Heuristic algorithms

J.L. Redondo

Outline

1. Algorithm UEGO

2. Algorithm FEMOEA

Outline

1. Algorithm UEGO

2. Algorithm FEMOEA

1. The algorithm UEGO

Introduction

- The main objective of any optimization technique is to find the global optimum (or optima) of any problem.

- Calculate:

$$\min f(x)$$

$$\text{subject to } x \in X$$

X is a closed set of $\mathbf{R}^n$

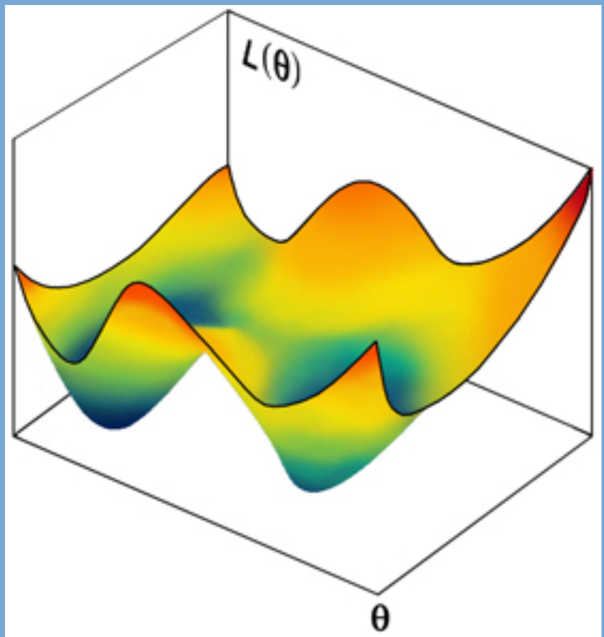
$f(.) : \Omega \to \mathbf{R}$ is a function in $\Omega \subset \mathbf{R}^n$

to find x^* y f^* such that

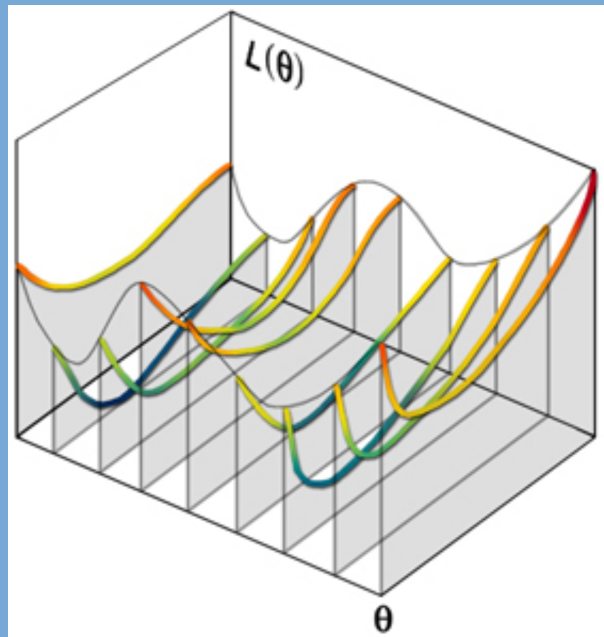
$$f^* = f(x^*) \leq f(x) \quad \forall x \in X$$

1. The algorithm UEGO

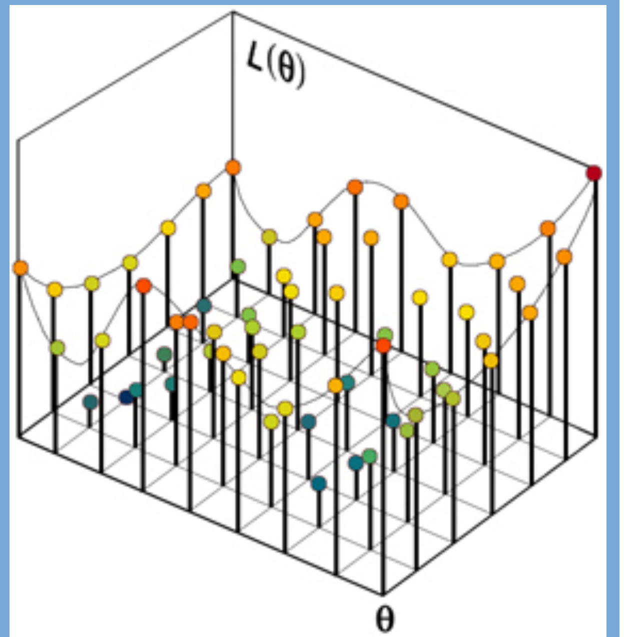
Introduction: Three common functions



Continuous



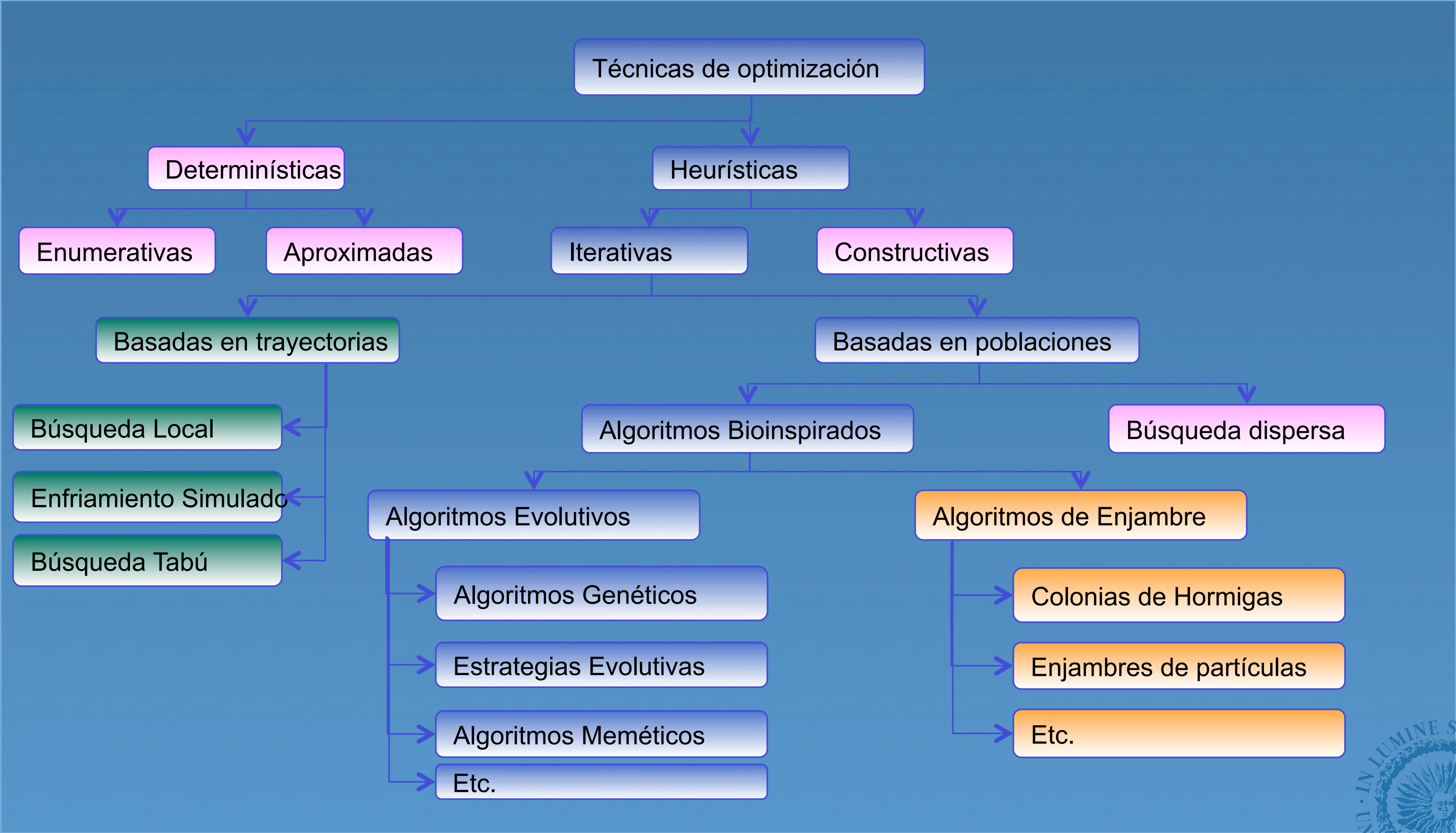
Discrete/Continuous



Discrete

1. The algorithm UEGO

Introduction: Classification of the global methods



1. The algorithm UEGO

Introduction: Classification of the global methods

Deterministic methods:

- They are guaranteed to obtain a solution to a given accuracy in a finite number of steps.
- They require an understanding of the mathematical structure of the problem to be solved.
- An example are the Branch-and-Bound methods that can be applied to any problem analytically describable.

1. The algorithm UEGO

Introduction: Classification of the global methods

Heuristic methods:

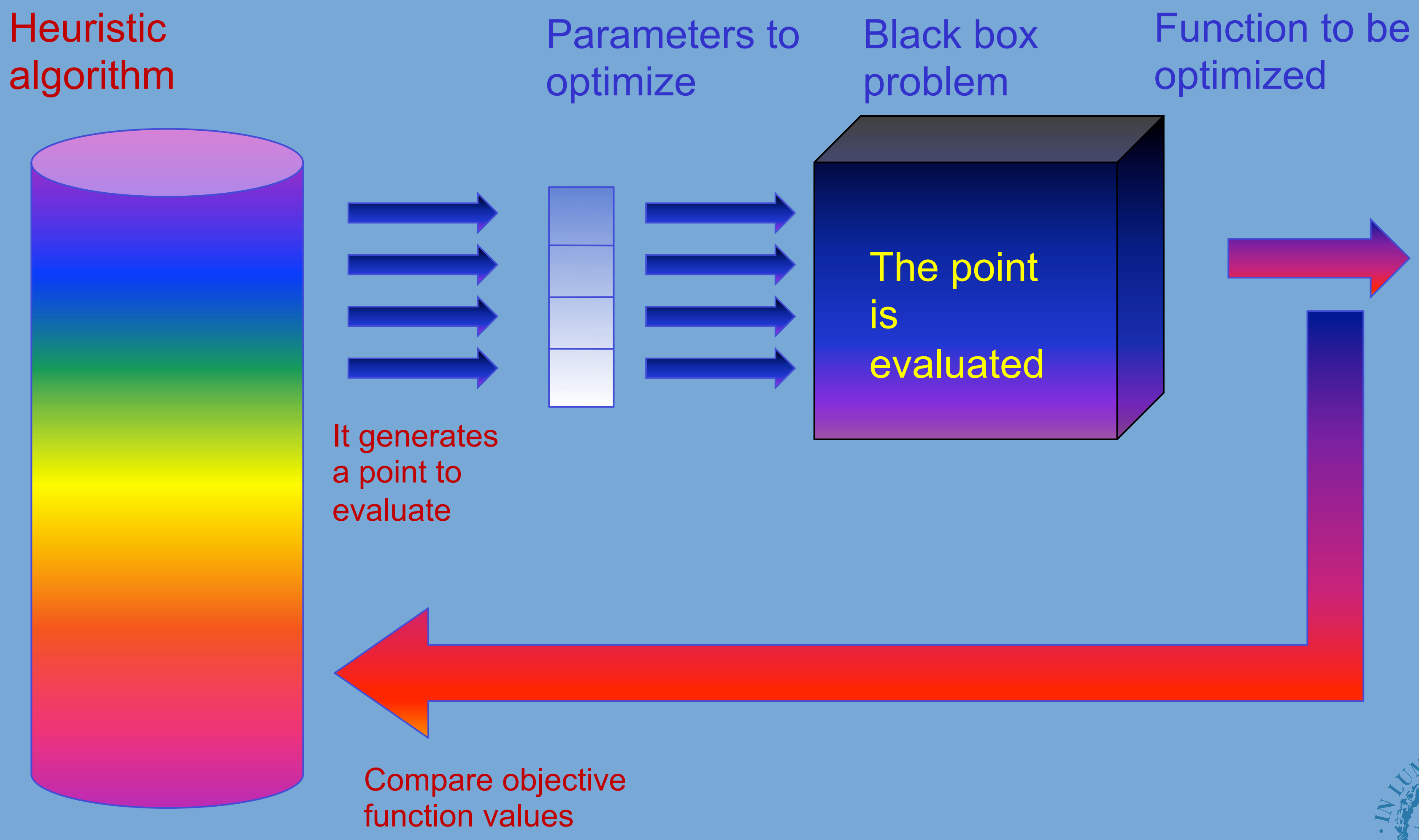
- They don't guarantee the absolute convergence towards a global solution. The probability of getting a solution in a global optimum environment tends to one when the number of sampled points tends to infinity.

