Pareto Explorer: A Global/Local Exploration Tool for Many Objective Optimization Problems

Dr. Oliver Schütze, Cinvestav-IPN
Glasgow, Sept. 17, 2015
Outline

Introduction
- Multi- and many objective optimization
- Background

Methods for the Treatment of M(a)OPs
- Pareto Tracer for MOPs
- Pareto Explorer for MaOPs

Application
- PE for laundry system
Multi-objective Optimization

Multi-objective Optimization Problem

\[
\min F = \begin{cases} 
  f_1 : Q \subset \mathbb{R}^n \to \mathbb{R} \\
  \vdots \\
  f_k : Q \subset \mathbb{R}^n \to \mathbb{R}
\end{cases} \quad \text{(MOP)}
\]

Definition: \( x \in Q \) is Pareto optimal if there exists no point \( y \in Q \setminus \{x\} \) s.t. \( F(y) \leq F(x) \) and \( F(y) \neq F(x) \).

\( P_Q \) is a set of dimension \((k-1)\)

\( F(P_Q) = \text{image of } P_Q \) (\textit{Pareto front})
Optimality Conditions

Theorem ([Kuhn, Tucker, 1951])

Let $x$ be a Pareto point of (MOP), then:

$$1,\ldots, k \geq 0: \sum_{i=1}^{k} a_i = 1 \quad \text{and} \quad \sum_{i=1}^{k} \nabla f_i(x) = 0 = J^T, \quad (C)$$

where

$$J = \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_k(x)^T \end{pmatrix} \in \mathbb{R}^{k \times n}$$

denotes the Jacobian of $F$ at $x$.

Remark $\alpha$ is orthogonal to the linearized Pareto front at $F(x)$

Definition If $x \in Q$ satisfies (C), then it is called a Karush Kuhn Tucker (KKT) point.
The Pareto Set is a Manifold

\[ \tilde{F} : R^{n+k} \rightarrow R^{n+1} \]

\[ \tilde{F}(x, \alpha) = \left( \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \right) \]

\[ \angle \sum_{i=1}^{k} \alpha_i - 1 \]

[Kuhn,Tucker 1951]

The Pareto set is contained in \( \tilde{F}^{-1}(0) \)

[Hillermeier 2001]

Pareto set: \((k-1)\)-dim manifold
Impact on Decision Making and Algorithm Design

Finite size approximation of the entire Pareto set/front: exponential growth in $k$ ($N = m^{k-1}$)
Example: 3 points per dimension, $k=14 \rightarrow N = 3^{13} \approx 1,600,000$
Impact on Decision Making and Algorithm Design

Definition:
- $k \leq 3$ multi-objective optimization problem (MOP)
- $k > 3$ many objective optimization problem (MaOP)
Steering the Search in MOO

**Given** point $x \in R^n$, direction $v \in R^n$, step size $t \in R_+$. New candidate found via line search

$$x_{new} = x + tv$$

**Then** the according movement in objective space (for infinitesimal step sizes) is given by

$$Jv$$

**“Proof”**

$$\lim_{t \to \infty} \frac{f_i(x + tv) - f_i(x)}{t} = \langle \nabla f_i(x), v \rangle = (Jv)_i$$
Methods for the Treatment of M(a)OPs

Mathematical Programming Techniques
- Scalarization methods
- Descent methods
- Interactive methods

Set based methods
- Evolutionary strategies
- Subdivision techniques
- Cell mapping techniques

Continuation Methods
- Perform a movement along the Pareto set/front (done by MP and SON methods)
Scalarization Methods

Scalarization methods transform the given MOP into a ‘suitable’ SOP
→ one solution expected, according to the given setting

Examples:
• Weighted sum method
• ε-Constraint method
• Reference point methods
  • Normal boundary intersection (NBI)
  • Pascoletti-Serafini
  • ...

