

Multi-Objective Evolutionary Algorithms

Jiří Kubařík

Czech Institute of Informatics, Robotics, and Cybernetics
Czech Technical University in Prague



<http://cw.felk.cvut.cz/doku.php/courses/a0m33eoa/start>

Multi-Objective Evolutionary Algorithms

- **Pareto Archived Evolution Strategy (PAES)**

Knowles, J.D., Corne, D.W. (2000) Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2), pp. 149-172

- **Multiple Objective Genetic Algorithm (MOGA)**

Carlos M. Fonseca, Peter J. Fleming: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, In *Genetic Algorithms: Proceedings of the Fifth International Conference*, 1993

- **Niched-Pareto Genetic Algorithm (NPGA)**

Jeffrey Horn, Nicholas Nafpliotis, David E. Goldberg: A Niched Pareto Genetic Algorithm for Multiobjective Optimization, *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE World Congress on Computational Intelligence, 1994

- **SPEA2**

Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm For Multiobjective Optimization, In: *Evolutionary Methods for Design, Optimisation, and Control*, Barcelona, Spain, pp. 19-26, 2002

Fitness Sharing

Diversity preservation method originally proposed for solving multi-modal optimization problems so that GA is able to sample each optimum with the same number of solutions.

Idea – diversity in the population is preserved by degrading the fitness of similar solutions

Algorithm for calculating the shared fitness value of i -th individual in population of size N

1. calculate *sharing function* value with all solutions in the population according to

$$Sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^\alpha, & \text{if } d_{ij} \leq \sigma_{share} \\ 0, & \text{otherwise.} \end{cases}$$

2. calculate niche count nc_i as follows

$$nc_i = \sum_{j=1}^N Sh(d_{ij})$$

3. calculate *shared fitness* as

$$f'_i = f_i / nc_i$$

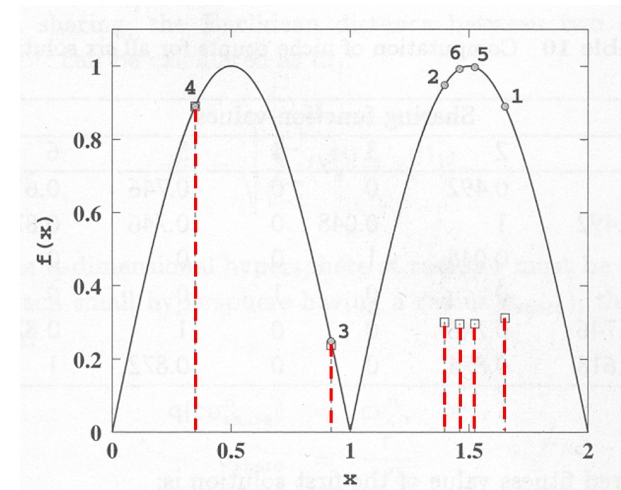
Remark: If $d = 0$ then $Sh(d) = 1$ meaning that two solutions are identical. If $d \geq \sigma_{share}$ then $Sh(d) = 0$ meaning that two solutions do not have any sharing effect on each other.

Fitness Sharing: Example

Bimodal function - six solutions and corresponding shared fitness functions

- $\sigma_{share} = 0.5, \alpha = 1.$

Sol. i	String	Decoded value	$x^{(i)}$	f_i	nc_i	f'_i
1	110100	52	1.651	0.890	2.856	0.312
2	101100	44	1.397	0.948	3.160	0.300
3	011101	29	0.921	0.246	1.048	0.235
4	001011	11	0.349	0.890	1.000	0.890
5	110000	48	1.524	0.997	3.364	0.296
6	101110	46	1.460	0.992	3.364	0.295



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

Let's take the first solution

- $d_{11} = 0.0, d_{12} = 0.254, d_{13} = 0.731, d_{14} = 1.302, d_{15} = 0.127, d_{16} = 0.191$
- $Sh(d_{11}) = 1, Sh(d_{12}) = 0.492, Sh(d_{13}) = 0, Sh(d_{14}) = 0, Sh(d_{15}) = 0.746, Sh(d_{16}) = 0.618.$
- $nc_1 = 1 + 0.492 + 0 + 0 + 0.746 + 0.618 = 2.856$
- $f'(1) = f(1)/nc_1 = 0.890/2.856 = 0.312$

NSGA: Fitness Assignment

Input: Set P of solutions with assigned objective values.

