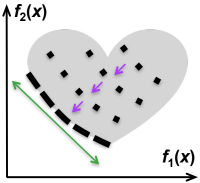
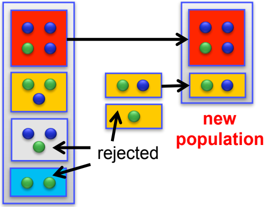




Introduction to Evolutionary Multiple Objective Optimization

Miłosz Kadziński

Institute of Computing Science
Poznan University of Technology, Poland

Multiple Objective Optimization

A general formulation of multiple objective optimization problems:

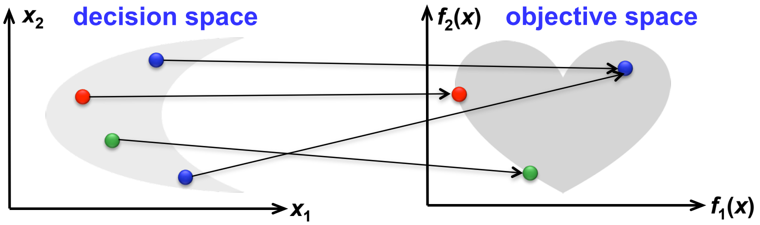
Minimize $f(x) = (f_1(x), f_2(x), \ldots, f_M(x))$ ⇒ *multiple (M) objectives*

subject to $x \in S$, ⇒ *constraints*

$g_j(x) \geq 0, \quad j = 1, \ldots, J$
$h_k(x) = 0 \quad k = 1, \ldots, K$
$x_i^L \leq x_i \leq x_i^U, \quad i = 1, \ldots, N$

where $x=(x_1, x_2, \ldots, x_N)$ is a solution (represented by decision variables) and $S \subset R^N$ is called the feasible set

- Accounting for **several conflicting objectives simultaneously**
- Need for considering the trade-offs between objectives

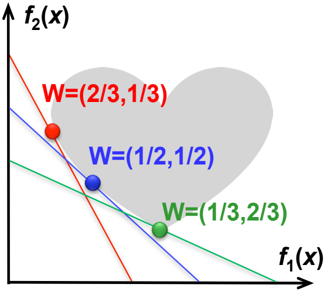


A Posteriori MOO Methods

Most a posteriori methods fall into two classes:

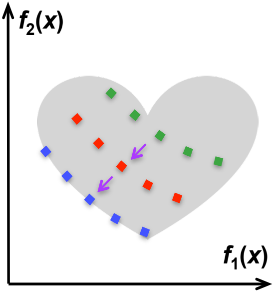
Classical methods solve multiple single-objective optimization problems that each produces one PO solution at a time

- Usually, multiple runs necessary to obtain a set of Pareto optimal solutions
- Usually, problem knowledge is necessary
- Last lecture: WSM, ECM, ASF

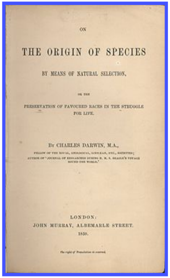


Evolutionary methods evolve a **population of solutions** simultaneously to obtain a representative set of diverse solutions in a **single run**

- **Mimic the process of natural evolution**
- Easy handling of problems with local Pareto fronts and discrete nature
- Pareto optimality of solutions cannot be guaranteed (lack of convergence proofs; fluctuations for the sake of diversity)

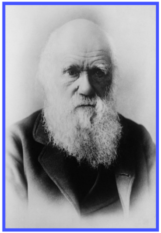


The Origins of Species – Evolutionary Perspective (1)



Charles Darwin
1859

- **Individuals** within species **are variable**
- Some of the variations are **passed on to offspring**
- In every generation, more offspring are produced than can survive
- The survival and reproduction of individuals are **not random**: the individuals who survive and go on to reproduce, or who reproduce the most, are those with the most favorable variations - **they are naturally selected**