O. Schütze
Example: Reference Point Methods

If the Decision Maker has a rough idea about his/her product, one can look for the solution which is ‘closest’ to a given reference point $Z$ (defined in objective space).

$$\min_x d(F(x), Z) \quad (*)$$

But: does the solution of (*) give us (exactly) what we need? Note that the solution depends on

- The reference point $Z$
- The chosen distance $d$
- The shape of the Pareto front.

Similar problems with all scalarizing functions (e.g., ASF)

- The shape of the Pareto front.
Predictor Corrector (PC) Methods

Given: \( H: \mathbb{R}^{N+1} \rightarrow \mathbb{R}^N \), \( H(x) = 0 \)

\( x_0 \in \mathbb{R}^{N+1} \) s.t. \( H(x_0) = 0 \), \( \text{rk}(H'(x_0)) = N \)

IFT: \( \exists \ c: (-\varepsilon, \varepsilon) \rightarrow \mathbb{R}^{N+1} \) \( c(0) = x_0 + H(c(s)) = 0 \)

Differentiation: \( H'(c(s)) \cdot c'(s) = 0 \)

Orthogonalization: \( H'(x_0) = QR = (q_1 \cdots q_{N+1})R \)

Predictor: \( p = x_0 + t_0 q_{N+1} \)

Corrector: Newton method on \( H(x) = 0 \) starting with \( p \)
PC Method of Hillermeier

[Hillermeier, 2001] Use PC method on

\[ \tilde{F} : \mathbb{R}^{n+k} \rightarrow \mathbb{R}^{n+1} \]

\[ \tilde{F}(x, \alpha) = \left( \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \right) \]

\[ \left( \sum_{i=1}^{k} \alpha_i - 1 \right) \]

Possible drawbacks

• Consideration of compound \((x, \alpha)\) space may increase nonlinearity

• No treatment of inequality constraints discussed
Pareto Tracer [Martin, Sch., 2014]

Basic Idea:
Use PC method as above, but separate decision \((x)\) and weight \((\alpha)\) space whenever possible

- Directly compute kernel vectors of F-tilde and utilize ‘steering’ in objective space
- Use Newton method for MOPs by [Fliege & Svaiter, 2009]
Pareto Tracer: Predictor (1/3)

Kernel vector of $F'$

$$F'(x, \alpha) \begin{pmatrix} \nu \\ \mu \end{pmatrix} = \begin{pmatrix} H \alpha \\ 0 \end{pmatrix} \begin{pmatrix} J^T \end{pmatrix} \begin{pmatrix} \nu \\ \mu \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}$$

$$i = 0 \quad \text{(equation II)}$$

$$H = J^T \quad \text{(equation I)}$$

Steer the search in direction $d$  \quad $R^k$ : compute  \quad $R^n$ s.t. $J = d$

$I$:  \quad $d = JH^{-1}J^T \cdot d$

$II$:  \quad $i = 0$

$JH^{-1}J^T \cdot d = d \div \frac{1}{2}$

$$d = JH^{-1}J^T \cdot d$$
Pareto Tracer: Predictor (2/3)

**Task:** compute $d$ duch that a movement along the *linearized* Pareto front is performed.

$x$ KKT point $R^k$ s.t. $J^T = 0$

$$a = QR = (q_1 \ldots q_k) \begin{bmatrix} r_{11} & \vdots \\ 0 & 1 \end{bmatrix}$$

$$= r_{11}q_1 \text{ span}\{q_1\} \{q_2, \ldots, q_k\} \text{ forms an orthonormal basis (ONB) of}$$

choose directions $d_i := q_{i+1}, \ i = 1, \ldots, k$ 1

and use $d_i$ as predictors
Pareto Tracer: Predictor (3/3)

**Theorem:** Let \( x \) be a KKT point of \((MOP)\) with weight vector \( \alpha > 0 \). Further, let \( \nabla f_i(x) \neq 0, \ i=1,\ldots,k \), \( \text{rank}(J)=k-1 \) and \( H_\alpha \) be regular. Let \( d_i \) and \( \mu_d \) as above, then

\[
\begin{align*}
_i &= JH^{-1} J^T d_i, \quad i=1,\ldots,k-1
\end{align*}
\]

point along the linearized Pareto set at \( x \) and the vectors

\[
J_i, \quad i=1,\ldots,k-1
\]

form an ONB of the linearized Pareto front at \( F(x) \).
Pareto Tracer: Corrector

