Unit: Population-based Multiobjective Optimization

Learning Goals

1. What is a Metaheuristic?
2. What is an Evolutionary Algorithm (EA)?
3. Basic operators: Initialization, recombination, mutation, and selection.
4. What is probabilistic convergence? What are the convergence properties of EA?
5. How to establish ranking in multiobjective optimization?
6. State-of-the-art evolutionary multiobjective optimization:
    1. NSGA-II: diversity and convergence
    2. SMS-EMOA: hypervolume based
7. Software and recent trends in algorithm design.

Metaheuristic and Evolutionary Optimization Algorithms

Heuristic algorithms apply smart rules to solve a problem or find approximative solutions.

They often find good solutions or improvements of existing solutions, but cannot guarantee optimality.

Metaheuristics are heuristic methods that can be applied generically to a larger class of problems.

Evolutionary are metaheuristics that are mimicking adaptation processes in biological evolution.

In biological evolution the interplay between selection and variation (recombination, mutation) leads to a gradual improvement of individuals in a population with respect to how well they are adapted to their environment.

Typically a constant environment is assumed represented by the objective function and a penalty for violated constraints (fitness function).

(μ+λ)-Evolutionary Algorithm

Initialization Initialize (randomly) population with μ individuals, for instance bitstrings or real vectors.

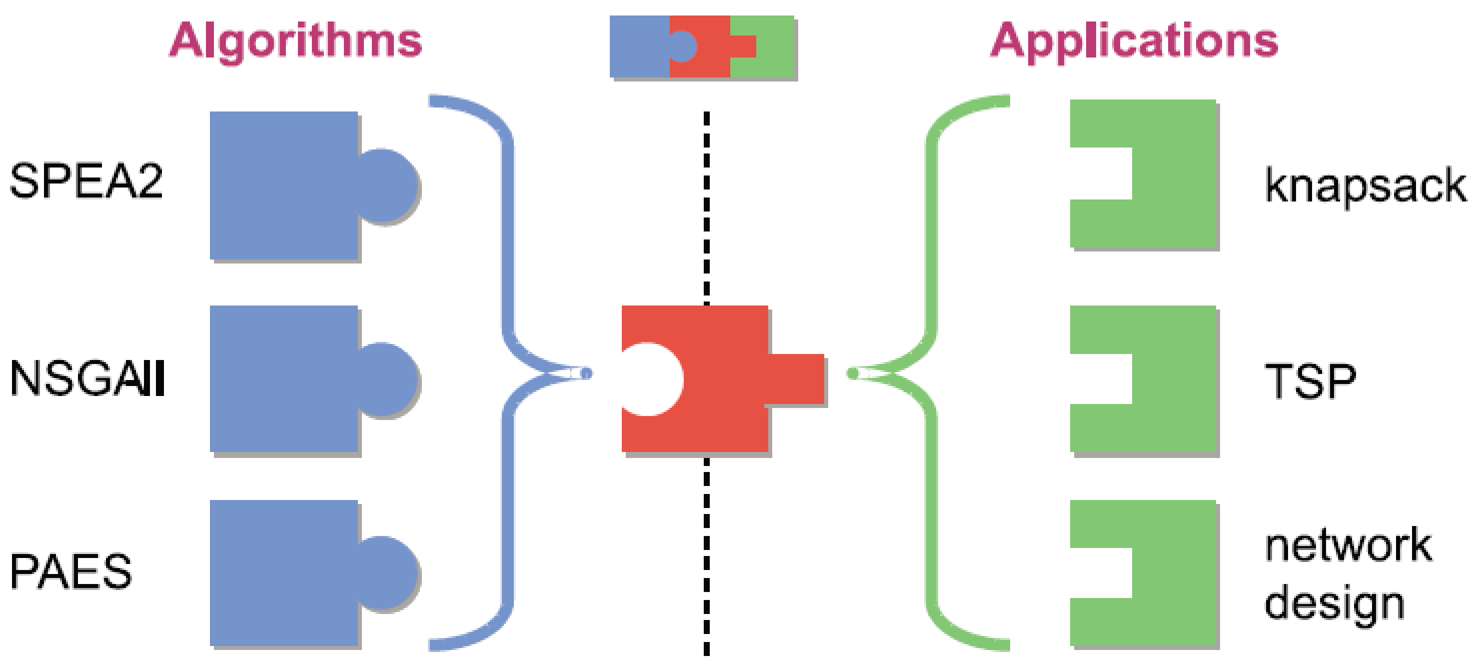
For $t = 0, 1, 2, \ldots$

- **Mating selection:** Select ρ individuals in X_t, that will serve as 'templates' for the individuals in X_{t+1}.
- **Recombination:** Combine the information of the selected individuals (e.g. by means of random crossover or of averaging) in order to create a new population, O_t, of λ offspring individuals.
- **Mutation:** Perturb (some of) the offspring individuals in O_t by means of small random modifications.
- **Fitness assignment:** Evaluate the fitness of each offspring individual, considering the corresponding objective function value (and, possibly, other criteria).
- **Environmental selection:** Select μ best individuals from $O_t \cup X_t$, in order to form the next generation parent population X_{t+1} (of size μ).

Metaheuristic algorithm design – Example: Multiobjective Optimization Library PISA

Metaheuristic (Generic)
(Evolutionary, Population,
Selection)

Instantiation (Specific)
Mutation, Recombination
Representation



The concept of PISA, a library for multicriteria optimziation (ETH Zuerich)

Selection operator

- Mating selection
  - Proportional to fitness (objective function value)
  - Proportional to rank due to objective function value
  - Uniformly randomly (Evolution strategies)
  - Tournament: Draw q individuals out of population and select best two individuals

- Replacement Selection (survivor selection)
  - (μ, λ) selection: Select μ best out of λ new variants
  - $(\mu + \lambda)$ selection: Select μ best out of λ new variants and μ solutions in current population P_t
  - $(\mu + 1)$ selection (steady-state selection): Add new solution and eliminate worst variant in population

Variation operators

- Initialization
  - 0-ary operator: init: $\Omega \rightarrow \mathbb{S}$
  - Distribute initial population in search space
  - Example: Initialize uniformly random in search space.
- Mutation
  - 1-ary (unary) operator: mutate: $\mathbb{S} \times \Omega \rightarrow \mathbb{S}$
  - Generate random variation of original solution
  - Example: Add normally distributed perturbation.
- Recombination
  - N-ary operator (N>1): $\mathbb{S} \times \mathbb{S} \times \Omega \rightarrow \mathbb{S}$
  - Combine information of N, typically 2, individuals
  - Example: Chose each solution component randomly from one of the parents.

Initialization

- Uniform initialization:

  $x_i = x_i^{min} + \text{uniform}(x_i^{max} - x_i^{min}), i = 1, \ldots, d$

- Initialization around a starting point:

  $x_i = x_i^{start} + \sigma \cdot \text{normal}(0, 1), i = 1, \ldots, d$

- Space filling designs (e.g. distance maximizing):

  $P = \max_{\mathbf{X} \subset_{\mu} \mathbb{S}} \min_{\mathbf{x}, \mathbf{x}' \in \mathbb{S}}(|\mathbf{x} - \mathbf{x}'|)$

  - Other designs: Latin hypercube designs, blocking designs maximal entropy designs, sequential designs etc.

Mutation of bitstrings and real vectors

- Bit-flip mutation

  - Flip-every bit with (small) probability p_m. Recommended (Bäck, 1996) $1/n$, where n is length of vector.

- Single bit mutation

  - Flip one randomly chosen bit

  - Search process might get trapped.

- Gaussian mutation

  - $x_i = x_i + \sigma \cdot \mathsf{Normal}(0, 1)$

  - Covariance matrix mutation $\mathbf{x} + \mathsf{Normal}(0, \Sigma)$.

  - Control σ and Σ based on success rate and autocorrelation of the more recent points in the evolution path. Needs to be adapted for multiobjective optimization; no optimal step-size.

Recombination: crossover operators

```
  1 1 1 1 | 1 1 1 1
+ 0 0 0 0 | 0 0 0 0
-------------------
= 1 1 1 1 | 0 0 0 0
```

1-point crossover: choose randomly crossover point. Choose first entries from first parent and last from second.

```
  1 1 | 1 1 | 1 1 1 1
+ 0 0 | 0 0 | 0 0 0 0
-------------------
= 1 1 | 0 0 | 1 1 1 1
```

N-point crossover: Alternate parent index at each crossover point.

```
  1 1 1 1 1 1 1 1
+ 0 0 0 0 0 0 0 0
-------------------
= 1 0 1 0 1 1 0 1
```

Uniform crossover: Choose each position randomly from one of the parents.

$$
\begin{array}{cccccccccc}
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
+ & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
= & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{array}
$$

Intermediate crossover: Choose average of parent vector position (only applicable for continuous spaces).

Theoretical analysis of recombination difficult and importance of it is discussed controversically: genetic repair hypothesis (Beyer 2001), schema theorem (Goldberg 1986), red queen hypothesis (theoretical biology)).