- They don't need to know the mathematical structure of problem to solve because they are based on the random generation of feasible points and local optimization routines.

- Thus can be applied to any problem, even black box or unstructured problems.

1. The algorithm UEGO

Introduction: Classification of the global methods

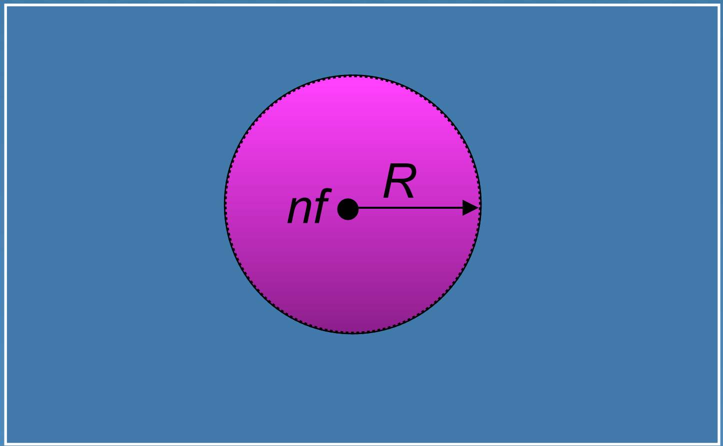


1. The algorithm UEGO

The algorithm UEGO

Basic concepts:

- The concept of cluster or individual is named species. A species is defined by its center and a radius (and a fitness value).



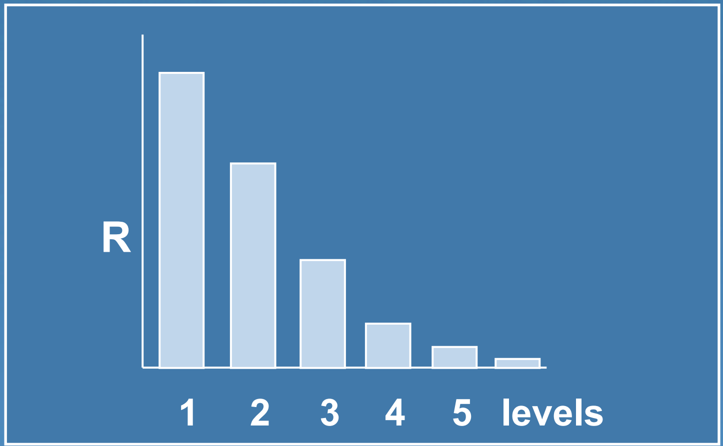
- The parameter "*levels*" indicates the maximum number of iterations in the algorithm.
- Radius list is a list with the decreasing value of the radii associated to each level.
- During the optimization process, UEGO manages a list of species.

1. The algorithm UEGO

The algorithm UEGO

Basic concepts:

- The concept of cluster or individual is named species. A species is defined by its center and a radius (and a fitness value).
- The parameter "*levels*" indicates the maximum number of iterations in the algorithm.



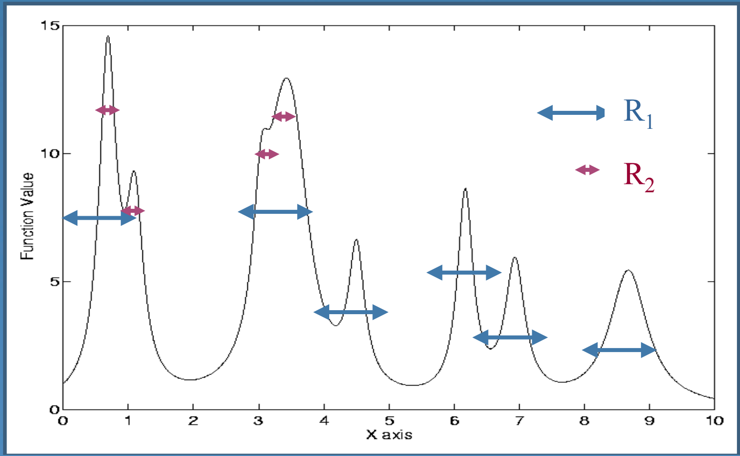
- Radius list is a list with the decreasing value of the radii associated to each level.
- During the optimization process, UEGO manages a list of species.

1. The algorithm UEGO

The algorithm UEGO

Basic concepts:

- The concept of cluster or individual is named species. A species is defined by its center and a radius (and a fitness value).



- The parameter *"levels"* indicates the maximum number of iterations in the algorithm.

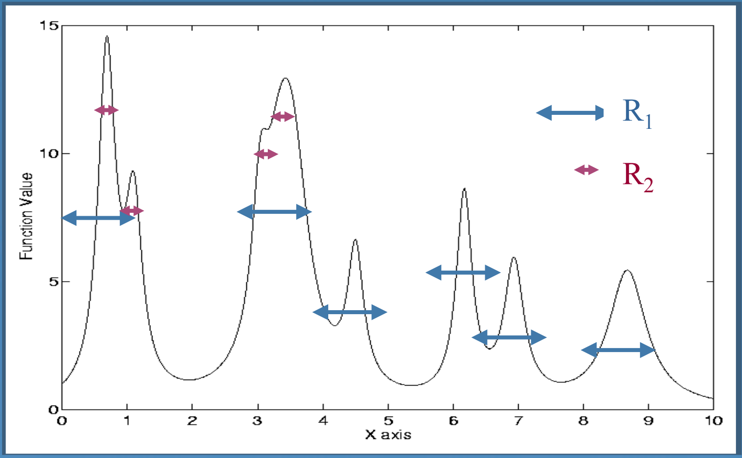
- Radius list is a list with the decreasing value of the radii associated to each level.

- During the optimization process, UEGO manages a list of species.

The algorithm UEGO

Basic concepts:

- The concept of cluster or individual is named species. A species is defined by its center and a radius (and a fitness value).



- The parameter "*levels*" indicates the maximum number of iterations in the algorithm.

- Radius list is a list with the decreasing value of the radii associated to each level.

- During the optimization process, UEGO manages a list of species.

1. The algorithm UEGO

The algorithm UEGO

User given parameters

Evals (N): The maximum number of function evaluations for the whole optimization process.

levels (L): The maximum number of levels.

max_spec_ num (M): The maximum length of the species list.

min r (R_L): The radius that is associated with the maximum level.

Parameter at each level

R_i : Radius associated with level i.

new_i : Maximum number of function evaluations allowed when creating new species.

n_i : Maximum number of function evaluations allowed when optimizing species.

1. The algorithm UEGO

The algorithm UEGO

Algorithm: Algorithm UEGO

Init species list
Optimize species(n_1)
FOR i = 2 to L
    Determine R_i , new_i , n_i
    Create species(new_i)   # budget per species = new_i /length(species_list$_i$)
    Fuse species(R_i)
    Shorten species list(M)
    Optimize species(n_i)   # budget per species = n_i /max_spec_num
    Fuse species(R_i)

1. The algorithm UEGO

The algorithm UEGO

Algorithm: Algorithm UEGO

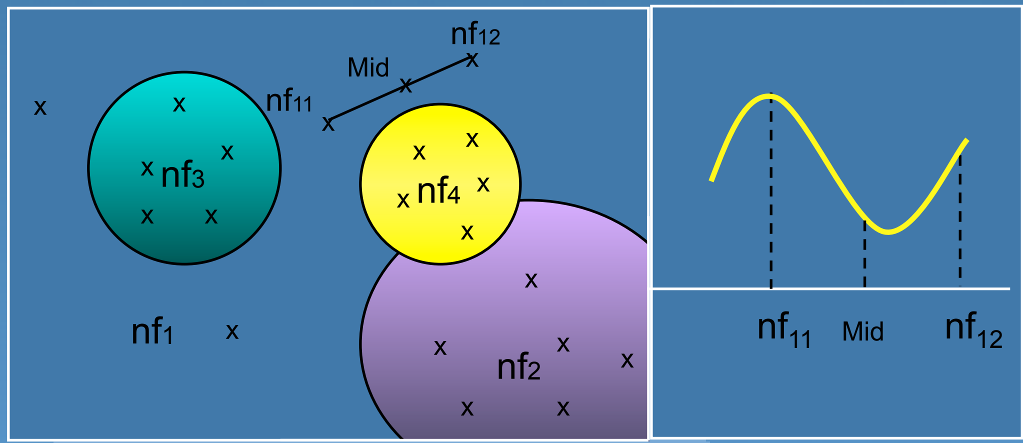
Init species list
Optimize species(n_1)
FOR i = 2 to L
    Determine R_i , new_i , n_i
    Create species(new_i)
    Fuse species(R_i)
    Shorten species list(M)
    Optimize species(n_i)
    Fuse species(R_i)

1. The algorithm UEGO

The algorithm UEGO

Algorithm: Algorithm UEGO

Init species list
Optimize species(n_1)
FOR i = 2 to L
    Determine R_i , new_i , n_i
    Create species(new_i)
    Fuse species(R_i)
    Shorten species list(M)
    Optimize species(n_i)
    Fuse species(R_i)

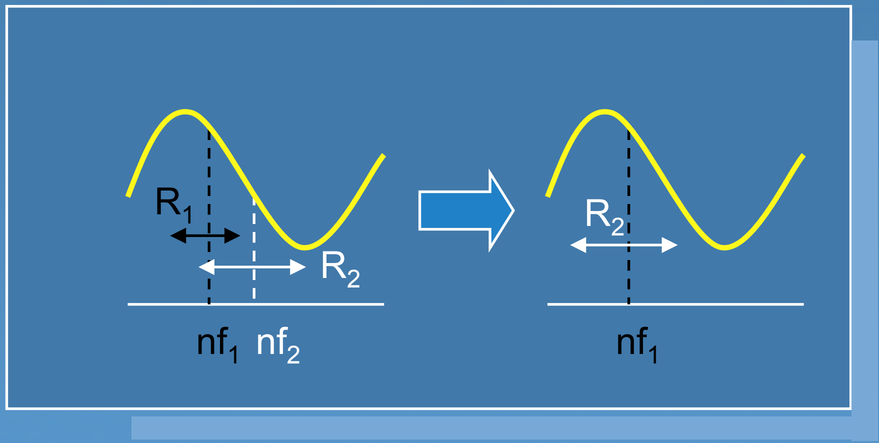


1. The algorithm UEGO

The algorithm UEGO

Algorithm: Algorithm UEGO

```
Init species list
Optimize species(n_1)
FOR i = 2 to L
        Determine R_i , new_i , n_i
        Create species(new_i )
        Fuse species(R_i )
        Shorten species list(M)
        Optimize species(ni)
        Fuse species(R_i )
```