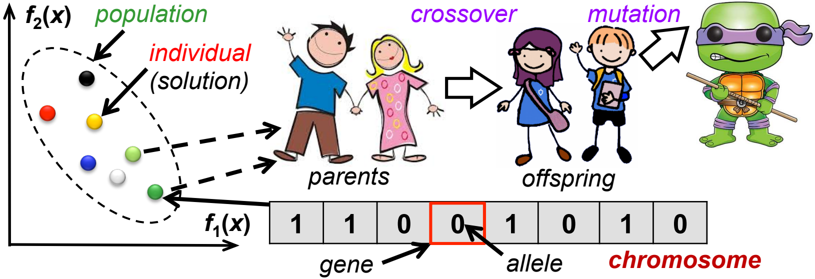


Charles Darwin

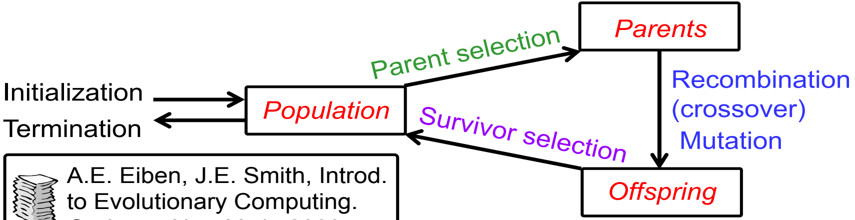
- Offspring created by reproduction, mutation, etc.
- Natural evolution **acts on the individuals**, but the consequences occur in the population
- Populations adapt to their environment
- Natural selection is a **guided search procedure** (**individuals suited to the environment survive, reproduce and pass their genetic traits to offspring**)
- Evolution acts on the existing traits, but can **produce new traits**
- Variations accumulate over time to generate new species
- Results of the evolution are ”creative”, ”surprising”, and ”highly adapted” to ”environmental niches”

Glossary

- **Individual** – carrier of the genetic information (**chromosome**); it is characterized by its state in the search space, its fitness; an instance of solution for a problem dealt with by the algorithm
- **Population** – pool of individuals which allows the application of genetic operators
- **Genetic operators** – source of variation (mutation, crossover)
- **Fitness function** – function giving the value of an individual
- **Survival of the fittest** – selection
- **Generation** – (natural) time unit (iteration) of the algorithm



General Scheme of Evolutionary Algorithm



A.E. Eiben, J.E. Smith, Introd. to Evolutionary Computing. *Springer*, New York, 2003

Initialize Population (e.g., with random candidate solutions or using a known heuristic for the problem)

Evaluate all individuals

Repeat Until (*termination condition is satisfied*)

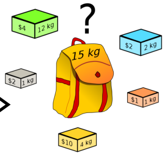
- **Select** parents
- **Recombine** (pairs of) parents (crossover)
- **Mutate** the resulting offspring
- **Evaluate** new candidates
- **Select** individuals for the next generation

Representation of Solutions (Individuals)

Binary representation

1	1	0	0	1	0	1	0

- Simplest and most common
- Natural for problems concerning Boolean decision variables
- Can be used to encode integer and real numbers



Integer representation

1	3	2	3	3	4	1	2

- For integer variables or discrete values (cardinal or ordinal)
- {train, truck Euro IV, truck Euro V, truck Euro VI} = {0, 1, 2, 3}
- {small, medium, large} = {0, 1, 2}

Decoding = transforming a solution from genotype to phenotype space

Real-valued represen.

0.5	0.2	0.6	0.8	0.7	0.3	0.9	0.4

- Continuous rather than discrete variables

Permutation represen.

1	5	8	6	4	3	2	7

- Sequence of events (**order** – job-shop scheduling problem, **adjacencies** – traveling salesman problem)

A	B	C	D
3	1	2	4

B → C → A → D
or
C → A → B → D

Mutation is a variation operator that alters one or more gene values in a chromosome from its initial state

- **Create one offspring from one parent**
- Mutation occurs during evolution according to a user-definable probability
- The probability should be set low (with too high value, the search turns into a primitive random search); typically, $1/pop_{size}$ or $1/chromosome_{length}$

For binary representation

	1	1	0	0	1	0	1	0
bitwise	1	1	0	1	1	0	1	0
flip bit	0	0	1	1	0	1	0	1

For integer representation

	1	2	3	4	5	6	7	8
random *resetting*	1	2	3	6	5	6	7	8
creep *mutation (add small value to gene value)*	1	2	3	2	5	6	7	8

4-2=2

Mutation – Variation Operator (2)

For floating representation

Gene values come from a continuous distribution:

- **uniform mutation** – similar to bit string or random resetting
- **non-uniform mutation** with a fixed distribution (e.g., Gaussian) – similar to creep mutation

For permutation representation

	1	2	3	4	5	6	7	8
swap *mutation*	1	5	3	4	2	6	7	8
insert *mutation*	1	2	5	3	4	6	7	8
scramble *mutation*	1	3	5	4	2	6	7	8
inversion *mutation*	1	5	4	3	2	6	7	8

Mutation is exploitative - it creates random small diversions, thereby staying near the area of the parent (only mutation can introduce new information)

Crossover – Variation Operator (1)

Recombination is a process for creating new individuals from two or more parents (**crossover** mostly refer to 2 parents)

- **Primary mechanism for creating diversity**
- Crossover rate p_c – typically in the range [0.5,1]; acts on a parent pair
- A random variable drawn from [0,1] - if value < p_c, creates offspring through recombination, else offspring created asexually (copy of parents)

For binary representation

0	0	0	0	1	0	0	0	0

1	1	0	1	0	0	0	0	1

one-point crossover (split parents at the crossover point and exchange tails)