Recombination: advanced concepts

- Linkage learning in genetic algorithms (Goldberg, 2004)
  - With each bit also the position of the bit gets learned e.g. $(1, b_1), (2, b_3), \ldots, (d, b_{i_d})$
  - Correlated variables are more likely inherited in a coupled manner
- Simulated binary crossover (SBX) (Deb, 2001)
  - Mutation and recombination within one step
  - Distance between individuals determines radius of mutation
  - SBX is commonly used in evolutionary multiobjective optimization.

Evolutionary multicriterion optimization (EMO)

EMO Algorithms generate a series of populations $P_t, t = 1, 2, \ldots$ that gradually move towards a well distributed set of points on the Pareto front.

Popular EMO variants are NSGA-II (Deb et al. 2001), SPEA-II (Zitzler et al. 2003), SMS-EMOA (Emmerich et al. 2005), and MOEA/D (Zhang et al. 2009)

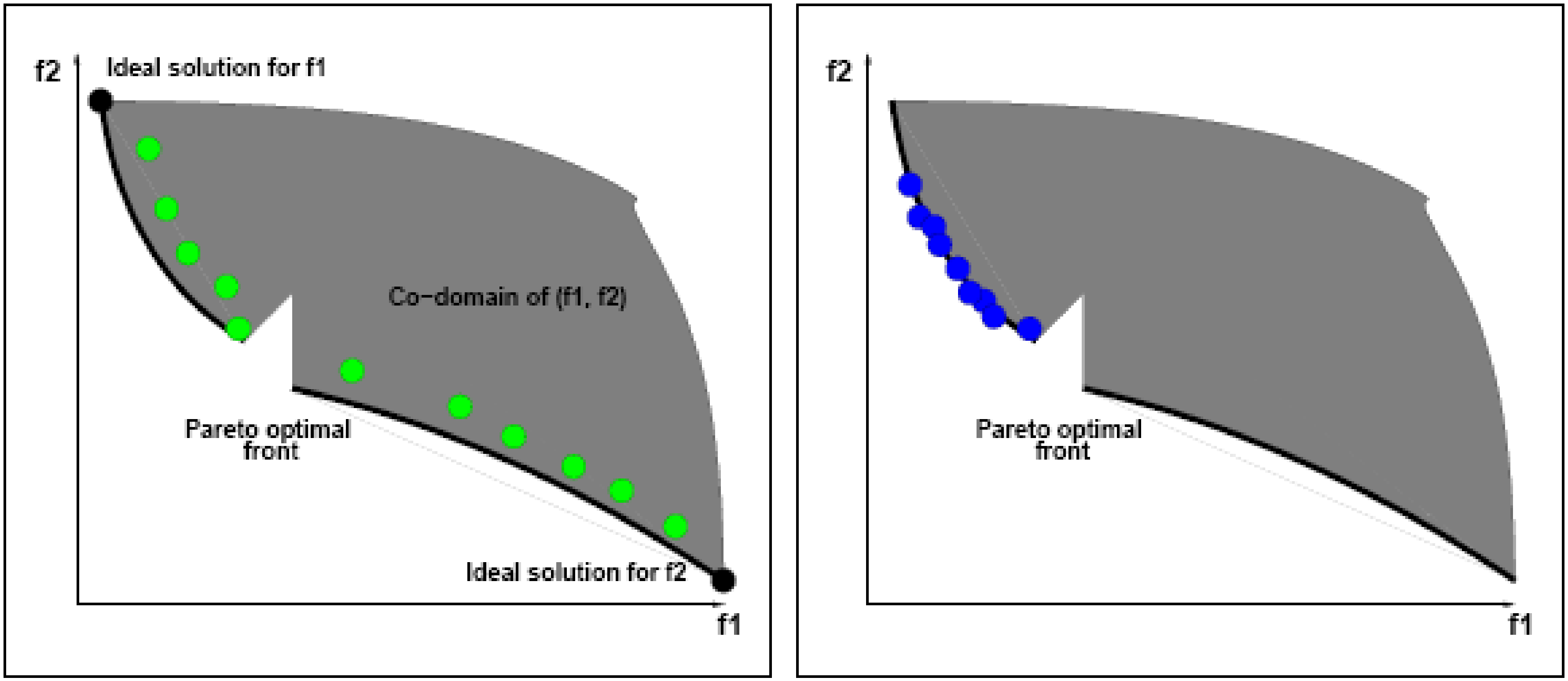
Salient topics are: (1) Statistical performance measures (2) Convergence reliability and dynamics (3) Integration in multicriteria decision making

Bi-annual international conference - EMO: 2001 (Zurich), 2003 (Faro), 2005 (Guanajuato), 2007 (Matsushima), 2009 (Nantes), 2011 (Ouro Preto), 2012 (Sheffield), 2014 (Ljubljana)

Software: JMetal (JAVA), Shark (C#), PygMOO (Python), and more ...

Evolutionary multiobjective optimization: Diversity and convergence

- The algorithm develops or evolves a finite set of search points



- Strive for good coverage and convergence to the pareto front !

Non-dominated sorting genetic algorithm (NSGA-II)

Problem: How can we assign a rank to each point of a population, if we have multiple objective functions?

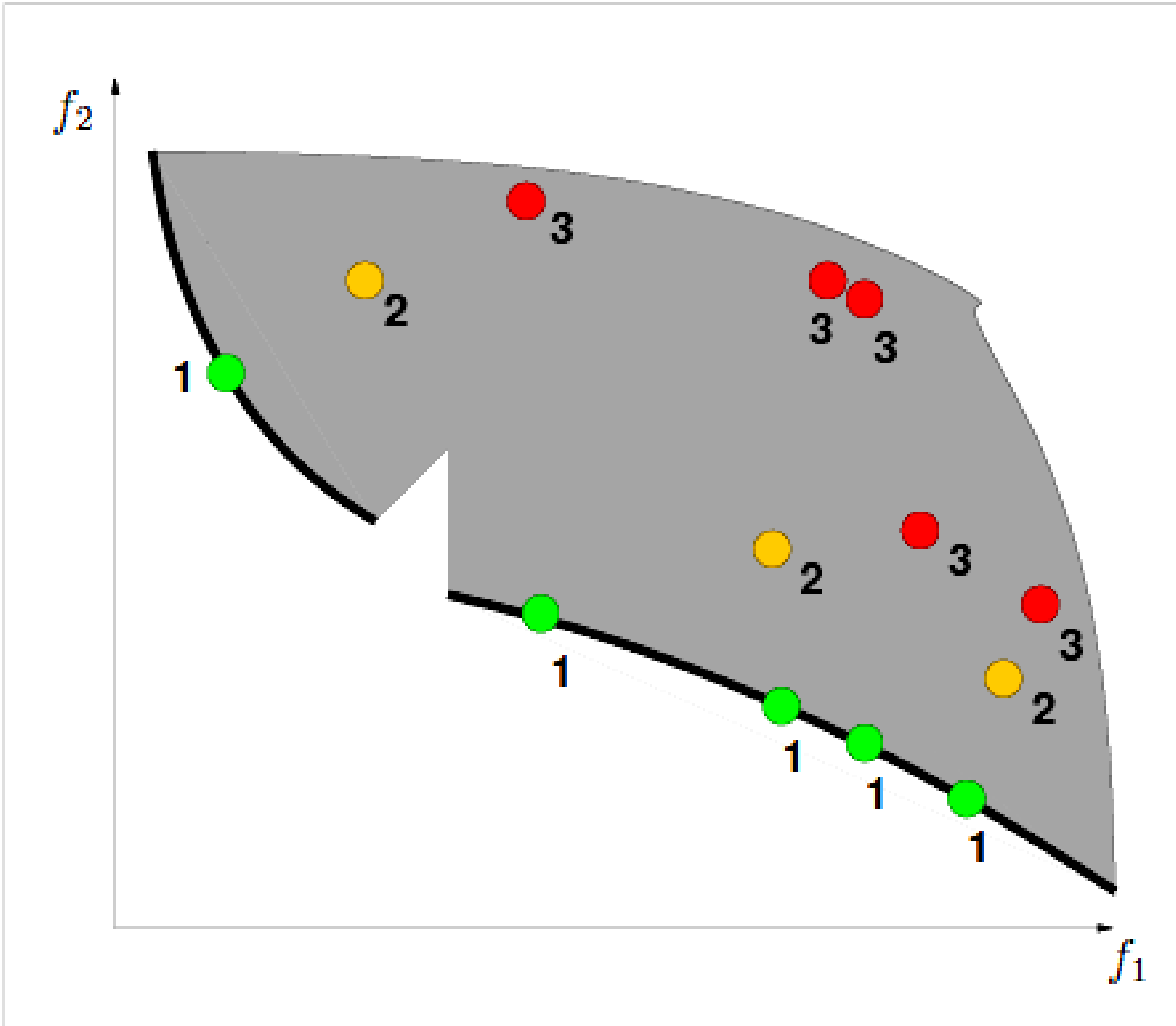
Ranking is done in two steps:

> **Non-dominated sorting**: The population is partitioned in a sequence of κ subsets (Q_1, . . . , Q_κ) based on the Pareto order, such that individuals in P_i never are dominated by individuals in P_j for $i < j$.
>
> **Crowding-distance sorting**: The subpopulations Q_i are sorted using a diversity measure - the crowding distance. Solutions that are in less crowded regions are ranked higher.