1. The algorithm UEGO

The algorithm UEGO

Algorithm: Algorithm UEGO

```
Init species list
Optimize species(n_1)
FOR i = 2 to L
        Determine R_i , new_i , n_i
        Create species(new_i )
        Fuse species(R_i )
        Shorten species list(M)
        Optimize species(n_i)
        Fuse species(R_i )
```

nf_1	R_1	$\Pi(nf_1)$
nf_2	R_2	$\Pi(nf_2)$
nf_3	R_2	$\Pi(nf_3)$
nf_4	R_3	$\Pi(nf_4)$

$\Rightarrow$

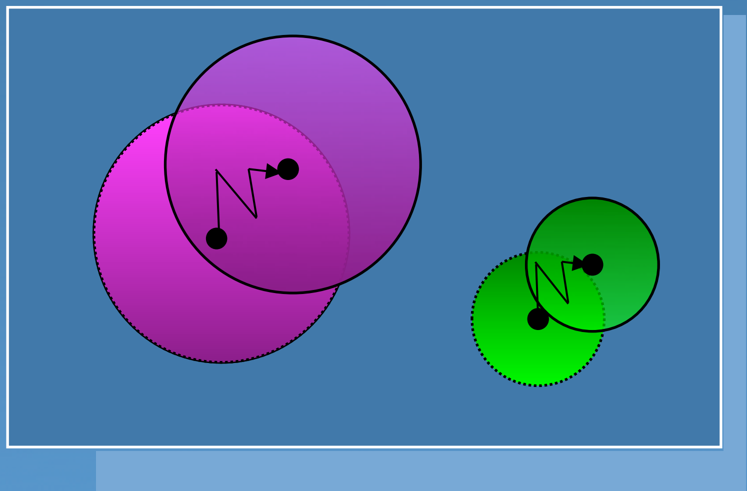
nf_1	R_1	$\Pi(nf_1)$
nf_2	R_2	$\Pi(nf_2)$
nf_3	R_2	$\Pi(nf_3)$

1. The algorithm UEGO

The algorithm UEGO

Algorithm: Algorithm UEGO

1 Init species list
2 Optimize species(n_1)
3 FOR i = 2 to L
4 Determine R_i , new_i , n_i
5 Create species(new_i)
6 Fuse species(R_i)
7 Shorten species list(M)
8 Optimize species(n_i)
9 Fuse species(R_i)



1. The algorithm UEGO

HPC with the algorithm UEGO

Algorithm: Algorithm UEGO

```
Init species list
Optimize species(n1)
FOR i = 2 to L
        Determine R_i , new_i , n_i
        Create species(new_i )
        Fuse species(R_i )
        Shorten species list(M)
        Optimize species(ni)
        Fuse species(R_i )
```

1. The algorithm UEGO

HPC with the algorithm UEGO

Algorithm: Algorithm UEGO

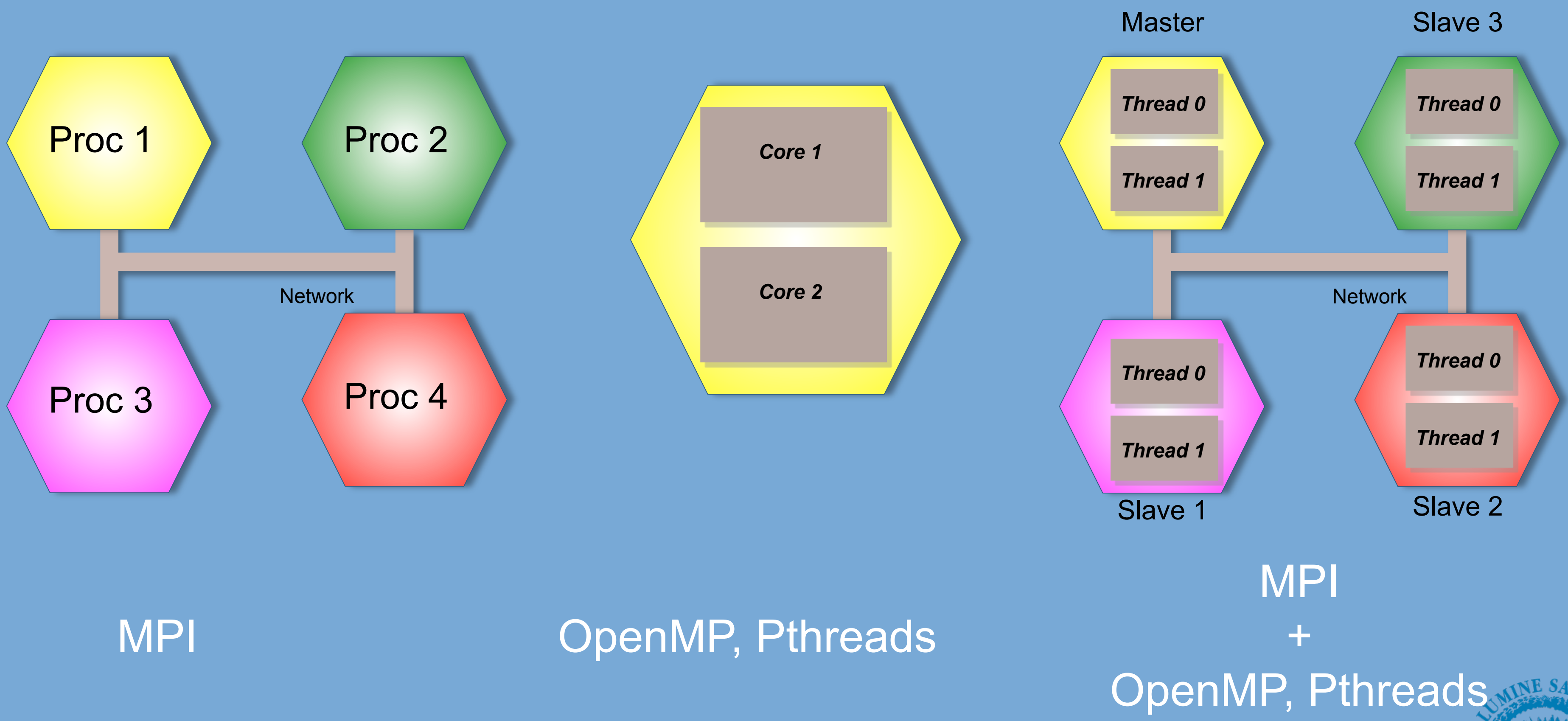
```
Init species list
Optimize species(n1)
FOR i = 2 to L
        Determine R_i , new_i , n_i
        Create species(new_i )
        Fuse species(R_i )
        Shorten species list(M)
        Optimize species(ni)
        Fuse species(R_i )
```

STOP

STOP

1. The algorithm UEGO

HPC with the algorithm UEGO



1. The algorithm UEGO

Applications

- Location problems
- Electron tomography
- Multiscaling problems
- Layout of tower solar plants.
- Optimization of brain parameters.
- Etc.

Outline

1. Algorithm UEGO

2. Algorithm FEMOEA

2. Algorithm FEMOEA

Mathematical formulation

Problem to solve

$$\min \{\pi_1(nf),\ldots,\pi_m(nf)\}$$
$$s.t.\quad nf \in S \subseteq \Re^n$$

$\pi_1,\ldots,\pi_m : \Re^n \to \Re$ .... are *m* real-valued functions

$\pi(nf) = (\pi_1(nf),\ldots,\pi_m(nf))$ .... Vector of the objective functions

$Z = \pi(S)$ .... Image of the feasible region

2. Algorithm FEMOEA

Previous heuristic approaches

NSGA-II [1]

SPEA2 [2]

MOEAD [3]

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2): 182-197, 2002.

[2] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, K. D. Papailiou, and T. Fogarty, editors, Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, pages 95-100, Athens,Greece, 2002. International Center for Numerical Methods in Engineering (CIMNE).

[3] Q. Zhang and H. Li. A multi-objective evolutionary algorithm based on decomposition. IEEE Transaction on evolutionary computation, 11(6), 712-731, 2007.

2. Algorithm FEMOEA

Heuristic previous approaches

NSGA-II [1]

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2): 182-197, 2002.

SPEA2 [2]

[2] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, K. D. Papailiou, and T. Fogarty, editors, Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, pages 95-100, Athens,Greece, 2002. International Center for Numerical Methods in Engineering (CIMNE).

MOEAD [3]

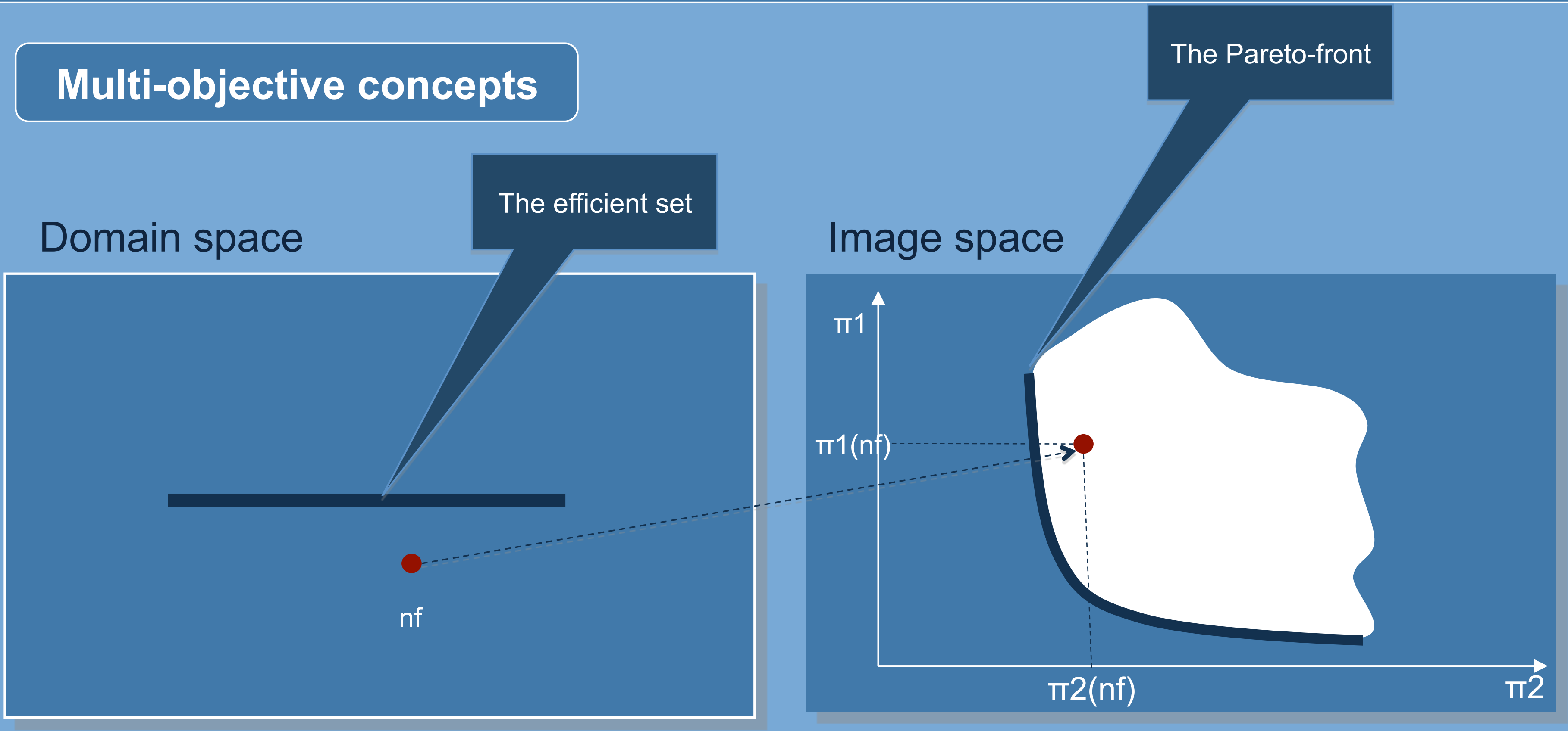
[3] Q. Zhang and H. Li. A multi-objective evolutionary algorithm based on decomposition. IEEE Transaction on evolutionary computation, 11(6), 712-731, 2007.

