Deterministic and Stochastic Algorithms for Multiobjective Optimization

Overview and Recent Advances

Michael T. M. Emmerich
Natural Computing Group
Leiden Center for Advanced Computer Science
Leiden University
The Netherlands

November 24, 2009
Deterministic and Stochastic Algorithms for Multiobjective Optimization
Overview and Recent Advances

Michael T. M. Emmerich
Natural Computing Group
Leiden Center for Advanced Computer Science
Leiden University
The Netherlands

November 24, 2009
Table of Contents
- Thomas Bäck (Professor, Natural Computing, Group Leader)
- Michael Emmerich (Assistant Professor, Multiobjective Optimization)
- Andre Deutz (Lecturer, Multiobjective Optimization Theory)
- Rui Li (Post Doc, Multiobjective Optimization, Ant Opt.)
- Johannes Kruisselbrink (PhD Student, Robust Optimization)
- Lars Willmes (PhD Student, Physical Systems Optimization)
- Ingo Heinle (PhD Student, BMW)
- Ron Breukelaer (PhD Student, Cellular Automata Optimization)
- Lorenzo Sella (PostDoc, Mathematical Biology), 10 master students
Teaching (Master Level)

- Natural Computing (Th. Bäck)
- Evolutionary Algorithms (Th. Bäck)
- Multiobjective Optimization and Decision (M. Emmerich)
- Optimization in Bioinformatics (M. Emmerich, A. Deutz)
- Nonlinear Computer Simulation (A. Deutz, M. Emmerich)
Projects at LIACS

- Multiobjective De-novo Drug Design (Andreas Bender, Michael Emmerich, www.pharma-it.net)
- Multiobjective Molecular Alignment (Michael Emmerich, Lars Willmes, AMOLF/FOM)
- Simulated Evolution (Lorenzo Sella, NWO)
- Evolutionary Mixed Integer Optimization (Rui Li, EU Grant)
- Robust Design Optimization (Johannes Kruisselbrink, NWO)
- Theory of Multiobjective Optimization (Andre Deutz)
- Industrial Optimization in Building Engineering, Cars, etc.

More info: https://natcomp.liacs.nl
Multiobjective Optimization

Minimize $f_1(x)$, Minimize $f_2(x)$, ..., Minimize $f_m(x)$, $x \in S$

- $x$ is the feasible search space (e.g. $\mathbb{R}^d$, $\mathbb{Z}^d$, or space of molecule graphs)
- $f_1, ..., f_m$ are the objective functions (In case of maximization: Minimize $-f(x)$).
Conflicting objectives

- Consider the case of disjoint set of minimizers for some $f_i, f_j, i \neq j$

$$\arg \min \{ f_i(x) | x \in X \} \cap \arg \min \{ f_j(x) | x \in X \} = \emptyset$$

- in that case $f_i$ and $f_j$ are conflicting
- we cannot minimize both at the same time ...
Dealing with conflicting objectives

In case of conflicting objectives we may ...

1. State some objectives as constraints, or
2. Aggregate objectives by means of an utility function $u$
   
   Maximize $u(f_1(x), ..., f_m(x))$, or

3. Prioritize objective functions (lexicographic order), or
4. Compute all Pareto optimal solutions and analyze trade-offs, then decide

Classification

- 1, 2, 3 are *a-priori methods*: decision-making before optimization
- 4 is a *a-posteriori* method: decision-making after optimization
- *Progressive* allow user interaction during search and can combine 1, 2, 3 or 4
Pareto domination

- The concept of Pareto domination defines a transitive order relation on $X$

$$\forall x \in X : x \text{ dominates } x', \text{ in symbols } x \prec x' \text{ iff:}$$

- for all $i \in \{1, \ldots, m\} : f_i(x) \leq f_i(x')$, and
- exists $j \in \{1, \ldots, m\} : f_j(x) < f_j(x')$.