Ranking is used in $SurvivorSelection$.

K. Deb, "MultiObjective Optimization using Evolutionary Algorithms", Wiley & Sons, Chichester, 2001

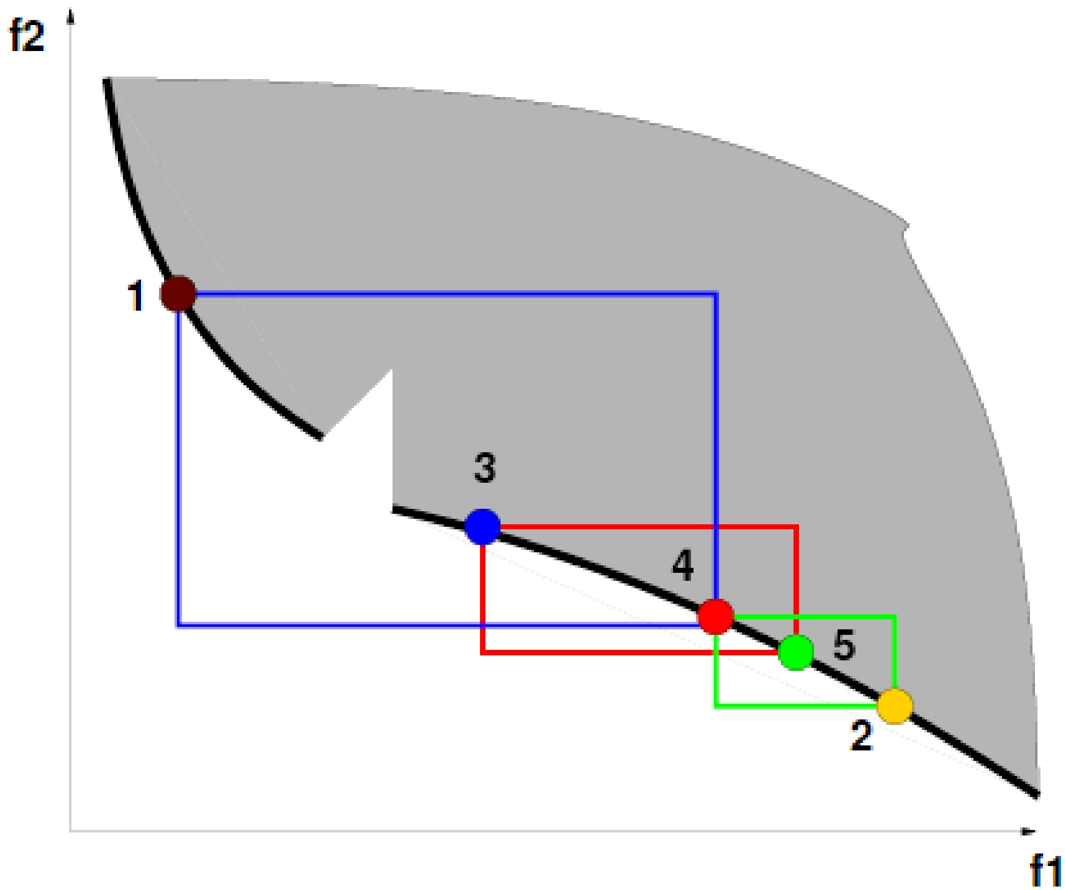
A: Non-dominated sorting



Algorithm: Non-dominated sorting

1. input: population P
2. output: ranked partitions $R_1, ..., R_k$
3. $k \leftarrow 0$
4. repeat
5. $\quad k \leftarrow k + 1$
6. $\quad R_k \leftarrow$ nondominated solutions in P
7. $\quad P \leftarrow P \setminus R_k$
8. until $P = \emptyset$

B: Crowding distance sorting

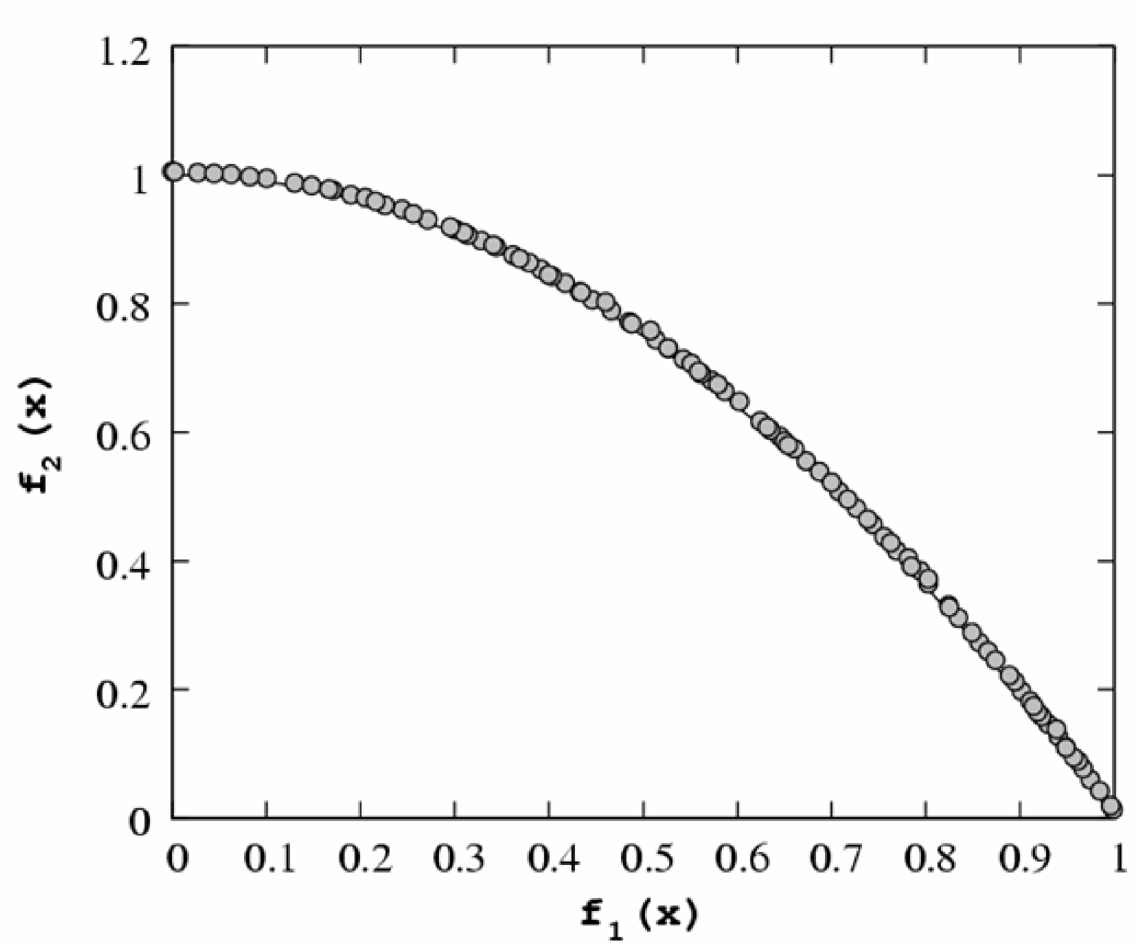


The crowding distance is only used to sort each of the partitions R_i!