Output: Set of solutions with assigned fitness values (the bigger the better).

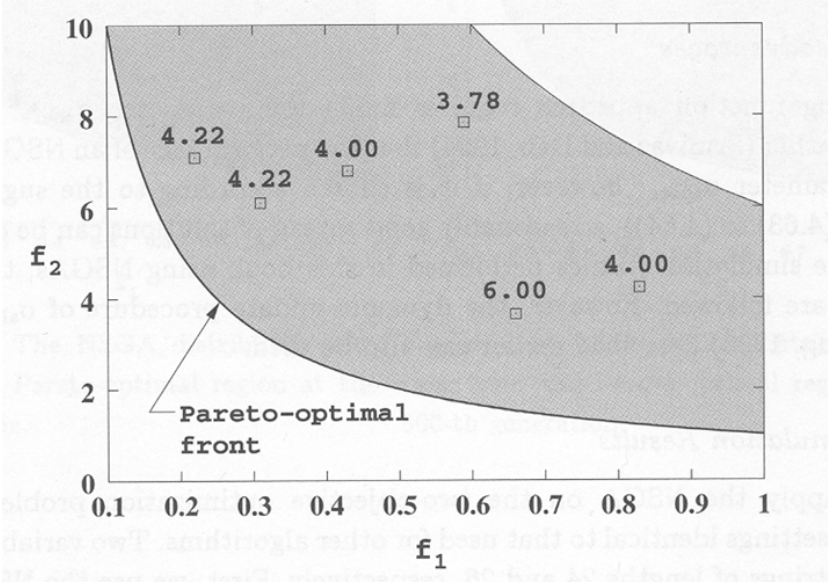
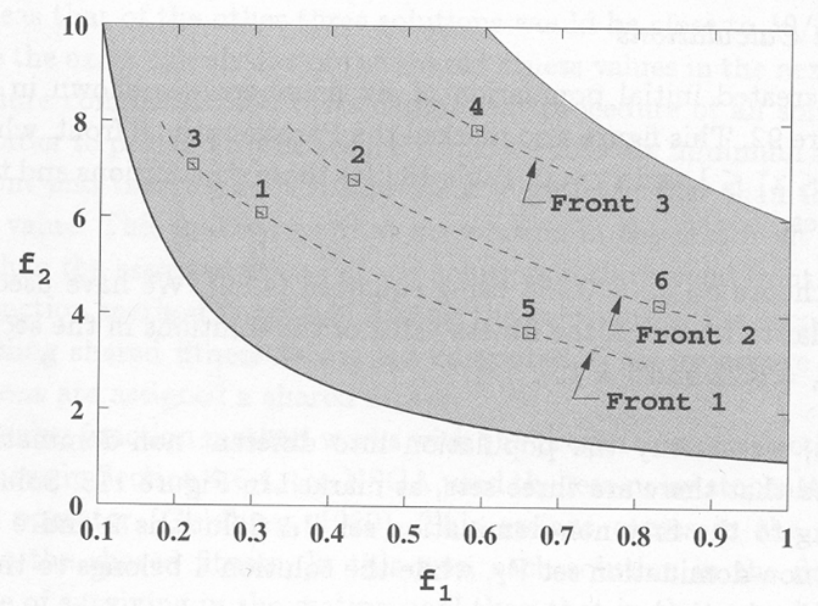
1. Choose sharing parameter σ_{share} , small positive number ϵ , initialize $F_{max} = PopSize$ and front counter $front = 1$
2. Find set $P' \subset P$ of non-dominated solutions
3. For each $q \in P'$
 - assign fitness $f(q) = f_{max}$,
 - calculate sharing function with all solutions in P' niche count nc_q among solutions of P' only, the normalized Euclidean distance d_{ij} is calculated
 - calculate shared fitness $f'(q) = f(q)/nc_q$.
4. $f_{max} = \min(f'(q) : q \in P') - \epsilon$
 $P = P \setminus P'$
 $front = front + 1$
5. If not all solutions are assessed go to step 2, otherwise stop.

$$d_{ij} = \sqrt{\sum_{k=1}^M \left(\frac{f_k^{(i)} - f_k^{(j)}}{f_k^{max} - f_k^{min}} \right)^2}$$

NSGA: Fitness Assignment cont.