0	0	0	0	0	0	0	0	1

1	1	0	1	1	0	0	0	0

N-point crossover (choose N random crossover point, split along these points)

0	0	0	0	0	0	0	0	0

1	1	0	1	1	0	0	0	1

Uniform crossover (treat each gene independently; random choice of parent)

0	1	0	0	0	0	0	0	1

1	0	0	1	1	0	0	0	0

Crossover – Variation Operator (2)

For floating-point representation

(integers below are used only to save space; all *k* and *α* are exemplary)

1	2	3	4	5	6	7	8	9	3	2	3	2	3	2	3	2	3

simple arithmetic recombination (e.g., after *k*=6 gene, use *α*=0.5 in the average)

1	2	3	4	5	6	5	5	6	3	2	3	2	3	2	5	5	6

single arithmetic recombination (e.g., for *k*=8 gene, use *α*=0.5 in the average)

1	2	3	4	5	6	7	5	9	3	2	3	2	3	2	3	5	3

whole arithmetic recombination (e.g., for all genes, use *α*=0.5 in the average)

2	2	3	3	4	4	5	5	6	2	2	3	3	4	4	5	5	6

For permuations representations: partially mapped crossover, edge crossover, order crossover, cycle crossover, …

Recombination is explorative – it makes a big jump in the area somewhere ”in-between” the (parent) areas – it combines information from the parents

- there exist multi-parent recombination operators

Selection

Selection can occur in two places:

- **Parent selection**: selection of individual(s) from the current population to take part in mating based on its/their fitness
- **Survivor selection**: replacement, selection from population (e.g., parents + offspring) to go into the next generation

the most distinctive aspect of EMOAs (will be discussed in the context of specific algorithms)

Parent selection

Selection pressure: *degree to which better individual are favored*

- if higher, better individual favored more
- if too high, premature convergence
- if too low, possible stagnation

Selection scheme: *process that selects an individual to go into the mating pool*

- **Fitness proportionate** (e.g., **roulette**): Absolute fitness F_i of individual compared to fitness of all individuals $\sum_{j=1,\ldots,N} F_j$
- **Ordinal based** (e.g., **tournament**) Ordering relation to rank any n individuals

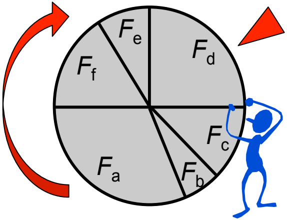
Parent Selection – Example Schemes

Aim: select n inidividuals to take part in the mating

sol	*a*	*b*	*c*	*d*	*e*	*f*
F_i	5	1	2	4	1.33	2.67

Roulette wheel selection

- Better individuals get a higher chance (chances are proportional to fitness)
- Assign to each individual a part of the roulette wheel, and spin the wheel n times to select n individuals



Tournament selection

Most widely used approach

- Tournament size k (low k, low selection pressure; higher k, increased pressure)
- Pick k individuals randomly and pick the best one by comparing their fitness
- Play n tournaments

$k = 3$
$n = 2$

1st *tournament*

a	5
d	4
e	1.33

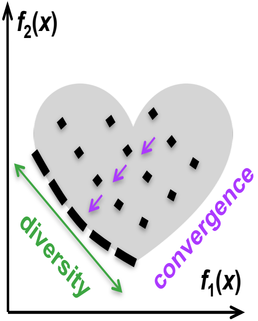
2nd *tournament*

f	2.67
c	2
e	1.33

Evolutionary Multiple Objective Optimization

Goals in Evolutionary Multiple Objective Optimization

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

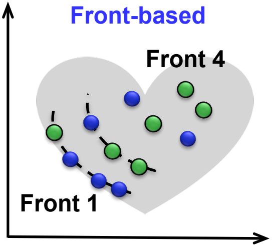


Main Differences with respect to Single Objective EAs

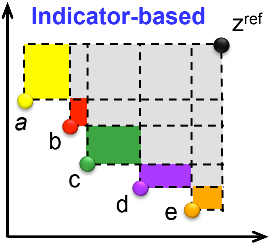
- Fitness computation
- Non-dominated solutions are preferred to attain convergence, i.e., maintain drive towards the Pareto front
- Emphasis given to less crowded or isolated solution to maintain diversity in the population

- How to maintain diversity and obtain a diverse set of Pareto optimal solutions
- How to maintain non-dominated solutions and the push towards the Pareto front?