Algorithm: Crowding distance sorting

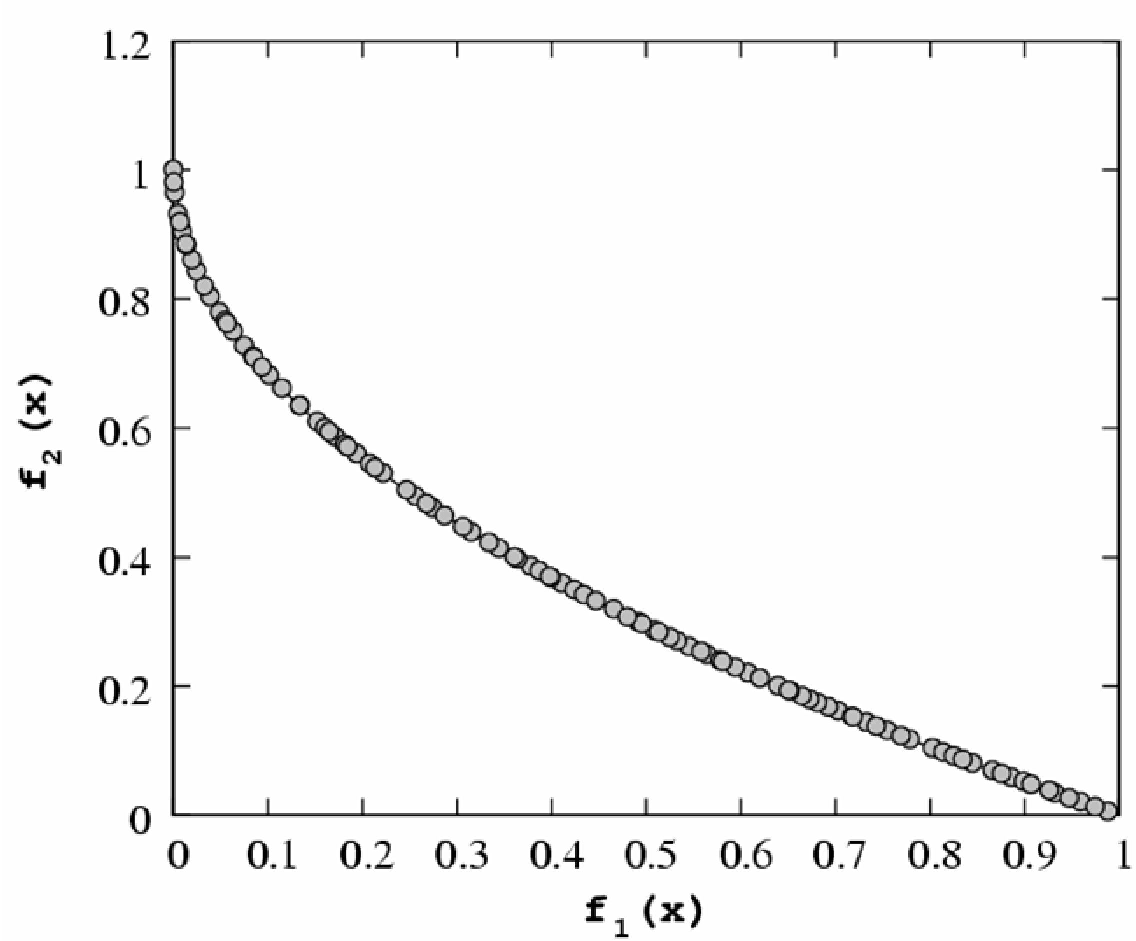
1. Input: sub-population of equal non-dominance rank R_l of size $r = |R|$
2. for $i = 1, \ldots, |R_l|$
   (a) $c_i \leftarrow 0$
   (b) set $\mathbf{x}$ to i-th point in R_l
   (c) for $j = 1, \ldots, m$ (all objectives)
      i. $L \leftarrow$ sort $R_l \backslash \{\mathbf{x}\}$ by j-th objective ascendingly
      ii. $\ell \leftarrow$ next lower j-th coordinate to x_j in L
      iii. $u \leftarrow$ next higher j-th coordinate to $x_j$in L
      iv. $c_i \leftarrow c_i + 2(u - \ell)$
3. Return crowding distances c_1, ..., c_r;

NSGA-II ZDT1*



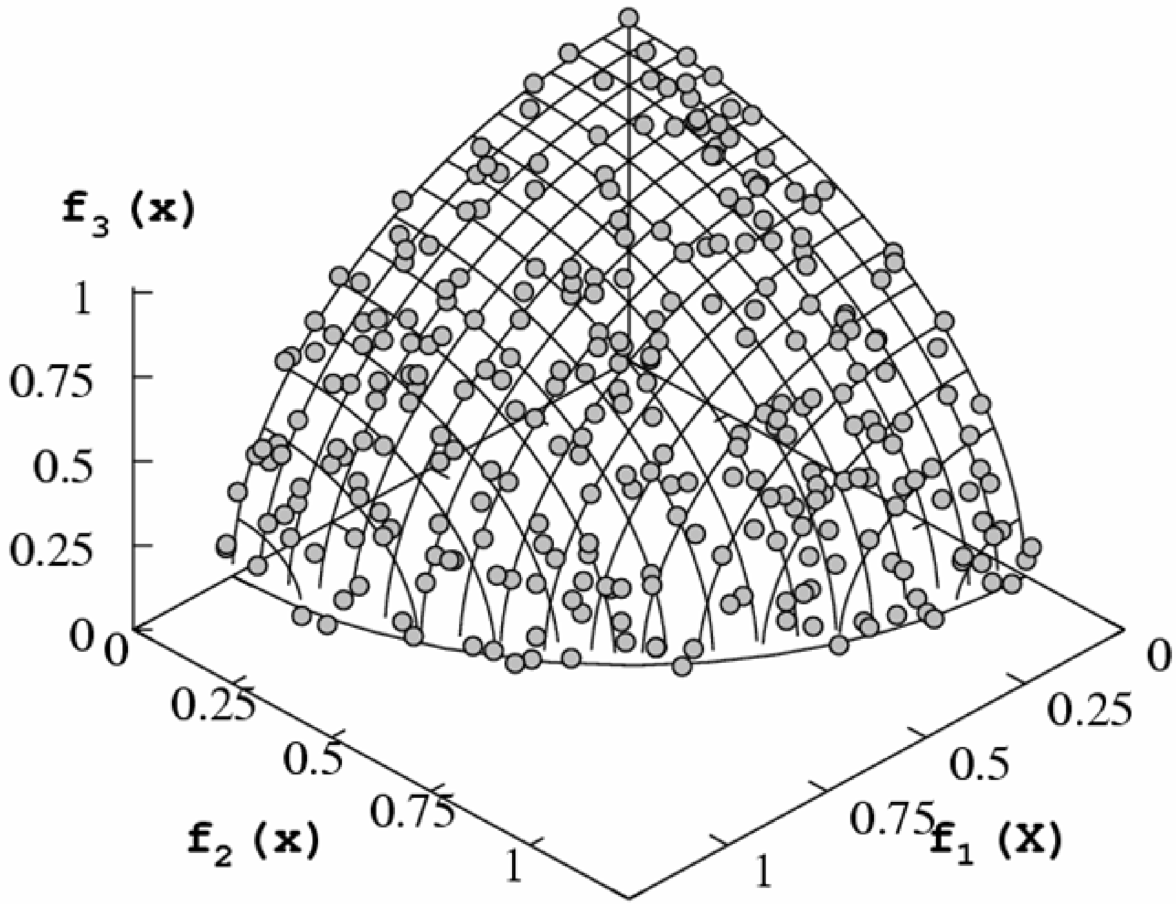
*DTLZ and ZDT are abbreviations for standard test problems in evolutionary multicriteria optimization.

NSGA-II ZDT2*



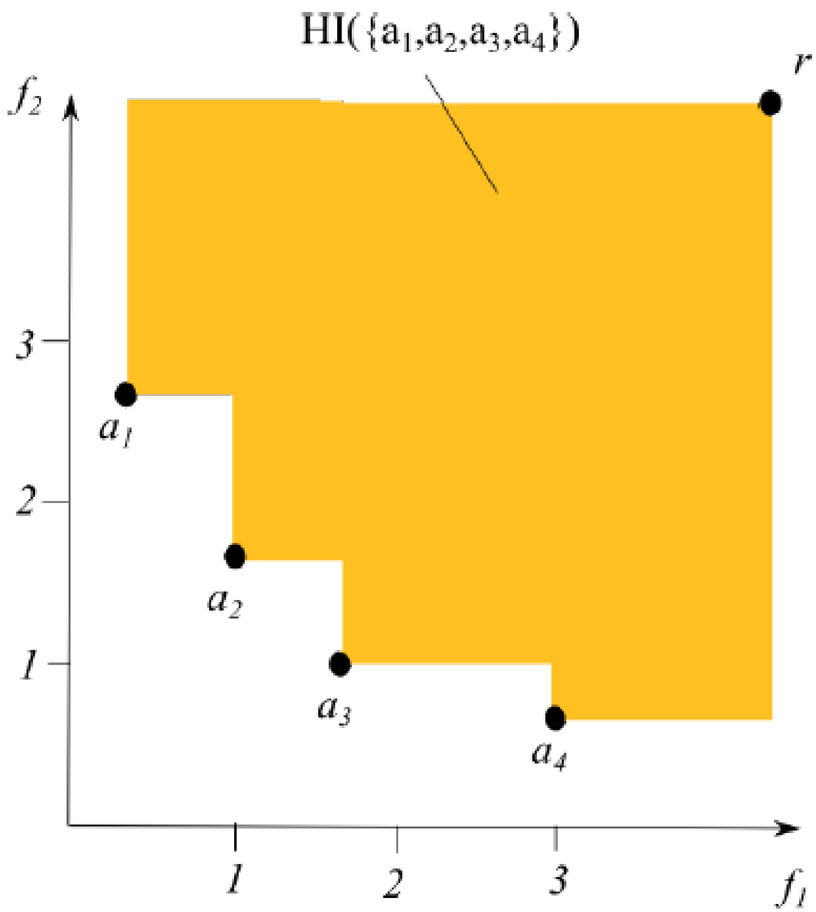
*DTLZ and ZDT are abbreviations for standard test problems in evolutionary multicriteria optimization.

