Computationally Expensive Multi-objective Optimization

Juliane Müller

Lawrence Berkeley National Lab

IMA Research Collaboration Workshop
University of Minnesota, February 23, 2016
Outline

- Multi-objective optimization problem
- Surrogate models
- Algorithm SOCEMO
- Numerical experiments
Outline

- Multi-objective optimization problem
- Surrogate models
- Algorithm SOCEMO
- Numerical experiments
Examples of multi-objective optimization

Design optimization: airfoil design

- maximize lift
- minimize drag
Examples of multi-objective optimization

Design optimization: airfoil design

- maximize lift
- minimize drag

Flight choice

- minimize ticket price
- minimize number/time of layovers
Multi-objective optimization problem

Minimize $f(x) = [f_1(x), f_2(x), \ldots, f_k(x)]^T$ \hspace{1cm} \text{(1a)}

$-\infty < x_j^l \leq x_j \leq x_j^u < \infty, \; j = 1, \ldots, d,$ \hspace{1cm} \text{(1b)}

where

- $f_i : \mathbb{R}^d \mapsto \mathbb{R}, \; i = 1, \ldots, k, \; k \geq 2$
- $f_i$ are generally in conflict (improving one objective will impair another objective)
- There does not exist one optimal solution
- Goal: find the best trade-off (Pareto-optimal) solutions
Pareto front (objective space), biobjective problem

The points $\mathbf{x}_A$ and $\mathbf{x}_B$ that correspond to A and B, respectively, are non-dominated.

The point $\mathbf{x}_C$ that corresponds to C is dominated.
Decision vector domination and Pareto optimality

- **Decision vector domination**
  Decision vector $\mathbf{x}_1$ dominates $\mathbf{x}_2$ ($\mathbf{x}_1 \prec \mathbf{x}_2$) iff
  \[
  f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \quad \forall i = 1, \ldots, k \quad \text{and}
  \]
  \[
  \exists m = 1, \ldots, k : \quad f_m(\mathbf{x}_1) < f_m(\mathbf{x}_2)
  \]

- **Pareto-optimal set**
  \[
  P^* = \{ \mathbf{x}^* : \nexists \mathbf{x} \text{ such that } \mathbf{x} \prec \mathbf{x}^* \}
  \]

- **Pareto-optimal front**
  \[
  PF^* = \{ \mathbf{f} = [f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \ldots, f_k(\mathbf{x}^*)]^T : \mathbf{x}^* \in P^* \}
  \]
Difficulties of solving problem (1)

- Evaluating the objective functions is computationally extremely expensive (one evaluation may take several hours)
- $f_i$ are computed by black-box simulations: we do not have an analytic description and no derivative information
- ‘Traditional’ methods for solving (1) generally require thousands of function evaluations
'Traditional’ solution methods using reformulation

For computationally cheap problems

- Linear scalarization of (1a):

\[
\min_x \sum_{i=1}^{k} w_i f_i(x), \quad w_i > 0 \tag{2}
\]

- \(\epsilon\)-constraint method

\[
\min_x f_m(x) \tag{3a}
\]

subject to \(f_i(x) \leq \epsilon_i, \quad i \neq m \tag{3b}\)

Solve to optimality and obtain one Pareto-optimal point

Not efficient for computationally expensive problems
Goal of this research

Devise an efficient algorithm that finds a good approximation of the Pareto front and that only uses few hundred function evaluations.
Outline

- Multi-objective optimization problem
- Surrogate models
- Algorithm SOCEMO
- Numerical experiments
Surrogate models (single objective)

Use surrogate models to approximate the expensive objective functions $f_1, \ldots, f_k$:

$$f_i(x) = s_i(x) + e_i(x), i = 1, \ldots, k,$$

- $f_i(x)$: the $i$th computationally-expensive objective function
- $s_i(x)$: the $i$th surrogate model output (cheap to evaluate)
- $e_i(x)$: the difference between both
Need some initial data to fit a surrogate model
\[ S = \{x_1, \ldots, x_n\} \]
\[ f_i(x_1), \ldots, f_i(x_n) \quad \forall i = 1, \ldots, k \]
Fit a surrogate model \( s_i \) to each of the data pairs
\( (x_l, f_i(x_l)) \), \( l = 1, \ldots, n \), \( i = 1, \ldots, k \)
Surrogate models

- Need some initial data to fit a surrogate model
  \[ S = \{x_1, \ldots, x_n\} \]
  \[ f_i(x_1), \ldots, f_i(x_n) \quad \forall i = 1, \ldots, k \]
- Fit a surrogate model $s_i$ to each of the data pairs $(x_l, f_i(x_l))$, $l = 1, \ldots, n$, $i = 1, \ldots, k$
Surrogate models

- Need some initial data to fit a surrogate model
  \[ S = \{ x_1, \ldots, x_n \} \]
  \[ f_i(x_1), \ldots, f_i(x_n) \text{ } \forall i = 1, \ldots, k \]
- Fit a surrogate model \( s_i \) to each of the data pairs \((x_l, f_i(x_l))\), \(l = 1, \ldots, n, i = 1, \ldots, k\)
Radial basis function (RBF) surrogate model

\[
s(x) = \sum_{l=1}^{n} \lambda_l \phi(\|x - x_l\|_2) + p(x),
\]

where

- \( \phi(r) = r^3 \) the cubic radial basis function
- \( p(x) = a + b^T x \) the polynomial tail
- \( x_l, l = 1, \ldots, n \), the already evaluated points
- determine parameters \( \lambda_l \in \mathbb{R}, l = 1, \ldots, n, a \in \mathbb{R}, \) and
  \( b = [b_1, \ldots, b_d]^T \in \mathbb{R}^d \) by solving a linear system of equations
Outline

- Multi-objective optimization problem
- Surrogate models
- Algorithm SOCEMO
- Numerical experiments
Algorithm steps

1: Initial experimental design, $\mathbf{x}_1, \ldots, \mathbf{x}_{n_0}$.