FEMOEA

1. A new termination rule
2. A new local method
3. A decreasing radius method to search in the space.

2. Algorithm FEMOEA

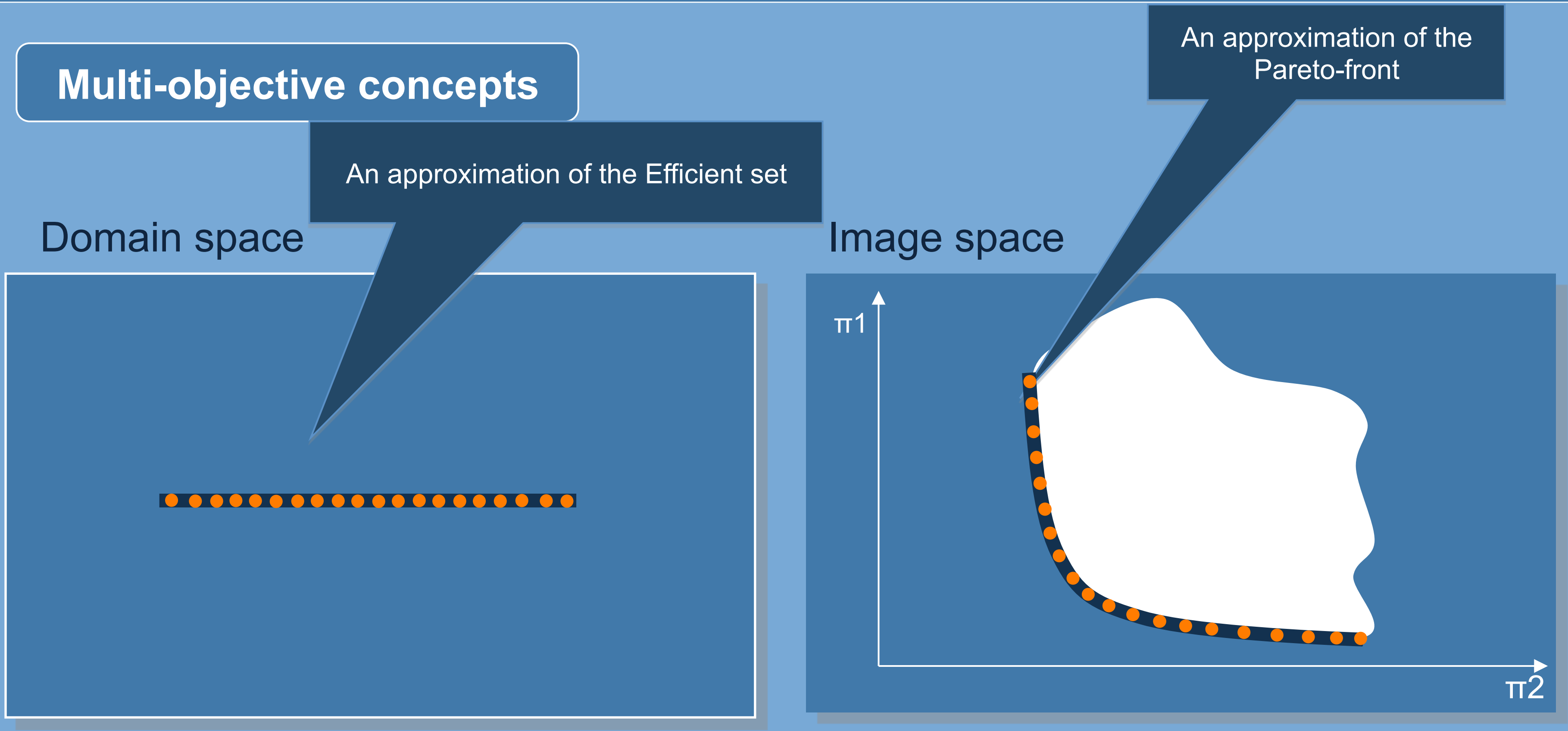
Multi-objective concepts



- **If nf and nf' are two feasible points and**
- **$f_l(nf) \leq f_l(nf')$ with one of the inequalities being strict** ➡ **nf dominates nf'**

2. Algorithm FEMOEA

Multi-objective concepts



- **If nf and nf' are two feasible points and**
- **$f_l(nf) \leq f_l(nf')$ with one of the inequalities being strict** → **nf dominates nf'**

L_{max} The number of solutions which must compose the final approximation of the Pareto-front.

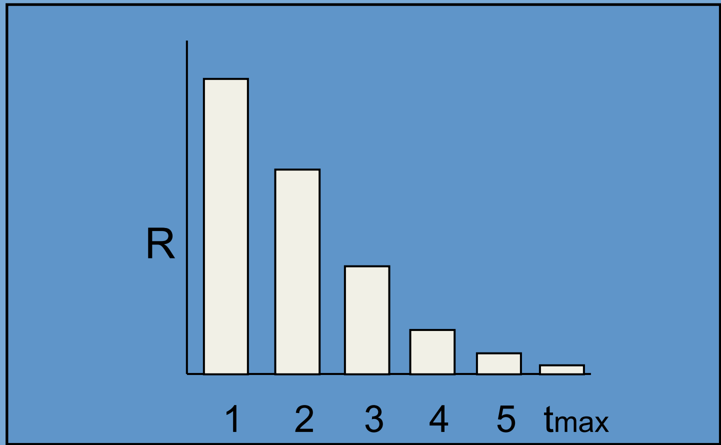
2. Algorithm FEMOEA

Basic concepts

Species concept

population_list

	nf	R	d_{rank}	c_{dist}
Lmax	nf_1	R_1	0	3
	nf_2	R_2	0	2
	nf_3	R_3	0	1
	nf_4	R_4	1	1



Domain space

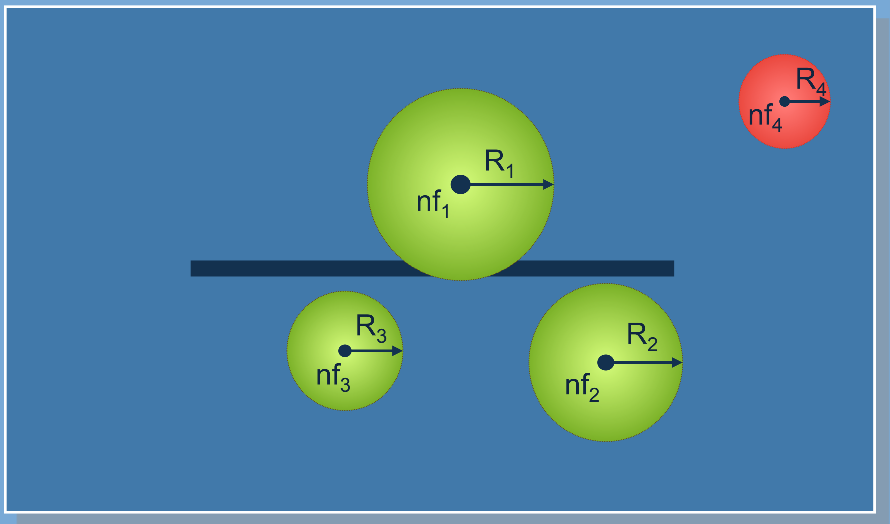
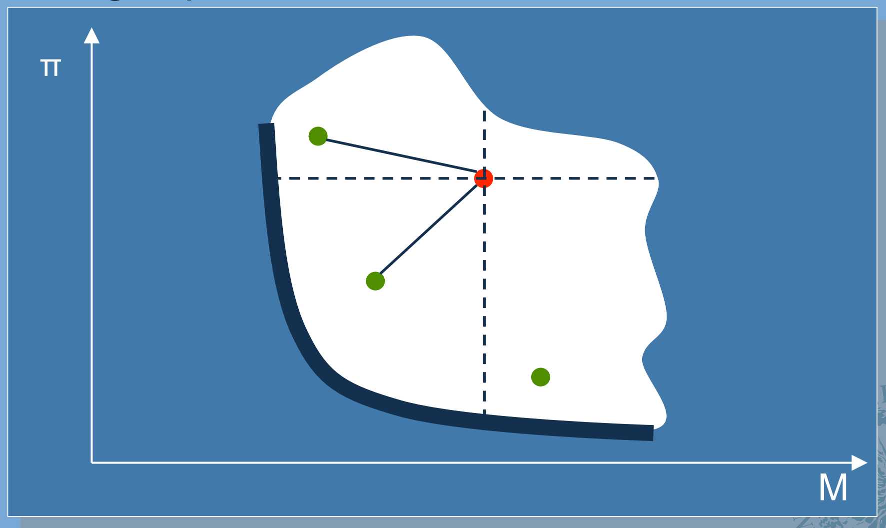


Image space



2. Algorithm FEMOEA

FEMOEA: Crowded comparison operator

A solution sp_i is preferable to another $sp_{i'}$ if

- $d^{i}_{rank} < d^{i'}_{rank}$ or
- $d^{i}_{rank} = d^{i'}_{rank}$ and $c^{i}_{dist} > c^{i'}_{dist}$

population_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	3
nf_2	R_2	0	2
nf_3	R_3	1	1
nf_4	R_4	2	2

external_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	5
nf_2	R_2	0	4
nf_3	R_3	0	3
nf_4	R_4	0	2

FEMOEA: user given parameters

L_{max}	The number of solutions which must compose the final approximation of the Pareto-front.
t_{max}	The maximum number of levels (or iterations).
R_1, $R_{t_{max}}$	The radius associated with the minimum and maximum level, respectively.
tol	The tolerance associated with the termination criteria.