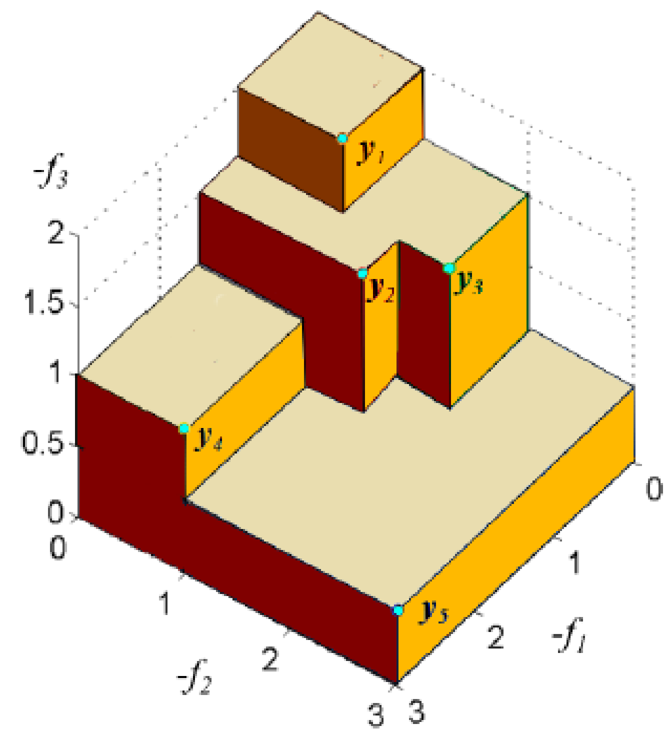
Results ZDTL4*



*DTLZ and ZDT are abbreviations for standard test problems in evolutionary multicriteria optimization.

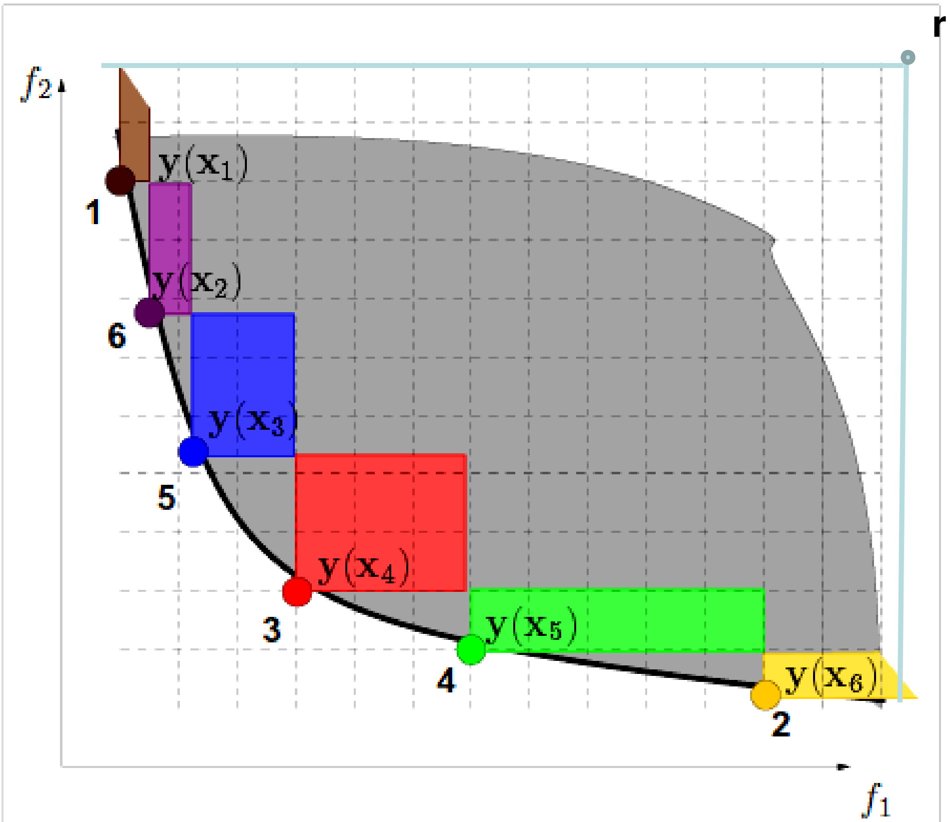
Hypervolume Indicator for assessing the quality of Pareto fronts

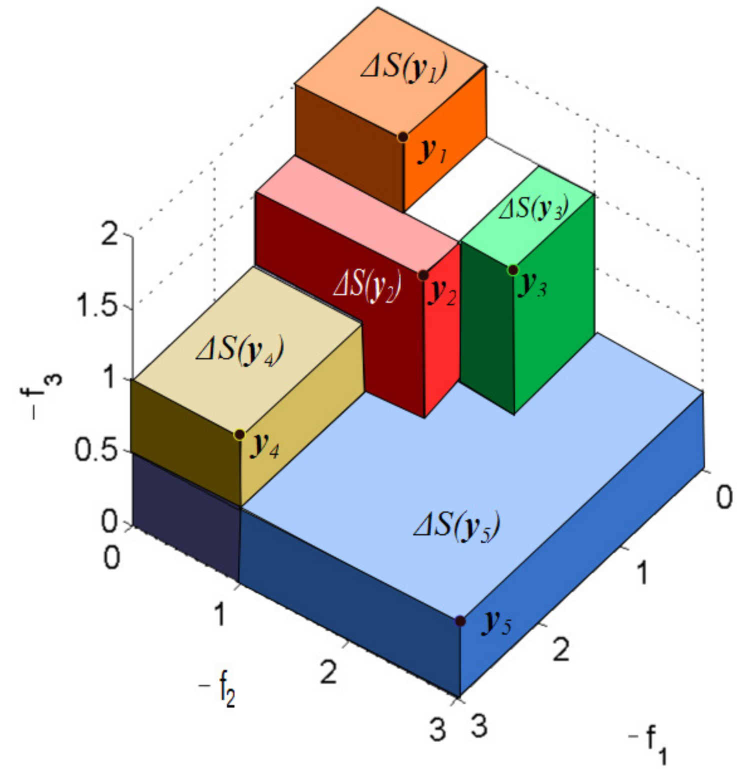




Used for performance assessment of Pareto front approximation. Computational complexity: $\Theta(\mu \log(\mu))$ for $m = 2, 3$ (Beume et al. 2010) and $\mathcal{O}(n^{m/3})$, $m > 3$ (Chang, 2013).

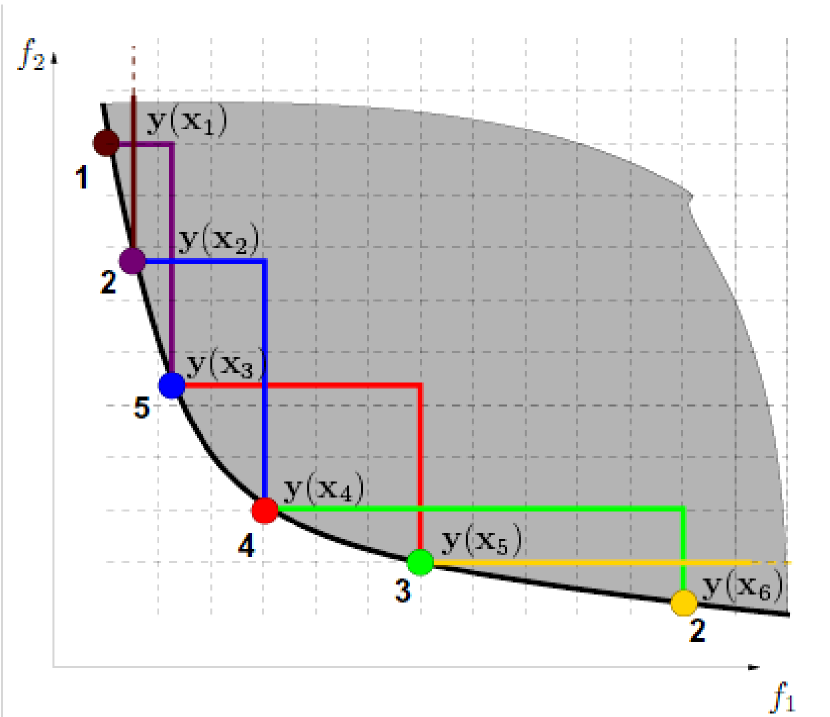
Hypervolume contributions

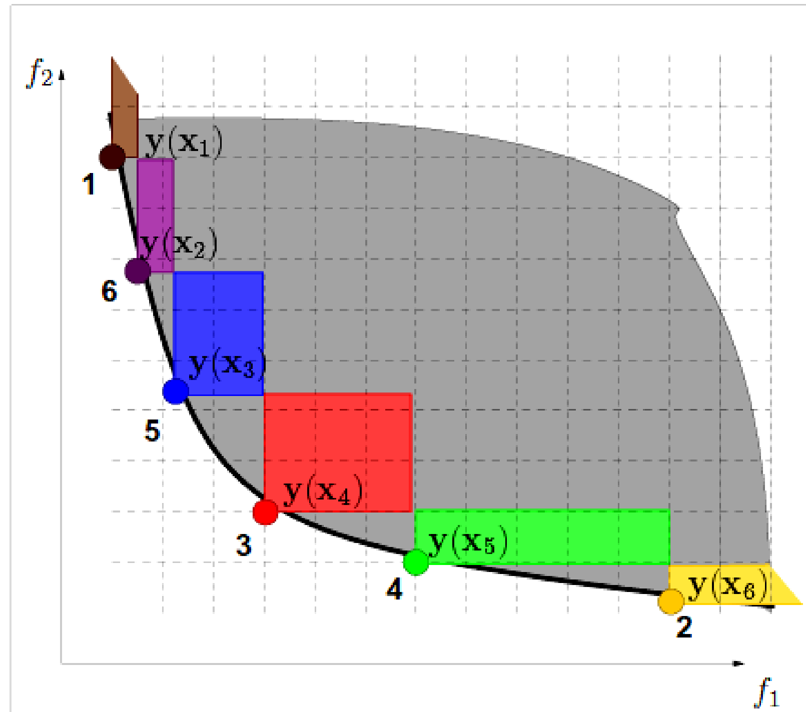




Hypervol. contribution: $\Delta\mathcal{S}(\mathbf{y}_i, Y) = \mathcal{S}(Y) - \mathcal{S}(Y \setminus \{y_i\})$.