We also define:

- $x$ is indifferent to $x'$ in symbols: $x \sim x' \text{ iff: } f(x) = f(x')$
- $x \preceq x' \text{ iff: } x \sim x' \text{ or } x \prec x'$
- $x$ is incomparable to $x'$ in symbols: $x \parallel x' \text{ iff: } x \not\preceq x' \text{ and } x' \not\preceq x$
Cone order in objective space

- Define \( y \oplus S = \{ y + s | s \in S \} \) (Minkowski sum).
- We can define Pareto domination in the objective space as:
  \[
y \preceq y' \text{ iff: } y' \in y \oplus (\mathbb{R}_+,0)^m
\]
- \( \preceq \) defines a partial order (transitive, antisymmetric) on \( \mathbb{R}^2 \)
Cone order in objective space

- The relation $y \preceq y'$ is a special example of a cone order.
- It is preserved under monotonous transformations:
  \[ y \preceq y' \implies m(y) \preceq m(y'), \quad m \text{ is a strictly monotonous function.} \]
- ⇒ optimal solutions in search space will remain optimal if objective functions are transformed by $m$. 

Michael Emmerich, LIACS
Deterministic and Stochastic Algorithms for Multiobjective Optimization
Efficient set and Pareto front

For a multiobjective optimization problem the *efficient set* $X_E$ is defined as

$$X_E = \{ x \in X | \text{not exits } x' \in X : x' \prec x \}$$

For a set $Y$ and the order $\prec$ the non-dominated set is defined as

$$Y_N = \{ y \in Y | \text{not exits } y' \in X : y' \prec y \}$$

For a multiobjective optimization problem we define the Pareto front as $(f(X))_N$ or the non-dominated set of the attainable subspace $f(X)$ of $\mathbb{R}^m$
Example for a Pareto front for a biobjective problem and a 3-D search space. (OV = objective function value.)
Existence of Pareto fronts

- If the attainable set $f(X)$ is compact (bounded and closed) the Pareto front is non-empty.
- For unbounded, but also for bounded non-closed sets $f(X)$ it may not exist (e.g. $f(X) = (0, 1]^m$).
- For the dimension of the Pareto front it holds that $\dim(\text{PF}) \leq m - 1$.
- For the dimension of the efficient set in general only $\dim(X_E) \leq \dim(X)$ holds.
Functional Dependencies in Pareto fronts

- Let $Y_{i_1}, \ldots, Y_{i_{m-1}}$ define $m - 1$ different coordinates of the Pareto front, then the remaining coordinate $Y_{i_m}$ is functionally dependent on $Y_{i_1}, \ldots, Y_{i_{m-1}}$

- In 2-objective case we can plot $f_2$ as a function of $f_1$ on a (subset of) $\mathbb{R}$

- In 3-objective case we can plot $f_3$ as a function of $f_1$ and $f_2$ over a (subset of) $\mathbb{R}^2$
Bird’s eye view on a 3-D Pareto front

For moves in positive direction of $f_1$ and $f_2$ (north-east bound) the value of $f_3$ will decrease.
Convexity of Pareto fronts

- Define the dominated subspace of $Y_N$ as $\text{dom}^+(Y_N) = \bigcup_{y \in Y_N} y \oplus (\mathbb{R}^+)^m$.
- A Pareto front $Y_N$ is called convex, iff $\text{dom}^+(Y_N)$ is convex.
- Define the dominating subspace of $Y_N$ as $\text{dom}^-(Y_N) = \bigcup_{y \in Y_N} y \ominus (\mathbb{R}^+)^m$.
- A Pareto front $Y_N$ is called concave, iff $\mathbb{R}^m \setminus \text{dom}^-(Y_N)$ is a convex set.
- A Pareto front can be convex, convex, both (linear shape), or consists of convex and concave partitions.
Shapes of Pareto fronts for bi-objective problems

Convex pareto front

$f_2$

$\gamma_{SE}$

$f_1$

Concave pareto front

$f_2$

$\gamma_{SE}$

$f_1$

Convex and concave parts

$f_2$

$\gamma_{SE}$

$f_1$

Disconnected Pareto front

$f_2$

$\gamma_{SE}$

$f_1$
Trade-off Analysis

- Left Picture: A region of good compromise solution exists
- \( M1 \) is unfavorable move: little gain for \( f_1 \), large loss for \( f_2 \)
- \( M2 \) presents a move with a balanced trade-off
- Right Picture: Either \( f_1 \) or \( f_2 \) can be improved (\( M3, M4 \)). Not both at the same time.
Deterministic Algorithms