2. Algorithm FEMOEA

Algorithm: FEMOEA (Fast and Efficient Multi-Objective Evolutionary Algorithm)

```
Init _species_lists
WHILE termination criteria are not satisfied DO
        Create_new_species (evals)
        IF ( length (population_list) > L_max ) THEN
                Select_species (population_list)
        Improve_species (population_list)
        Update_external_list
        IF ( length (external_list) > L_max ) THEN
                Select_species (external_list)
        Improve_species (external_list)
IF (length (external_list) < L_max) THEN
        Compose_pareto
```

population_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	3
nf_2	R_1	0	2
nf_3	R_1	1	1
nf_4	R_1	2	1

$L_{max} = 4$

external_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	3
nf_2	R_1	0	2

2. Algorithm FEMOEA

Algorithm: FEMOEA (Fast and Efficient Multi-Objective Evolutionary Algorithm)

Init _species_lists
WHILE termination criteria are not satisfied DO
    Create_new_species (evals)
    IF (length (population_list) > L_{max}) THEN
        Select_species (population_list)
    Improve_species (population_list)
    Update_external_list
    IF (length (external_list) > L_{max}) THEN
        Select_species (external_list)
    Improve_species (external_list)
IF (length (external_list) < L_{max}) THEN
    Compose_pareto

2. Algorithm FEMOEA

Creation new species

population_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	4
nf_2	R_2	0	2
nf_3	R_3	1	1
nf_4	R_4	2	1
nf_{15}	R_t	3	2
nf_{16}	R_t	4	1

nf_{11} nf_{12} nf_{13} → nf_{14} nf_{15} nf_{16}

external_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	4
nf_2	R_2	0	3
nf_3	R_3	0	2

Domain space

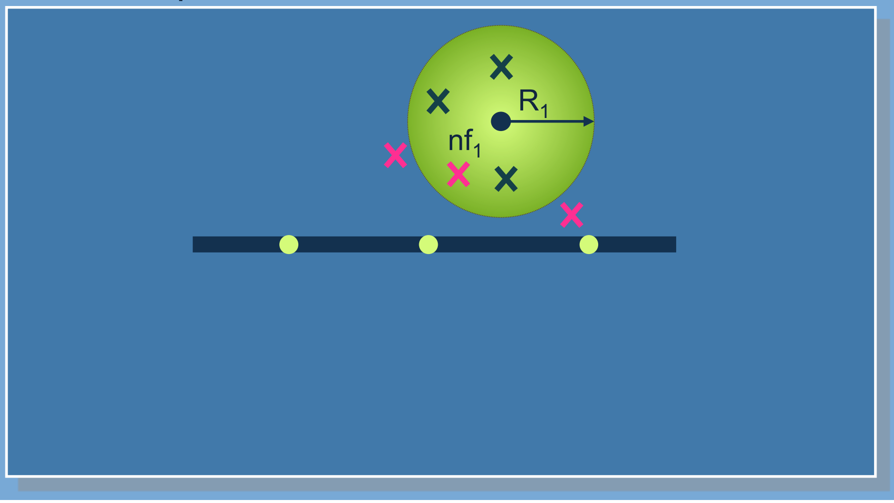
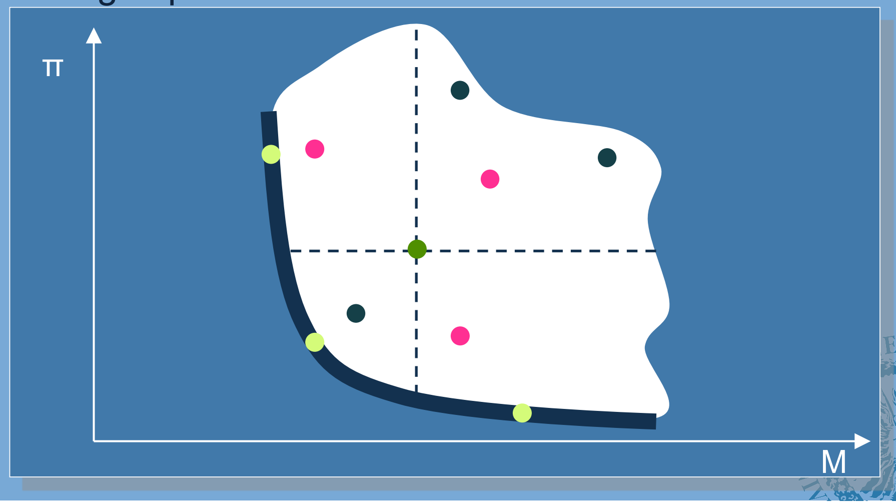


Image space



2. Algorithm FEMOEA

Algorithm: FEMOEA (Fast and Efficient Multi-Objective Evolutionary Algorithm)

```
Init _species_lists
WHILE termination criteria are not satisfied DO
        Create_new_species (evals)
        IF ( length (population_list) > L_max ) THEN
                Select_species (population_list)
        Improve_species (population_list)
        Update_external_list
        IF ( length (external_list) > L_max ) THEN
                Select_species (external_list)
        Improve_species (external_list)
IF (length (external_list) < L_max) THEN
        Compose_pareto
```

population_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	7
nf_{32}	R_t	0	5
nf_2	R_2	0	2
nf_{14}	R_t	1	4
nf_3	R_3	1	3
…			
nf_{13}	R_t	3	[illegible]
nf_4	R_4	4	1

$=L_{max}$

$> L_{max}$

external_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	10
nf_2	R_2	0	9
nf_3	R_3	0	8
nf_4	R_4	0	7
nf_5	R_5	0	6
…			
[illegible]	R_8	0	2
nf_9	R_9	0	1

$=L_{max}$

$> L_{max}$

Crowded comparison operator

2. Algorithm FEMOEA

Algorithm: FEMOEA (Fast and Efficient Multi-Objective Evolutionary Algorithm)

```
Init _species_lists
WHILE termination criteria are not satisfied DO
    Create_new_species (evals)
    IF ( length (population_list) > L_max ) THEN
        Select_species (population_list)
    **Improve_species (population_list)**
    Update_external_list
    IF ( length (external_list) > L_max ) THEN
        Select_species (external_list)
    **Improve_species (external_list)**
IF (length (external_list) < L_max) THEN
    Compose_pareto
```

FEMOEA can be adapted to another problem by adapting the improving method

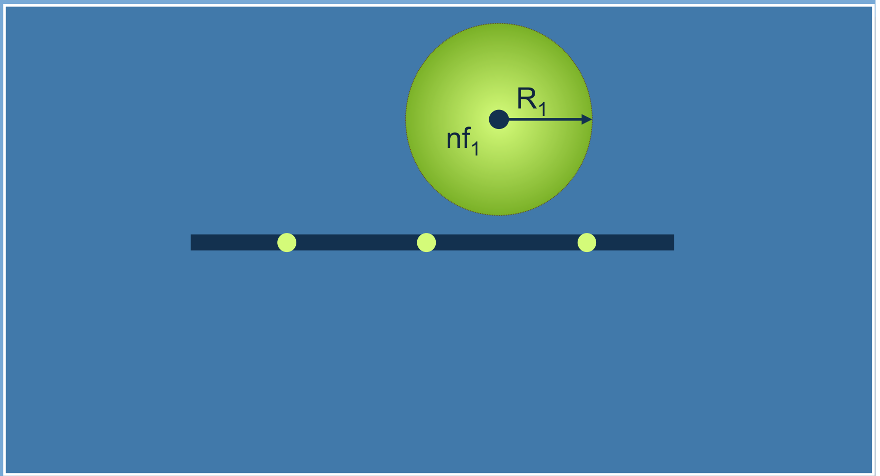
2. Algorithm FEMOEA

Improve_species (list)

list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	3
nf_2	R_2	0	2
nf_3	R_3	1	1
nf_4	R_4	2	3

Domain space



Algorithm 3 Algorithm MO_SASS(y, σ_{ub}, bel)

Set $ic = 1$, $y^{(ic)} = y$, $b^{(ic)} = 0$, $scnt = 0$, $fcnt = 0$, $\sigma^{(0)} = \sigma_{ub}$, $\sigma_{lb} = \max\{\sigma_{ub}/1000, 10^{-5}\}$
Fix ex, ct, $Scnt$, $Fcnt$, $Maxfcnt$, ic_{max}
while $ic < ic_{max}$ and $fcnt < Maxfcnt$
  $\sigma^{(ic)} = \sigma^{(ic-1)}$
  if $scnt > Scnt$
    $\sigma^{(ic)} = ex \cdot \sigma^{(ic-1)}$
  if $fcnt > Fcnt$
    $\sigma^{(ic)} = ct \cdot \sigma^{(ic-1)}$
  if $\sigma^{(ic)} < \sigma_{lb}$
    $\sigma^{(ic)} = \sigma_{ub}$ and $b^{(ic)} = 0$
  if $\sigma^{(ic)} > \sigma_{ub}$
    $\sigma^{(ic)} = \sigma_{ub}$
  Generate a multivariate Gaussian random vector $\xi_{aux}^{(ic)} = N(b^{(ic)}, \sigma^{(ic)} I)$
  if $y^{(ic)} + \xi^{(ic)}$ dominates $y^{(ic)}$
    $y^{(ic+1)} = y^{(ic)} + \xi^{(ic)}$; $scnt = scnt + 1, fcnt = 0$
  else
    if $bel = 0$ and $y^{(ic)} + \xi^{(ic)}$ is not dominated by any point on the *external_list*
      Include $y^{(ic)} + \xi^{(ic)}$ in *external_list*; $scnt = 0, fcnt = fcnt + 1$; $b^{(ic+1)} = 0.4\xi_{aux}^{(ic)} + 0.2b^{(ic)}$
    else
      if $y^{(ic)} - \xi^{(ic)}$ dominates $y^{(ic)}$
        $y^{(ic+1)} = y^{(ic)} - \xi^{(ic)}$; $scnt = scnt + 1, fcnt = 0$
      else
        if $bel = 0$ and $y^{(ic)} - \xi^{(ic)}$ is not dominated by any point on the *external_list*
          Include $y^{(ic)} - \xi^{(ic)}$ in *external_list*; $scnt = 0, fcnt = fcnt + 1$; $b^{(ic+1)} = b^{(ic)} - 0.4\xi_{aux}^{(ic)}$
        else
          $b^{(ic+1)} = 0.5b^{(ic)}$, $fcnt = fcnt + 1$, $scnt = 0$
  $ic = ic + 1$
Return $y^{(ic)}$

2. Algorithm FEMOEA

Algorithm: FEMOEA (Fast and Efficient Multi-Objective Evolutionary Algorithm)

```
Init _species_lists
WHILE termination criteria are not satisfied DO
        Create_new_species (evals)
        IF ( length (population_list) > L_max ) THEN
                Select_species (population_list)
        Improve_species (population_list)
        Update_external_list
        IF ( length (external_list) > L_max ) THEN
                Select_species (external_list)
        Improve_species (external_list)
IF (length (external_list) < L_max) THEN
        Compose_pareto
```

population_list

nf	R	d_{rank}	c_{dist}
nf_3	R_3	0	4
nf_2	R_2	0	2
nf_4	R_4	0	1
nf_1	R_1	1	3

external_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	10
nf_2	R_2	0	9
nf_3	R_3	0	8
nf_4	R_4	0	7
nf_5	R_1	0	6
nf_6	R_1	0	5
nf_7	R_2	0	4

2. Algorithm FEMOEA

Algorithm: FEMOEA (Fast and Efficient Multi-Objective Evolutionary Algorithm)

```
Init _species_lists
WHILE termination criteria are not satisfied DO
        Create_new_species (evals)
        IF ( length (population_list) > L_max ) THEN
                Select_species (population_list)
        Improve_species (population_list)
        Update_external_list
        IF ( length (external_list) > L_max ) THEN
                Select_species (external_list)
        Improve_species (external_list)
IF (length (external_list) < L_max) THEN
        Compose_pareto
```

1. Modified Hausdorff distance: how far two sets are from each other.

$$hd(PF_1, PF_2) = \frac{\dfrac{\sum_{a \in PF_1} \min\{\, d(a,b) : b \in PF_2 \}}{\max\{ d(a,a') : a, a' \in PF_1 \}} + \dfrac{\sum_{b \in PF_2} \min\{\, d(a,b) : a \in PF_1 \}}{\max\{ d(b,b') : b, b' \in PF_2 \}}}{2}$$

2. t_{max}: maximum number of iterations

2. Algorithm FEMOEA

Algorithm: FEMOEA (Fast and Efficient Multi-Objective Evolutionary Algorithm)

```
Init _species_lists
WHILE termination criteria are not satisfied DO
        Create_new_species (evals)
        IF ( length (population_list) > L_max ) THEN
                Select_species (population_list)
        Improve_species (population_list)
        Update_external_list
        IF ( length (external_list) > L_max ) THEN
                Select_species(external_list)
        Improve_species (external_list)
IF (length (external_list) < L_max) THEN
        Compose_pareto
```