Crowding distance vs. Hypervolume contribution in two dimensions





Computation time $\Theta(\mu \log \mu)$ for all contributions in 2-D and 3-D. (Emmerich and Fonseca, 2011)

Computing hypervolume contributions in 2D

Input: $(x_1, y_1), \dots, (x_s, y_s)$.

/ These are* s *Mutually non-dominated vectors in objective space* (f_1, f_2) *vectors sorted by first coordinate */*

Output $c_1, \dots, c_s$ (hypervolume contributions, least contributor to be discarded in SMS-EMOA)

$(x_0, y_0) \leftarrow (\infty, r_1)$

$(x_{s+1}, y_{s+1}) \leftarrow (r_{s+1,\infty})$

For $i = 1, \dots, s$

$$c_i \leftarrow (x_{i+1} - x_i)(y_{i-1} - y_i)$$

End for

Return $(c_1, \dots, c_s)$

For 3-D optimal code, see: Emmerich, Michael TM, and Carlos M. Fonseca. "Computing hypervolume contributions in low dimensions: Asymptotically optimal algorithm and complexity results." *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, Berlin, Heidelberg, 2011.

SMS EMOA

- S-Metric Selection Evolutionary Multiobjective Optimization Algorithm (Emmerich et al. 2005, Beume et al. 2007).
- Basic idea: Find population with maximal hypervolume coverage.
- The hypervolume indicator measures the quality of a Pareto front approximation; Roughly, it is the size of the space that is dominated by the population.
- SMS-EMOA uses non-dominated sorting but crowding distance ranking is exchanged by hypervolume contribution.
- Uses a $(\mu + 1)$ selection scheme (one offspring per generation).
- On popular benchmarks SMS-EMOA clearly outperforms NSGA-II and other state-of-the-art EMOA.
- Fast SMS-EMOA (Hupkens et al. 2013) replaces non-dominated sorting by queuing and features $\mathcal{O}(\log(\mu))$ processing time per generation;

Approximations of Pareto fronts achieved with SMS-EMOA on GSP test problems problems

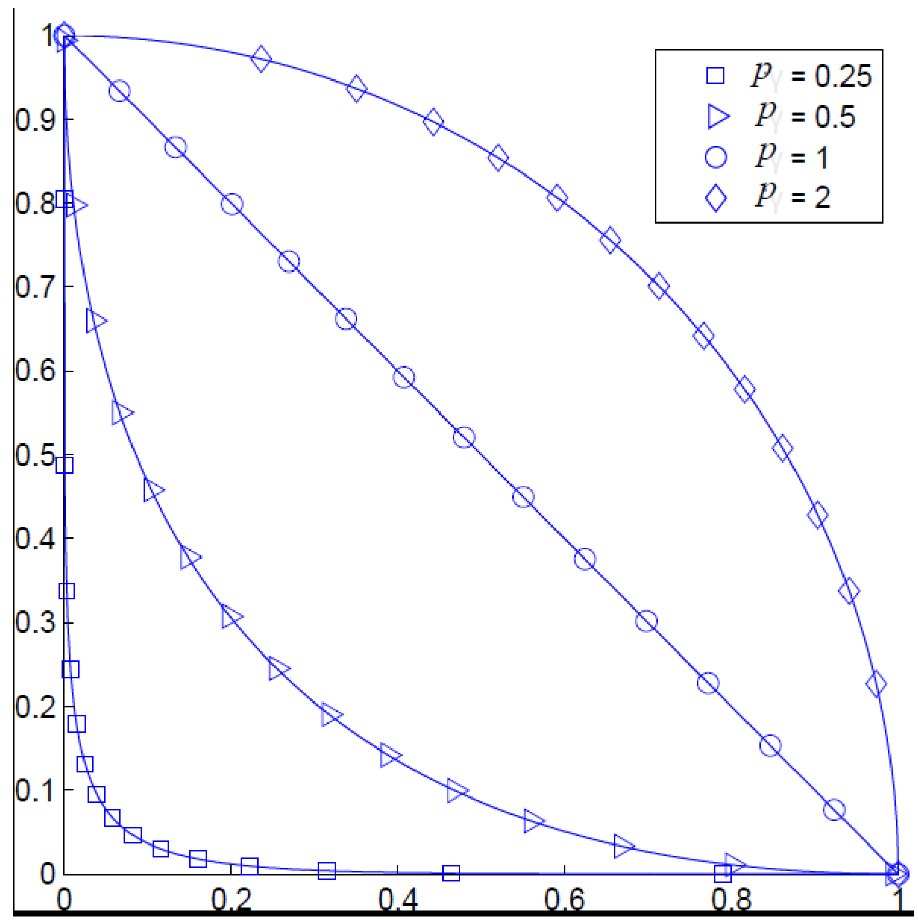
$$f_1(\mathbf{x}) = 1/d^\gamma \sum_{i=1}^{d} (|x_i|^\gamma)$$

$$f_2(\mathbf{x}) = 1/d^\gamma \sum_{i=1}^{d} (|x_i - 1|^\gamma)$$

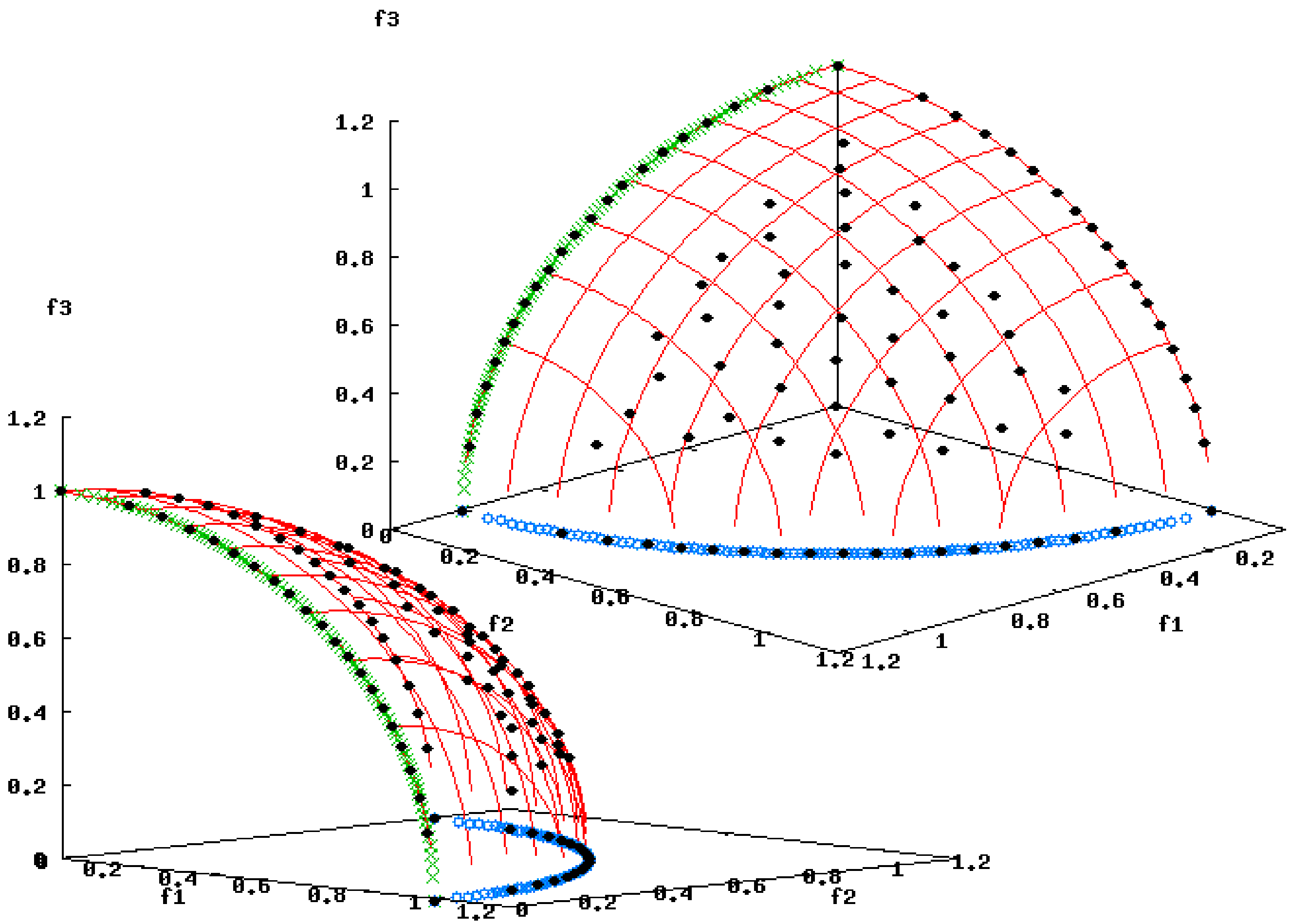
$$\gamma = 1/p$$

The parameter p determines curvature of Pareto front.

