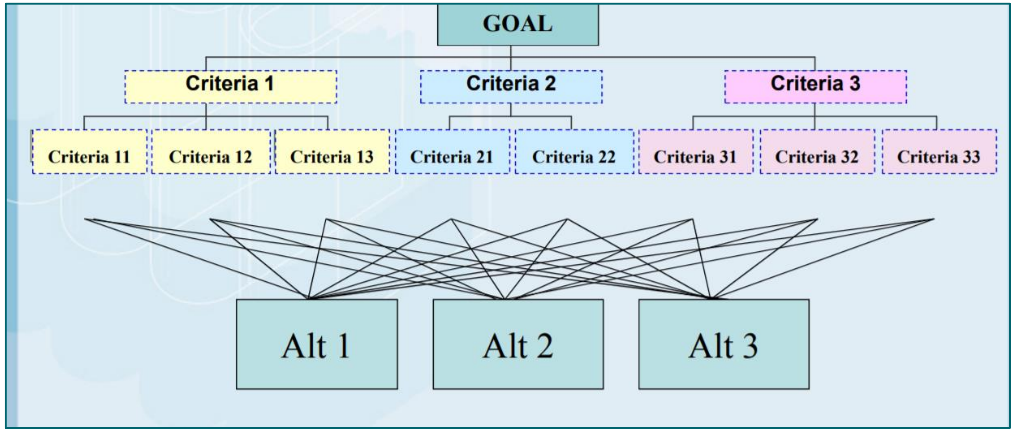
# Ranking of Interventions: AHP vs MOO

# Ranking with AHP and MOO

SIEMENS
Ingenuity for life

## UJ3:Analytical Hierarchical Process-AHP

- ❑ Decompose the decision-making problem into a hierarchy of criteria and alternatives



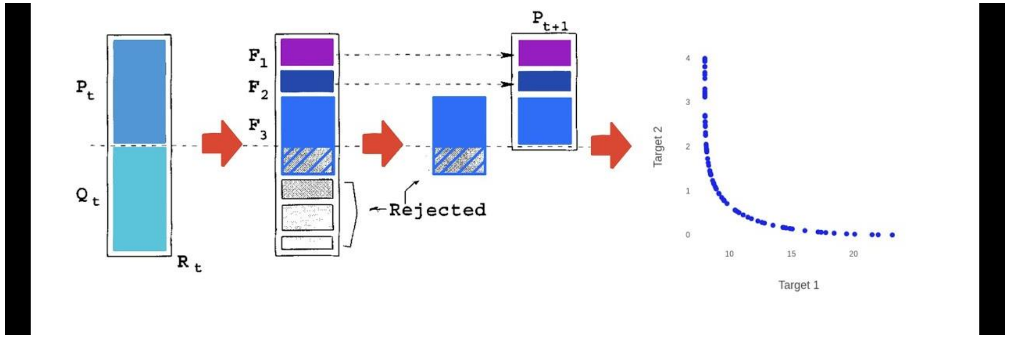

- ❑ Level 1 is the goal of the analysis. Level 2 is multi-criteria that consist of several criterions, You can also add several other levels of sub-criteria. The last level is the alternative choices
- ❑ In the end, the best alternative (in the maximization case) is the one that has the greatest value in the following expression:

$$AHP_i = \sum_{j=1}^{n} \frac{a_{ij}}{\sum_{i=1}^{m} a_{ij}} \times w_j$$

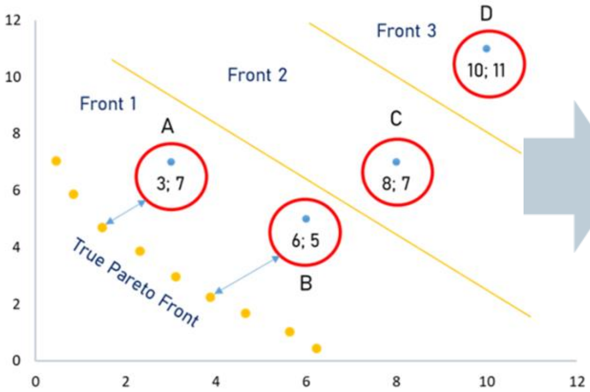
| Alternative | AHP Score | Rank |
|---|---|---|
| C | 0.0835 | 1 |
| B | 0.0823 | 2 |
| I | 0.0798 | 3 |
| L | 0.0784 | 4 |
| M | 0.0773 | 5 |
| A | 0.0695 | 6 |
| K | 0.0659 | 7 |
| G | 0.0642 | 8 |

## UJ5: Multi-Objective Optimization-MOO

- ❑ Since MOO has more than one objective function, we have something called pareto-dominance
- ❑ A solution is called dominant if it has at least one value of objective function that is better than the other while have better or equal value on another objective function




- ❑ Solution Looks like this




Ranking

| ALT./CRIT. | Cosst (min) | Reliability (max) | Risk (min) | Rank |
|---|---|---|---|---|
| Intervention 1 | 180 | 0.99 | 0.1 | 1 |
| Intervention 2 | 200 | 1 | 0.07 | 3 |
| Intervention 3 | 240 | 1 | 0.03 | 2 |
| Intervention 4 | 200 | 0.99 | 0.08 | 4 |
| Intervention 5 | 220 | 1 | 0.08 | 5 |

# Comparison between AHP and MOO

SIEMENS
Ingenuity for life

## AHP:

- **Input:**
  - List of interventions (User/Intervention Library)
  - Pairwise criteria comparisons (User)
  - Structure of decision flow (Coded)
- **Process:**
  - Calculate weighted scores for each criteria and Interventions to get importance
  - Get decision scores as a function of criteria and Interventions importance
- **Output:**
  - A ranking of the interventions based on the AHP score.
- **Decision:**
  - If unsure of decision, do sensitivity analysis (User/Coded)
- **Pros:**
  - Simple
  - Input could be easily adjusted
- **Cons:**
  - Required input scales quadratically with number of criteria
  - Highly subjective input

## MOO:

- **Input:**
  - List of interventions (User/Intervention Library)
  - Input constraints (User/Coded)
- **Process:**
  - Algorithm searches the search space defined by constraints to obtain optimal solutions defined by the Pareto front
- **Output:**
  - Pareto front
- **Decision:**
  - Best decision by distance to Pareto front (Coded)
  - If unsure of decision, inspect results based on different distance metrics and performance indicators (User/Coded)
- **Pros:**
  - Objectivity, minimal user input
- **Cons:**
  - Black-box behavior

# MOO

# Multi-Objective Optimization

# Multi-objective optimization techniques

SIEMENS
Ingenuity for life

1) **a priori Preference Articulation**: take decisions before searching (decide ⇒ search). This group of techniques includes those approaches that assume that either a certain desired achievable goals or a certain pre-ordering of the objectives can be performed by the decision maker prior to the search.
2) **a posteriori Preference Articulation**: search before making decisions (search ⇒ decide). These techniques do not require prior preference information from the decision maker. Some of the techniques included in this category are among the oldest multi-objective optimization approaches proposed.
3) **Progressive Preference Articulation**: integrate search and decision making (decide⇔search). These techniques normally operate in three stages:
    1) find a non dominated solution,
    2) get the reaction of the decision maker regarding this non dominated solution and modify the preferences of the objectives accordingly,
    3) repeat the two previous steps until the decision maker is satisfied or no further improvement is possible.
4) A different kind of approach is represented by Evolutionary Algorithms that are based on Darwin's theory of survival of the fittest. They are found on the idea that as the population evolves in a genetic algorithm, solutions that are non-dominated are chosen to remain in the population.

# But first let's talk multiple objectives…

SIEMENS
Ingenuity for life

A. Multi-objective optimization problems deals with conflicting objectives, i.e. while one objective increases the other decreases.
B. There is no unique global solution but a set of solutions.
C. In general we are interested in the following mathematical problem type:

minimize/maximize: $f_m(\mathbf{x}), m = 1, 2, \ldots, M$

Subject to: $g_j(\mathbf{x}) \geq 0, j = 1, 2, \ldots, J$

$h_k(\mathbf{x}) = 0, k = 1, 2, \ldots, K$

$x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, \ldots, n$

A solution is a vector of *n* decision variables :

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$$

A solution is a vector of *n* decision variables :

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$$

a) Furthermore, the problem is subjected to ***J*** inequality constraints and ***K*** equality constraints. Additionally, each variable has an upper and/or lower bound associated with it.
b) A solution that satisfies all constraints and variable bounds is called a **feasible solution**.
c) The set of all feasible solutions is called the **feasible region,** or ***S***.
d) The term "search space" is also used as a synonym to feasible region.
e) The **objective space** is constituted by the possible values of the ***M*** objectives functions for all solutions in ***S***.

# Definitions

SIEMENS
*Ingenuity for life*

A **Domination**

A solution $\mathbf{x}^{(1)}$ is said to dominate the other solution $\mathbf{x}^{(2)}$ if both condition 1 and 2 below are true:

- Condition 1: $\mathbf{x}^{(1)}$ is no worse than $\mathbf{x}^{(2)}$ for all objectives
- Condition 2: $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ in at least one objective

The mathematical notation for $\mathbf{x}^{(1)}$ dominates $\mathbf{x}^{(2)}$ is:
$x^{(1)} \preceq x^{(2)}$

B **Non-Dominated set:**

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

C **Globally Pareto-optimal set:**

The non-dominated set of the entire feasible search space $S$ is the globally Pareto-optimal set.

# The concept of “Dominated”(1)

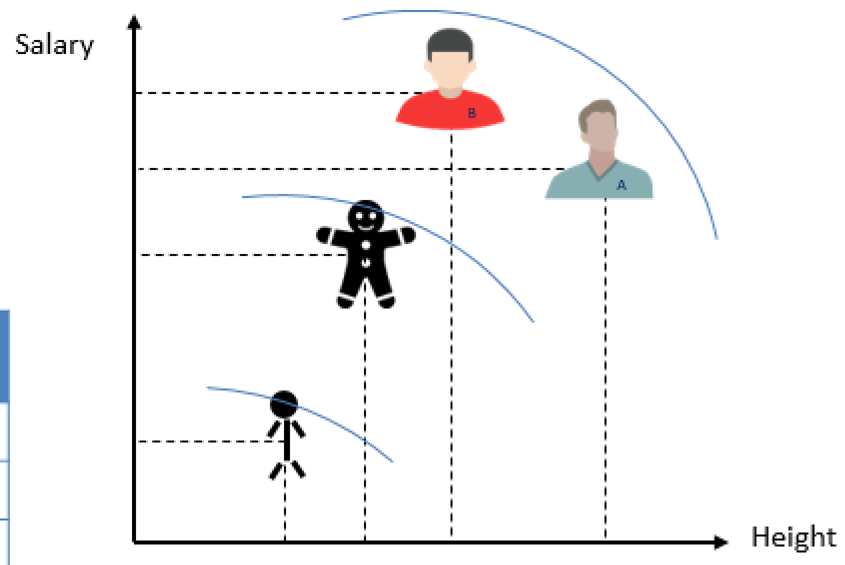
SIEMENS
*Ingenuity for life*

| Basic Concept | In a single-objective optimization problem, it’s easy to identify the quality of a solution. For example, the larger score is a better solution. However, if the problem is a multi-objective optimization problem, the quality of solution is not easy to identify. When there are conflicts between objectives, we can use the concept of “Domination” to judge if the solution is good or not. |
|---|---|

| Name | Height | Salary |
|---|---|---|
| A | 190 | 80K |
| B | 170 | 85K |
| C | 165 | 70K |
| D | 160 | 22K |

Non-dominated
Non-dominated
Dominated
Dominated

| Name | Height | Salary | Pareto Optimal | Front Level | $n_p$ | $S_p$ |
|---|---|---|---|---|---|---|
| A | 190 | 80K | ✔ | 1 | 0 | C, D |
| B | 170 | 85K | ✔ | 1 | 0 | C, D |
| C | 165 | 70K | - | 2 | 1 | D |
| D | 160 | 22K | - | 3 | 2 | - |



# The concept of “Dominated”(2)

SIEMENS
Ingenuity for life

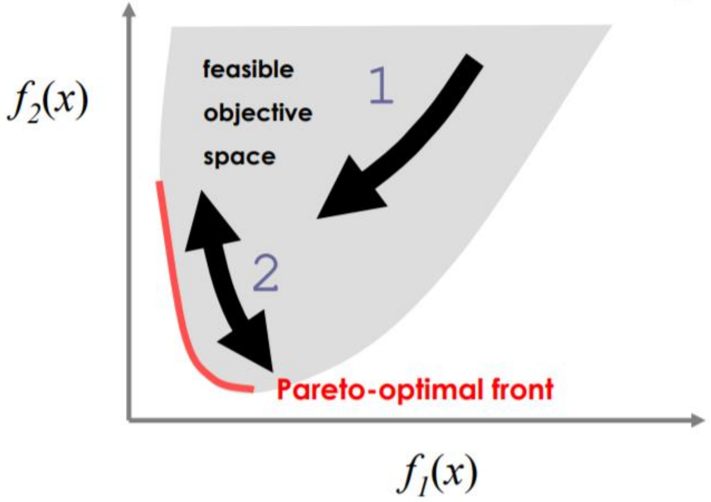
A. This is a simple case that illustrates the concept of Pareto-Optimal. In the **previous slide**, we have 4 members **A, B, C** and **D** with two features: Height and Salary. Now, if we try to compare their height and salary simultaneously, we will find that it’s not quite intuitive because they have multiple objectives.
B. Now that these two objectives are **the larger the better**, we can simply compare each of them. At first, we observe that **A** and **B** have both higher number them **C** and **D**, so we say **A** and **B** “**dominate**” **C** and **D** with respect to height and salary. With the same idea, **C** dominates **D** and **D** is dominated by all of them.
C. How about **A** and **B**? **A** has higher height than **B** but lower salary. In the contrary, **B** confronts the same situation. We call the situation “**non-dominated**”.
D. If we can find a set of solutions that they don’t dominate each other and not dominated by any other solutions, we call them **“Pareto-optimal**” solutions.
E. In the case above, **A and B are just on the Pareto-optimal front**.