[Fliege & Svaiter, 2009]

Idea: given $x$ and (MOP), compute the Newton direction $\nu$ via solving

$$\max \min_{i=1,...,k} \nabla f_i(x)^T + \frac{1}{2} T \nabla^2 f_i(x)$$

Remark: necessary condition for $k=1$: $\nabla^2 f(x) + \nabla f(x) = 0$

$$\min_{t} \quad t$$

$$\text{s.t.} \quad \nabla f_i(x)^T + \frac{1}{2} T \nabla^2 f_i(x) \leq 0, \quad i = 1,..,k$$
Hessian Free Realization

Successive update of the Hessians via quasi Newton strategies, i.e., approximation of $H_\alpha$ by

$$B_\alpha = \sum_{i=1}^{k} \alpha_i B_i,$$

where $B_i \approx \nabla^2 f_i(x)$

→ Only gradient information required
Example PT1: Unconstr. MOP

\[ f_1(x) = \frac{1}{2}(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} + x_1 x_2) + e^{(x_1 x_2)^2} \]

\[ f_2(x) = \frac{1}{2}(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} - x_1 + x_2) + e^{(x_1 x_2)^2} \]

<table>
<thead>
<tr>
<th></th>
<th>Hillermeier</th>
<th>PT-N</th>
<th>PT-QN</th>
<th>PT-SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solutions</td>
<td>65.0000</td>
<td>63.0000</td>
<td>63.0000</td>
<td>63.0000</td>
</tr>
<tr>
<td>Avg. corrector iterations</td>
<td>2.3594</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Function evaluations</td>
<td>65.0000</td>
<td>63.0000</td>
<td>63.0000</td>
<td>63.0000</td>
</tr>
<tr>
<td>Jacobian evaluations</td>
<td>216.0000</td>
<td>63.0000</td>
<td>63.0000</td>
<td>63.0000</td>
</tr>
<tr>
<td>Hessian evaluations</td>
<td>432.0000</td>
<td>126.0000</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
PT2: Eq. Constr. MOP, n=100

\[ f_j(x) = \sum_{i=1, i \neq j}^{100} (x_i - a_i^j)^2 + (x_j - a_i^j)^4, \quad j = 1, 2 \]

s.t. \[ \|x - c\|^2 = r^2 \]

<table>
<thead>
<tr>
<th></th>
<th>Hillermeier</th>
<th>PT-N</th>
<th>PT-QN</th>
<th>PT-SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solutions</td>
<td>63.0000</td>
<td>64.0000</td>
<td>64.0000</td>
<td>63.0000</td>
</tr>
<tr>
<td>Avg. corrector iterations</td>
<td>3.1746</td>
<td>1.8254</td>
<td>1.9688</td>
<td>4.7778</td>
</tr>
<tr>
<td>Function evaluations</td>
<td>64.0000</td>
<td>179.0000</td>
<td>192.0000</td>
<td>365.0000</td>
</tr>
<tr>
<td>Jacobian evaluations</td>
<td>264.0000</td>
<td>179.0000</td>
<td>191.0000</td>
<td>365.0000</td>
</tr>
<tr>
<td>Hessian evaluations</td>
<td>528.0000</td>
<td>358.0000</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
PT3: Eq. + Box Constr. MOP

\[ f_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \]
\[ f_2(x) = 3x_1 + 3x_2 \frac{x_3}{3} + 0.01(x_4, x_5)^3 \]
\[ s.t. \quad x_1 + 2x_2, x_3, 0.5x_4 + x_5 = 2 \]
\[ 4x_1, 2x_2 + 0.8x_3 + 0.6x_4 + 0.5x_5^2 = 0 \]
\[ x_i \in [-2, 2], i = 1, \ldots, 5 \]