- Linear Weights/Homotopy Method
- Continuation Method
- S-Metric Gradient
Pareto front approximation

- Continuous Pareto fronts defined by black-box functions can be approximated by a finite *approximation set*.
- Pareto front (PF) approximation algorithms seek to find finite approximations of the Pareto front with a good *coverage* and *closeness*.
Linear weighting

► A naïve approach for approximating the Pareto front is to minimize the linear scalarization

\[ f_L(x) = w_1 f_1(x) + \ldots + w_m f_m(x) \rightarrow \min \]

for a well spread set of weights \( w_1, \ldots, w_m \in [0, 1]^m \)

► it is easy to prove that if \( 0 \prec w \) the solution is Pareto optimal, but

► solutions in concave parts of the PF cannot be obtained this way
Convex case

\[ y_2 = C - \frac{w_1}{w_2} y_1 \]

\[ y_1 U^{\text{linear}}_{w_1,w_2} = \omega_1 f_1 + \omega_2 f_2 \]
Concave case

\[ y_2 = C - \frac{w_1}{w_2} y_1 \]

\[ \nabla U_{\omega_1, \omega_2}^{linear}(y) \]

\[ U_{\omega_1, \omega_2}^{linear} = \omega_1 f_1 + \omega_2 f_2 \]
Homotopy method

We want to compute a uniformly spaced set of Pareto optimal points on the Pareto front which can be described implicitly as the path $x(\lambda), \lambda \in [0, 1]$ with

$$x(\lambda) = \arg\min_{x \in \mathbb{R}^n} (1 - \lambda)f_1(x) + \lambda f_2(x)$$  \hspace{1cm} (2)

and, assuming convexity and continuous differentiability, this can be expressed as:

$$(1 - \lambda)\nabla f_1(x) + \lambda \nabla f_2(x) = 0$$  \hspace{1cm} (3)
Homotopy method

Given a point $x(\lambda_t)$ we can compute the next point $x(\lambda_{t+1})$ as follows. First we compute a search direction:

$$\tilde{v} := x(\lambda_t + \epsilon) - x(\lambda_t)$$

(4)

where

$$x(\lambda_t + \epsilon) = \arg \min_{x \in \mathbb{R}^n} (1 - \lambda_t - \epsilon) f_1(x) + (\lambda_t + \epsilon) f_2(x)$$

(5)

and $\epsilon$ is an appropriately small positive number. The normalized $v := \frac{\tilde{v}}{||\tilde{v}||}$ is used as the search direction.
Homotopy method

Compute the step size $h \in \mathbb{R}_+$ along $v$ in the decision space $x_{t+1} = x_t + hv$ such that $\|F(x_t) - F(x_{t+1})\|_\infty = \Theta \Delta$ (where $\Theta \in (0, 1)$ is a safety factor). In case $F$ is Lipschitz continuous we know that there exists an $L \geq 0$ such that

$$\forall x, x' \in \mathcal{X}, \|F(x) - F(x')\| \leq L\|x - x'\|$$  \hspace{1cm} (6)

The Lipschitz constant around $x_t$ can be estimated by:

$$L_{x_t} := \|DF(x_t)\|_\infty = \max_{i=1}^2 \|\nabla f_i(x_t)\|_1$$  \hspace{1cm} (7)

Combining 6 and 7, using $h = \|x_t - x_{t+1}\|$, and assuming $h$ is sufficiently small, we obtain the following estimate:

$$h \approx \frac{\Theta \Delta}{L_{x_t}}$$  \hspace{1cm} (8)
Homotopy method

Find the next Pareto optimal point $x_{t+1}$. Apply the $\epsilon$-constraint method where the constant $\epsilon$ is computed as the second coordinate of the expression

$$\epsilon := \pi_2(F(x_t) + F'(hv)) \quad (9)$$

From which we get: $x_{t+1} := \text{arg min} \, f_1(x)$ such that $f_2(x) = \epsilon$. 
Continuation Methods

Assume for we are given a differentiable mapping (of class $C^r$, $r > 1$):

$$H : S \subset \mathbb{R}^{n+k} \rightarrow \mathbb{R}^n$$

and want to find all points for which $H(x) = 0$.