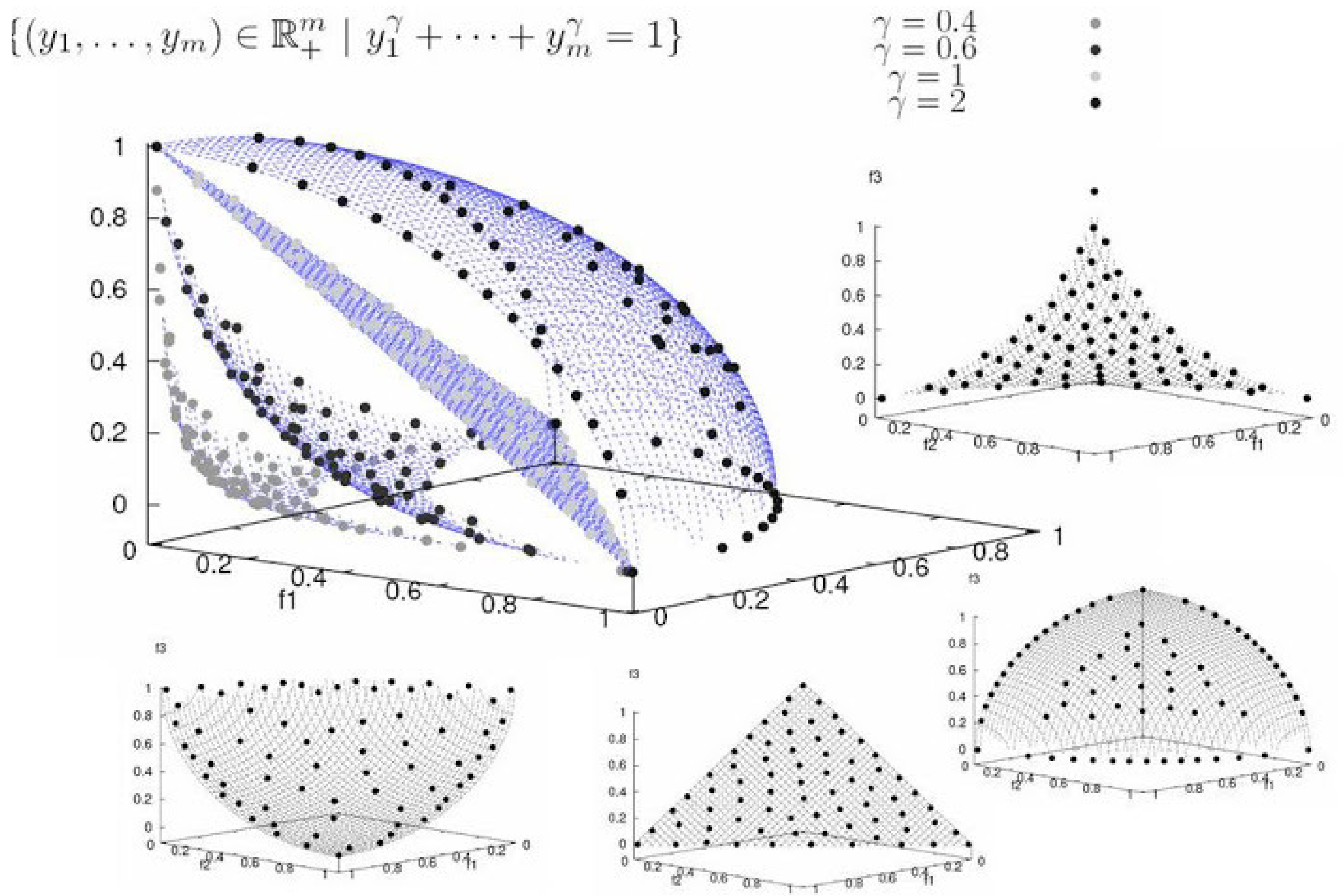
For test problems, see: Emmerich, Deutz: Test Problems based on Lamé Superspheres, EMO, Matsushima, 2007



SMS-EMOA results, DTLZ1



Distribution of points achieved with SMS-EMOA



Emmerich, Deutz:Test Problems based on Lamé Superspheres, EMO, Matsushima, JP, 2007

Simplified NSGA-II and SMS-EMOA

1. Input: t_{max}, mutation stepsizes: $\sigma_1, \ldots, \sigma_d$
2. Initialize population $P_0 = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\mu)})$ by generating μ random vectors in feasible domain for input variables $[x_1^{(min)}, x_1^{(max)}] \times \cdots \times [x_d^{(min)}, x_d^{(max)}] \in \mathbb{R}^d$. .
3. Evaluate objectives and constraints for points in P_0

initialization

4. For $t = 1, 2, ..., t_{max}$
   - (a) Set $\mathbf{x}^{new} \in \mathbb{R}^d$ by selecting random index $s \in \{1, \ldots, \mu\}$ and set $x_i^{new} = x_i^{(s)} + \sigma_i N(0,1)$, $i = 1, \ldots, d$.
   - (b) Remark: N(0,1): generates standard normal distributed random number.
   - (c) Evaluate objectives and constraints for $\mathbf{x}^{new}$

   mutation

   - (d) Set $P_t \leftarrow P_{t-1} \cup \{\mathbf{x}^{new}\}$;
   - (e) If P_t contains dominated or infeasible solutions then delete randomly one of these, otherwise delete individual with smallest crowding distance (NSGA-II) or hypervolume contribution (SMS-EMOA) among the non-dominated points.

   selection

5. Return feasible solutions in P_t

Generalization to multiobjective optimization

Convergence Analysis of $(\mu + \lambda) - EA$ for multiobjective optimization

Rudolph, G.; Agapie, A., "Convergence properties of some multi-objective evolutionary algorithms," in Evolutionary Computation, 2000. Proceedings of the 2000 Congress on , vol.2, no., pp.1010-1016 vol.2, 2000;
doi: 10.1109/CEC.2000.870756

For finding minimal sets of general partial orders:

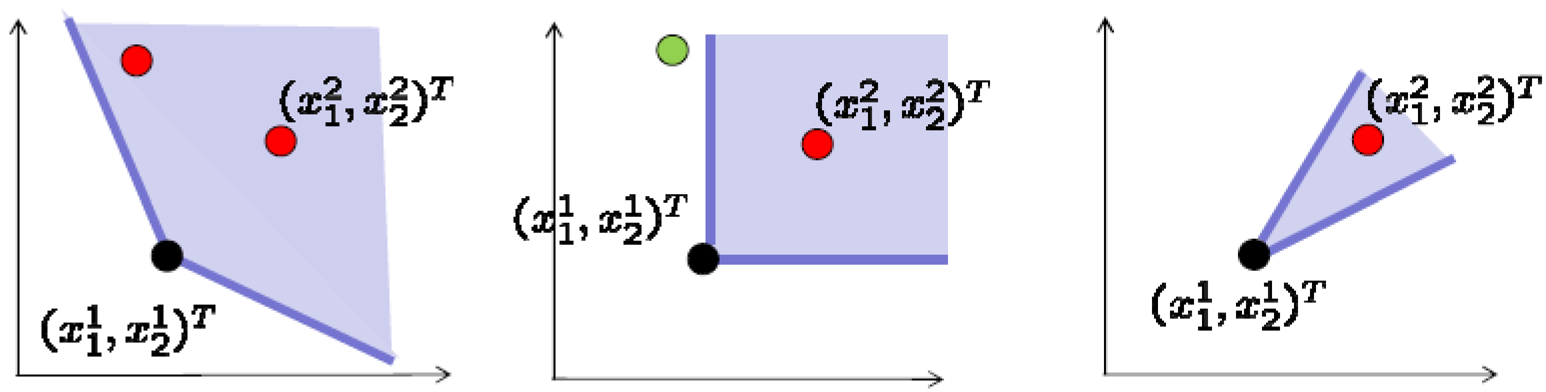
Rudolph, G. (2001). A partial order approach to noisy fitness functions. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on* (Vol. 1, pp. 318-325). IEEE

On the 0-1 knapsack problem:

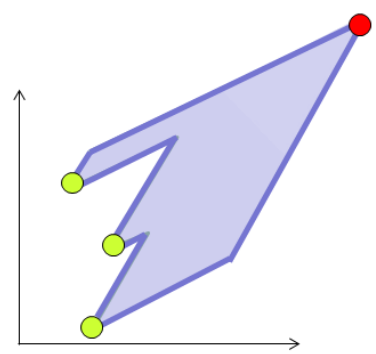
Kumar, Rajeev, and Nilanjan Banerjee. "Analysis of a multiobjective evolutionary algorithm on the 0–1 knapsack problem." *Theoretical Computer Science* 358.1 (2006): 104-

Some new results ...