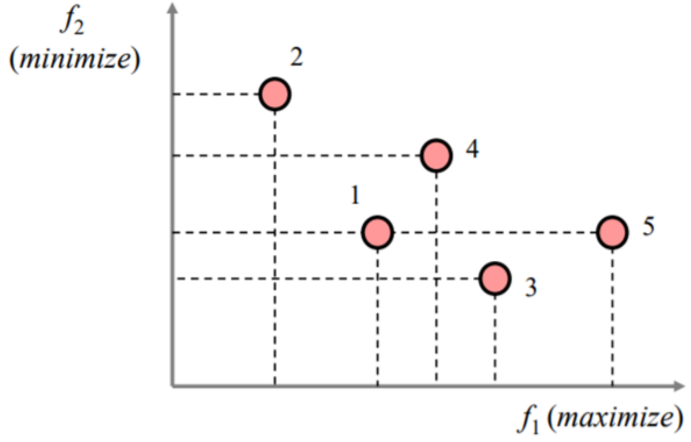
# Goals in MOO

SIEMENS
Ingenuity for life

- Find a set of solutions that they don't dominate each other and not dominated by any other solutions, we call them **"Pareto-optimal"** solutions

- Find set of solutions as close as possible to Pareto-optimal front
- To find a set of solutions as diverse as possible

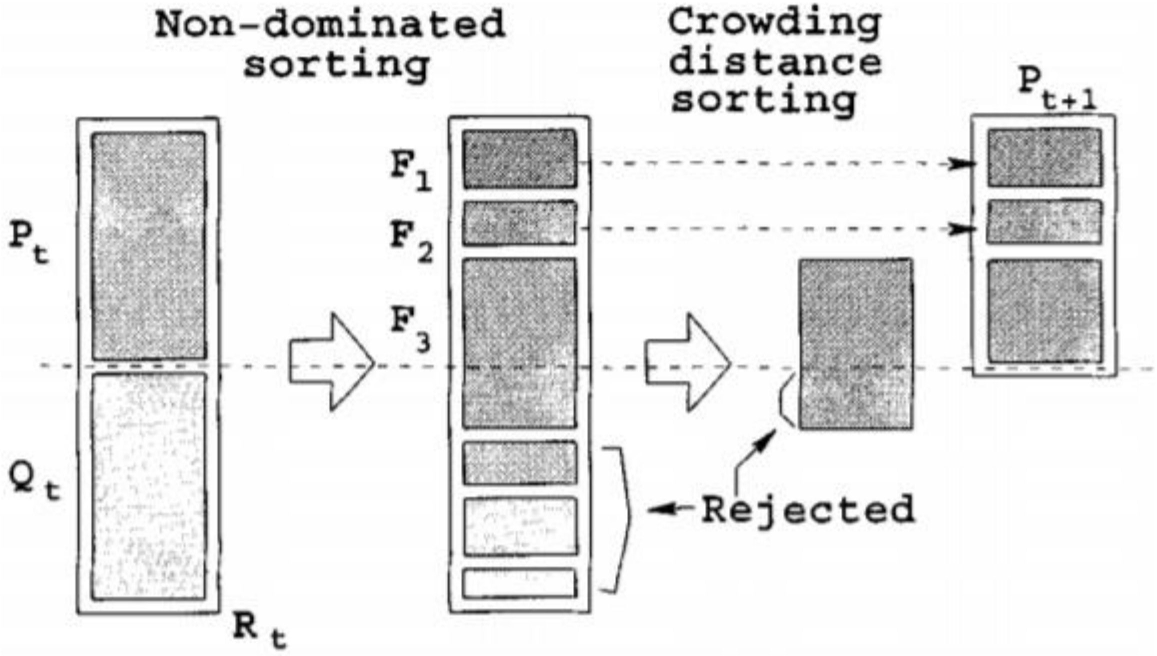



- 1 Vs 2: 1 dominates 2
- 1 Vs 5: 5 dominates 1
- 1 Vs 4: Neither solution dominates

# NSGA-2 Algorithm-1

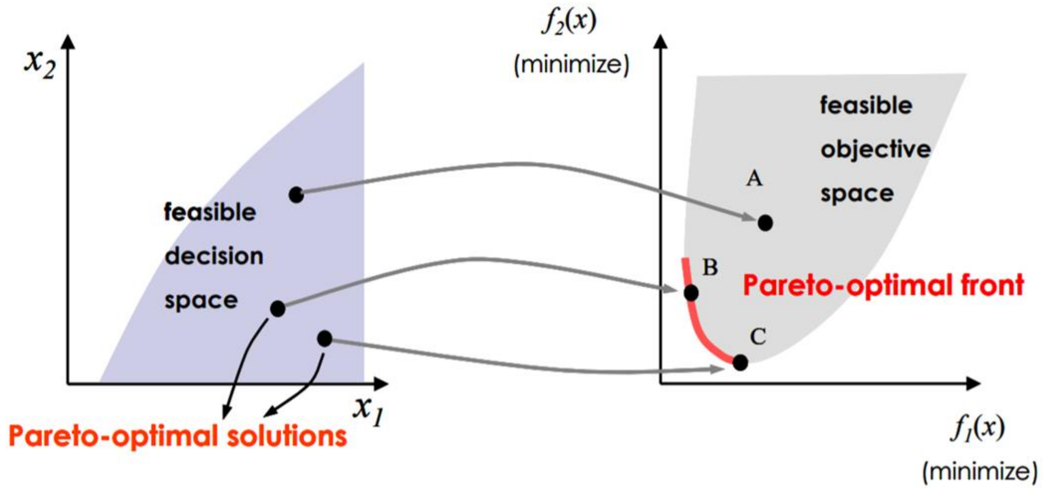
- In an attempt to find the optimal set of alternatives , one would solve for these vectors using the NSGA-II genetic algorithm




i. Let me briefly introduce the NSGA-II algorithm. At first, we have a bunch of individuals as parents in GA with multiple objectives.
ii. In each iteration, we combine the parent and the offspring after GA operations.
iii. **Through the Non-dominated Sorting, we classify all individuals to different Pareto-optimal front level**.
iv. We then select individuals as the next population from Pareto-optimal front in the order of different levels.
v. As for diversity preservation, the "**Crowding Distance**" is also computed.
vi. The crowded distance comparison guides the selection process at the various stages of the algorithm toward a uniformly spread-out Pareto-optimal front.

# Pareto Front & Pareto Optimal Solution

1) A Pareto optimal solution is a feasible solution, though; it cannot be improved without degrading at least one set of objectives. Therefore, no feasible solution can dominate it.
2) These alternatives, or Pareto optimal solutions, can be referred to as the Pareto front




- **Non-dominated solution set**
  - Given a set of solutions, the non-dominated solution set is a set of all the solutions that are not dominated by any member of the solution set
- The non-dominated set of the entire feasible decision space is called the **Pareto-optimal set**
- The boundary defined by the set of all point mapped from the Pareto optimal set is called the **Pareto-optimal front**

# NSGA-2 Algorithm-2

SIEMENS
*Ingenuity for life*

| Name | Height | Salary | Pareto Optimal | Front Level | $n_p$ | $S_p$ |
|---|---|---|---|---|---|---|
| A | 190 | 80K | ✔ | 1 | 0 | C, D |
| B | 170 | 85K | ✔ | 1 | 0 | C, D |
| C | 165 | 70K | - | 2 | 1 | D |
| D | 160 | 22K | - | 3 | 2 | - |

i. The identities of **np** and **Sp** are shown at the end of each row. **np** means "How many people dominate you?" and **Sp** means "Who are you dominating?".
ii. Because **A** and **B** are not dominated by any solution and not dominating each other, their **np** equals to 0 and **Sp** contains **C** and **D**.
**iii.** **C** is dominated by **A** and **B**, its **np** equals to 1. C also dominates D, so its Sp contains D.

| | |
|---|---|
| $R_t = P_t \cup Q_t$ | combine parent and offspring population |
| $\mathcal{F} = \texttt{fast-non-dominated-sort}(R_t)$ | $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$, all nondominated fronts of $R_t$ |
| $P_{t+1} = \emptyset$ and $i = 1$ | |
| until $\lvert P_{t+1}\rvert + \lvert\mathcal{F}_i\rvert \le N$ | until the parent population is filled |
| $\quad\texttt{crowding-distance-assignment}(\mathcal{F}_i)$ | calculate crowding-distance in $\mathcal{F}_i$ |
| $\quad P_{t+1} = P_{t+1} \cup \mathcal{F}_i$ | include $i$th nondominated front in the parent pop |
| $\quad i = i + 1$ | check the next front for inclusion |
| $\text{Sort}(\mathcal{F}_i, \prec_n)$ | sort in descending order using $\prec_n$ |
| $P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - \lvert P_{t+1}\rvert)]$ | choose the first $(N - \lvert P_{t+1}\rvert)$ elements of $\mathcal{F}_i$ |
| $Q_{t+1} = \texttt{make-new-pop}(P_{t+1})$ | use selection, crossover and mutation to create a new population $Q_{t+1}$ |
| $t = t + 1$ | increment the generation counter |

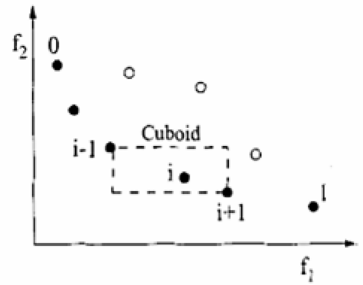
# Selection mechanism



A. Finally, each solution in the population will have two attributes:
   1) Non-domination Rank
   2) Crowding Distance
B. The next population will be selected follow the principles:
   1) The individuals with higher non-domination rank (**smaller number**) will have higher priority
   2) If individuals have identical non-domination rank, the ones with larger **Crowding Distance** have higher priority