The solution set $M = \{x \in S|H(x) = 0, x = \text{regular}\}$ is a $k$ dimensional $C^r$ manifold.

Continuation methods approximate $M$ numerically. Starting from an initial point $x^* \in M$ they use predictor-corrector procedure to compute further points:

- **P** Predict a set $\{p_1, \ldots, p_s\}$ of distinct and well distributed points which are near both to $x^*$ and to $M$.

- **C** Use $p_i$ as a starting point for local minimization of $\|H(x_i^*)\|$. 
Continuation Methods

Karush Kuhn Tucker Conditions for Pareto Optima: Let $x^*$ be a locally efficient point. Then there exists $\alpha_1, \ldots, \alpha_m \geq 0$, with $\sum_{i=1}^{m} \alpha_i = 1$ such that $\sum_{i=1}^{m} \alpha_i \nabla f_i(x^*) = 0$
**Linear Level sets**

- if the levelcurves are (locally) linear and intersecting in $x$, then $x$ cannot be efficient
- there exists a cone that includes dominating solutions (convex combinations of $\nabla f_1(x)$ and $\nabla f_2(x)$)
Pareto front as level set of $H : \mathbb{R}^{d+m} \rightarrow \mathbb{R}^{d+1}$

Find all points $H(x_1, \ldots, x_d, \alpha_1, \ldots, \alpha_m) \in \mathbb{R}^d \times \mathbb{R}_{+,0}^m$ for which

$$H(x_1, \ldots, x_d, \alpha_1, \ldots, \alpha_m) = \left( \begin{array}{c} \frac{\partial \sum_{i=1}^{m} \alpha_i f_i(x)}{\partial x_1} \\ \vdots \\ \frac{\partial \sum_{i=1}^{m} \alpha_i f_i(x)}{\partial x_d} \\ 1 - \sum_{i=1}^{m} \alpha_i \end{array} \right) = 0$$

- Under smoothness conditions the $m - 1$ dimensional manifold for which this equation is fulfilled contains all Pareto optimal solutions.
- Continuation methods can be used to explore this manifold.
Continuation Methods

Algorithm CONT-Recover\(^\dagger\)

1. compute initial set \( \mathcal{B} \) of boxes \( B(c_i, d) \) around points \( c_i = (x^B, \alpha^B)^T \), \( H(c_i) \approx 0 \)
2. mark all boxes
3. for all marked boxes \( B \in \mathcal{B} \)
   - unmark box \( B \) with center \( (x^B, \alpha^B)^T \)
   - compute a set of \( k - 1 \) orthonormal vectors that span the tangent space \( T_{x^B, \alpha^B} M \) by QR factorization of \( H'(x^B, \alpha^B) \)
   - generate predictors \( p_1, \ldots, p_{k-1} \) for each basis vector direction
   - starting from \( p_1, \ldots, p_{k-1} \) compute correctors \( c_1, \ldots, c_{k-1} \) with \( H(c_i) \approx 0 \)
   - add all \( B(c_i, d) \) to \( \mathcal{B} \) for which \( B(c_i, d) \not\in \mathcal{B} \)
Continuation Method Box Recover

from: [Schütze et al. Dagstuhl MOO, 2005]
Continuation Method Box Recover

Fig. 4. Computation of MOP (16): The figures show all projections of the Pareto set in the image space for $n = 10$.

from: [Schütze et al. Dagstuhl MOO, 2005]
S-Metric Gradient Method

- S-Metric as quality indicator for Pareto front approximations

\[ S(P) = \text{Vol}\{\mathbf{y} \mid \exists \mathbf{y}^{(i)} \in P : \mathbf{y}^{(i)} \prec \mathbf{y} \land \mathbf{y} \prec \mathbf{f}^{\text{max}}\} \]
Maximizing the S-Metric

