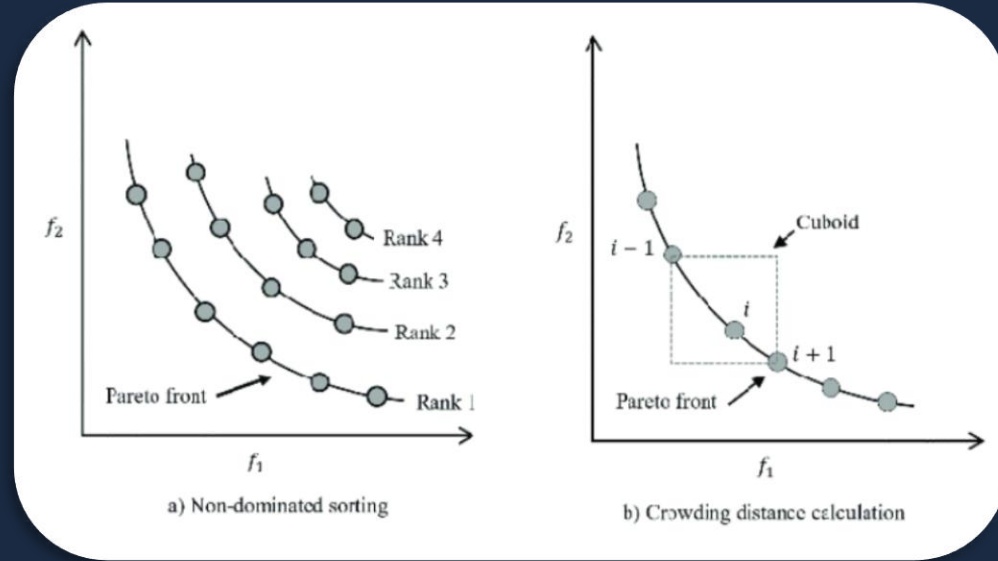
To divide the population into different levels of Pareto fronts based on the *concept of dominance*.

solution A is said to dominate another solution B if:

1. A is no worse than B in all objectives, and
2. A is strictly better than B in at least one objective.

### Strategy:

The population is divided into different fronts. The first front (F1) contains all non-dominated solutions. The second front (F2) contains solutions that are dominated by solutions in F1 but not by each other, and so on.



## Crowding Distance Assignment

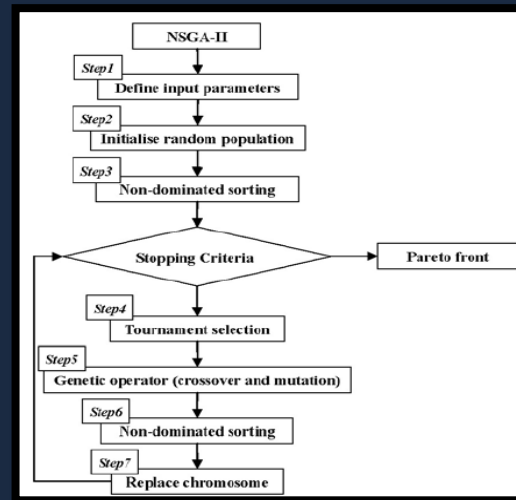
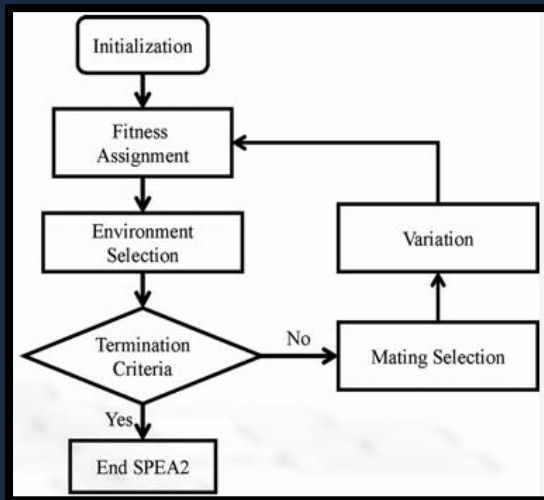
### Objective:

To maintain diversity among solutions within the same Pareto front.

### Strategy:

For each solution in a front, a crowding distance is calculated based on the Euclidean distance between adjacent solutions in the objective space.