$\underline{\texttt{fast-non-dominated-sort}(P)}$

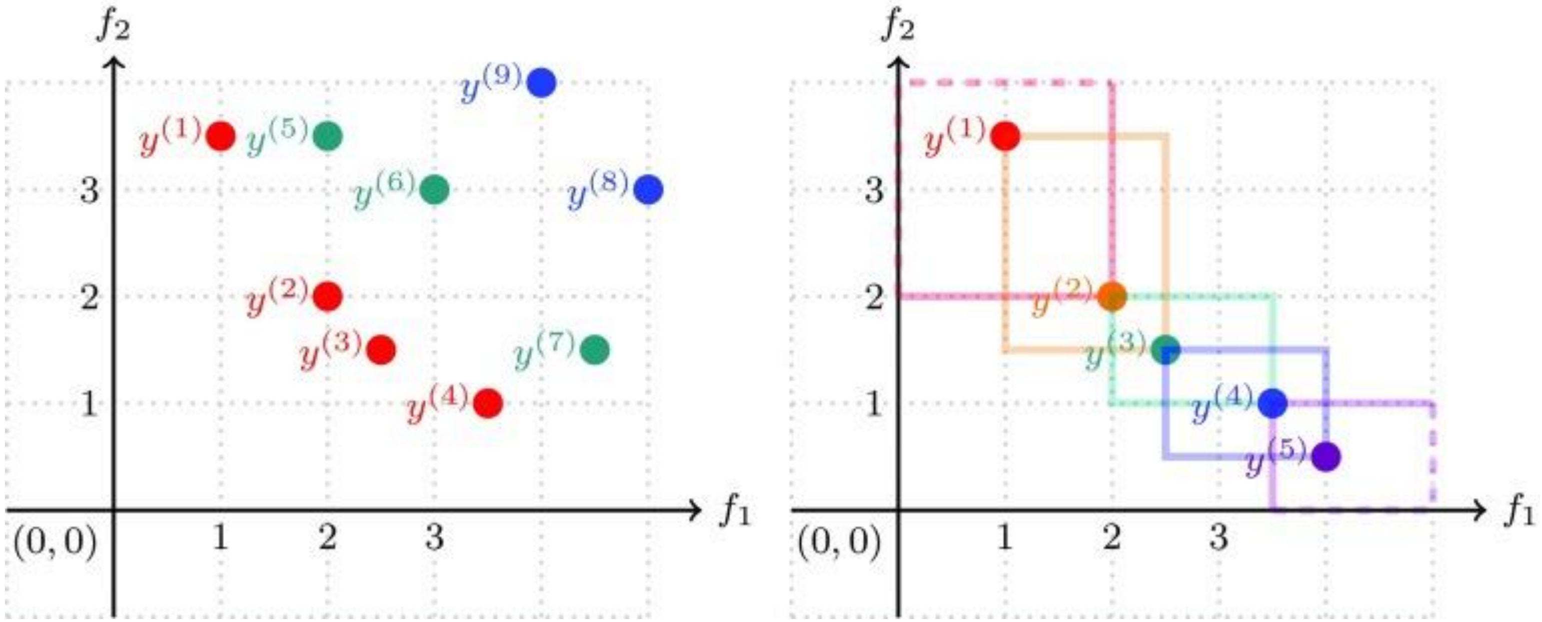
| | |
|---|---|
| for each $p \in P$ | |
|   $S_p = \emptyset$ | |
|   $n_p = 0$ | |
|   for each $q \in P$ | |
|     if $(p \prec q)$ then | If $p$ dominates $q$ |
|       $S_p = S_p \cup \{q\}$ | Add $q$ to the set of solutions dominated by $p$ |
|     else if $(q \prec p)$ then | |
|       $n_p = n_p + 1$ | Increment the domination counter of $p$ |
|   if $n_p = 0$ then | $p$ belongs to the first front |
|     $p_{\text{rank}} = 1$ | |
|     $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$ | |
| $i = 1$ | Initialize the front counter |
| while $\mathcal{F}_i \neq \emptyset$ | |
|   $Q = \emptyset$ | Used to store the members of the next front |
|   for each $p \in \mathcal{F}_i$ | |
|     for each $q \in S_p$ | |
|       $n_q = n_q - 1$ | |
|       if $n_q = 0$ then | $q$ belongs to the next front |
|         $q_{\text{rank}} = i + 1$ | |
|         $Q = Q \cup \{q\}$ | |
|   $i = i + 1$ | |
|   $\mathcal{F}_i = Q$ | |




K.Deb, A.Pratap, S.Agarwal, T.Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,IEEE Trans. Evol. Comput.6(2)(2002)182

# Illustration of non-dominated sorting (left) and crowding distance (right)

- The *ranking procedure of NSGA-II* consists of two levels.
- First, non-dominated sorting is performed. This ranking solely depends on the Pareto order and does not depend on diversity.
- Secondly, individuals which share the same rank after the first ranking are then ranked according to the crowding distance criterion which is a strong reflection of the diversity.



# NSGA-2: Crowding Distance-1

- To estimate the density of solutions surrounding a particular solution *i* in the population, we take the average distance of two solutions on either side of solution *i* along each of the objectives.
- For each front $F_i$ , n is the number of individuals.
- Initialize the distance to be zero for all the individuals i.e. $F_i(d_j) = 0$, where $d_j$ corresponds to the $j^{th}$ individual in Front $F_i$ .
- For each objective function m
    - Sort the individuals in front $F_i$ based on objective m i.e. $I = sort(F_i , m)$.
    - Assign infinite distance to boundary values for each individual in $F_i$ i.e. $I(d_1) = \infty$ and $I(d_n) = \infty$

# NSGA-2: Crowding Distance-2

SIEMENS
*Ingenuity for life*

## Diversity Preservation / Crowding Distance (CD)

1. Sort all 'l' solution in a Pareto front in ascending order of $f_m$ and compute:

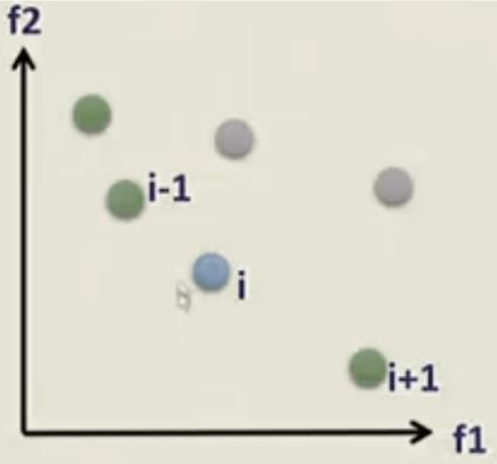
$$CD_{im} = \frac{f_m(x_{i+1}) - f_m(x_{i-1})}{f_m(x_{max}) - f_m(x_{min})}, \; i = 2, \ldots\ldots, (l\text{-}1)$$

2. Repeat step 1 for each objective and find the crowding distance of solution i as:

$$CD_i = \sum_{m=1}^{M} CD_{im}$$

3. Given two solutions i and j, solution i is preferred to solution j if:

$$R_i < R_j \text{ or } (R_i = R_j \text{ and } CD_i > CD_j)$$




*More the Crowding Distance, more the solution is likely to be retained for diversity preservation*

# (More) Constrained NSGA-II

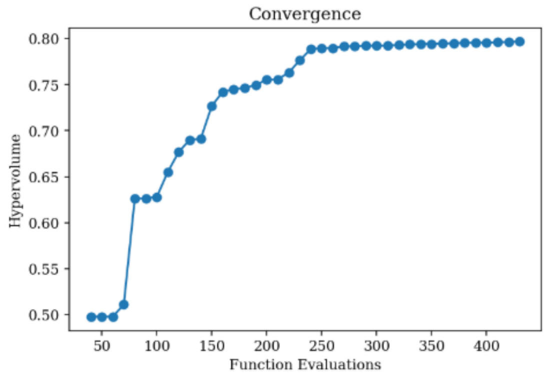
SIEMENS
Ingenuity for life

A. This constraint-handling method uses the binary tournament selection, where two solutions are picked from the population and the better solution is chosen. In the presence of constraints, each solution can be either feasible or infeasible. Thus, there may be at most three situations:
    1) both solutions are feasible;
    2) one is feasible and other is not; and
    3) both are infeasible.
B. For single objective optimization, we used a simple rule for each case.
    1) Choose the solution with better objective function value.
    2) Choose the feasible solution.
    3) Choose the solution with smaller overall constraint violation.
C. Since in no case constraints and objective function values are compared with each other, there is no need of having any penalty parameter, a matter that makes the proposed constraint-handling approach useful and attractive.

# Performance Measures

SIEMENS
Ingenuity for life

A. Unlike in single-objective optimization, there are two goals in a multiobjective optimization:
    1) convergence to the Pareto-optimal set and
    2) maintenance of diversity in solutions of the Pareto-optimal set.
B. These two tasks cannot be measured adequately with one performance metric.
C. Many performance metrics have been suggested .
D. Here, we define two performance metrics that are more direct in evaluating each of the above two goals in a solution set obtained by a multiobjective optimization algorithm
    1) The first metric measures the extent of convergence to a known set of Pareto-optimal solutions. Since multiobjective algorithms would be tested on problems having a known set of Pareto-optimal solutions, the calculation of this metric is possible
    2) The second metric measures the extent of spread achieved among the obtained solutions. Here, we are interested in getting a set of solutions that spans the entire Pareto-optimal region



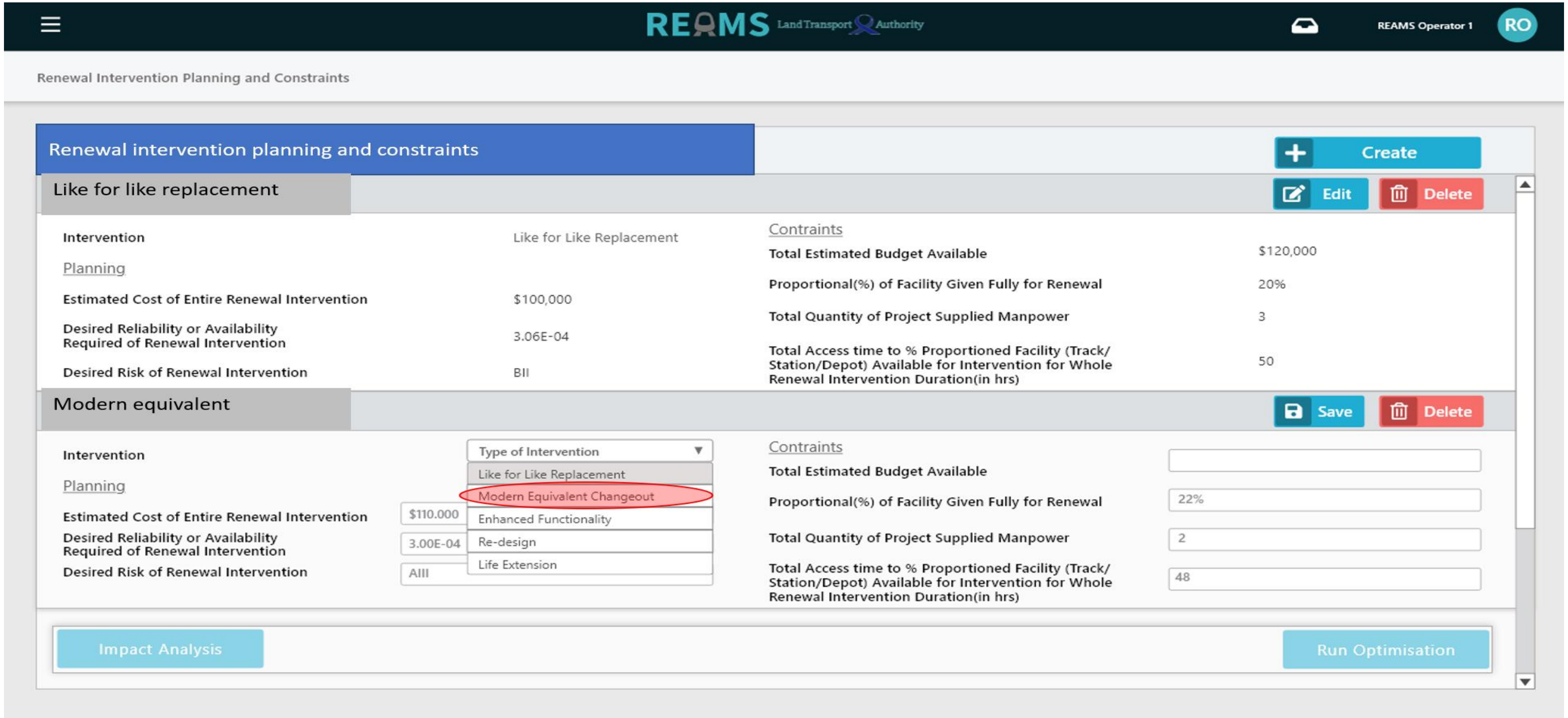
# MOO

# REAMS Example:

## Ranking of Asset Renewal Options

# Data Capture for MOO

SIEMENS
Ingenuity for life



# REAMS Example:
# Ranking of Asset Renewal Intervention Options

SIEMENS
Ingenuity for life

## Intervention Options

| Assumptions of Intervention type | re-design (assuming done at year 20, remaining life 10 years) | | | | |
|---|---|---|---|---|---|
| **Intervention Type** | **re-design** | **Like for like** | **Modern equivalent** | **Continued maintenance until OEM design life (assuming done at year 20, remaining life 10 years)** | **Remarks** |
| OEM design life | 30 | 30 | 30 | 30 | |
| Performance (FPMK for > 5 min event) | 0.9589 | 0.9772 | 0.8793 | 1.277* | |
| Safety Risk | AIII | AIII | AIV | CII* | |
| Service Disruption Risk | AIII | AIII | AIV | CII* | |
| Common track equipment (Locomotive qty) | 0 | 0 | 0 | 0 | |
| Common track equipment (wagon qty) | 0 | 0 | 0 | 0 | |
| calculated duration (in facility) due constraints (months) | 8.717 | 7.1875 | 9.602 | assume into maintenance hours | |
| demand manhours per mth (constrained facility Access time x supply manpower) per unit of RS (whole train = sum of all subsystems). | 121 | 98 | 123 | assume into maintenance current support | |
| | | | | | |
| Acquisition cost: CAPEX | $ 459,599,615 | $ 797,079,435 | $ 824,473,234 | $ 120,826,667 | |
| Total Operations (traction energy cost) | $ 471,057,539 | $ 434,554,925 | $ 425,429,272 | $ 160,030,213 | using $0.20 per KWh |
| Total OPEX | $ 158,003,131 | $ 365,636,452 | $ 334,200,289 | $ 149,498,929 | Sum of Annual |