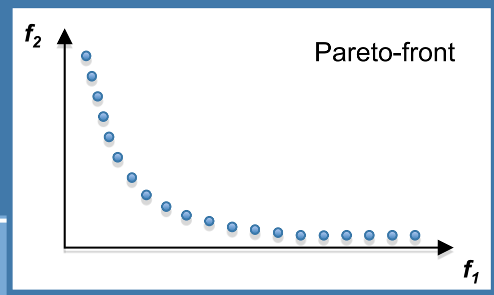
population_list

nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	2
nf_2	R_2	1	4
nf_3	R_3	2	2
nf_4	R_4	3	1

external_list

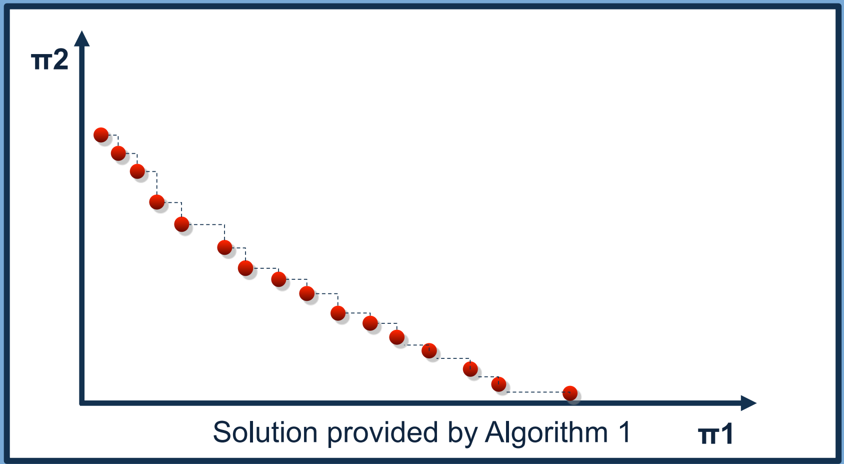
nf	R	d_{rank}	c_{dist}
nf_1	R_1	0	5
nf_2	R_2	0	4
nf_3	R_3	0	1

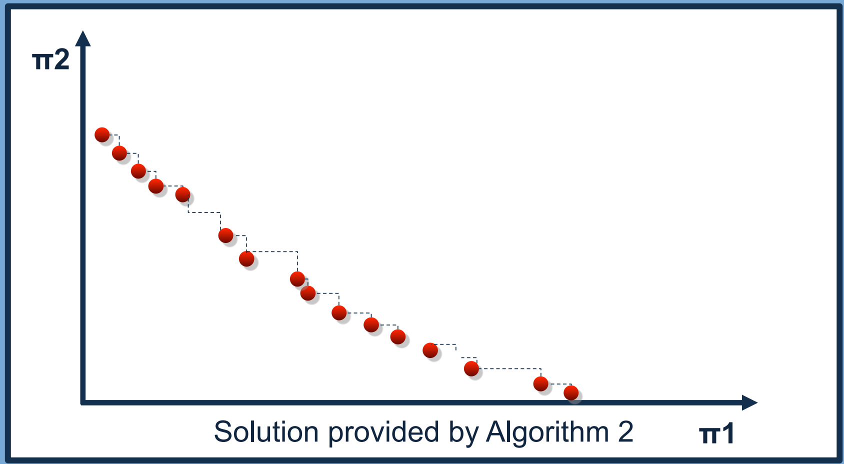
$L_{max} = 4$



2. Algorithm FEMOEA

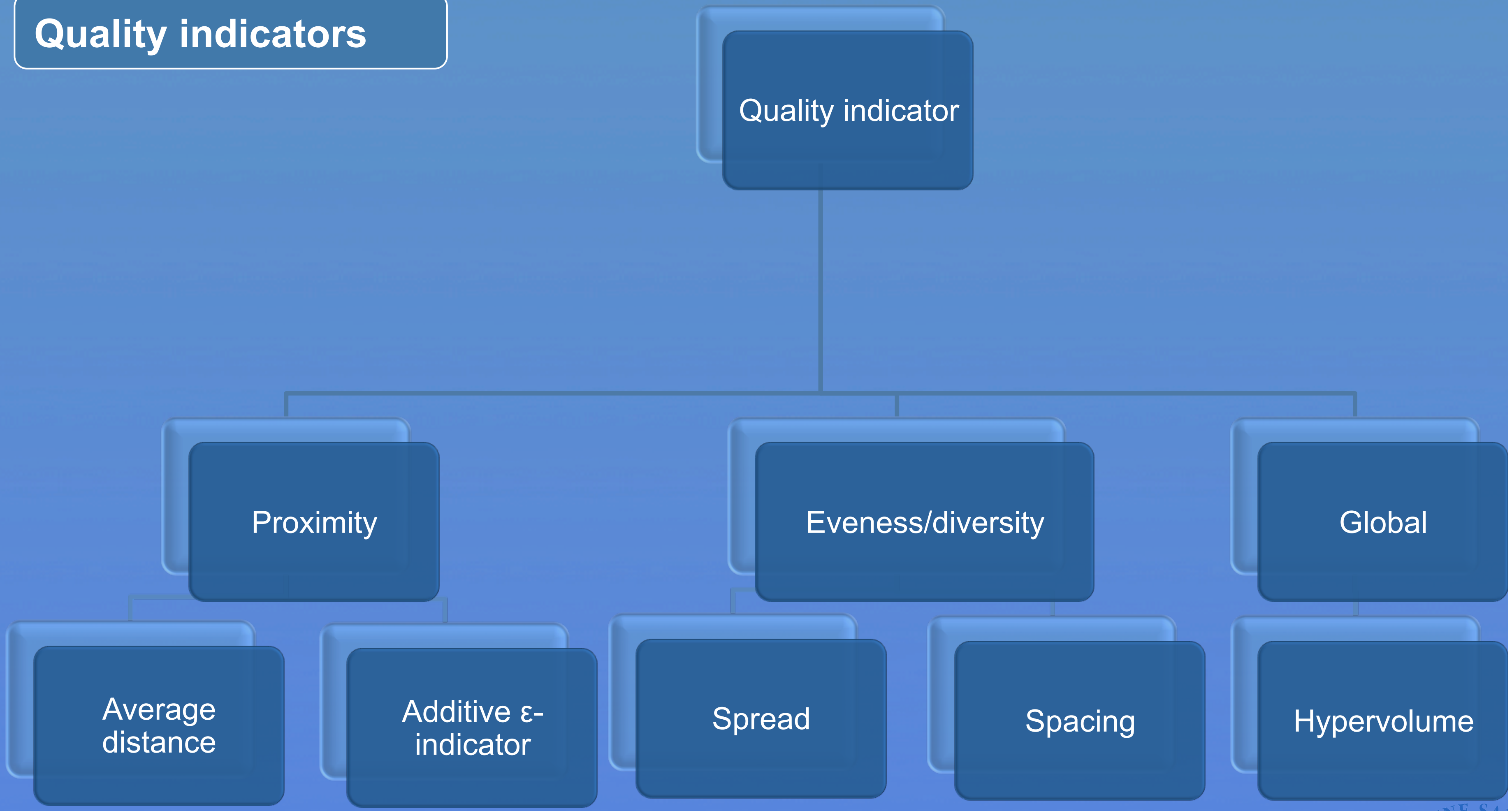
Which is the best Pareto front?