- Cone-base hypervolume for trade-off bounded optimization

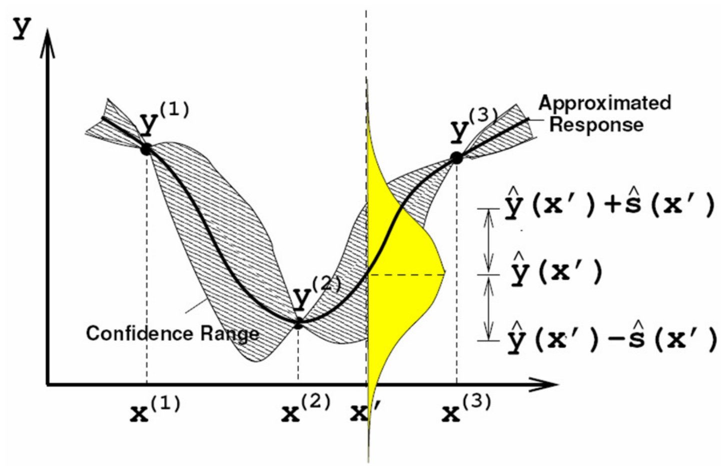


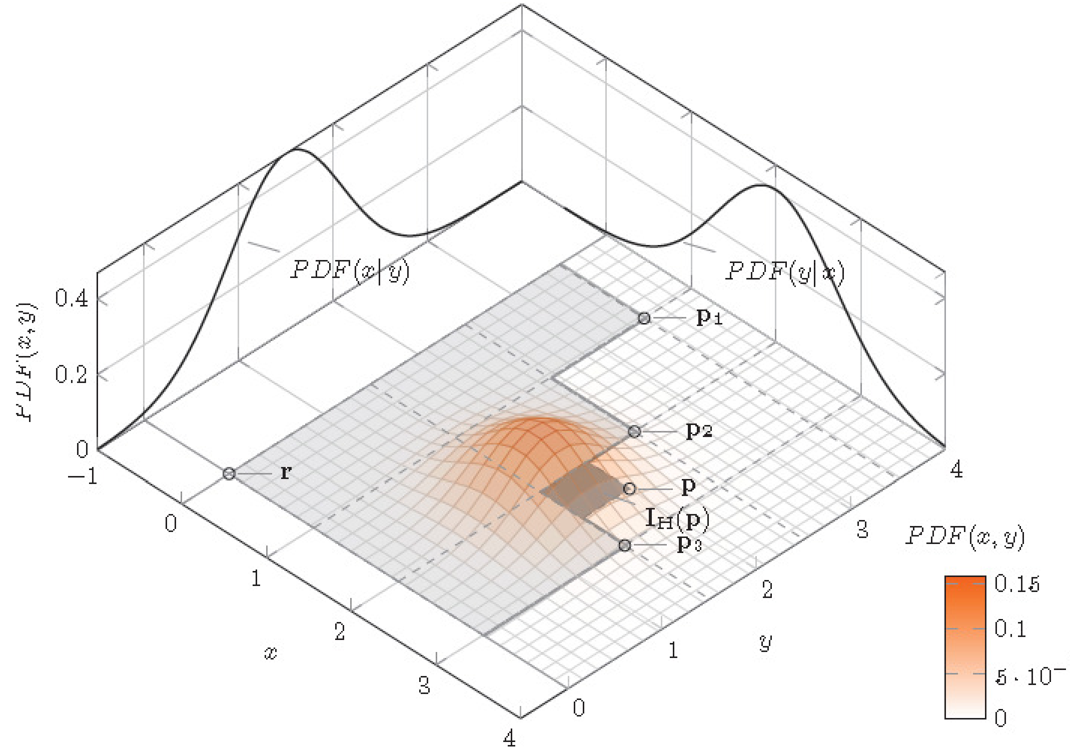
- Cone-based hypervolume



Shukla, P. K.; Emmerich, M. & Deutz: A Theoretical Analysis of Curvature Based Preference Models, *EMO, Springer,* **2013**, *7811*, 367-382

New results ...



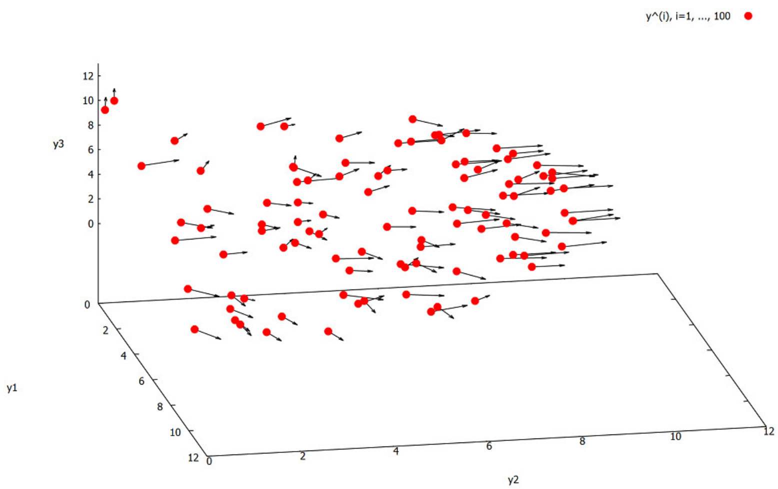


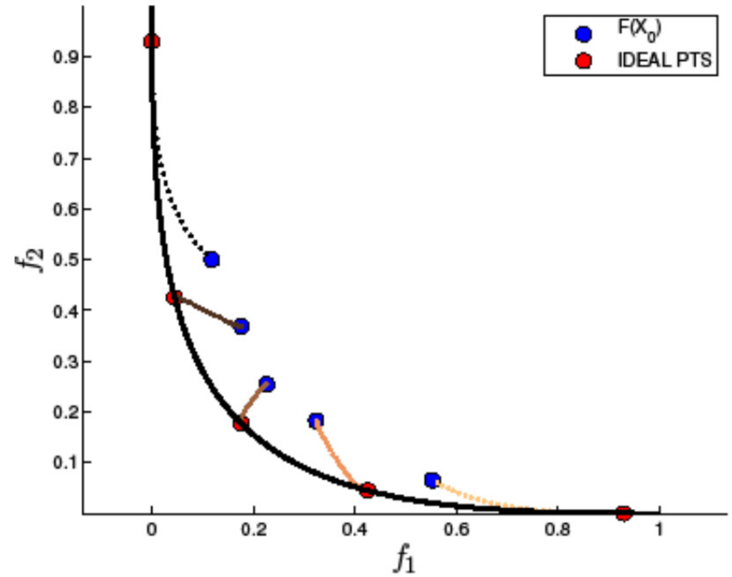
- Evaluations are costly: An archive of previous evaluations is kept
- Before evaluating it, the result is estimated from a machine learning tool based on previous evaluation including uncertainty of prediction
- Most promising point is evaluated and added to archive

Iris Hupkens, Kaifeng Yang, André Deutz, Michael Emmerich: Faster Computation of the Expected Improvement, ArXiV, 2014.

Some new results ...

- Hypervolume gradient/set based Newto's method





- Steering the population in the direction of steepest hypervolume ascent.
- The population is seen as a vector and the partial derivatives to all coordinates of all points with respect to the hypervolume are taken
- Emmerich, M. & Deutz: Time Complexity and Zeros of the Hypervolume Indicator Gradient Field, EVOLVE, 2014, 500, 169-193

Take home message

- Distinguish: Exact methods (guarantee optimality), Heuristic (smart methods for finding good solutions when exact methods are not available), Metaheuristics (generic heuristics).
- Evolutionary algorithms are population based metaheuristics using selection, recombination, mutation operators. They implement stochastic processes on population state space and probabilistically converge to optimal solutions → suitable for Pareto front approximation; very flexible (real valued, integer variables)
- NSGA-II uses non-dominated sorting for ranking based on dominance; and diversity based ranking: crowding distance
- Hypervolume indicator measures the dominated (hyper)volume
- SMS-EMOA maximizes the hypervolume indicator; crowding distance is replaced by hypervolume contribution; yields more regular distribution than NSGA-II and progress can be analyzed.
- Evolutionary multiobjective optimization is active field of research; analysis and extensions to non-standard problems are hot topics.

Key references

- Emmerich, M. T., & Fonseca, C. M. (2011, January). Computing hypervolume contributions in low dimensions: asymptotically optimal algorithm and complexity results. In *Evolutionary Multi-Criterion Optimization* (pp. 121-135). Springer Berlin Heidelberg.
- Bäck, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms* (Vol. 996). Oxford: Oxford university press.
- Rudolph, G. (1996, May). Convergence of evolutionary algorithms in general search spaces. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on* (pp. 50-54). IEEE.
- Chan, Timothy M. "Klee’s Measure Problem Made Easy." (Accepted for: Foundations of Computer Science, 2013)
- Beume, N., Fonseca, C. M., López-Ibáñez, M., Paquete, L., & Vahrenhold, J. (2009). On the complexity of computing the hypervolume indicator. *Evolutionary Computation, IEEE Transactions on*, *13*(5), 1075-1082.

Key references

- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, *6*(2), 182-197.
- Hupkens, I., & Emmerich, M. (2013). Logarithmic-Time Updates in SMS-EMOA and Hypervolume-Based Archiving. In *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV* (pp. 155-169). Springer International Publishing.
- Beume, N., Naujoks, B., & Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, *181*(3), 1653-1669.
- Custódio, A. L., Emmerich, M., & Madeira, J. F. A. (2012). Recent Developments in Derivative-free Multiobjective Optimization.