Approach: Maximize S-Metric in order to find well spread Pareto front approximation
Maximizing the S-Metric in 3-D
S-Metric gradient for population

- We represent a population $P$ of size $\mu$, $P \subseteq \mu \mathcal{X}$, as a vector of length $\mu \cdot d$:

$$ p = (x_1^{(1)}, \ldots, x_d^{(1)}, \ldots, x_1^{(\mu)}, \ldots, x_d^{(\mu)})^\top = (p_1, \ldots, p_{\mu \cdot d})^\top. $$  

- The mapping from $\mu d$-vectors to populations is defined as:

$$ \Psi(p) = \{ (x_1^{(i)}, \ldots, x_d^{(i)})^\top | i \in \{1, \ldots, \mu\} \}. $$

- Different $\mu d$-vectors may represent the same population (but not vice-versa).

- Every non-empty population $P \subseteq \mu \mathcal{X}$ is represented by at least one tuple of the form above.
S-Metric gradient for population

- We represent a population \( P \) of size \( \mu \), \( P \subseteq \mu \mathcal{X} \), as a vector of length \( \mu \cdot d \):
  \[
p = (x_1^{(1)}, \ldots, x_d^{(1)}, \ldots, x_1^{(\mu)}, \ldots, x_d^{(\mu)})^\top = (p_1, \ldots, p_{\mu \cdot d})^\top.
\]
  \( (12) \)

- The mapping from \( \mu d \)-vectors to populations is defined as:
  \[
  \Psi(p) = \{ (x_1^{(i)}, \ldots, x_d^{(i)})^\top \mid i \in \{1, \ldots, \mu\} \}.
  \]
  \( (13) \)

- Different \( \mu d \)-vectors may represent the same population (but not vice-versa).

- Every non-empty population \( P \subseteq \mu \mathcal{X} \) is represented by at least one tuple of the form above.
S-Metric gradient for a population

A general definition of the gradient for the space of $\mu d$-vectors is

$$\nabla_p S = \left( \frac{\partial S}{\partial p_1}, \ldots, \frac{\partial S}{\partial p_{\mu d}} \right)^T$$  \hspace{1cm} (14)

the following structure of the composition of mappings is applied:

$$\mathbb{R}^{\mu d} \xrightarrow{\text{F}} \mathbb{R}^{\mu m} \xrightarrow{S} \mathbb{R}^+.$$  \hspace{1cm} (15)
S-Metric gradient for a population

1. How can the S-metric gradient be computed?
2. Chain rule: Let $x^{(1)}, x^{(2)}, \ldots, x^{(\mu)}$ be $\mu$ points in the decision space, then $\nabla S(p)$ can be written as:

$$
S' \text{ at } \begin{pmatrix}
    f(x^{(1)}) \\
    f(x^{(2)}) \\
    \vdots \\
    f(x^{(\mu)})
\end{pmatrix} \circ \begin{pmatrix}
    f' \text{ at } x^{(1)} & 0 & 0 & \cdots & 0 \\
    0 & f' \text{ at } x^{(2)} & 0 & \cdots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & 0 & 0 & f' \text{ at } x^{(\mu)}
\end{pmatrix}
$$
S-Metric gradient for a population

- A more detailed description of this matrix is given as:

$$\nabla S(y^{(1)},...,y^{(\mu)})^T \cdot \begin{pmatrix} \frac{\partial f_1(x^{(1)})}{\partial x_1^{(1)}} & \cdots & \frac{\partial f_1(x^{(1)})}{\partial x_d^{(1)}} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(x^{(1)})}{\partial x_1^{(1)}} & \cdots & \frac{\partial f_m(x^{(1)})}{\partial x_d^{(1)}} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \frac{\partial f_1(x^{(\mu)})}{\partial x_1^{(\mu)}} & \cdots & \frac{\partial f_1(x^{(\mu)})}{\partial x_d^{(\mu)}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \frac{\partial f_m(x^{(\mu)})}{\partial x_1^{(\mu)}} & \cdots & \frac{\partial f_m(x^{(\mu)})}{\partial x_d^{(\mu)}} \end{pmatrix} \cdot F'(x^{(1)},...,x^{(\mu)})$$
Maximizing the S-Metric