Example:

- First, 10 solutions are classified into different non-dominated fronts.
- Then, the fitness values are calculated according to the fitness sharing method.
 - The sharing function method is used front-wise.
 - Within a front, less dense solutions have better fitness values.



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

NSGA: Conclusions

Computational complexity

- Governed by the non-dominated sorting procedure and the sharing function implementation.
 - **non-dominated sorting** – complexity of $O(MN^3)$.
 - **sharing function** – requires every solution in a front to be compared with every other solution in the same front, total of $\sum_{j=1}^{\rho} |P_j|^2$, where ρ is a number of fronts.
Each distance computation requires evaluation of n differences between parameter values.
In the worst case, when $\rho = 1$, the overall complexity is of $O(nN^2)$.

Advantages

- Assignment of fitness according to non-dominated sets – makes the algorithm converge towards the Pareto-optimal region.
- Sharing allows phenotypically diverse solutions to emerge.

Disdvantages

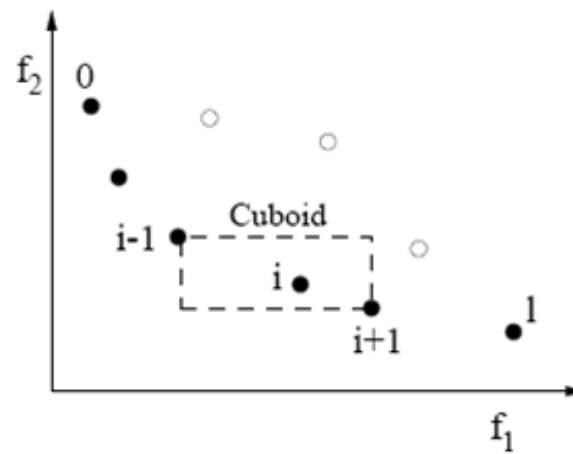
- non-elitist
- sensitive to the sharing method parameter σ_{share} .
requires some guidelines for setting the σ

$\sigma_{share} = \frac{0.5}{\sqrt[q]{q}}$, for example based on the expected number of optima q



NSGA-II: Diversity preservation

Density estimation – crowding distance estimates how much unique the solution is.



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

Crowded comparison operator

Every solution in the population has two attributes

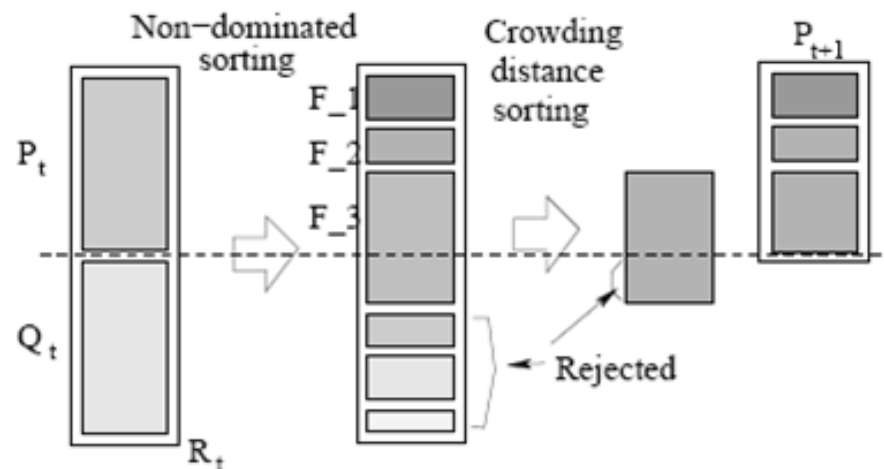
1. non-domination rank (i^{rank}), and
2. crowding distance ($i^{distance}$).

A partial order \prec_n is defined as:

$$i \prec_n j \text{ if } (i^{rank} < j^{rank}) \text{ or } ((i^{rank} = j^{rank}) \text{ and } (i^{distance} > j^{distance}))$$

NSGA-II: Evolutionary Model

1. Current population P_t is sorted based on the non-domination
Each solution is assigned a fitness equal to its non-domination level (1 is the best).
2. The usual binary tournament selection, recombination, and mutation are used to create a child population Q_t of size N.
3. Combined population $R_t = P_t \cup Q_t$ is formed.
Elitism is ensured.
4. Population P_{t+1} is formed according to the following schema



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

NSGA-II: Constraint Handling Approach

Binary tournament selection with modified domination concept is used to choose the better solution out of the two solutions i and j , randomly picked up from the population.