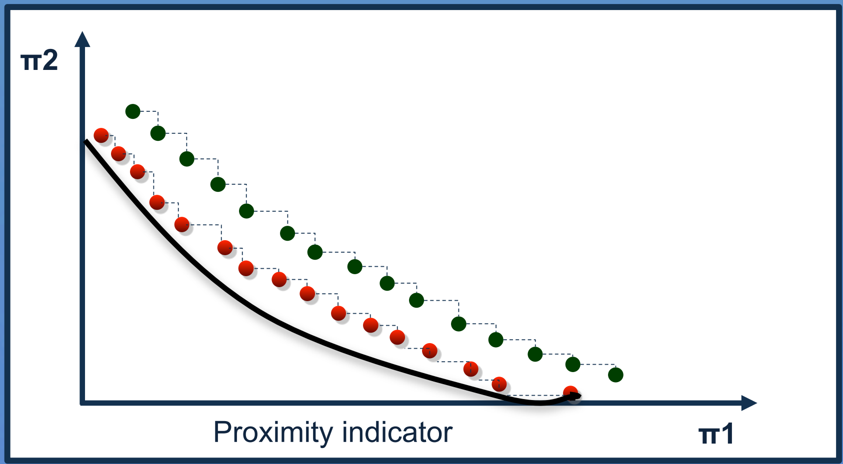


2. Algorithm FEMOEA

Quality indicators



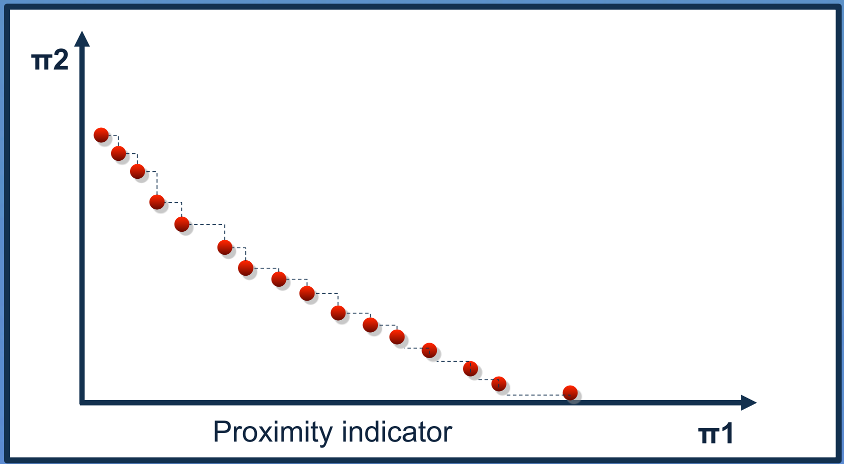
2. Algorithm FEMOEA

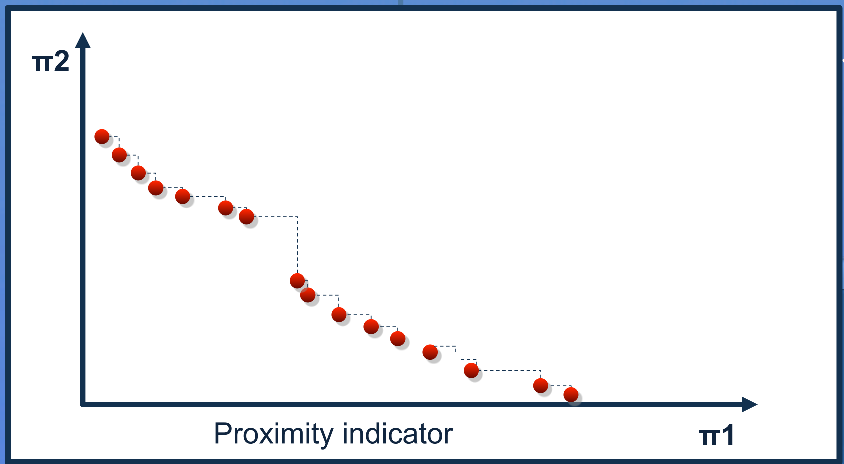


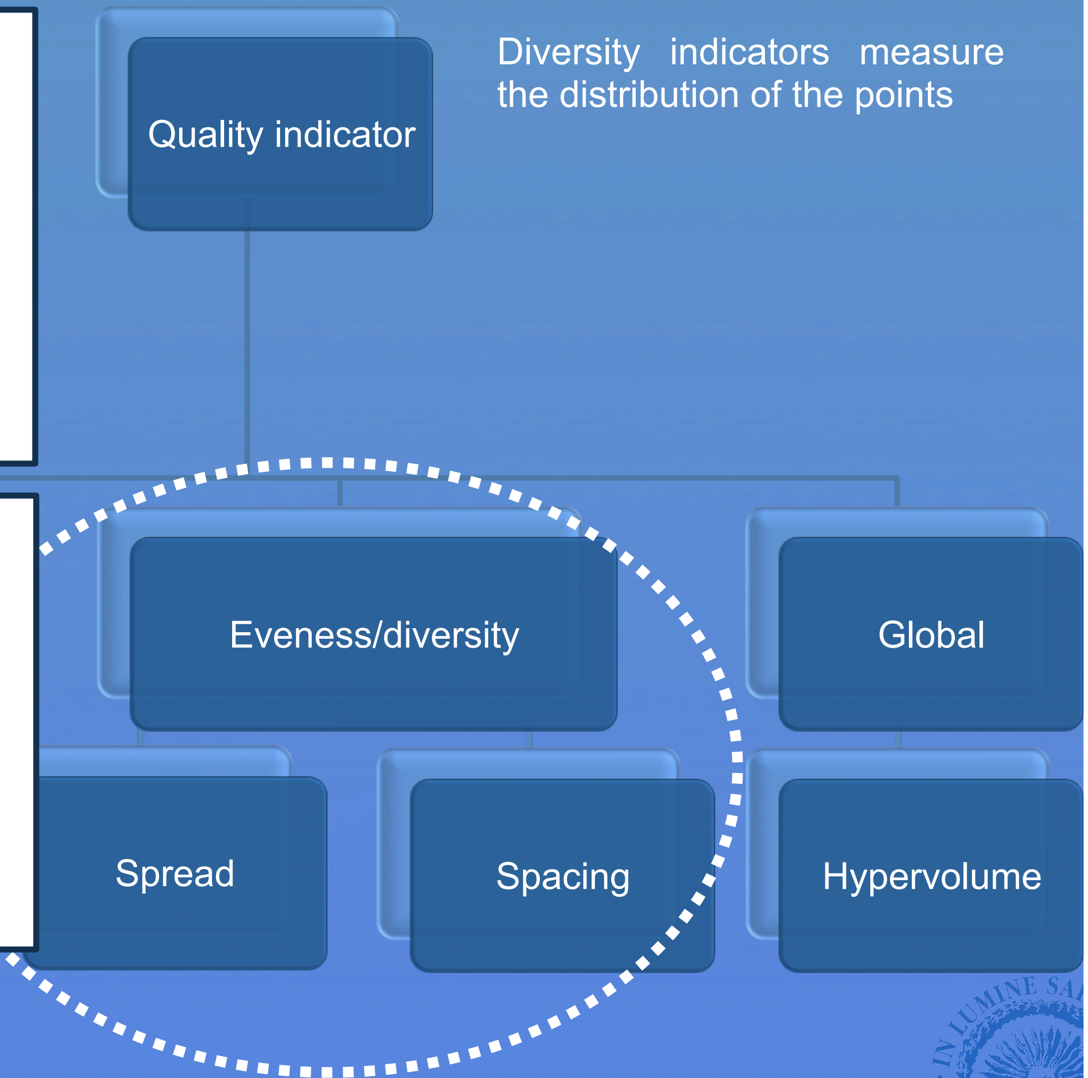
Proximity indicators measure the distance between the appoximation set and the reference set.

- Quality indicator
  - Proximity
    - Average distance
    - Additive ε-indicator
  - Eveness/diversity
    - Spread
    - Spacing
  - Global
    - Hypervolume

2. Algorithm FEMOEA

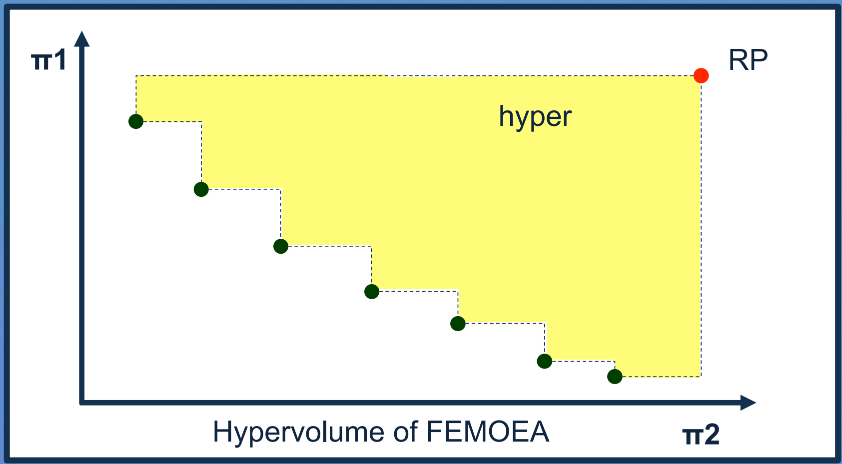


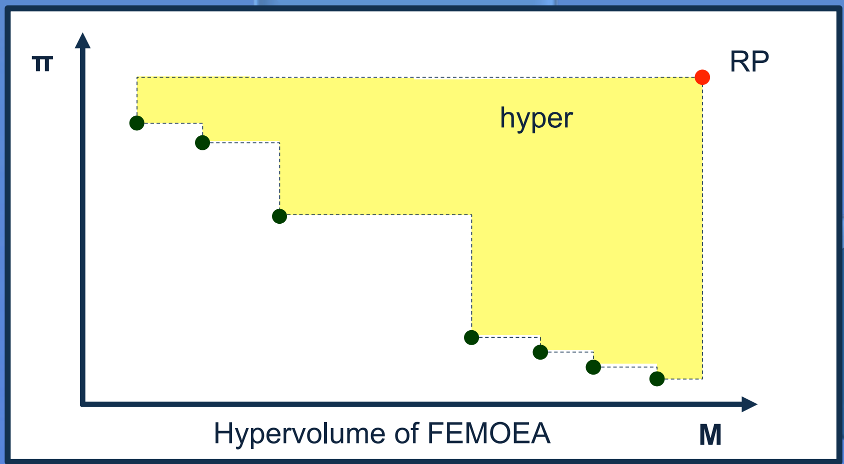




Diversity indicators measure the distribution of the points

2. Algorithm FEMOEA





Quality indicator

Hypervolume measures the hypervolume of the portion of the criterion space that is weakly dominated by the approximation set.

Eveness/ diversity

Spread

Spacing

Global

Hypervolume

2. Algorithm FEMOEA

COMPUTATIONAL EXPERIMENTS

- All the computational results have been carried out on a 4-core processor HP PorLiant ML330 G6 to 2.00GHz abd 7.8GB memory (using one of its cores).
- FEMOEA has been compararerd to MOEAD, NSGA-II and SPEA2. The implementations provided by the frameworks jMetal have been used http://jmetal.sourceforge.net/
- The algoritms have been tested on a set of 20 standardard benchmark problems.
- Each particular instance has been executed 100 times and average values have been computed.
- MOEAD, NSGA-II and SPEA2 were run with the same number of functions evaluations as FEMOEA.
- The number of points in the PF were set to 100 for 2-objectives problems and 300 for 3-objetives problems.

2. Algorithm FEMOEA

COMPUTATIONAL EXPERIMENTS: Proximity indicator → Additive

Cuadro 1: Average $I^1_{\epsilon+}$ values. MOEAD, NSGA-II and SPEA2 were run with the same number of functions evaluations as FEMOEA.

	FEMOEA	MOEAD	NSGA-II	SPEA2
zdt1	0.000000000e+00	1.504400000e-07	0.000000000e+00	0.000000000e+00
zdt2	0.000000000e+00	1.265890000e-06	0.000000000e+00	0.000000000e+00
zdt3	2.778970000e-06	1.000000083e-11	0.000000000e+00	0.000000000e+00
zdt4	2.294000001e-08	2.652280000e-06	1.043170000e-04	2.246170000e-05
zdt6	0.000000000e+00	4.999999526e-11	8.898320000e-05	1.614573000e-04
dtlz1	8.728999998e-08	1.430000001e-09	6.135110000e-05	3.039370000e-05
dtlz2	0.000000000e+00	3.036830000e-06	2.871138000e-04	7.223479000e-04
dtlz3	2.700000024e-10	1.153800000e-07	3.797690000e-05	7.841600000e-06
dtlz5	0.000000000e+00	3.318920000e-06	4.466775000e-04	2.963108000e-04
dtlz6	0.000000000e+00	7.352990000e-06	2.574523000e-04	1.128326000e-04
dtlz7	6.765032000e-05	1.910000000e-09	0.000000000e+00	0.000000000e+00
viennet	2.413874000e-05	4.199999992e-10	1.086685000e-03	2.982108300e-03
viennet2	1.805013400e-03	8.469999591e-10	1.724744300e-03	1.731114400e-03
deb	1.280946000e-05	1.962423143e-01	4.185533450e-02	6.005316230e-02
deb1	2.037800000e-06	3.668000001e-08	3.424000000e-07	0.000000000e+00
fonseca	5.718940000e-06	2.653630000e-06	1.071085700e-03	7.526463000e-04
kursawe	4.645415000e-05	1.011272800e-04	9.036141000e-04	1.494671200e-03
poloni	0.000000000e+00	1.805540000e-06	2.769843800e-03	4.223474000e-04
qv	1.460881100e-04	1.489579600e-04	5.000320000e-04	1.140905500e-03
schaffer	1.443578000e-05	1.190647211e-01	0.000000000e+00	0.000000000e+00
Average	1.063618085e-04	1.577897569e-02	2.559777680e-03	3.496530050e-03

2. Algorithm FEMOEA

COMPUTATIONAL EXPERIMENTS: Proximity indicator → Average distance

Cuadro 2: Mean average distance values. MOEAD, NSGA-II and SPEA2 were run with the same number of functions evaluations as FEMOEA.

	FEMOEA	MOEAD	NSGA-II	SPEA2
zdt1	6.193869500e-05	1.894623000e-07	0.000000000e+00	0.000000000e+00
zdt2	5.115316650e-05	1.514529000e-07	6.755000000e-09	0.000000000e+00
zdt3	8.785943606e-05	9.590841807e-07	1.724172663e-05	1.427912710e-06
zdt4	3.209638265e-04	2.153703244e-06	2.310333302e-04	4.196763812e-03
zdt6	1.702045631e-03	4.794867752e-12	7.544440597e-05	1.601046870e-04
dtlz1	2.498083038e-04	6.111900315e-10	2.396015843e-04	3.030887391e-02
dtlz2	1.065620683e-04	4.862261461e-07	5.146346890e-04	4.052772878e-04
dtlz3	2.374174112e-04	1.026246133e-06	5.008061456e-02	5.900285185e-03
dtlz5	1.029929831e-04	5.028926485e-07	5.748154348e-04	3.624701780e-04
dtlz6	2.455000798e-12	1.183288988e-07	6.459219943e-04	2.076751624e-04
dtlz7	6.354147956e-06	1.985848545e-05	4.774262083e-06	2.058007311e-07
viennet	4.013300170e-04	2.996916361e-10	1.038610545e-03	8.576754682e-04
viennet2	2.648152640e-04	6.054923177e-11	2.353006952e-04	2.925364324e-04
deb	3.589488940e-04	4.402295399e-02	6.955192176e-02	9.391611809e-02
deb1	1.155426594e-05	3.319413481e-06	2.284629421e-07	3.235173922e-07
fonseca	1.406290637e-04	1.610561572e-06	1.103567489e-03	5.539194626e-04
kursawe	4.342922322e-04	2.509442179e-05	1.133945116e-03	6.882308807e-04
poloni	4.525207437e-05	3.305108729e-04	1.506540157e-03	1.390697082e-03
qv	3.193440191e-04	1.756089793e-04	8.254123915e-04	5.670035004e-04
schaffer	1.015810048e-06	2.965291295e-01	1.036481620e-06	1.017440742e-06
Average	2.452138656e-04	1.705568373e-02	6.389032592e-03	6.990530291e-03

2. Algorithm FEMOEA

COMPUTATIONAL EXPERIMENTS: Dispersion indicator →Spread

Cuadro 4: Average spread values. MOEAD, NSGA-II and SPEA2 were run with the same number of functions evaluations as FEMOEA.