Approach: Maximize S-Metric in order to find well spread Pareto front approximation
Gradient computation in 3-D

- ... in 3-D the gradient values are given by area segments of the attainment are to be computed:
Gradient-based S-metric maximization

1: **input variables**: initial population as $\mu d$ vector $p$
2: **control variables**: accuracy of line search $\alpha_{\text{min}}$, step reduction rate $\tau \in (0, 1)$
3: $i \leftarrow 0; p^{\text{best}} \leftarrow p^0$
4: $d^{(0)} \leftarrow \nabla S(p^{\text{best}}) \{\text{Initialize search direction}\}$
5: **while** $|d^{(i)}| > \epsilon$ \{Gradient larger than $\epsilon$** do**
   6: $p^{\text{best}} \leftarrow p^{\text{best}} + \alpha^* d^{(i)}$
   7: $d^{(i+1)} \leftarrow \nabla S(p^{\text{new}}), i \leftarrow i + 1 \{\text{Compute new gradient direction}\}$
8: **end while**
9: **return** $p^{\text{best}}$


Michael Emmerich, LIACS Deterministic and Stochastic Algorithms for Multiobjective Optimization
Gradient-based S-metric maximization

- GSP-Problem:
  \[ f_1(x) = \frac{1}{d^\alpha} (\sum_{i=1}^{d} x_i^2)^\alpha \rightarrow \min \]
  \[ f_2(x) = \frac{1}{d^\alpha} (\sum_{i=1}^{d} (1 - x_i)^2)^\alpha \rightarrow \min \]

- for \( \alpha = 0.5 \) problem has a linear Pareto front
  \[ \{(y_1, y_2) \mid y_2 = 1 - y_1 \text{ and } y_1 \in [0, 1]\} \]

- the points maximizing the S-metric are equidistant (with two occupying the end points) - proof: see paper
Gradient-based S-metric maximization

- **GSP-Problem:**
  \[
  f_1(x) = \frac{1}{d^\alpha} \left( \sum_{i=1}^{d} x_i^2 \right)^\alpha \rightarrow \min \text{ and }
  f_2(x) = \frac{1}{d^\alpha} \left( \sum_{i=1}^{d} (1 - x_i)^2 \right)^\alpha \rightarrow \min
  \]

  - for \( \alpha = 0.5 \) problem has a linear Pareto front
    \[\{(y_1, y_2) \mid y_2 = 1 - y_1 \text{ and } y_1 \in [0, 1]\}\]
  - the points maximizing the S-metric are equidistant (with two occupying the end points) - proof: see paper
Convergence order on GSP, $\alpha = 0.5$

- the convergence (after a reasonable starting population has been found by the SMS-EMOA) is linear or almost linear.
- running time grows linearly with dimension and population size
- Reason: Gradient computation complexity: $\Theta(\mu d)$ function evaluations.
Stochastic Algorithms

- S-Metric Selection-EMOA
- Metamodel-Assisted EMOA
- Application for Molecular Alignment
Evolutionary Algorithms (EA)

- Stochastic search processes inspired by biological evolution
- More robust than gradient based techniques - can be used for experimental optimization
- Adaptation of a population (set) of individuals (solutions) to an artificial environment (objective functions, constraints) by means of variation and selection mechanisms.
- Variation mechanisms are mutation (random small modification), recombination (mixing two solutions) and others (e.g. local search).
- General iteration scheme:

\[ P_{t+1} = \text{select}_{\text{environment}}(\text{variate}(\text{select}_{\text{mating}}(P_t))) \]
SMS-EMOA

- SMS-EMOA\(^3\) \(^4\) stands for \textit{S-Metric Selection Evolutionary Multiobjective Optimization Algorithm}
- The SMS-EMOA is a simple but effective EA for multiobjective optimization
- It combines the non-dominated sorting scheme from NSGA-II with the hypervolume maximization principle
**SMS-EMOA - Basic Algorithm**