## User Input

| Constraints | Targets |
|---|---|
| Cost | |
| Performance | |
| Risk | |

## User Input

| Objectives | Goal |
|---|---|
| Cost | Minimize |
| Performance | Maximize |
| Risk | Minimize |

# Mathematical Formulation

SIEMENS
Ingenuity for life

## Objective Function

Minimize $\sum_i^N \sum_t^T c_{it} x_{it}$ ......... **Minimize Total Cost**

Minimize $\sum_i^N \sum_t^T \mathrm{Risk}_{it} x_{it}$ ......... **Minimize Risk**

Maximize $\sum_i^N \sum_t^T \mathrm{Reliability}_{it} x_{it}$ ....... **Maximize Reliability**

## Constraints

$$\sum_i^N \sum_t^T c_{it} x_{it} \le B$$

Total Cost <= B

$$\sum_i^N \sum_t^T c_{it} R_{it} \ge R$$

Reliability >= R

$$\sum_i^N \sum_t^T c_{it} R_{it} \le Risk$$

Risk <= Ri

## Python Implementation

In [1]:

```
import numpy as np
import autograd.numpy as anp
import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D
from pymoo.factory import get_termination
from pymoo.optimize import minimize
from pymoo.model.problem import Problem
from pymoo.configuration import Configuration
from pymoo.factory import get_performance_indicator
from pymoo.algorithms.nsga2 import NSGA2
from pymoo.factory import get_sampling, get_crossover, get_mutation

Configuration.show_compile_hint = False

%matplotlib inline
```

In [2]:

```
class UJ5poc(Problem):

    def __init__(self):
        super().__init__(n_var=5, n_obj=3, n_constr=3, xl=0, xu=100)

    def _evaluate(self, x, out, *args, **kwargs):
        # Define objective function here.
        f1 = 100*x[:, 0] +150*x[:, 1]+ 600*x[:,2]+540*x[:,3]+340*x[:,4]
        f2 = 0.98*x[:,0]+0.90*x[:,1]+0.93*x[:,2]+0.89*x[:,3]+0.99*x[:,4]
        f3 = 0.1*x[:,0]+0.1*x[:,1]+0.13*x[:,2]+0.21*x[:,3]+0.23*x[:,3]


        # Define constains here.
        g1 = f1 - 700
        g2 = f2 - .99
        g3 = f3 - 0.21

        out["F"] = np.column_stack([f1,f2,f3])
        out["G"] = np.column_stack([g1,-g2,g3])

    def _calc_pareto_front(self, flatten=True, **kwargs):
        '''
        Theoritcal Pareto front for display only.
        '''
        f1 = 0
        f2 = 0
        f3 = 0
        return np.column_stack([f1, f2, f3])

    def _calc_pareto_set(self, flatten=True, **kwargs):
        '''
        Theoritcal Pareto front set for display only.
        '''
        x1 = 0
        x2 = 0
        x3 = 0
        return np.column_stack([x1, x2, x3])

problem = UJ5poc()
```

# Results of MOO

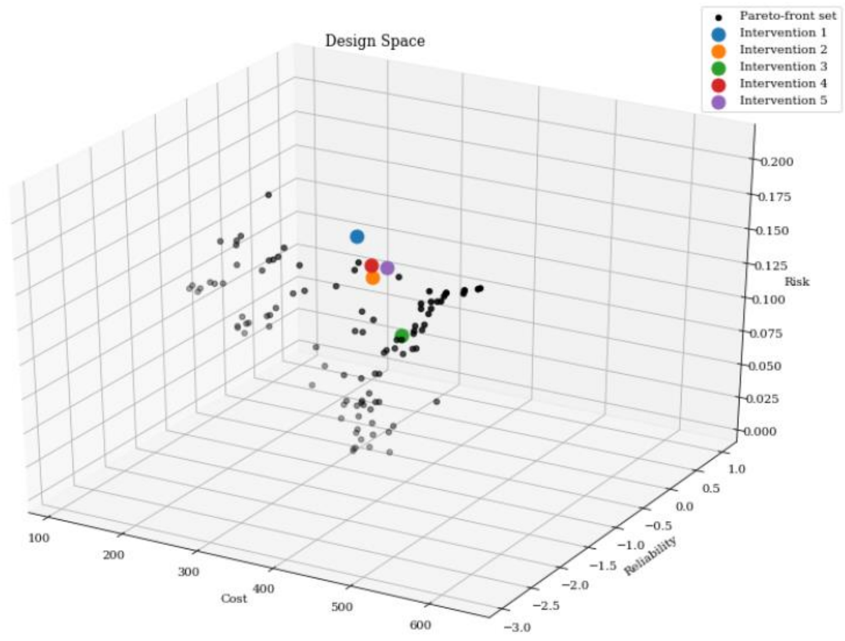
SIEMENS
Ingenuity for life

**Non-Dominated Sorting**

In [175]:
```
algorithm = NSGA2(
    pop_size=100,
    n_offsprings=10,
    sampling=get_sampling("real_random"),
    crossover=get_crossover("real_sbx", prob=0.9, eta=15),
    mutation=get_mutation("real_pm", eta=20),
    eliminate_duplicates=True
)

#termination = get_termination("n_gen", 1000)
termination = MultiObjectiveSpaceToleranceTermination(tol=0.0025,
                                                      n_last=30,
                                                      nth_gen=5,
                                                      n_max_gen=None,
                                                      n_max_evals=None)
res = minimize(problem,
               algorithm,
               termination,
               seed=1,
               pf=problem.pareto_front(),
               save_history=True,
               verbose=True)
pf = res.F

plot = Scatter()
plot.add(res.F, color="black")
plot.show()
```

| n_gen | n_eval | cv (min) | cv (avg) | n_nds | delta_ideal | delta_nadir | delta_f |
|---|---|---|---|---|---|---|---|
| 1 | 100 | 0.00000E+00 | 2.15561E+02 | 6 | - | - | - |
| 2 | 110 | 0.00000E+00 | 1.66598E+02 | 7 | 0.00000E+00 | 0.00000E+00 | 0.052174631 |
| 3 | 120 | 0.00000E+00 | 1.26791E+02 | 8 | 0.572775384 | 0.00000E+00 | 0.136604502 |
| 4 | 130 | 0.00000E+00 | 9.78803E+01 | 10 | 0.050882035 | 0.325912075 | 0.085240973 |
| 5 | 140 | 0.00000E+00 | 7.30866E+01 | 13 | 0.205940060 | 0.127842239 | 0.087748698 |
| 6 | 150 | 0.00000E+00 | 5.40442E+01 | 14 | 0.036401671 | 0.00000E+00 | 0.017728475 |
| 7 | 160 | 0.00000E+00 | 3.78357E+01 | 15 | 0.048267106 | 0.018274913 | 0.019928363 |
| 8 | 170 | 0.00000E+00 | 2.14993E+01 | 17 | 0.161837086 | 0.038780114 | 0.036014564 |
| 9 | 180 | 0.00000E+00 | 1.06768E+01 | 18 | 0.00000E+00 | 0.025790749 | 0.015962339 |
| 10 | 190 | 0.00000E+00 | 3.747916129 | 18 | 0.098833629 | 0.023857047 | 0.015527507 |
| 11 | 200 | 0.00000E+00 | 0.504360865 | 21 | 0.00000E+00 | 0.037545873 | 0.021246236 |
| 12 | 210 | 0.00000E+00 | 0.042777259 | 23 | 0.031595234 | 0.00000E+00 | 0.006179440 |
| 13 | 220 | 0.00000E+00 | 0.022981643 | 27 | 0.00000E+00 | 0.00000E+00 | 0.008107230 |
| 14 | 230 | 0.00000E+00 | 0.012361376 | 33 | 0.00000E+00 | 0.016398907 | 0.007257909 |
| 15 | 240 | 0.00000E+00 | 0.005768164 | 34 | 0.00000E+00 | 0.016672315 | 0.019603070 |
| 16 | 250 | 0.00000E+00 | 0.001996491 | 34 | 0.00000E+00 | 0.00000E+00 | 0.007320656 |

[155]: <matplotlib.legend.Legend at 0x275baf89a20>




In [ ]:
```
#The black dots represents the analytically derived Pareto set and front in the corresponding space
# and the circles solutions found by the algorithm
```

# Ranking of Interventions

SIEMENS
Ingenuity for life

**Intervention -- Pareto front distance: Generational Distance & Generational Distance Plus**

The GD performance indicator measures the distance from solution to the Pareto-front. Let us assume the points found by our algorithm are the objective vector set $A = \{a_1, a_2, \ldots, a_{|A|}\}$ and the reference points set (Pareto-front) is $Z = \{z_1, z_2, \ldots, z_{|Z|}\}$. Then,

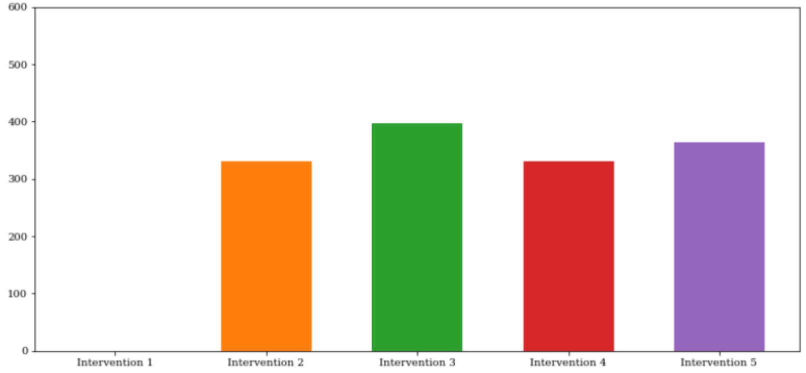
$$\mathrm{GD}(A) = \frac{1}{|A|}\left(\sum_{i=1}^{|A|} d_i^p\right)^{1/p}$$

Where $d_i$ represents the euclidean distance (p=2) from $a_i$ to its nearest reference point in $Z$. Basically, this results in the average distance from any point $A$ to the closest point in the Pareto-front.

$$\mathrm{GD}^+(A) = \frac{1}{|A|}\left(\sum_{i=1}^{|A|} {d_i^+}^2\right)^{1/2}$$

where for minimization $d_i^+ = max\{a_i - z_i, 0\}$ represents the modified distance from $a_i$ to its nearest reference point in $Z$ with the corresponding value $z_i$.

```
]: gd_plus = get_performance_indicator("gd+", res.F,normalize=True)

plt.figure(figsize=[13,6])

plt.bar([1],[gd_plus.calc(A)], width = 0.6)
plt.bar([2],[gd_plus.calc(B)], width = 0.6)
plt.bar([3],[gd_plus.calc(C)], width = 0.6)
plt.bar([4],[gd_plus.calc(D)], width = 0.6)
plt.bar([5],[gd_plus.calc(E)], width = 0.6)

plt.xticks([1,2,3,4,5],['Intervention 1','Intervention 2','Intervention 3','Intervention 4','Intervention 5'])
plt.ylim([0,600]);
```




```
In [157]: ## GD Distance for the different Interventions

print("GD", gd.calc(A))
print("GD", gd.calc(B))
print("GD", gd.calc(C))
print("GD", gd.calc(D))
print("GD", gd.calc(E))

GD 11.778740101057378
GD 165.5057993154978
GD 184.3474053052114
GD 165.50574062828727
GD 174.9816513739253
```

```
In [96]: gd_plus

Out[96]: <pymoo.performance_indicator.gd_plus.GDPlus at 0x275b8fa3c50>
```