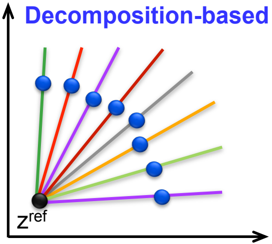
Agenda – EMO Algorithms (EMOAs)



NSGA-II, SPEA2



SMS EMOA



MOEA/D

Population model: steady state (SMS EMOA) or generational (NSGA-II, SPEA2)

Fitness assignment:
- objective-based (VEGA)
- aggregation-based (MOEA/D)
- front-based (NSGA-II, SPEA2)
- indicator-based (SMS EMOA)

EMOAs

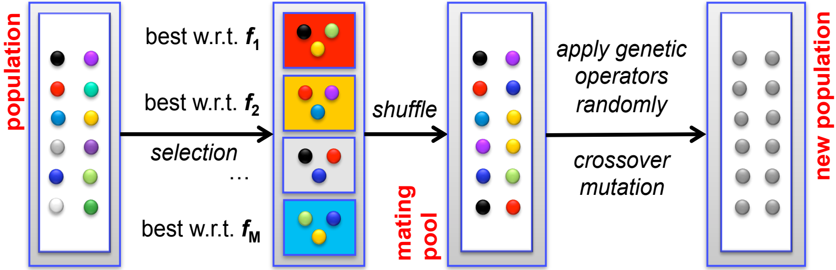
Diversity preservation:
- crowding distance (NSGA-II)
- k-nearest neighbor (SPEA2)
- hypervolume (SMS EMOA)
- decomposition (MOEA/D)

Elitism: non-elitism (VEGA), with archive (SPEA2, MOEA/D)
or without archive (NSGA-II, SMS EMOA)

VEGA – Vector Evaluated Genetic Algorithm

- **First implementation** of an evolutionary multi-objective optimization algorithm
- Objective-based selection phase

- Subpopulations are created from the old population according to each different objective separately (*multiple selections of the same solution are possible*)
- Crossover and mutation applied randomly after combining and shuffling the subpopulations



J.Schaffer, *Multiple Objective Optimization* with Vector Evaluated Genetic Algorithms. Proceedings of the 1st Intern. Conference on Genetic Algorithms, pp. 93-100, *1985*

VEGA – Pros and Cons

Advantages

- Simple idea, easy to implement, efficient
- Simple single objective genetic algorithm can be easily extended to handle multi-objective optimization problems
- When it is desirable, the algorithm has a tendency to produce solutions near the individual best for every objective

Disadvantages

- Each solution is evaluated only with respect to one objective (in MOO, all objectives are important)
- Individuals may be stuck at local optima of individual objectives
- It does not necessarily produce non-dominated solutions
- It does not have an explicit mechanism to maintain diversity
- Non-elitism: the problem of losing good solutions during the optimization process due to random effects (new population (i.e., a modified mating pool) replaces the old one)

NSGA-II – Selection Pressure

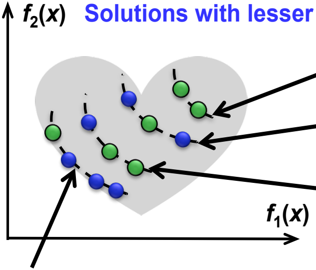
NSGA-II = Non-dominated Sorting Genetic Algorithm–II

Three features:

- It uses an elitist principle
- It emphasizes non-dominated solutions
- It uses an explicit diversity preserving mechanism

Kalyanmoy Deb

Solutions with lesser non-dominance rank are preferred



Front 4 = solutions dominated by solutions from Fronts 1, 2, and 3 (**non-dominance rank = 4**)

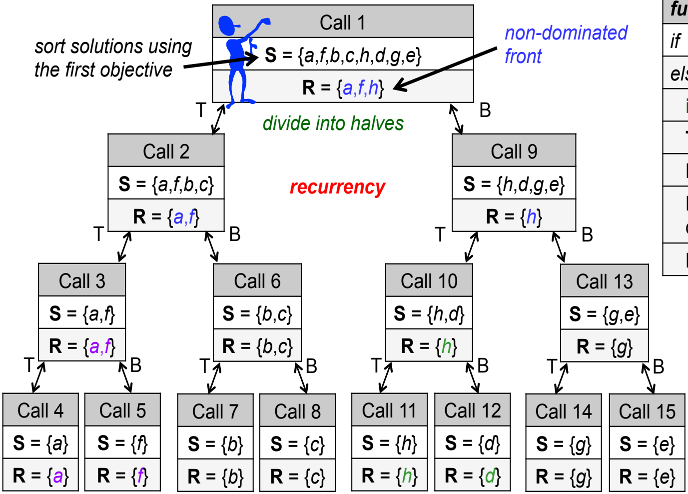
Front 3 = solutions dominated by solutions from Fronts 1 and 2 (**non-dominance rank = 3**)

Front 2 = solutions dominated by solutions from Front 1 (solutions that are non-dominated after eliminating solutions from Front 1) (**non-dominance rank = 2**)

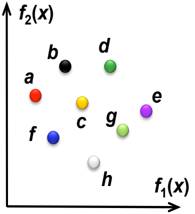
Front 1 = non-dominated solutions (**non-dominance rank = 1**)