# NSGA-II vs. SPEA2



## Principal differences

### Fitness Calculation:

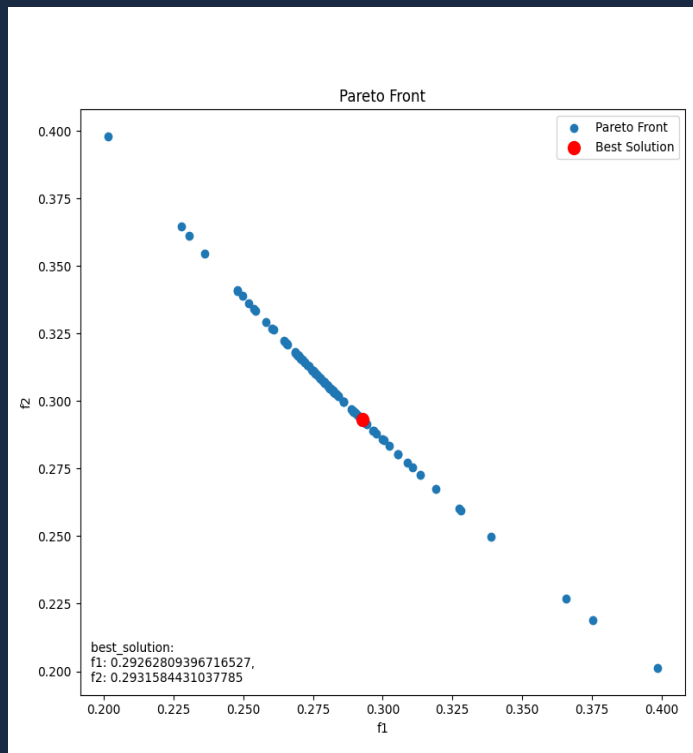
- In NSGA-II, fitness is based on **dominance ranking** and crowding distance.
- In SPEA2, fitness is based on strength, **raw fitness**, and **density**.

### Environmental Selection:

- NSGA-II uses a selection process based on **dominance** and **crowding distance**.
- SPEA2 uses a selection process based on combined **fitness** and **density truncation**.

# Experiments

## NSGA-II



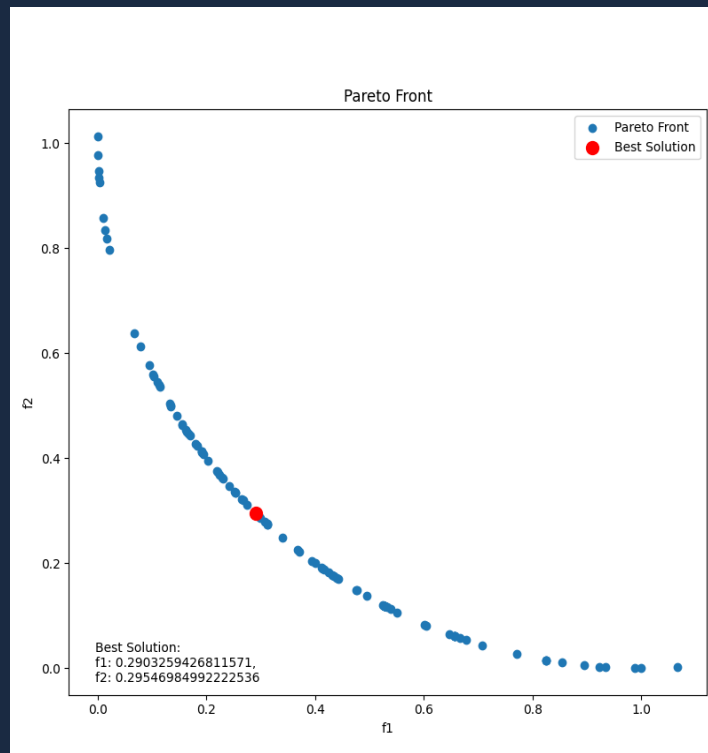
## Hyperparameters:

Population size = 100

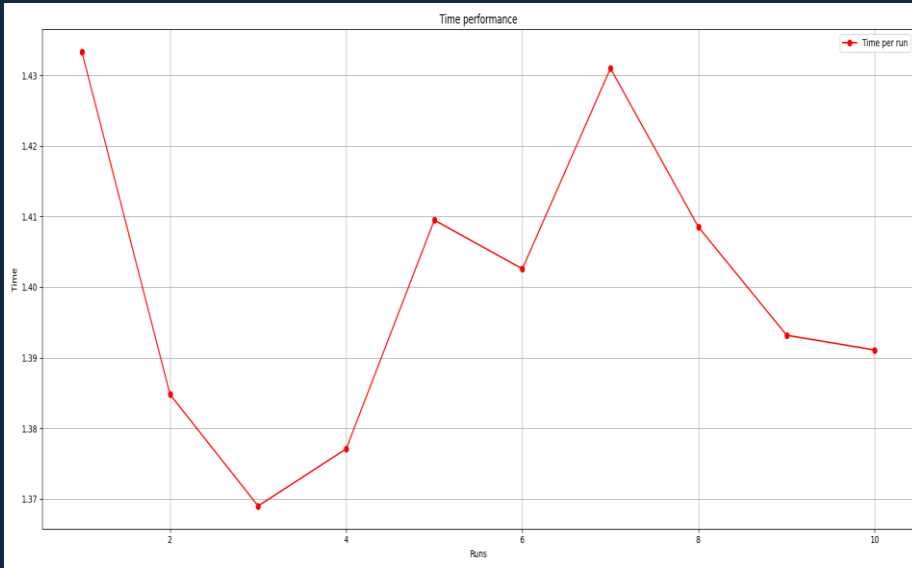
Generations = 100

Mutation rate = 0.05

## SPEA2



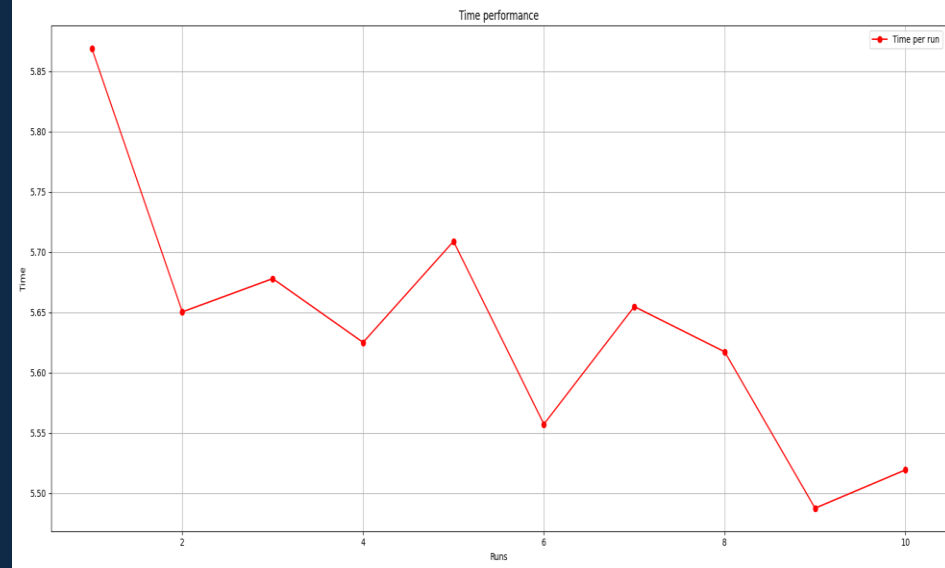
## NSGA-II

[illegible]

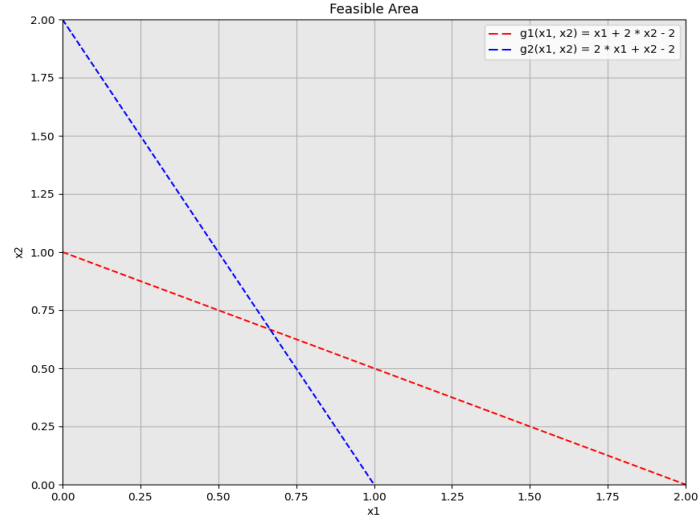
## SPEA2

[illegible]

# SPEA2



## Problem 2



$$\begin{aligned} \max f_1(x_1, x_2) &= -x_1 + 3x_2 \\ \max f_2(x_1, x_2) &= 3x_1 + x_2 \\ \text{s. t. } g_1(x_1, x_2) &= x_1 + 2x_2 - 2 \leq 0 \\ g_2(x_1, x_2) &= 2x_1 + x_2 - 2 \leq 0 \\ x_1, x_2 &\geq 0 \end{aligned}$$