In the presence of constraints each solution in the population can be either **feasible** or **infeasible**, so that there are the following three possible situations:

1. both solutions are feasible,
2. one is feasible and other is not,
3. both are infeasible.

NSGA-II: Constraint Handling Approach

Binary tournament selection with modified domination concept is used to choose the better solution out of the two solutions i and j , randomly picked up from the population.

In the presence of constraints each solution in the population can be either **feasible** or **infeasible**, so that there are the following three possible situations:

1. both solutions are feasible,
2. one is feasible and other is not,
3. both are infeasible.

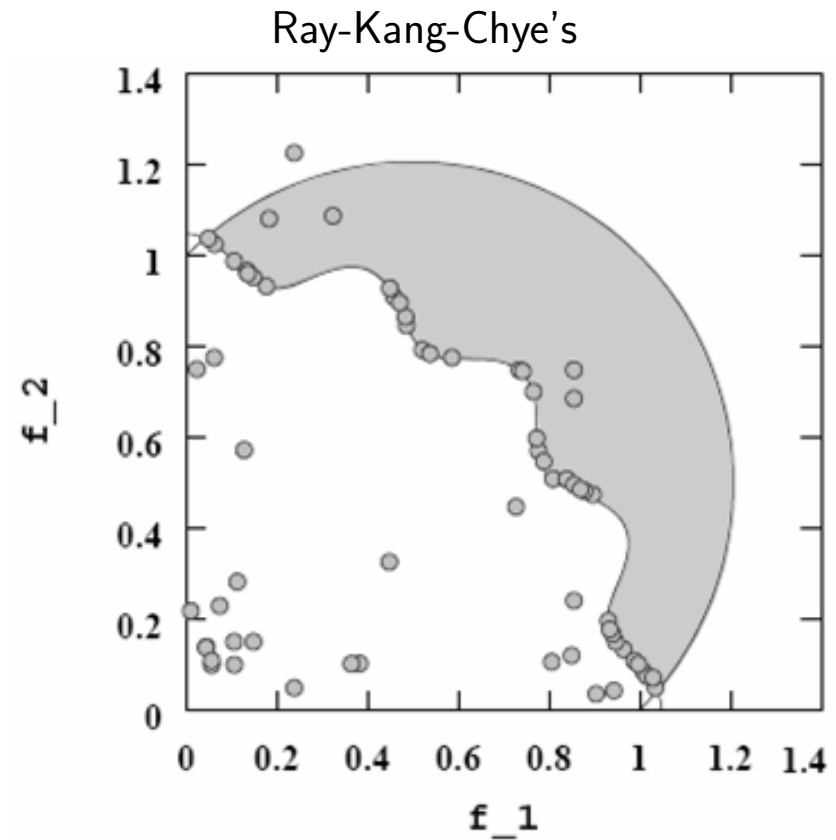
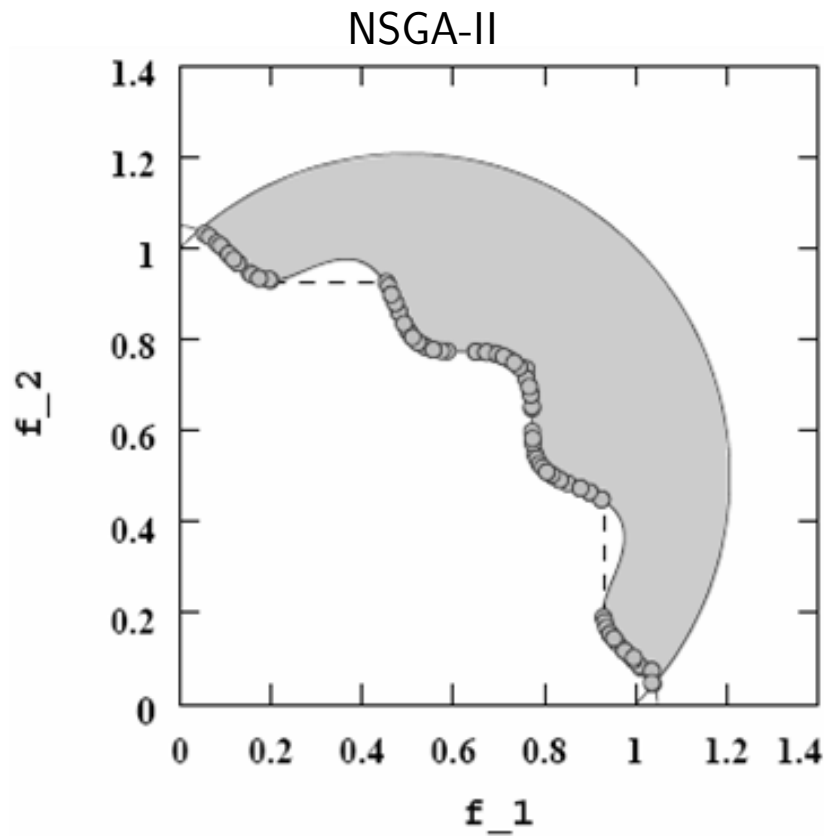
Constrained-domination: A solution i is said to constrained-dominate a solution j , if any of the following conditions is true