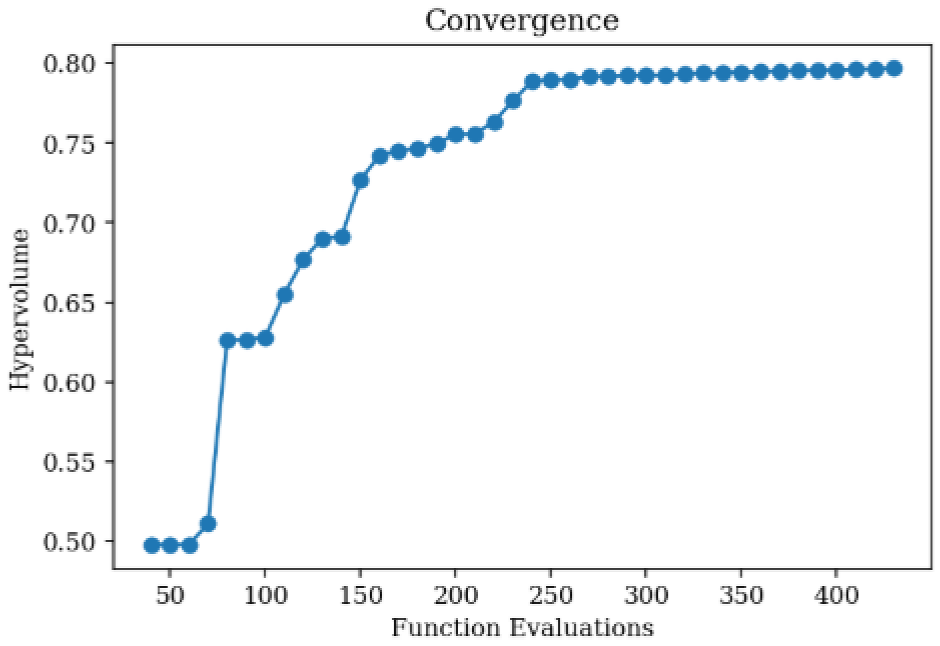
Since intervention 1 is closest to Pareto front, it should be selected as the best action to be taken.

# Performance Tracking

SIEMENS
Ingenuity for life

a) If the optimization scenario is repetitive it makes sense to track the performance of the algorithm.
b) Because we have stored the history of the optimization run, we can now analyze the convergence over time.
c) To measure the performance, we need to decide what metric to be used.
d) Here, we are using Hypervolume.

```python
import matplotlib.pyplot as plt
from pymoo.performance_indicator.hv import Hypervolume

# create the performance indicator object with reference point (4,4)
metric = Hypervolume(ref_point=np.array([1.0, 1.0]))

# collect the population in each generation
pop_each_gen = [a.pop for a in res.history]

# receive the population in each generation
obj_and_feasible_each_gen = [pop[pop.get("feasible")[:,0]].get("F") for pop in pop_each_gen]

# calculate for each generation the HV metric
hv = [metric.calc(f) for f in obj_and_feasible_each_gen]

# function evaluations at each snapshot
n_evals = np.array([a.evaluator.n_eval for a in res.history])

# visualze the convergence curve
plt.plot(n_evals, hv, '-o')
plt.title("Convergence")
plt.xlabel("Function Evaluations")
plt.ylabel("Hypervolume")
plt.show()
```



# AHP

# Analytics Hierarchical Process

# The Analytical Hierarchy Process - AHP

SIEMENS
Ingenuity for life

- AHP is one of the multiple criteria decision-making method that was originally developed by Prof. Thomas L. Saaty (1977).
- Provides measures of judgement consistency
- Derives priorities among criteria and alternatives
- Simplifies preference ratings among decision criteria  using pair wise comparisons

# Analytic hierarchy process: How does it work?

SIEMENS

Ingenuity for life

A. Users of the AHP **first decompose their decision problem into a hierarchy of more easily comprehended sub-problems, each of which can be analyzed independently.**
B. The elements of the hierarchy can be related to any aspect of the decision problem tangible or intangible, carefully measured or roughly estimated, well or poorly understood anything at all that applies to the decision at hand.
C. Once the hierarchy is built, **the decision makers systematically evaluate its various elements by comparing them with one another two at a time, with respect to their impact on an element above them in the hierarchy**.
D. In making the comparisons, the decision makers can use concrete data about the elements, but **they typically use their judgements about the elements' relative meaning and importance**. It is the essence of the AHP that human judgments, and not just the underlying information, can be used in performing the evaluations.
E. The AHP converts these evaluations to numerical values that can be processed and compared over the entire range of the problem.
F. A numerical weight or priority is derived for each element of the hierarchy, allowing diverse and often incommensurable elements to be compared to one another in a rational and consistent way.
G. This capability distinguishes the AHP from other decision making techniques.
H. In the final step of the process, numerical priorities are calculated for each of the decision alternatives.
I. These numbers represent the alternatives' relative ability to achieve the decision goal, so they allow a straightforward consideration of the various courses of action.

# The basic procedure is as follows:

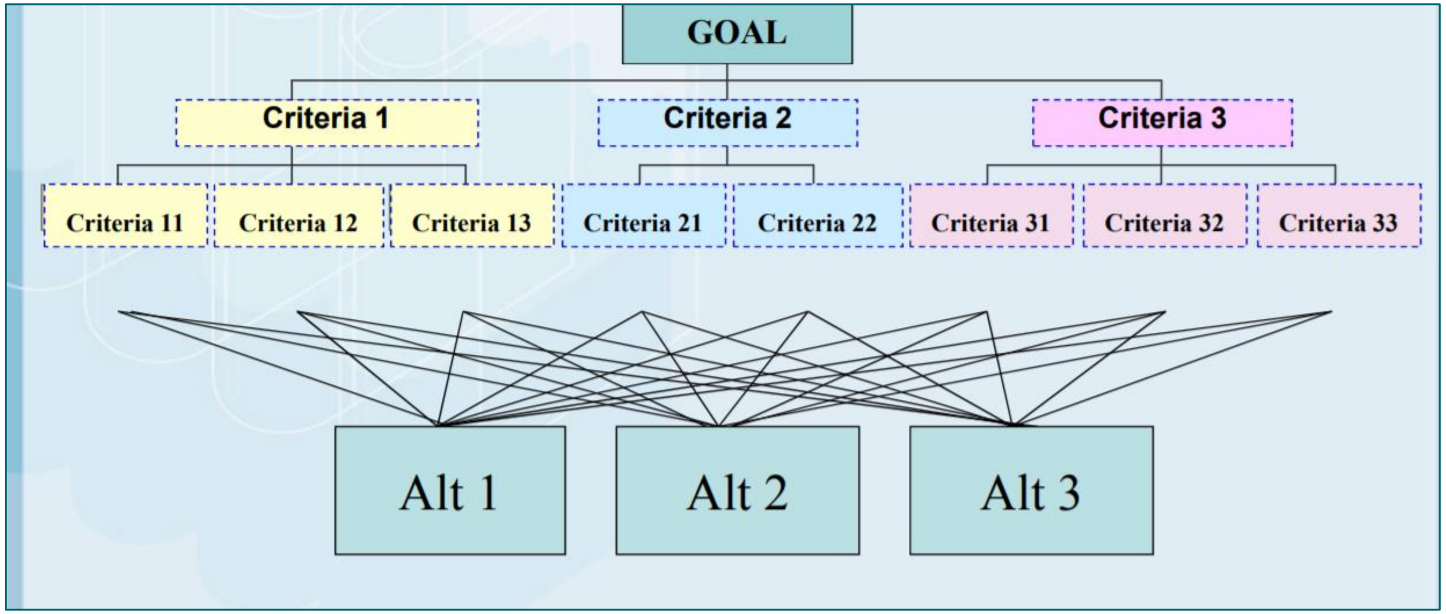
SIEMENS
Ingenuity for life

1. Develop the ratings for **each decision alternative** for each criterion by
   a. developing a pair wise comparison matrix for each criterion
   b. normalizing the resulting matrix
   c. averaging the values in each row to get the corresponding rating
   d. calculating and checking the consistency ratio
2. Develop the weights **for the criteria** by
   a. developing a pairwise comparison matrix for each criterion
   b. normalizing the resulting matrix
   c. averaging the values in each row to get the corresponding rating
   d. calculating and checking the consistency ratio
3. Calculate the weighted average rating for each decision alternative. Choose the one with the highest score.

# Structure the Hierarchy

SIEMENS
Ingenuity for life

- ❑ Decompose the decision-making problem into a hierarchy of criteria and alternatives




- ❑ Level 1 is the goal of the analysis.
- ❑ Level 2 is multi-criteria that consist of several criterions, You can also add several other levels of sub-criteria.
- ❑ The last level is the alternative choices

# Saaty's Scale for Pairwise Comparison

SIEMENS
Ingenuity for life

- The first step in the AHP procedure is to make pair wise comparisons between each criterion

| Scale | Degree of preference |
|---|---|
| 1 | Equal importance |
| 3 | Moderate importance of one factor over another |
| 5 | Strong or essential importance |
| 7 | Very strong importance |
| 9 | Extreme importance |
| 2,4,6,8 | Values for inverse comparison |

- Results of the comparison (for each factors pair) were described in term of integer values from 1 (equal value) to 9 (extreme different) where higher number means the chosen factor is considered more important in greater degree than other factor being compared with.

# The mathematics of AHP

SIEMENS

*Ingenuity for life*

## (1) Normalization: "Behind the scene"

For a matrix of pair-wise elements: $\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$

1) sum the values in each column of the pair-wise matrix

$$C_{ij} = \sum_{i=1}^{n} C_{ij}$$

2) divide each element in the matrix by its column total to generate a normalized pair-wise matrix

$$X_{ij} = \frac{C_{ij}}{\sum_{i=1}^{n} C_{ij}} \qquad \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}$$

3) divide the sum of the normalized column of matrix by the number of criteria used (n) to generate weighted matrix

$$W_{ij} = \frac{\sum_{j=1}^{n} X_{ij}}{n} \qquad \begin{bmatrix} W_{11} \\ W_{12} \\ W_{13} \end{bmatrix}$$

## (2) Consistency analysis : "Behind the scene"

Consistency Vector is calculated by multiplying the pair-wise matrix by the weights vector

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} * \begin{bmatrix} W_{11} \\ W_{21} \\ W_{31} \end{bmatrix} = \begin{bmatrix} Cv_{11} \\ Cv_{21} \\ Cv_{31} \end{bmatrix}$$

Then it is is accomplished by dividing the weighted sum vector with criterion weight

$$Cv_{11} = \frac{1}{W_{11}}[C_{11}W_{11} + C_{12}W_{21} \quad C_{13}W_{31}]$$

$$Cv_{21} = \frac{1}{W_{21}}[C_{21}W_{11} + C_{22}W_{21} \quad C_{23}W_{31}]$$

$$Cv_{31} = \frac{1}{W_{31}}[C_{31}W_{11} + C_{32}W_{21} \quad C_{33}W_{31}]$$

λ is calculated by averaging the value of the Consistency Vector

$$\lambda = \sum_{i=1}^{n} Cv_{ij}$$

$$CI = \frac{\lambda - n}{n-1}$$

CI measures the deviation

$$C_r = \frac{CI}{RI}$$

$C_r < 0.10$

RANDOM INCONSISTENCY INDICES (RI) FOR $N - 10$

| $N$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| RI | 0.00 | 0.00 | 0.58 | 0.9 | 1.12 | 1.24 | 1.32 | 1.41 | 1.46 | 1.49 |

Source: Satty (1980).

Source: Haas, R. and Meixner, N.