K. Deb, S. Agrawal, A. Pratap, T. Meyarivan. A fast and elitist multi-object. genetic algorithm: NSGA-II. *IEEE Trans. on Evolution. Computation*, 6(2), 182-197, 2002

Identification of Non-dominance Front (Kung et al., 1975)



***function* R = *front*(S)**
if **|S| = 1** *then* **R = S**
else
i = *floor*(**|S|/2**)
T = *front*(S$^{1\ldots i}$)
B = *front*(S$^{i+1\ldots\|S\|}$)
N = solutions of **B** not dominated by any sol. of **T**
R = T ∪ N



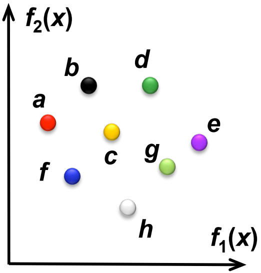
Non-dominated sorting of population

Step 1: Set all non-dominated fronts F_j, j = 1, 2, … as empty sets and set non-domination level counter j = 1

Step 2: Use some approach to find the non-dominated set P' of population P

Step 3: Update $F_j = P'$ and $P = P \backslash P'$

Step 4: If $P \neq \emptyset$, increment $j = j + 1$ and go to **Step 2**.
Otherwise, stop and declare all non-dominated fronts F_i, i = 1, 2, …, j.

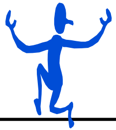


Iteration 1	Iteration 2	Iteration 3
S = {*a*,*f*,*b*,*c*,*h*,*d*,*g*,*e*}	**S** = {*b*,*c*,*d*,*g*,*e*}	**S** = {*d*,*e*}
R = {*a*,*f*,*h*}	**R** = {*b*,*c*,*g*}	**R** = {*d*,*e*}

Front 1 *Front 2* *Front 3*

NSGA-II – Diversity Preservation

- **NSGA-II** uses an **explicit diversity preserving mechanism**
- **Crowding distance** = an estimate of the density of solutions surrounding a particular solution



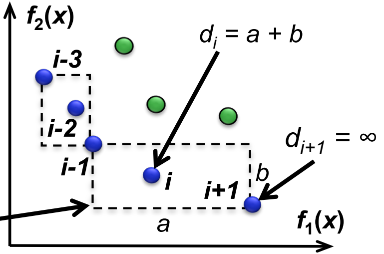
Crowding distance assignment procedure for a set of solution in front *F*

Step 1: Set crowding distance $d_i = 0$, $i = 1, 2, \dots, l$, where $l = |F|$.
Step 2: For every objective function $m = 1, 2, \dots, M$, sort the set in worse order of f_m or find sorted indices vector: $I^m = sort(f_m)$.
Step 3: For $m = 1, 2, \dots, M$, assign a large distance ∞ to boundary solutions, and for all other solutions $j = 2$ to $(l-1)$, assign as follows:

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{max} - f_m^{min}}$$

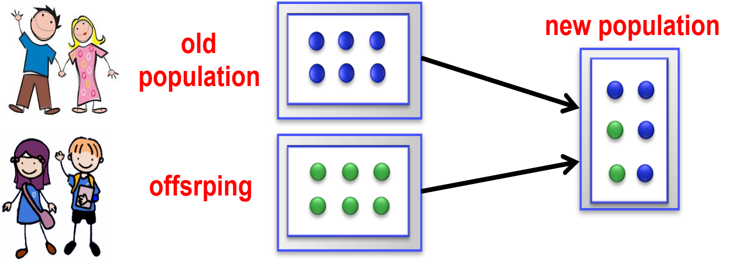
Solutions with greater crowding distance are preferred
(e.g., solution ***i*** is preferred to ***i-2***)

the perimeter of the cuboid formed by the nearest neighbors as the vertices



Elitism – Without Archive

- **Elitism** addresses the problem of losing good solutions during the optimization process due to random effects
- Combine the old population and the offspring to apply a deterministic sorting procedure – instead of replacing the old population with offspring

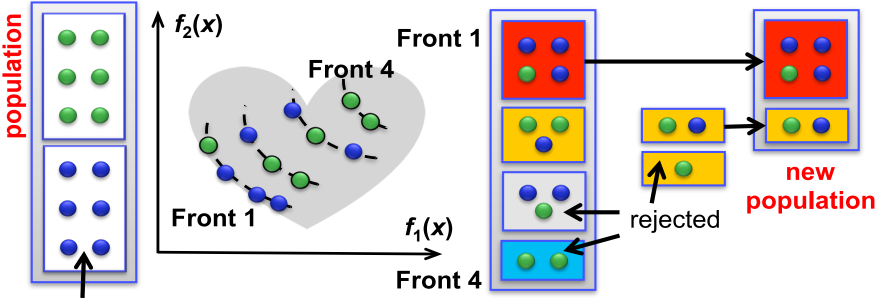


NSGA-II – Survivor Selection

Crowded tournament selection operator

Solution ***a*** wins a tournament with another solution ***b*** if any of the following conditions are true:

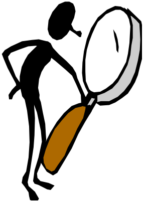
- If solution ***a*** has a better non-dominance rank than ***b***
- If they have the same rank, but solution ***a*** has a better (greater) crowding distance than solution ***b***



offspring (crossover & mutation)

elitism: by combining the old population and the offspring

NSGA-II – Pros and Cons



Advantages

- Explicit diversity preservation mechanism
- Overall complexity of NSGA-II is at most $O(MN^2)$
- Elitism does not allow an already found Pareto optimal solution to be deleted

Disadvantages

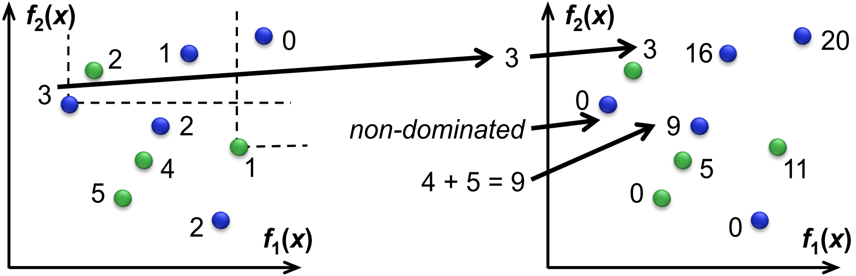
- Crowded comparison can restrict the convergence
- Non-dominated sorting on 2N size
- If the population size is small, the algorithm shows poor exploration power

SPEA2 – Selection Pressure

Fitness assignment

- Dominated and dominating solutions are taken into account
- **Strength value** = the number of dominated solutions
- **Raw fitness** = the sum of strengths of dominating solutions

S(***i***) = the number of solutions dominated by ***i***

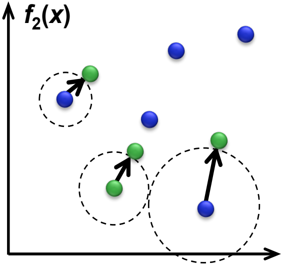


Small values of ***R***(***i***) are promoted

$$R(i) = \sum_{\text{solutions dominating } i} S(i)$$

Density information

- **Discrimination between solutions having identical raw fitness values** (***R***(***i***))
- Adaptation of the *k*-th nearest neighbor method
- σ_i^k = distance (in the objective space) of solution ***i*** to *k*-th nearest neighbor
- Most often: *k* = 1 or *k* = square root of the population size



example for k = 1

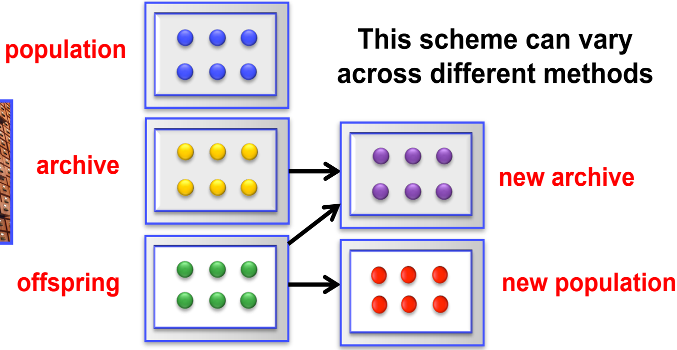
- Great values of σ_i^k are promoted
- It is possible to integrate ***R***(***i***) and σ_i^k into a fitness (then reciprocal of σ_i^k is considered)

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

E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *TIK-Report*, 2001

Elitism – With Archive

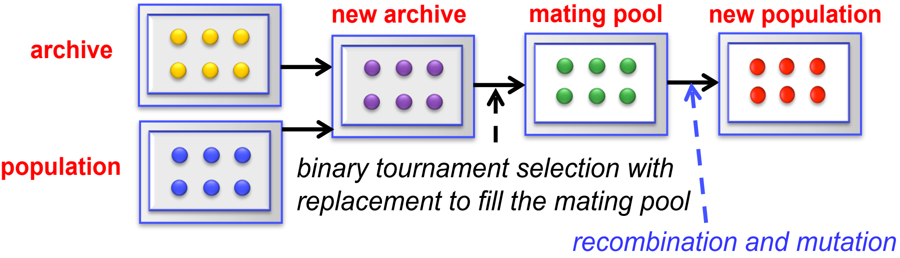
- Elitism is needed to preserve the promising solutions
- A second population, called **archive**, is maintained to which promising solutions in the population are copied in each generation
- The archive may be used as external storage or may be integrated into the algorithm by including archive members in the selection process