<table>
<thead>
<tr>
<th></th>
<th>PT-N</th>
<th>PT-QN</th>
<th>PT-SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solutions</td>
<td>62.0000</td>
<td>62.0000</td>
<td>62.0000</td>
</tr>
<tr>
<td>Avg. corrector iterations</td>
<td>1.0323</td>
<td>1.0323</td>
<td>2.3968</td>
</tr>
<tr>
<td>Function evaluations</td>
<td>128.0000</td>
<td>132.0000</td>
<td>219.0000</td>
</tr>
<tr>
<td>Jacobian evaluations</td>
<td>127.0000</td>
<td>127.0000</td>
<td>215.0000</td>
</tr>
<tr>
<td>Hessian evaluations</td>
<td>254.0000</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
PT4: Unconstr. MOP, $k=3$

$$f_j(x) = \sum_{i=1, i \neq j}^{100} \left( (x_i - a_{ij})^2 + (x_j - a_{ij})^4 \right), \quad j = 1, 2, 3$$

PT-QN: 4109 F, 2504 J
Pareto Explorer

**Basic Idea:**
Global/local exploration of the performance landscape via
1. Compute set of initial candidate solutions (via global method)
2. Explore solution manifold locally: starting from a given (optimal) solution, steer the search along the Pareto set/front into given user specified directions (decision space, objective space, weight space, dynamic reference points)
Neighborhood Exploration

Example: bias free neighborhood exploration in objective space for DTLZ2
PE: Steering in Objective Space

Given: $x$ KKT point, $R^k$ s.t. $J^T = 0$

$dy$ : user specified direction in obj. space

$$d_{new} = Q_2 Q_2^T dy$$ orthogonal projection onto linearized PF

$\rightarrow$ apply Pareto Tracer using $d_{new}$
PE: Steering in Decision Space (1)

Example: \( x = \begin{pmatrix} 100 \\ 1 \\ 2 \end{pmatrix} \) \( \vdots \) reduce \( x_1 \)

Given: \( x \) KKT point

\( x \) : user specified direction in decision space

\[ Q_x = \begin{pmatrix} 1 & \cdots & k & 1 \end{pmatrix} \] ONB of lin. Pareto set

\[ u_{new} = Q_x Q_x^T \] orth. proj. onto lin. Pareto set

\( \Rightarrow \) apply Pareto Tracer using \( u_{new} \)
PE: Steering in Decision Space (2)

Setting Decision Maker is satisfied with objective value \( F(x) \)

Possible solution find "best" direction \( v \) under the constraint that the movement in objective space is minimal \( (Jv = 0) \)

\[ \rightarrow \text{Predictor direction is solution of} \]

\[ \min_v -\langle v, d_x \rangle \]

s. t. \[ Jv = 0 \]

\[ ||v|| = 1 \]
Example: Laundry System

Simplified model for a laundry system

Objectives:
1. Wool grease type A
2. Wool grease type B
3. Red wine
4. Sebum type A
5. Sebum type B
6. Curry
7. Motor oil
8. Petroleum
9. Blood
10. Egg
11. Starch
12. Vegetable fat
13. Cocoa
14. Cost

Parameters:
1. Temperature
2. Amount of cleaner
3. Washing time
4. Frequency of the rotating unit
Laundry System: Steer in Obj. Space

**Start:** high quality solution (high cost)

**Steering:** reduce cost \((=f_{14}) \rightarrow d_y = -e_{14}\)
Laundry System: Steer in Dec. Space

**Interpretation:**

\[ x_3(t_0=0)=1.45, \ x_3(t_{\text{final}})=0.73 \]

\[ \rightarrow \text{Washing time can be reduced by approx. 50% without (significantly) changing the washing quality} \]
Conclusions and Future Work

**Tools**

- MOPs: Pareto Tracer
- MaOPs: Pareto Explorer

**Future Work**

- hybrids of EAs and continuation methods
- evolutionary continuation-like strategies
- applications to problems related to SMD!

**Questions?**