1. Solution i is feasible and solution j is not.
2. Solutions i and j are both infeasible, but solution i has a smaller overall constraint violation.
3. Solutions i and j are feasible, and solution i dominates solution j .

NSGA-II: Simulation Results cont.

Comparison of NSGA-II and Ray-Kang-Chye's Constraint handling approach

■ Ray, T., Tai, K. and Seow, K.C. [2001] "Multiobjective Design Optimization by an Evolutionary Algorithm", Engineering Optimization, Vol.33, No.4, pp.399-424



©Kalyanmoy Deb et al.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II.

Strength Pareto Evolutionary Algorithm 2 (SPEA2)

SPEA2 maintains two sets of solutions

- **regular population** of newly generated solutions, and
- **archive**, which contains a representation of the nondominated front among all solutions considered so far.

The archive size is fixed, i.e., whenever the number of nondominated individuals is less than the predefined archive size, the archive is filled up by *good* dominated individuals.

A **truncation method** is invoked when the nondominated front exceeds the archive limit.

A member of the archive is only removed if

1. a solution has been found that dominates it or
2. the maximum archive size is exceeded and the portion of the front where the archive member is located is overcrowded.

Using the archive makes it possible not to lose certain portions of the current nondominated front due to random effects.

All individuals in the archive participate in selection.

SPEA2: Algorithm

Input: N is the population size, \bar{N} is the archive size.

1. **Initialization:** Generate an initial population P_0 and create the empty archive $\bar{P}_0 = \emptyset$. Set $t = 0$.
2. **Fitness assignment:** Calculate fitness of individuals in P_t and \bar{P}_t .
3. **Environmental selection:** Copy all nondominated individuals in P_t and \bar{P}_t to \bar{P}_{t+1} .
If size of \bar{P}_{t+1} exceeds \bar{N} then reduce \bar{P}_{t+1} using the truncation operator.
If size of \bar{P}_{t+1} is less than \bar{N} then fill \bar{P}_{t+1} with dominated solutions in P_t and \bar{P}_t .
4. **Termination:** If $t \geq T$ then return nondominated solutions in \bar{P}_{t+1} . Stop.
5. **Mating selection:** Perform binary tournament selection with replacement on \bar{P}_{t+1} in order to fill the mating pool.
6. **Variation:** Apply recombination and mutation operators to the mating pool and fill P_{t+1} with the generated solutions,
increment generation counter $t = t + 1$,
go to Step 2.



SPEA2: Fitness Assignment

Fitness assignment (fitness is to minimized) – for each individual both dominating and dominated solutions are taken into account.