This scheme can vary across different methods

SPEA2 – Elitism With Archive

- The number of individuals contained in the archive is constant over time
- The truncation operator prevents boundary solutions being removed
- Copy all non-dominated solutions from the archive and population to the new archive

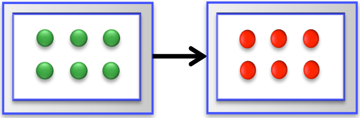


- If the archive is too small, the best solutions in the previous archive and population are copied to the new archive
- If the archive is too large, the solutions with minimum distance to another solution are systematically eliminated until the desired size is reached

Two general population management models

Generation model: each individual survives for exactly one generation

The entire set of parents is replaced by the offspring

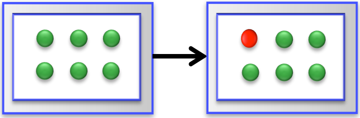


Steady-state model: a big part of solutions survive to next generation

In every generation, few high-quality solutions are selected for creating a new offspring

Low-quality solutions are removed, and the new offspring is placed in their place (not the whole population is replaced)

In the extreme case, one offspring is generated per generation and one member of the population is replaced

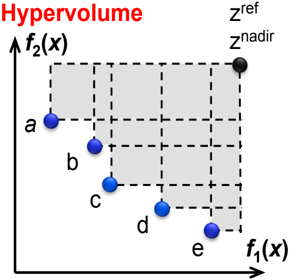


Generation gap = percentage of the population that is replaced

Hypervolume – The Most Famous Indicator in EMO

- Use **quality indicators** to assign each solution a single-objective fitness
- Typically, the **fitness** is defined by how much the quality indicator decreases if the solution is removed from the population (**indicator loss**)
- General principle: the quality indicator should be **coherent with "convergence" and "spread"**

Hypervolume

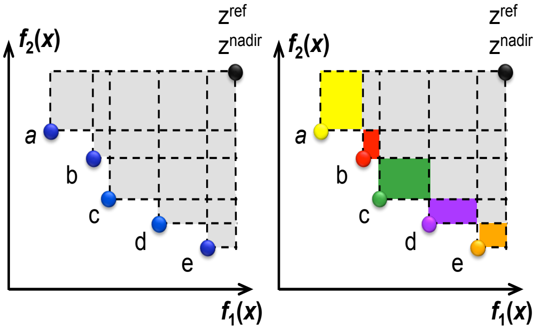


- Size (portion) of the space covered by the solutions in the population
- **Size of the dominated space**
- Lebesgue measure of the union of hypercubes defined by the non-dominated solutions
- Also called Klee's measure or *S*-metric
- Can be used to compare different sets of Pareto optimal solutions and individual solutions

A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Theory of the hypervolume indicator: optimal μ-distributions and the choice of the refer. point. FOGA'09, 87-102 2009

SMS EMOA - Indicator-based EMOA (1)

- Use quality indicators to assign each solution a single-objective fitness
- Typically, the fitness is defined by how much the quality indicator decreases if the solution is removed from the population (indicator loss)



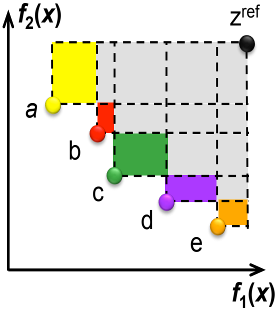
- **Unique contribution to the hypervolume**
- In some implementations, set to ∞ for the extreme solutions
- Strict Pareto compliance (captures convergence and spread)
- Favor large volume of the dominated portion

N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European J. of Oper. Research*, 181(3), 1653-69, 2007

SMS EMOA - Indicator-based EMOA (2)

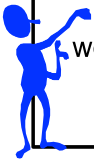
- Hypervolume has nicer mathematical properties than many other measures
- Hypervolume is expensive to calculate, though there exist several efficient implementations

Initialize Population (e.g., with random candidate solutions)
Repeat For a Pre-defined Number of Generations
Evaluate all solutions by means of their contribution to hypervolume
Generate an offspring by variation operators
Replace the worst solution in the population with the new offspring



Greedy selection scheme, which may perform arbitrarily bad if more than one solution is to be removed in each generation

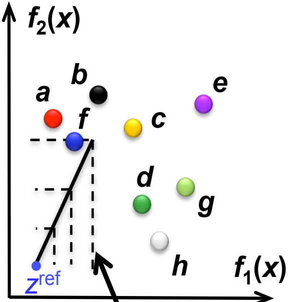
Achievement Scalarizing Function (ASF) - A Priori