2: Computationally expensive objective function evaluations, $f_i(\mathbf{x}_1), \ldots, f_i(\mathbf{x}_{n_0}), i = 1, \ldots, k$.

3: Identify the non-dominated points.

4: Fit the RBF surrogate models to the data.

5: Select the new sample point $\mathbf{x}_{\text{new}}$:
   - Balance between local and global search

5: Evaluate $f_i(\mathbf{x}_{\text{new}}), i = 1, \ldots, k$ and go to Step 3.
Algorithm steps

1: Initial experimental design, $x_1, \ldots, x_{n_0}$.
2: Computationally expensive objective function evaluations, $f_i(x_1), \ldots, f_i(x_{n_0}), i = 1, \ldots, k$.
3: Identify the non-dominated points.
4: Fit the RBF surrogate models to the data.
5. Select the new sample point $x_{\text{new}}$:
   - Balance between local and global search
5: Evaluate $f_i(x_{\text{new}}), i = 1, \ldots, k$ and go to Step 3.
Sampling based on decision and objective space

- Objective space
  - Sample data
  - Non-dominated

- Decision space

J. Müller (julianemueller@lbl.gov)
Target value sampling – objective space

Approximate the Pareto front with a piecewise linear function

Find the point \( \mathbf{x} \) for which \( f_1(\mathbf{x}) = 2 \) and \( f_2(\mathbf{x}) = 20 \)
Target value sampling – decision space

- Target values \((t_1, t_2) = (2, 20)\)
- Find the point \(x\) for which \(f_1(x) = 2\) and \(f_2(x) = 20\)
- Use the computationally cheap surrogate models \(s_1\) and \(s_2\) and solve a computationally cheap multi-objective optimization problem

\[
\min_x \left[ |s_1(x) - t_1|, |s_2(x) - t_2| \right]^T \tag{5a}
\]

\[-\infty < x^l_j \leq x_j \leq x^u_j < \infty, j = 1, \ldots, d. \tag{5b}\]
Perturbation of non-dominated points

Decision space

Sample points
Non-dominated points
Perturbation points
Minimum point of each surrogate model

- Find the minimum point of each surrogate model (computationally cheap) $\forall i$:

$$\min_x s_i(x)$$

s.t. $x_j^l \leq x_j \leq x_j^u$, $j = 1, \ldots, d$
Stochastic sampling and scoring

- Randomly generate points in the variable domain
- Score each random point and select the best one:
  - Use the surrogate models $s_1, \ldots, s_k$ to predict the objective function values (surrogate model score)
  - Compute the distance of each random point to the set of already evaluated points (distance score)
  - Compute a weighted sum of both scores
Solve the surrogate multi-objective problem

- Use a multi-objective genetic algorithm to find the Pareto-optimal points of the approximation problem

\[
\text{Minimize } s(x) = [s_1(x), s_2(x), \ldots, s_k(x)]^T \tag{6a}
\]

\[-\infty < x_j^l \leq x_j \leq x_j^u < \infty, \quad j = 1, \ldots, d, \tag{6b}\]

- May generate a large set of new sample points
Outline

- Multi-objective optimization problem
- Surrogate models
- Algorithm SOCEMO
- Numerical experiments
Example with $d = 2$, $k = 2$

$$\min f(x) = [f_1(x), f_2(x)]^T$$  \hspace{1cm} (7a)$$

$$f_1(x_1, x_2) = 1 - \exp \left\{- \sum_{j=1}^{2} \left[ x_j - \frac{1}{\sqrt{2}} \right]^2 \right\}$$  \hspace{1cm} (7b)$$

$$f_2(x_1, x_2) = 1 - \exp \left\{- \sum_{j=1}^{2} \left[ x_j + \frac{1}{\sqrt{2}} \right]^2 \right\}$$  \hspace{1cm} (7c)$$

$$-4 \leq x_1, x_2 \leq 4$$  \hspace{1cm} (7d)$$
Example with $d = 2, k = 2$
Example with $d = 2$, $k = 2$
Example with $d = 2, k = 2$
Example with $d = 2$, $k = 2$
Example with \( d = 2, k = 2 \)
Example with $d = 2, k = 2$
Example with $d = 2, k = 2$
Example with \( d = 2, k = 2 \)
Example with $d = 2$, $k = 2$
Example with $d = 2$, $k = 2$
Example with $d = 2$, $k = 2$
Example with $d = 2$, $k = 2$
Example with $d = 2, k = 2$
Comparison to MO-Genetic Algorithm (MOGA)

- We compared the algorithm on 58 benchmark problems, one application from airfoil design, one application from structural optimization
- 1-35 variables, 2-10 objective functions
- Pareto fronts: convex connected, concave connected, disconnected, unknown
Comparison to MO-Genetic Algorithm (MOGA)

Performance measures

- (a) Number of non-dominated solutions
- (b) Set coverage metric: the proportion of solutions from MOGA that are weakly dominated by solutions from SOCEMO and vice versa
- (c) Hypervolume: the size of the space covered
Comparison to MO-Genetic Algorithm (MOGA)

Large numbers are better

(a) Number of non-dominated solutions
(b) Set coverage metric
(c) Hypervolume
Conclusions

- We developed an algorithm for computationally expensive black-box multi-objective optimization problems
- We use computationally cheap surrogate models to approximate the expensive objective functions
- We use a combination of local and global search strategies
- Numerical experiments showed that SOCEMO is an efficient and effective approach