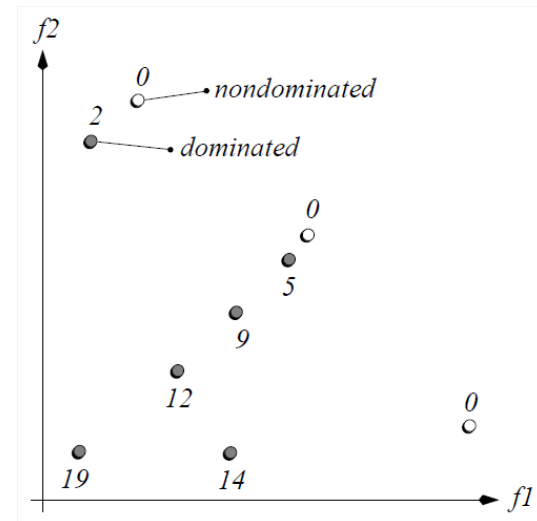
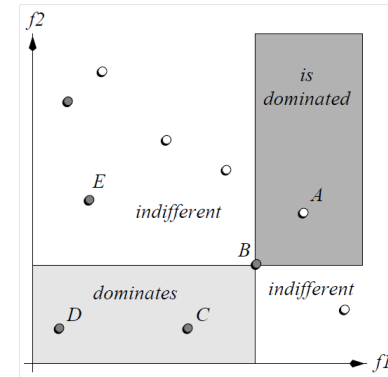
- Each individual i in the archive \bar{P}_t and the population P_t is assigned a **strength value** $S(i)$, representing the number of solutions it dominates.
- The raw fitness $R(i)$ of an individual i is calculated as

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

that is $R(i)$ is determined by the strengths of its dominators in both archive and population.

$R(i) = 0$ corresponds to a nondominated solution.

Since the **raw fitness assignment** is based on the concept of Pareto dominance, it **may fail when most individuals do not dominate each other**.



MOEA Performance Measures

The result of a MOEA run is not a single scalar value, but a collection of vectors forming a non-dominated set.

- Comparing two MOEA algorithms requires comparing the non-dominated sets they produce. However, there is no straightforward way to compare different non-dominated sets.

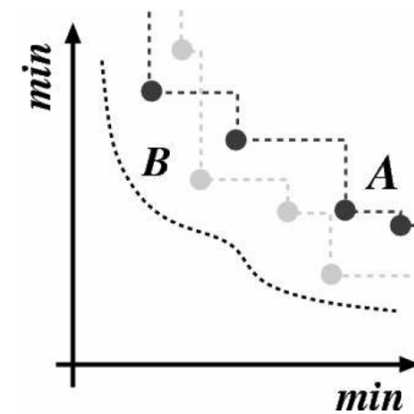
Three goals that can be identified and measured:

1. The distance of the resulting non dominated set to the Pareto-optimal front should be minimized.
2. A good (in most cases uniform) distribution of the solutions found is desirable.
3. The extent of the obtained non dominated front should be maximized, i.e., for each objective, a wide range of values should be present.

C Metric

Coverage of two sets $C(X, Y)$ – given two sets of non-dominated solutions X and Y found by the compared algorithms, the measure $C(X, Y)$ returns a ratio of a number of solutions of Y that are dominated by or equal to any solution of X to the whole set Y .

- It returns values from the interval $[0, 1]$.
- The value $C(X, Y) = 1$ means that all solutions in Y are covered by solutions of the set X . And vice versa, the value $C(X, Y) = 0$ means that none of the solutions in Y are covered by the set X .
- Always both orderings have to be considered, since $C(X, Y)$ is not necessarily equal to $1 - C(Y, X)$.



$$C(A, B) = 0.25, C(B, A) = 0.75$$

Properties:

- It has low computational overhead.
- If two sets are of different cardinality and/or the distributions of the sets are non-uniform, then it gives unreliable results.

Reading

- Kalyanmoy Deb: Multi-objective optimization using evolutionary algorithms
<http://books.google.com/books?id=OSTn4GSy2uQC&printsec=frontcover&dq=deb&hl=cs&cd=1>
- Kalyanmoy Deb et al.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, vol. 6, pp. 182–197, 2000.
<http://sci2s.ugr.es/docencia/doctobio/2002-6-2-DEB-NSGA-II.pdf>
- Eckart Zitzler et al.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm, 2001.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.5073&rep=rep1&type=pdf>
- Eckart Zitzler: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, 1999.
<ftp://ftp.tik.ee.ethz.ch/pub/people/zitzler/Zitz1999.ps.gz>
- Joshua Knowles and David Corne: On Metrics for Comparing Non-Dominated Sets, 2001.
<http://www.lania.mx/~ccoello/knowles02a.ps.gz>