	FEMOEA	MOEAD	NSGA-II	SPEA2
zdt1	1.863358045e-01	2.773606325e-01	4.203510381e-01	1.750379504e-01
zdt2	1.848858105e-01	1.354201840e-01	4.257036897e-01	1.715564605e-01
zdt3	7.009471091e-01	1.003305610e+00	7.547560977e-01	7.049793045e-01
zdt4	2.373313066e-01	2.586506705e-01	4.462643706e-01	3.462081974e-01
zdt6	1.820476634e-01	1.582906417e-01	4.220327358e-01	1.696521706e-01
dtlz1	1.729801815e-01	4.576504307e-04	4.599159444e-01	2.909007981e-01
dtlz2	1.919599743e-01	1.811535604e-01	3.862598848e-01	1.678993061e-01
dtlz3	4.289690327e-01	4.584144647e-01	6.520295597e-01	5.006143448e-01
dtlz5	1.918062185e-01	1.811501426e-01	4.063028207e-01	1.756125910e-01
dtlz6	1.890016877e-01	1.812455201e-01	4.346597362e-01	1.730222905e-01
dtlz7	5.259582615e-01	7.489013801e-01	6.634659348e-01	5.470887946e-01
viennet	3.539878062e-01	6.888246406e-02	5.935894445e-01	5.970729797e-01
viennet2	2.236858397e-01	8.801705351e-02	4.612284117e-01	2.377602266e-01
deb	3.824614952e-01	1.356314289e+00	7.327294309e-01	6.748643620e-01
deb1	7.044826982e-01	1.204757294e+00	7.687836871e-01	7.113754920e-01
deb	2.743351600e-01	6.850804584e-01	6.920269922e-01	6.272201585e-01
deb1	7.044826982e-01	9.187328664e-01	7.687836871e-01	7.113754920e-01
fonseca	1.916339970e-01	1.461751863e-01	4.134363233e-01	1.550532590e-01
kursawe	4.343619014e-01	7.282128594e-01	5.637210441e-01	4.359278434e-01
poloni	4.692007784e-01	6.513852055e-01	6.106987140e-01	5.018228024e-01
qv	2.446799170e-01	2.421809147e-01	4.090534156e-01	1.540008818e-01
schaffer	1.938720635e-01	1.432323781e+00	3.957628205e-01	1.718692439e-01
Average	3.141231606e-01	4.272670623e-01	5.190021333e-01	3.507337548e-01

2. Algorithm FEMOEA

COMPUTATIONAL EXPERIMENTS: Dispersion indicator → Spacing

Cuadro 5: Average TS values. MOEAD, NSGA-II and SPEA2 were run with the same number of functions evaluations as FEMOEA.

	FEMOEA	MOEAD	NSGA-II	SPEA2
zdt1	4.293577810e-01	5.239096233e-01	6.446590462e-01	4.161074748e-01
zdt2	4.277010093e-01	3.660649180e-01	6.485802919e-01	4.118915490e-01
zdt3	8.330283284e-01	9.966311025e-01	8.643619074e-01	8.354202669e-01
zdt4	4.845007193e-01	5.059201950e-01	6.643141099e-01	5.534493023e-01
zdt6	4.242312825e-01	3.957964025e-01	6.461376012e-01	4.097354119e-01
dtlz1	4.135643715e-01	1.897207650e-02	6.740091929e-01	4.825630219e-01
dtlz2	4.358082041e-01	4.234569164e-01	6.180685141e-01	4.073456704e-01
dtlz3	6.514876924e-01	6.736504298e-01	8.014532958e-01	6.949132228e-01
dtlz5	4.356497225e-01	4.234544193e-01	6.336399793e-01	4.167170799e-01
dtlz6	4.324271754e-01	4.235360396e-01	6.524529616e-01	4.091886358e-01
dtlz7	7.215774827e-01	8.609082003e-01	8.103901969e-01	7.359059067e-01
viennet	5.282494183e-01	8.769558581e-02	7.862348077e-01	6.744842099e-01
viennet2	2.384006105e-01	1.636370189e-01	6.046423197e-01	2.577703990e-01
deb	5.208153965e-01	8.172511244e-01	8.248765272e-01	7.831030624e-01
deb1	8.351272511e-01	9.536843831e-01	8.723320860e-01	8.391939899e-01
fonseca	4.354239955e-01	3.798699388e-01	6.394091901e-01	3.911051360e-01
kursawe	6.556991985e-01	8.489489737e-01	7.469083358e-01	6.568988006e-01
poloni	6.815402597e-01	8.025293210e-01	7.773802098e-01	7.046781177e-01
qv	4.559689873e-01	4.867193985e-01	6.360993054e-01	3.899776849e-01
schaffer	4.378812318e-01	1.204830348e+00	6.252710804e-01	4.123071308e-01
Average	5.239220059e-01	5.678733208e-01	7.085610480e-01	5.441378037e-01

2. Algorithm FEMOEA

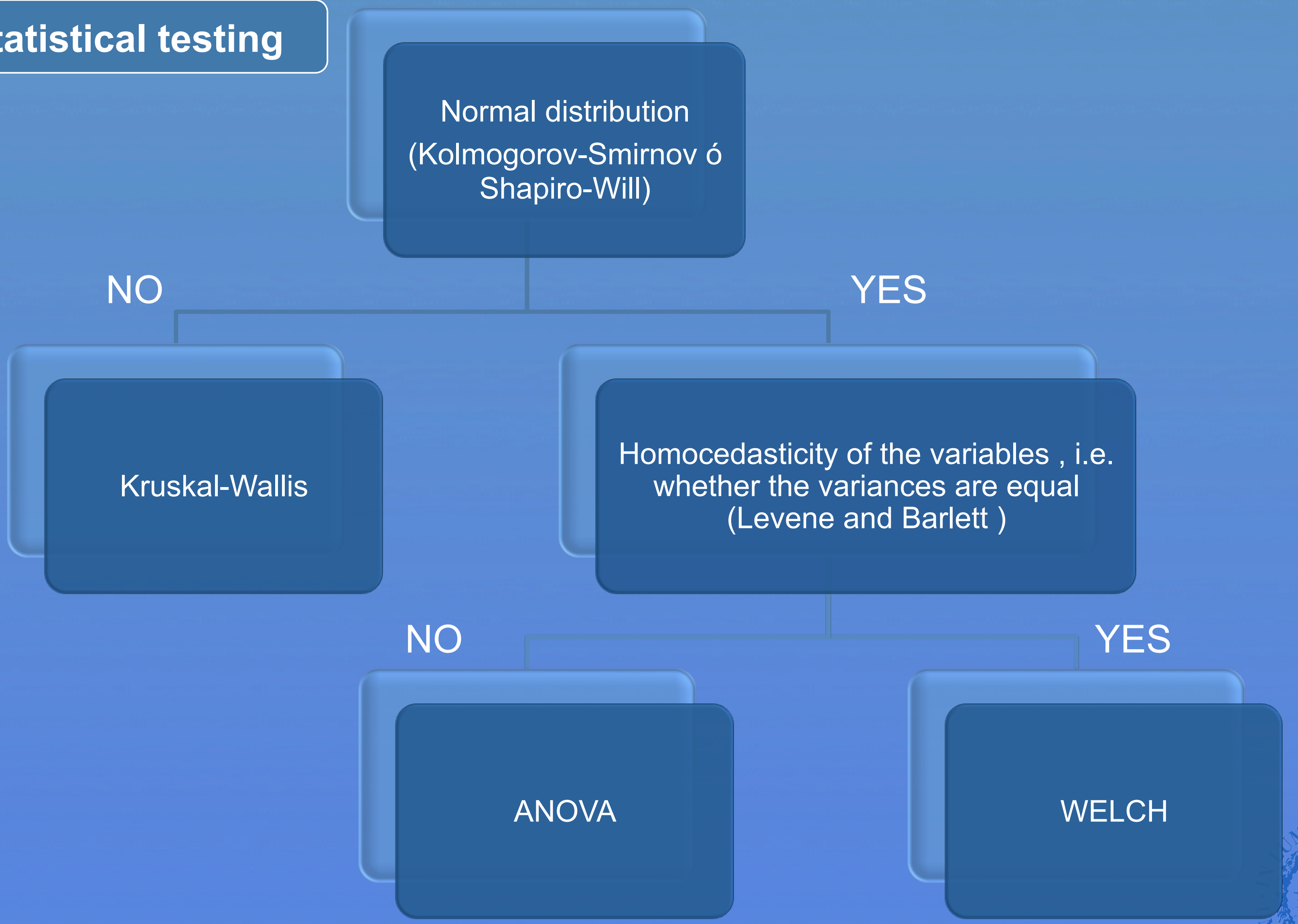
COMPUTATIONAL EXPERIMENTS: Global indicator → Hypervolume

Cuadro 3: Average hypervolume values. MOEAD, NSGA-II and SPEA2 were run with the same number of functions evaluations as FEMOEA.

	FEMOEA	MOEAD	NSGA-II	SPEA2
zdt1	6.616784320e-01	6.616611183e-01	6.604983740e-01	6.618078082e-01
zdt2	3.284214082e-01	3.284910601e-01	3.273222292e-01	3.285067891e-01
zdt3	5.156609080e-01	5.137373399e-01	5.153904940e-01	5.156947541e-01
zdt4	8.429148722e-01	8.430757657e-01	8.424701384e-01	8.431150869e-01
zdt6	4.480084410e-01	4.480934033e-01	4.465707427e-01	4.478009352e-01
dtlz1	9.812583499e-01	9.811982379e-01	9.811976195e-01	9.812667224e-01
dtlz2	2.102431197e-01	2.101300868e-01	2.087321064e-01	2.097226164e-01
dtlz3	9.759973416e-01	9.759623936e-01	9.759816324e-01	9.760049404e-01
dtlz5	2.102520503e-01	2.101300921e-01	2.086405810e-01	2.098039892e-01
dtlz6	2.459323291e-01	2.456073643e-01	2.445669507e-01	2.457704939e-01
dtlz7	3.346326466e-01	3.334696637e-01	3.338607474e-01	3.345401267e-01
viennet	8.399626496e-01	5.730952639e-01	8.324896596e-01	8.337762196e-01
viennet2	9.270504370e-01	7.276933653e-01	9.215935133e-01	9.267807404e-01
deb	9.001508486e-01	8.361252811e-01	8.645700589e-01	8.539537038e-01
deb1	3.265927309e-01	3.255647507e-01	3.262271324e-01	3.265111745e-01
fonseca	3.122747599e-01	3.126387250e-01	3.092504830e-01	3.115540668e-01
kursawe	4.025923336e-01	4.020596850e-01	4.006455951e-01	4.020836164e-01
poloni	2.191284181e-01	2.159935669e-01	2.156383613e-01	2.158969186e-01
qv	8.304565872e-02	8.259066237e-02	8.166120710e-02	8.258191371e-02
schaffer	9.955388209e-01	9.950186523e-01	9.955251296e-01	9.955399250e-01
Average	5.380668278e-01	5.111168239e-01	5.346416378e-01	5.351356271e-01

2. Algorithm FEMOEA

Statistical testing



Heuristic algorithms

J.L. Redondo