# Pairwise Comparison

SIEMENS
Ingenuity for life

- The matrix "A" represents the judgments or relative importance of alternatives as matrix, where "*n*" is the number of items being evaluated.
- The entries of matrix "A," i.e., $\mathbf{a}_{ij}$ are the relative judgments between the two alternatives *i* and *j* in such a way that the *i*th row corresponds to the *j*th column of "A."

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} & \cdots & \mathbf{a}_{1n} \\ \mathbf{a}_{21} & \mathbf{a}_{22} & \cdots & \mathbf{a}_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{a}_{n1} & \mathbf{a}_{n2} & \cdots & \mathbf{a}_{nn} \end{bmatrix}. \tag{1}$$

$$\mathbf{a}_{ii} = 1 \iff i = j, \tag{2}$$

$$\mathbf{a}_{ij} = \frac{1}{\mathbf{a}_{ij}}, \tag{3}$$

where $\mathbf{a}_{ij}$ can also be written as

$$\mathbf{a}_{ij} = \frac{w_i}{w_j}, \tag{4}$$

where $w_i$ shows the relative weight of the alternative $i$.

# Deriving Relative Weights

SIEMENS
Ingenuity for life

$$
\begin{array}{c|cccccc}
C & A_1 & A_2 & A_3 & \ldots\ldots & A_n \\
\hline
A_1 & w_1/w_1 & w_1/w_2 & w_1/w_3 & \ldots\ldots & w_1/w_n \\
A_2 & w_2/w_1 & w_2/w_2 & w_2/w_3 & \ldots\ldots & w_2/w_n \\
 & w_3/w_1 & w_3/w_2 & w_3/w_3 & \ldots\ldots & w_3/w_n \\
A_3 & & & & & \\
, & & & & & , \\
A_n & w_n/w_1 & w_n/w_2 & w_n/w_3 & \ldots\ldots & w_n/w_n
\end{array}, \qquad (5)
$$

here, the aim is to find eigenvalues "$w$," where $w$ is

$$w = (w_1, w_2, w_3, \ldots, w_n), \qquad (6)$$

where "$w$" is the eigenvector and a column matrix.

- This step requires the estimation of relative weights for each of the criteria and sub-criteria of decision hierarchy.
- In this method, the corresponding weights of decision elements are determined by comparing the normalized eigenvalue to the principal eigenvalue
- It is calculated by multiplying each row of the above matrix. As there is "*n*" number of entries, take the *n*th root of the multiplication. Finally, the normalized roots are obtained by deriving the total and subsequently divide them by the total outcome.

# Priority Vectors: Eigen-Vectors

SIEMENS
*Ingenuity for life*

- Consider $[Ax = \lambda_{max}x]$ where
  - A is the comparison matrix of size $n \times n$, for n criteria, also called the priority matrix.
  - x is the Eigenvector of size $n \times 1$, also called the priority vector.
  - $\lambda_{max}$ is the Eigenvalue.

- To find the ranking of priorities, namely the Eigen Vector X:
  1) Normalize the column entries by dividing each entry by the sum of the column.
  2) Take the overall row averages.

$$A = \begin{bmatrix} 1 & 0.5 & 3 \\ 2 & 1 & 4 \\ 0.33 & 0.25 & 1.0 \end{bmatrix} \xrightarrow{\text{Normalized Column Sums}} \begin{bmatrix} 0.30 & 0.28 & 0.37 \\ 0.60 & 0.57 & 0.51 \\ 0.10 & 0.15 & 0.12 \end{bmatrix} \xrightarrow{\text{Row averages}} X = \begin{bmatrix} 0.32 \\ 0.56 \\ 0.12 \end{bmatrix}$$

Column sums 3.33 1.75 8.00     1.00 1.00 1.00

Priority vector

# Calculation of Consistency Ratio

SIEMENS
Ingenuity for life

- The next stage is to calculate $\lambda_{max}$ so as to lead to the Consistency Index and the Consistency Ratio.
- Consider $[Ax = \lambda_{max}\, x]$ where x is the Eigenvector.

$$\underbrace{\begin{bmatrix} 1 & 0.5 & 3 \\ 2 & 1 & 4 \\ 0.333 & 0.25 & 1.0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 0.32 \\ 0.56 \\ 0.12 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 0.98 \\ 1.68 \\ 0.36 \end{bmatrix}}_{Ax} = \lambda_{max} \underbrace{\begin{bmatrix} 0.32 \\ 0.56 \\ 0.12 \end{bmatrix}}_{x}$$

- $\lambda_{max}$=average{0.98/0.32, 1.68/0.56, 0.36/0.12}=3.04
- Consistency index , CI is found by CI=($\lambda_{max}$-n)/(n-1)=(3.04-3)/(3-1)= 0.02

# Consistency Ratio: CR<= 0.1

$C.R. = C.I./R.I.$ where R.I. is the random index

| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| R.I. | 0 | 0 | .52 | .88 | 1.11 | 1.25 | 1.35 |

$C.I. = 0.02$

$n = 3$

$R.I. = 0.50$(from table)

So, $C.R. = C.I./R.I. = 0.02/0.52 = 0.04$

$C.R. \leq 0.1$ indicates sufficient consistency for decision.

# Development of Priority Ranking

SIEMENS
Ingenuity for life

- The overall priority for each decision alternative is obtained by summing the product of the criterion priority (i.e., weight) (with respect to the overall goal) times the priority (i.e., preference) of the decision alternative with respect to that criterion.
- Ranking these priority values, we will have AHP ranking of the decision alternatives.

Step 0B: Calculate priority vector for each matrix.

| | Cost | Reliability | Risk |
|---|---|---|---|
| Prj-1 | 0.123 | 0.087 | 0.593 |
| Prj-2 | 0.320 | 0.274 | 0.341 |
| Prj-3 | 0.557 | 0.639 | 0.066 |

| | Criterion |
|---|---|
| Cost | 0.398 |
| Reliability | 0.085 |
| Risk | 0.218 |
| | 0.299 |

Step 1: Sum the product of the criterion priority (with respect to the overall goal) times the priority of the decision alternative with respect to that criterion.

Overall car A priority = 0.398 (0.123) + 0.085 (0.087) + 0.218 (0.593) + 0.299 (0.265) = 0.265

Step 2: Rank the priority values.

| Alternative | Priority |
|---|---|
| Prj-1 | 0.421 |
| Prj-2 | 0.314 |
| Prj-3 | 0.265 |
| Total | 1.000 |

# AHP

# REAMS Example:

## Ranking of Interventions in Time Bundle in UJ3

# Concept of Renewal Schedule Planning

SIEMENS
*Ingenuity for life*

ST Engineering

## Step 1: Assign Interventions to time Bundles:

- All renewal interventions have an Start and End date (time) by which the project should be implemented.
- Bundle projects based on Start Date.

Time Bundles

| Class | Scope | Facility | Duration (in years) | Date Start | Date End | FY2020 | FY2021 | FY2022 | FY2023 | FY2024 | FY2025 | FY2026 | FY2027 | FY2028 | FY2029 | FY2030 | FY2031 | FY2032 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RS | renewal of RS brakes system | Depot | 12 | 1-Apr-20 | 31-Mar-32 | | | | | | | | | | | | | |
| RS | Midlife upgrade | Depot | 4 | 1-Apr-29 | 31-Mar-33 | | | | | | | | | | | | | |
| RS | Replacement of trains VDU (Liquid crystal display/LED). The useful life of the VDU is 7 years. | Depot | 4 | 1-Apr-25 | 31-Mar-29 | | | | | | | | | | | | | |
| RS | Replacement of TIMS | Depot | 4 | 1-Apr-25 | 31-Mar-29 | | | | | | | | | | | | | |
| SIG | To replace ATP/ATO System | Depot | 5 | 1-Apr-20 | 31-Mar-25 | | | | | | | | | | | | | |
| SIG | To replace CBI | Depot | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | |
| SIG | To replace 200 Point Measuring System (PMS) | Depot | 2 | 1-Apr-24 | 31-Mar-26 | | | | | | | | | | | | | |
| SIG | Point equipment | Depot | 5 | 1-Apr-30 | 31-Mar-35 | | | | | | | | | | | | | |
| SIG | To replace ATP/ATO System | Stations | 5 | 1-Apr-20 | 31-Mar-25 | | | | | | | | | | | | | |
| SIG | Emergency Stop Plunger (ESP) and associated interlocking | Stations | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | |
| SIG | To replace CBI | Stations | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | |
| SIG | To replace 200 Point Measuring System (PMS) | Stations | 2 | 1-Apr-24 | 31-Mar-26 | | | | | | | | | | | | | |
| SIG | Point equipment | Stations | 5 | 1-Apr-30 | 31-Mar-35 | | | | | | | | | | | | | |
| SIG | To replace ATP/ATO System | Track | 5 | 1-Apr-20 | 31-Mar-25 | | | | | | | | | | | | | |
| SIG | Emergency Stop Plunger (ESP) and associated interlocking | Track | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | |
| SIG | To replace CBI | Track | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | |
| SIG | To replace 200 Point Measuring System (PMS) | Track | 2 | 1-Apr-24 | 31-Mar-26 | | | | | | | | | | | | | |
| SIG | Point equipment | Track | 5 | 1-Apr-30 | 31-Mar-35 | | | | | | | | | | | | | |

## Step 2: Ranking with AHP to De-conflict

- Derives priorities among criteria and alternatives (Performance, Cost or Risk)

$$AHP_i = \sum_{j=1}^{n} \frac{a_{ij}}{\sum_{i=1}^{m} a_{ij}} \times w_j$$

| Alternative | AHP Score | Rank |
|---|---|---|
| C | 0.0835 | 1 |
| B | 0.0823 | 2 |
| I | 0.0798 | 3 |
| L | 0.0784 | 4 |
| M | 0.0773 | 5 |
| A | 0.0695 | 6 |
| K | 0.0659 | 7 |
| G | 0.0642 | 8 |

## Step 3: Reassigning of Low-ranked interventions after AHP

- After AHP, lower ranked project will be reassigned to another time bundle with no conflict
- Taking some rules into consideration, a logic will be coded to do the re-assignment of projects automatically.

Move RS project to FY2021 to DE-CONFLICT

| Class | Scope | Facility | Duration (in years) | Date Start | Date End | FY2020 | FY2021 | FY2022 | FY2023 | FY2024 | FY2025 | FY2026 | FY2027 | FY2028 | FY2029 | FY2030 | FY2031 | FY2032 | Rank FY2020 Based on Risk |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RS | renewal of RS brakes system | Depot | 12 | 1-Apr-20 | 31-Mar-32 | | | | | | | | | | | | | | 1 |
| RS | Midlife upgrade | Depot | 4 | 1-Apr-29 | 31-Mar-33 | | | | | | | | | | | | | | |
| RS | Replacement of trains VDU (Liquid crystal display/LED). The useful life of the VDU is 7 years. | Depot | 4 | 1-Apr-25 | 31-Mar-29 | | | | | | | | | | | | | | |
| RS | Replacement of TIMS | Depot | 4 | 1-Apr-25 | 31-Mar-29 | | | | | | | | | | | | | | |
| SIG | To replace ATP/ATO System | Depot | 5 | 1-Apr-20 | 31-Mar-25 | | | | | | | | | | | | | | 2 |
| SIG | To replace CBI | Depot | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | | |
| SIG | To replace 200 Point Measuring System (PMS) | Depot | 2 | 1-Apr-24 | 31-Mar-26 | | | | | | | | | | | | | | |
| SIG | Point equipment | Depot | 5 | 1-Apr-30 | 31-Mar-35 | | | | | | | | | | | | | | |
| SIG | To replace ATP/ATO System | Stations | 5 | 1-Apr-20 | 31-Mar-25 | | | | | | | | | | | | | | 3 |
| SIG | Emergency Stop Plunger (ESP) and associated interlocking | Stations | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | | |
| SIG | To replace CBI | Stations | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | | |
| SIG | To replace 200 Point Measuring System (PMS) | Stations | 2 | 1-Apr-24 | 31-Mar-26 | | | | | | | | | | | | | | |
| SIG | Point equipment | Stations | 5 | 1-Apr-30 | 31-Mar-35 | | | | | | | | | | | | | | |
| SIG | To replace ATP/ATO System | Track | 5 | 1-Apr-20 | 31-Mar-25 | | | | | | | | | | | | | | 4 |
| SIG | Emergency Stop Plunger (ESP) and associated interlocking | Track | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | | |
| SIG | To replace CBI | Track | 6 | 1-Apr-23 | 31-Mar-29 | | | | | | | | | | | | | | |
| SIG | To replace 200 Point Measuring System (PMS) | Track | 2 | 1-Apr-24 | 31-Mar-26 | | | | | | | | | | | | | | |
| SIG | Point equipment | Track | 5 | 1-Apr-30 | 31-Mar-35 | | | | | | | | | | | | | | |

# Analytic hierarchy process

SIEMENS

Ingenuity for life

A. Users of the AHP **first decompose their decision problem into a hierarchy of more easily comprehended sub-problems, each of which can be analyzed independently.**
B. The elements of the hierarchy can be related to any aspect of the decision problem tangible or intangible, carefully measured or roughly estimated, well or poorly understood anything at all that applies to the decision at hand.
C. Once the hierarchy is built, **the decision makers systematically evaluate its various elements by comparing them with one another two at a time, with respect to their impact on an element above them in the hierarchy**.
D. In making the comparisons, the decision makers can use concrete data about the elements, but **they typically use their judgements about the elements' relative meaning and importance**. It is the essence of the AHP that human judgments, and not just the underlying information, can be used in performing the evaluations.
E. The AHP converts these evaluations to numerical values that can be processed and compared over the entire range of the problem.
F. A numerical weight or priority is derived for each element of the hierarchy, allowing diverse and often incommensurable elements to be compared to one another in a rational and consistent way.
G. This capability distinguishes the AHP from other decision making techniques.
H. In the final step of the process, **numerical priorities are calculated for each of the decision alternatives.**
I. These numbers represent the alternatives' relative ability to achieve the decision goal, so they allow a straightforward consideration of the various courses of action.