*Utilize the Pareto ranking method for resolution and illustrate with graphs.*





## Simple GA using special operators

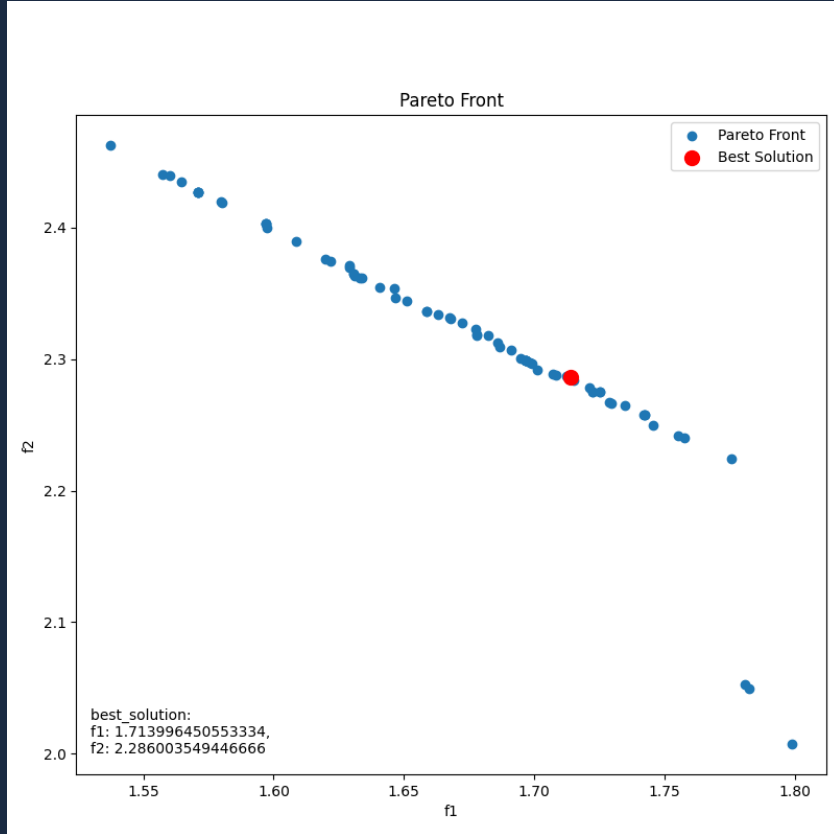
- methodology that provides a way of **handling constraints** (*linear constraints*).
- special “genetic” operators, which guarantee to keep all “chromosomes” within the constrained solution space.

Steps :

1. **Initialize population** : A portion of potential solutions is selected randomly from the space of the whole feasible region; The remaining part of potential solutions consists entirely of points on the boundary of the solution space.
2. **Selection** : Because this is a multi objective function optimization problem *Selection operator must* chose non – dominated individual as best individuals.
3. **Penalty system** : if an individual violates a constrain , it gets a penalty as a punishment , so we make sure that individual is not going to be chosen.
4. **Crossover and mutation** : should move the offspring towards the boundary of the solution space (*as arithmetical crossovers, boundary mutation*)
5. **Repair system** : The main idea of the chromosome-repairing scheme is that, for an **infeasible solution**, repair it for transforming it into a **feasible solution**.

# Experiments

## Simple GA using special operators



## Hyperparameters:

Population size = 100

Generations = 100

Mutation rate = 0.05

Generation 100: Best chromosome  $x_1 = 0.5144$ ,  $x_2 = 0.7428$ ,  $f_1 = 1.7140$ ,  $f_2 = 2.2860$ , penalty = 0.0000



## Conclusions

- For non-constrained multi objective functions problems NSGA II and SPEA 2 demonstrated to find the best solution in the pareto front ,however NSGA II due its *fast non-dominated sort system* ,it surpassed the performance of SPEA 2 in run time. However further observations could be done with different kind of problems and test also the quality of the solutions, since both algorithms have a different approach for selecting the quality and diversity in the best solutions, NSGA II uses *Crowding Distance Assignment method* for maintaining the variability in the solutions by calculating the *Euclidean distance*, in the other hand SPEA 2 uses a *Density truncation method* that eliminates the solutions that are crowded in a specific area.
- A simple GA using special operators can be a fast and effective algorithm for searching the best solution for linear constrained objective functions, maintaining the individuals within the feasible area of possible solutions . However it should be tested and compared with other more complex GA algorithms.

**Thank you!**