**algorithm** SMS-EMOA

1: \( P_0 \leftarrow \text{initialize()} \) \{initial set of \( \mu \) individuals\}
2: \( t \leftarrow 0 \)
3: \textbf{repeat}
4: \( x_{t+1} \leftarrow \text{generate}(P_t) \) \{Generate one offspring by variation operators\}
5: \( P_{t+1} \leftarrow \text{replace}_{\Delta S}(P_t \cup \{x_{t+1}\}) \) \{Select \( \mu \) individuals for the new population\}
6: \( t \leftarrow t + 1 \)
7: \textbf{until} stop criterium reached
S-Metric Selection

procedure replace\(\Delta S(Q)\)

1: \(\{R_1, \ldots, R_\ell\} \leftarrow\) non-dominated-sort\( (Q)\) \{all \(\ell\) partitions of \(Q\) in increasing order\}

2: for all \(x \in R_\ell\) do

3: \(\Delta S(x, R_\ell) \leftarrow S(R_\ell) - S(R_\ell \setminus \{x\})\)

4: end for

5: \(x \leftarrow \arg\min_{x \in R_\ell} [\Delta S(x, R_\ell)]\) \{detect element of \(R_\ell\) with lowest \(\Delta S(x, R_\ell)\}\)

6: \(Q' \leftarrow Q \setminus \{x\}\)

7: return \(Q'\)
S-Metric Selection for 2 objectives
SMS-EMOA Results on Superspheres

\( \ell_1 \)-Superspheres\(^5\):
\[
f_1(x) = (\sum_{i=1}^{n} |x_i|)^\gamma \cdot n^{-\gamma}, \quad f_2(x) := (\sum_{i=1}^{n} |x_i - 1|)^\gamma \cdot n^{-\gamma}, \quad x \in [0, 1]^d,
\]
\[
y_2(y_1, \gamma) = (1 - y_1^{1/\gamma})^\gamma, \quad \gamma > 0, \quad y_1 \in [0, 1]
\]
SMS-EMOA on 3-D Superspheres

\[(y_1, \ldots, y_m) \in \mathbb{R}_+^m \mid y_1^\gamma + \cdots + y_m^\gamma = 1\]
SMS-EMOA and Kriging

- Problem: Time expensive objective function evaluations
- Metamodels can be used to predict objective function based on previous evaluations
- Metamodel-Assisted EA use Kriging interpolations to predict outcome
Expected Improvement in Hypervolume

\[ I(y) = S(\{y\} \cup P^{(t)}) - S(P^{(t)}) \]

\[ \text{ExI}(x) = \int_{\mathbb{R}^m} I(y) \text{PDF}(y) dy \]
Molecular Alignment Example (with Prof. Vrakking)

Femtosecond laser pulse shaping, $\omega \in \mathbb{R}^{80}$ (pixels)

$$F_1(\omega) = \langle \cos^2(\phi) \rangle \rightarrow \text{max}$$

$$F_2(\omega) = \text{SHG} = \int_0^T |E(t)|^4 \, dt$$


Michael Emmerich, LIACS

Deterministic and Stochastic Algorithms for Multiobjective Optimization
Molecular Alignment Example

- Parallel SMS-EMOA vs. NSGA-II
- Simulations were carried out in parallel on 12 computers
Conclusions

- The efficient set contains the superset of all solutions that can be found with aggregation (utility) functions
- The Pareto-front of a solution reveals the nature of the conflict and trade-offs
- Continuation Methods, Homotopy Methods, and S-Metric gradient methods compute precise approximations for regular and differentiable problems
- They can guarantee local efficiency of solutions
- Evolutionary methods can compute Pareto fronts for a wider class of problems; Advanced EMOA yields satisfactory precision
- They are robust and can deal with noise, disconnectedness and escape local optima
- Metamodels (kriging) can reduce number of function expensive evaluations
Research directions

- Theory of Pareto landscapes
- Dealing with noise and robustness
- Many-objective optimization ($m \gg 3$), see also Wagner et al.\(^8\)
- Experimental quantum control with $m = 3$ objectives