# REAMS Example: Ranking of Interventions within a time Bundle

SIEMENS
Ingenuity for life

Summary table:

| Criteria | Alternatives | Evaluators |
|---|---|---|
| Cost | Intervention 1 | Michael |
| Reliability | Intervention 2 | |
| Risk | Intervention 3 | |
| | Intervention 4 | |
| | Intervention 5 | |
| | Intervention 6 | |

Table of Saaty:

| Value | Definition | Comments |
|---|---|---|
| 1 | Equal importance | Two elements contribute equally to the objective |
| 3 | Moderate importar | Judgment slightly favors one element over another |
| 5 | Strong importance | Judgment strongly favors one element over another |
| 7 | Very strong importa | Judgment strongly favors one element over another, its dominance is demonstrated by experience |
| 9 | Extreme importanc | The dominance of one element over another is demonstrated and absolute |
| 2, 4, 6, 8 | can be used to expr | |
| Reciprocity | If the element i has | |

*Use the Saaty table values to evaluate the set of comparison matrices below.*

*A value x of Saaty on the line i and the column j of a matrix means that the element i has an importance of the value x over the element j. On the contrary, the element of line j and column i has a value of 1/x.*

*Only cells above the diagonal must be entered.*

Assuming Interventions 1-6 are in a time Bundle and Interventions (1,3,5) are conflicting with Interventions(2,4,6)

Which Projects to reschedule?

Setting up an AHP analysis

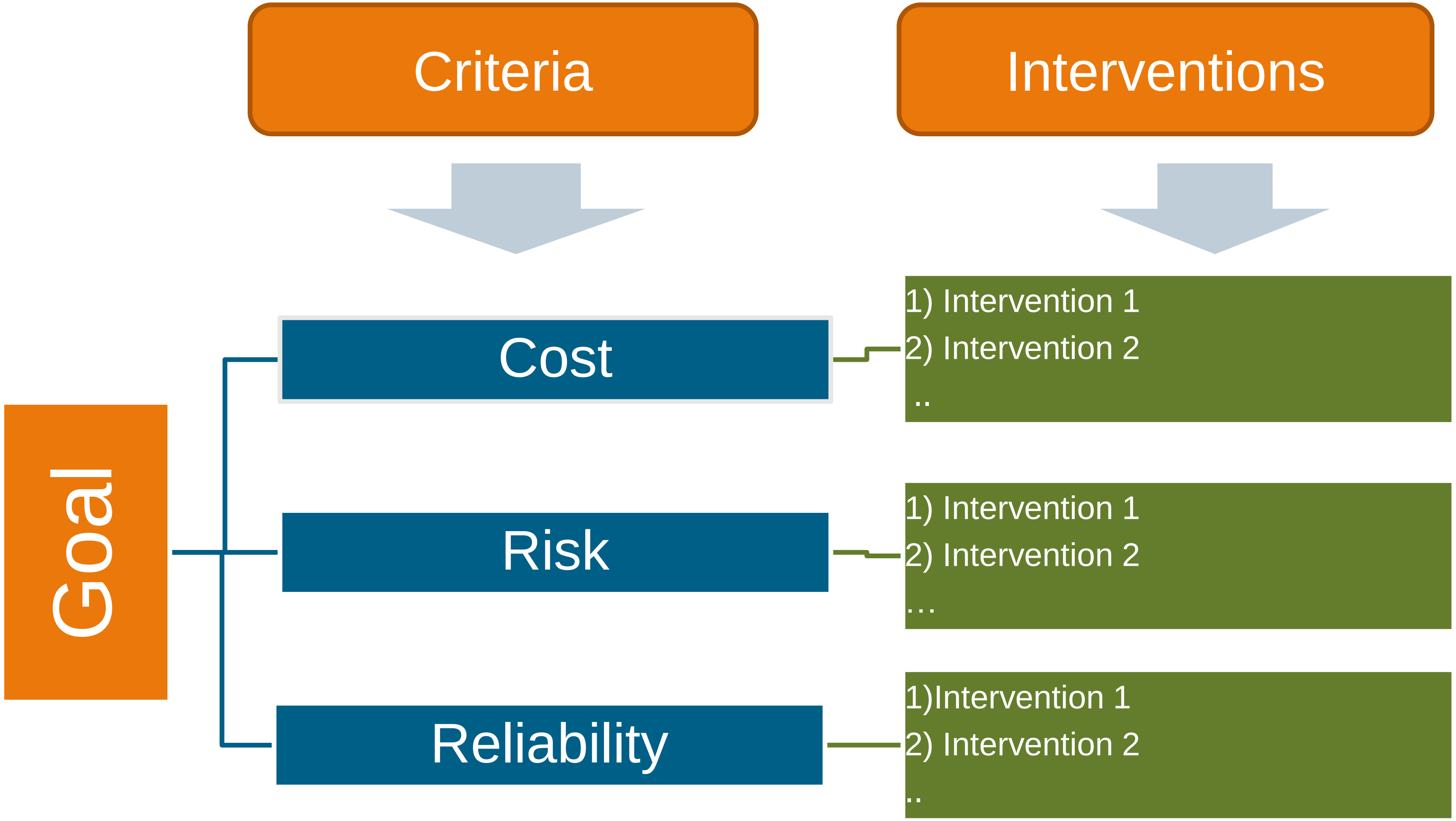
**Step 1: Generate the AHP design**

**Step 2: Run the AHP analysis**

Interpreting the results of an AHP analysis

# First decompose decision problem into a hierarchy of more easily comprehended  sub-problems

SIEMENS
Ingenuity for life



# Pairwise Comparison

## Comparison Matrix for Criteria and Alternatives

Once the hierarchy is built, **the decision makers systematically evaluate its various elements by comparing them with one another two at a time, with respect to their impact on an element above them in the hierarchy**

Comparative matrices of evaluator Michael:

On criteria:

| Criteria | Cost | Reliability | Risk |
|---|---|---|---|
| Cost | 1 | 3 | 7 |
| Reliability | 0.333333333 | 1 | 9 |
| Risk | 0.142857143 | 0.11111111 | 1 |

Alternatives for criterion Cost:

| Alternatives | Intervention 1 | Intervention 2 | Intervention 3 | Intervention 4 | Intervention 5 | Intervention 6 |
|---|---|---|---|---|---|---|
| Intervention 1 | 1 | 3 | 5 | 7 | 1 | 9 |
| Intervention 2 | 0.333333333 | 1 | 5 | 5 | 7 | 3 |
| Intervention 3 | 0.2 | 0.2 | 1 | 3 | 5 | 7 |
| Intervention 4 | 0.142857143 | 0.2 | 0.33333333 | 1 | 1 | 5 |
| Intervention 5 | 1 | 0.14285714 | 0.2 | 1 | 1 | |
| Intervention 6 | 0.111111111 | 0.33333333 | 0.14285714 | 0.2 | | 1 |

Alternatives for criterion Reliability:

| Alternatives | Intervention 1 | Intervention 2 | Intervention 3 | Intervention 4 | Intervention 5 | Intervention 6 |
|---|---|---|---|---|---|---|
| Intervention 1 | 1 | 5 | 9 | 7 | 3 | 1 |
| Intervention 2 | 0.2 | 1 | 5 | 3 | 5 | 7 |
| Intervention 3 | 0.11111111 | 0.2 | 1 | 9 | 7 | 3 |
| Intervention 4 | 0.14285714 | 0.33333333 | 0.111111111 | 1 | 9 | 5 |
| Intervention 5 | 0.33333333 | 0.2 | 0.142857143 | 0.111111111 | 1 | 9 |
| Intervention 6 | 1 | 0.14285714 | 0.333333333 | 0.2 | 0.11111111 | 1 |

Alternatives for criterion Risk:

| Alternatives | Intervention 1 | Intervention 2 | Intervention 3 | Intervention 4 | Intervention 5 | Intervention 6 |
|---|---|---|---|---|---|---|
| Intervention 1 | 1 | 5 | 7 | 9 | 5 | 3 |
| Intervention 2 | 0.2 | 1 | 3 | 5 | 7 | 9 |
| Intervention 3 | 0.142857 | 0.333333 | 1 | 3 | 9 | 5 |
| Intervention 4 | 0.111111 | 0.2 | 0.333333 | 1 | 3 | 7 |
| Intervention 5 | 0.2 | 0.142857 | 0.111111 | 0.333333 | 1 | 5 |
| Intervention 6 | 0.333333 | 0.111111 | 0.2 | 0.142857 | 0.2 | 1 |

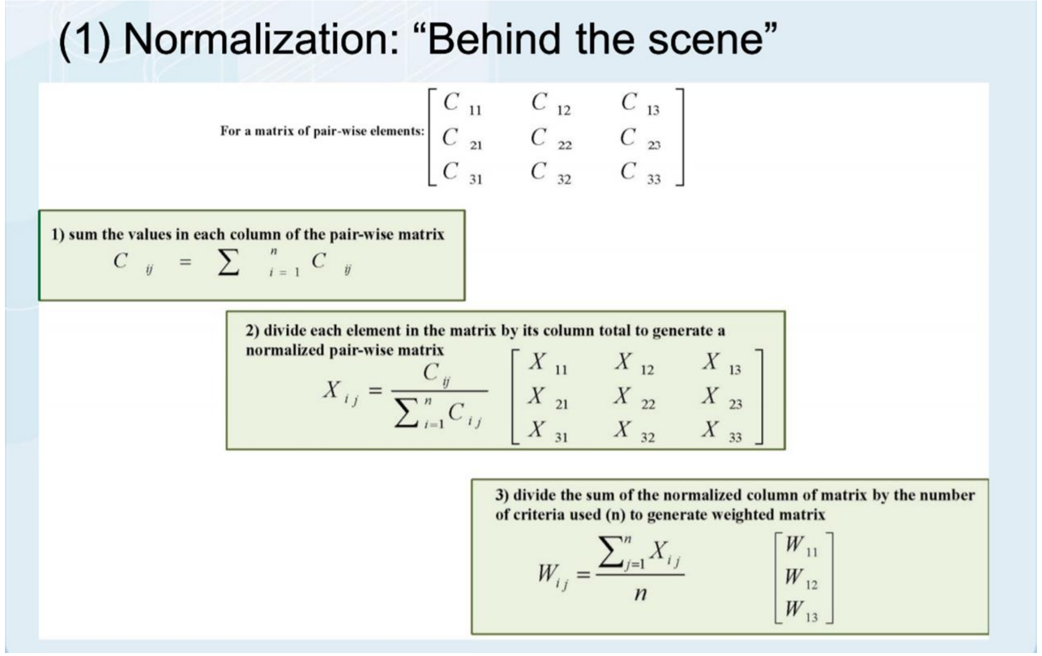
**Run the analysis**

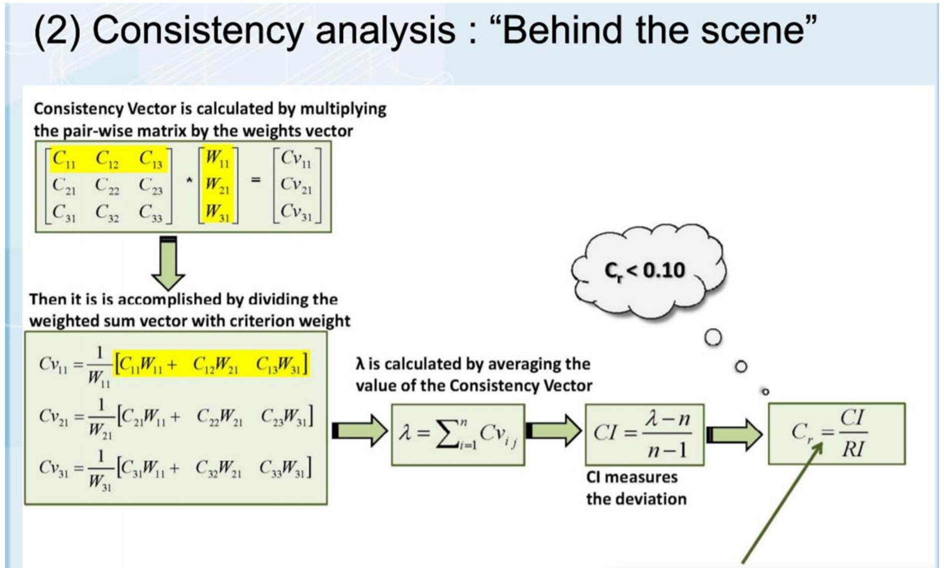
*Once you have completed the matrices you can click the button above to start the AHP analysis*