Minimize the weighted Chebyshev distance (i.e., maximal weighted distance on any objective) from the reference point $\mathbf{z}^{\mathbf{ref}}$:

$$\textbf{Minimize} \;\; \max_{i=1,\dots,M} w_i \cdot |f_i(x) - z_i^{ref}|$$
$$\textbf{subject to} \;\; x \in S$$

- $\mathbf{z}^{\mathbf{ref}}$ indicates the aspiration levels (desired values that the DM would like to have) for all objectives
- Solution which has the least weighted Chebyshev distance is selected
- Objective weights determine a direction of the isoquant
- Solutions with equal distances from $\mathbf{z}^{\mathbf{ref}}$ are situated on the same isoquant

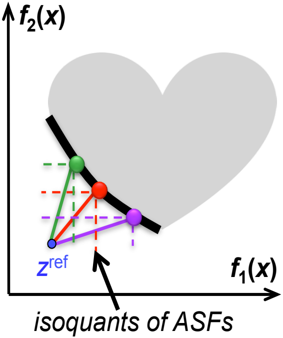


isoquant of ASF
(w_1=2/3 and w_2=1/3)

Minimize the weighted Chebyshev distance (i.e., maximal weighted distance on any objective) from the reference point $\boldsymbol{z}^{ref}$:

$$\textbf{\textit{Minimize}}\ \ max_{i=1,\ldots,M}\ w_i \cdot |f_i(x) - z_i^{ref}|$$
$$\textbf{\textit{subject to}}\ x \in S$$

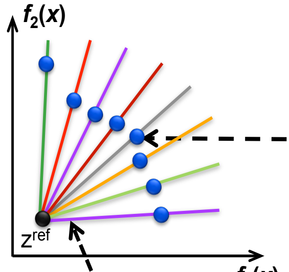
- Transformed into a posteriori method by finding the best solutions for different weight vectors
- $\boldsymbol{z}^{ref}$ can be set to the ideal point $\boldsymbol{z}^{ideal}$ (no DM's preferences)
- *Property: for any* $z^{ref} \in R^N$ and any weight vector w an optimal solution is weakly Pareto-optimal (WPO)
- *Advantage*: when using the ideal point $\boldsymbol{z}^{ideal}$ all PO solutions can be found
- *Disadvantage*: requires $\boldsymbol{z}^{ref}$ (or $\boldsymbol{z}^{ideal}$)



A. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimiz. problems. *OR Spektrum*, 8, 73-87, 1986

MOEA/D – Decomposition-based EMOA (1)

- Multiple Objective Evolutionary Algorithm based on Decomposition (MOEA/D)
- "*Old things become new again*": **decomposition converts a multiple objective problem into several single objective sub-problems**
- Decomposition is a basic idea behind many traditional mathematical programming methods for MOO problems



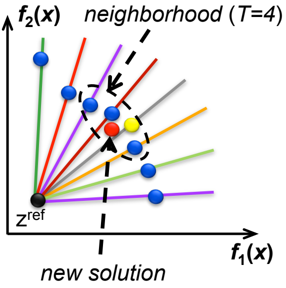
different weight vectors determining directions of the isoquants of ASFs

- Weighted Chebyshev approach (maximal weighted distance from the reference point)
- **Generate *N* uniformly distributed weight vectors (each defines a sub-problem)**
- Each solution corresponds to a sub-problem

- **Population consists of the best solutions found so far for each sub-problem**
- The algorithm is initiated with the randomly generated population
- Maintain an **external archive** to store all non-dominated solutions found during evolution

MOEA/D – Decomposition-based EMOA (2)

- **Neighborhood**: ***T*** closest weight vectors (e.g., Euclidean distance) in the neighborhood of each weight vector (including itself)
- **Hypothesis**: the optimal solutions of two sub-problems should be close to each other if the weight vectors are close to each other
- New solutions are generated by applying variation operators to solutions of neighboring sub-problems



- For each solution, randomly select a pair of neighboring sub-problems, applying variation operators to two corresponding solutions
- If the new solution is better than a solution corresponding to a neighboring sub-problem, replace that solution with the new one
- Update the reference point (in MOEA/D – it is composed of the minimal objective values in the population)
- Update the external archive

Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. on Evolutionary Computation*, 11(6), 712-31, 2007

Evolutionary Multiple Objective Optimization – Summary

Evolutionary Multiple Objective Optimization

- Evolve a **population of solutions** simultaneously
- Finally, a solution that suits the real-world application needs to be selected
- Pace of scientific developments is enormous (NSGA-II – citations!)
- Numerous concrete implementations can be developed by incorporating: existing **operators** (mutation, reproduction, mating selection); **schemes** (front-, indicator-, or decomposition-based along with population model, fitness assignment, diversity preservation, elitism, …)
- **Thousands of real-world practical applications**
- The most successful stories of EMO algorithms are for problems with at most 5 objectives, but the current trend is to make them suitable for dealing with more objectives and accounting for preference information)