i. The **Saaty** table provides the values to be used by the decision-maker in order to fill in the comparison tables.
ii. Above is an example of filling in the criteria comparison table by the Decision-Maker Michael.

# The AHP converts these evaluations to numerical values that can be processed and compared over the entire range of the problem

SIEMENS
Ingenuity for life




RANDOM INCONSISTENCY INDICES (RI) FOR N − 10

| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| RI | 0.00 | 0.00 | 0.58 | 0.9 | 1.12 | 1.24 | 1.32 | 1.41 | 1.46 | 1.49 |

Source: Satty (1980).

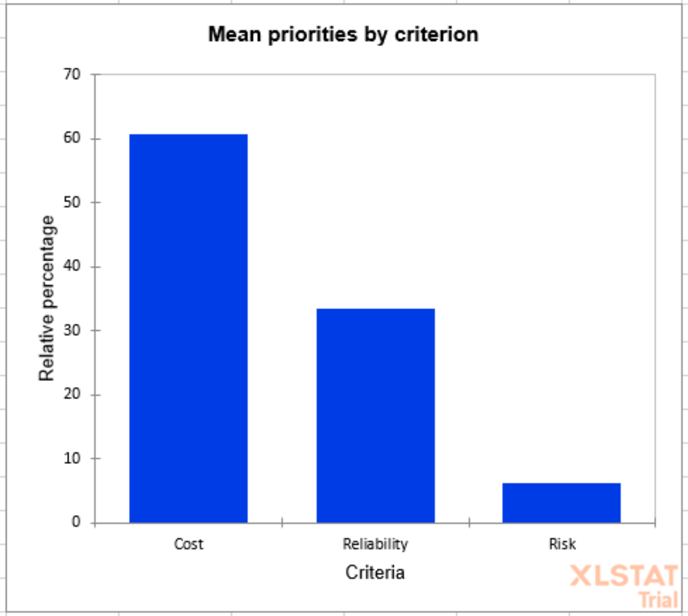
$$\text{final weight of criteria} = \sum \left[ (\text{weight of alternatives w.r.t criteria}) \times (\text{importance of criteria}) \right].$$

# Synthesizing the results:
## Mean priorities by Criterion & Alternatives

*Results obtained by replacing the missing data by the equity value 1*

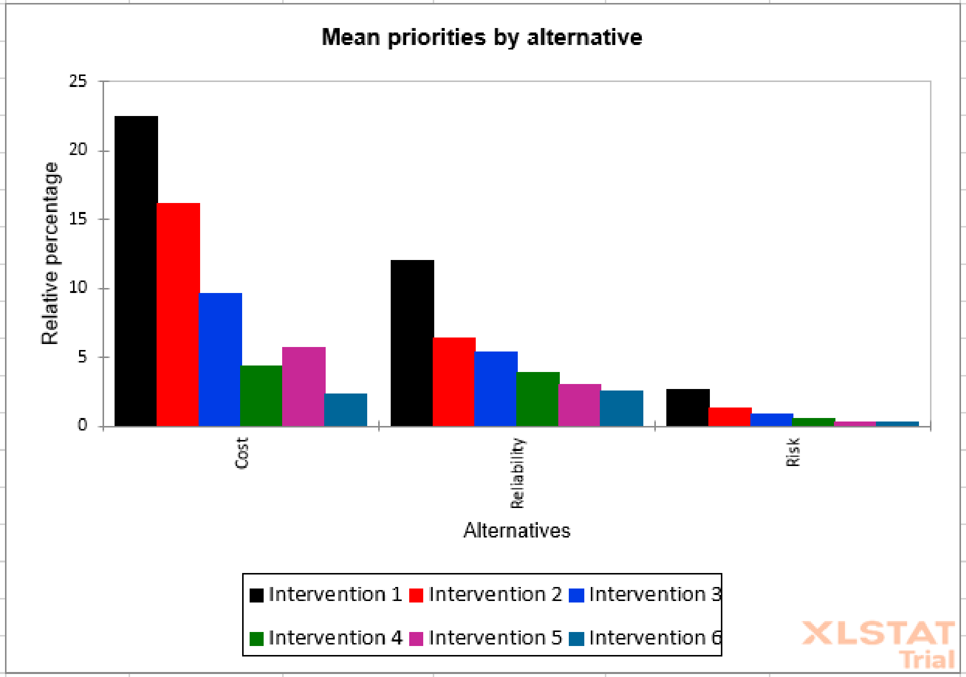
Mean priorities by criterion:

| Criteria | % |
|---|---|
| Cost | 60.63 |
| Reliability | 33.28 |
| Risk | 6.09 |




Mean priorities by alternative:

| Crit./Alt. | Intervention 1 | Intervention 2 | Intervention 3 | Intervention 4 | Intervention 5 | Intervention 6 |
|---|---|---|---|---|---|---|
| Cost | 22.41 | 16.13 | 9.65 | 4.38 | 5.70 | 2.35 |
| Reliability | 12.01 | 6.40 | 5.38 | 3.92 | 3.05 | 2.52 |
| Risk | 2.66 | 1.37 | 0.91 | 0.53 | 0.36 | 0.25 |



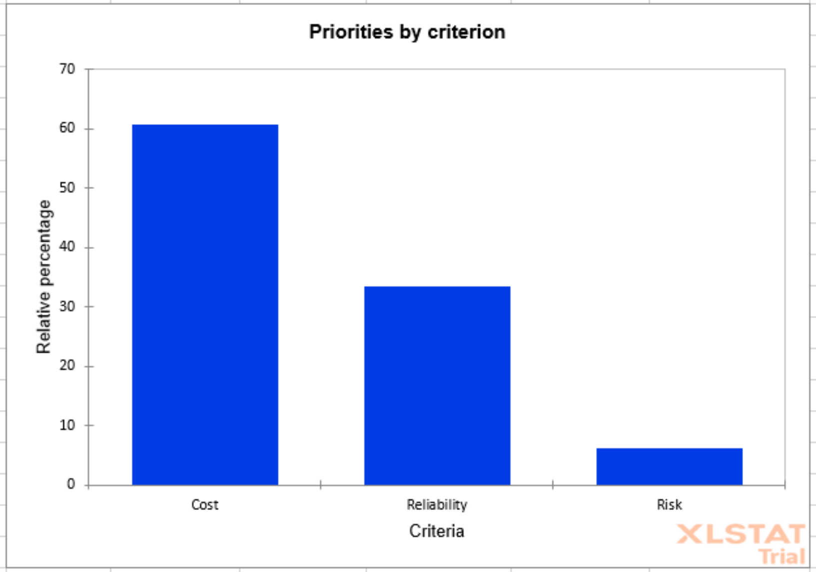
# Checking the Consistency Ratio

SIEMENS
Ingenuity for life

Priorities by criterion:

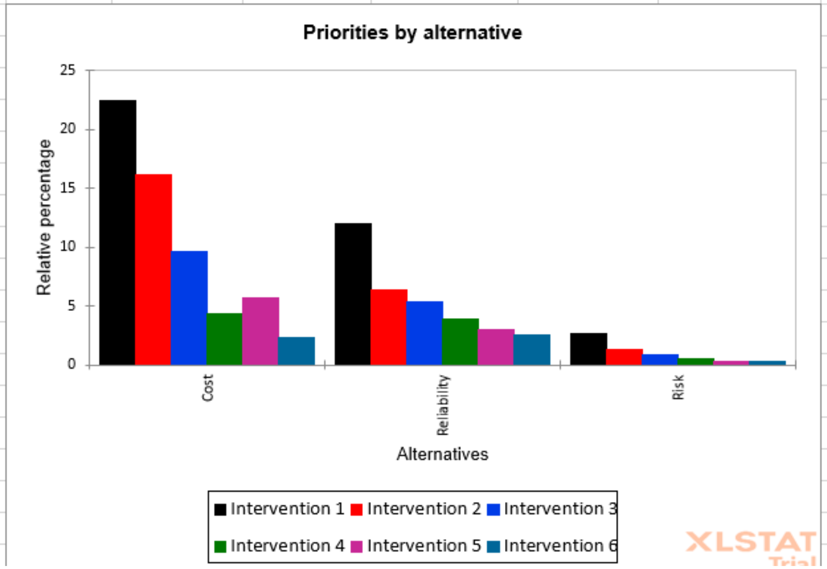
| Criteria | % |
|---|---|
| Cost | 60.63 |
| Reliability | 33.28 |
| Risk | 6.09 |

*IC = 0.106 ; RC = 18.2%*




Priorities by alternative:

| Crit./Alt. | Intervention 1 | Intervention 2 | Intervention 3 | Intervention 4 | Intervention 5 | Intervention 6 | IC | RC |
|---|---|---|---|---|---|---|---|---|
| Cost | 22.41 | 16.13 | 9.65 | 4.38 | 5.70 | 2.35 | 0.34 | 27.40 |
| Reliability | 12.01 | 6.40 | 5.38 | 3.92 | 3.05 | 2.52 | 1.04 | 83.61 |
| Risk | 2.66 | 1.37 | 0.91 | 0.53 | 0.36 | 0.25 | 0.40 | 32.05 |




a. In order to calculate the consistency ratio, an index was formulated to measure the consistency of weights.
b. In this regard, the acceptable range of **CR should be equal to or less than 0.10**.
c. However, **a revision in the pairwise comparison is compulsory, if CR is greater than this boundary value**

# AHP Ranking

SIEMENS
Ingenuity for life

a. The final step starts from the summation of relative values for each set of alternatives on all hierarchy levels.
b. These values are combined together to establish the overall score or criteria weight of each alternative.
c. As an outcome, the normalized local priority vectors are obtained due to this additional function.

Priorities by alternative:

| Crit./Alt. | Intervention 1 | Intervention 2 | Intervention 3 | Intervention 4 | Intervention 5 | Intervention 6 | IC | RC |
|---|---|---|---|---|---|---|---|---|
| **Cost** | 22.41 | 16.13 | 9.65 | 4.38 | 5.70 | 2.35 | 0.34 | 27.40 |
| **Reliability** | 12.01 | 6.40 | 5.38 | 3.92 | 3.05 | 2.52 | 1.04 | 83.61 |
| **Risk** | 2.66 | 1.37 | 0.91 | 0.53 | 0.36 | 0.25 | 0.40 | 32.05 |
| | 37.09 | 23.91 | 15.93 | 8.83 | 9.11 | 5.13 | | |
| **Rank** | 1 | 2 | 3 | 5 | 4 | 6 | | |

**Priorities by alternative**

$$\text{final weight of criteria} = \sum \left[ (\text{weight of alternatives w.r.t criteria}) \times (\text{importance of criteria}) \right].$$