

Focusing Search in Multiobjective Evolutionary Optimization through Preference Learning from User Feedback

Thomas Fober **Weiwei Cheng** **Eyke Hüllermeier**

AG Knowledge Engineering & Bioinformatics
Fachbereich Mathematik und Informatik



Multikriterielle Optimierung

- gleichzeitige Optimierung einer vektorwertigen Funktion

$$f: \mathcal{S} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$$

- Wertebereich: Keine totale Ordnung, jedoch Halbordnung „ $<_p$ “
 - $f(x) <_p f(x') \Leftrightarrow [f(x)]_i \geq [f(x')]_i$ für alle i und $[f(x)]_j > [f(x')]_j$ für ein j
 - einzelne Lösungen des Suchraums können unvergleichbar sein
- suche alle Lösungen x , die nicht-dominiert werden, also für die gilt:

$$\nexists x' \in \mathcal{S} : f(x) <_p f(x')$$

Multikriterielle Optimierung: Lösungsansätze

- vollständige Approximation der Paretofront (z.B. NSGA-II):
 - Paretofront kann sehr groß sein, Optimierung entsprechend teuer
 - dann ist die Auswahl der endgültigen Lösung eher schwierig für den Benutzer
 - maximal präferierte Lösung nicht unbedingt in endlicher Approximation enthalten

- Skalarisierung (z.B. gewichtete Summe):

- a-priori Spezifizierung eines Gewichtsvektors w und Optimierung von

$$f(x) = w_1 \cdot f_1(x) + \dots + w_m \cdot f_m(x)$$

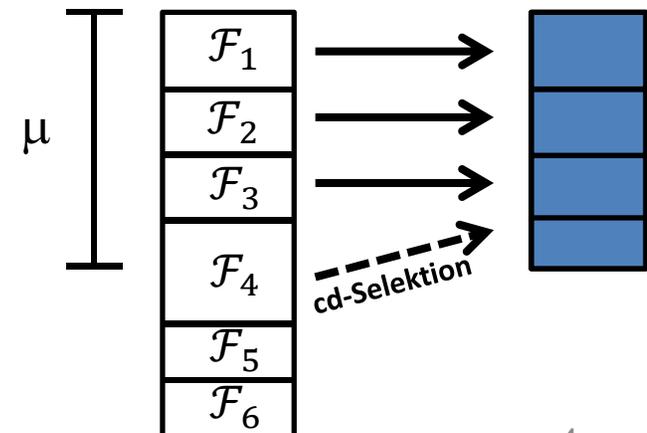
- einkriterielle Optimierung ist allerdings kein einfaches Problem
- Gewichte können falsch gewählt werden
- unterschiedliche Funktionen können verschiedene Wertebereiche oder Skalen haben

- interaktive Methoden:

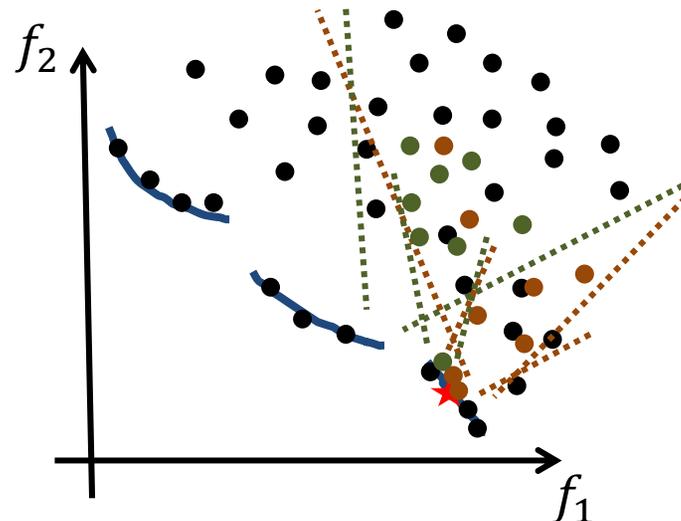
- Benutzer wird in die Optimierung einbezogen (z.B. indem er nach jeder Iteration befragt wird)
- seine Entscheidungen lenken die Suche, oder transformieren das Problem in ein einkriterielles Optimierungsproblem

Selektion im NSGA-II Algorithmus

- die Selektion verfolgt zwei Ziele:
 1. finden von Lösungen die Pareto-optimal sind
 2. möglichst gleichmäßiges und vollständiges Approximieren der Paretofront
- Sortierung der Individuen nach Dominanz
 - \mathcal{F}_1 Menge der Individuen, die nicht dominiert werden
 - \mathcal{F}_2 Menge der Individuen, die nur durch die Menge \mathcal{F}_1 dominiert werden
 - \mathcal{F}_3 Menge der Individuen, die nur durch die Menge \mathcal{F}_2 dominiert werden
 - ...
- Crowding Distance (cd)
 - zur vollständigen und gleichmäßigen Approximierung der Paretofront

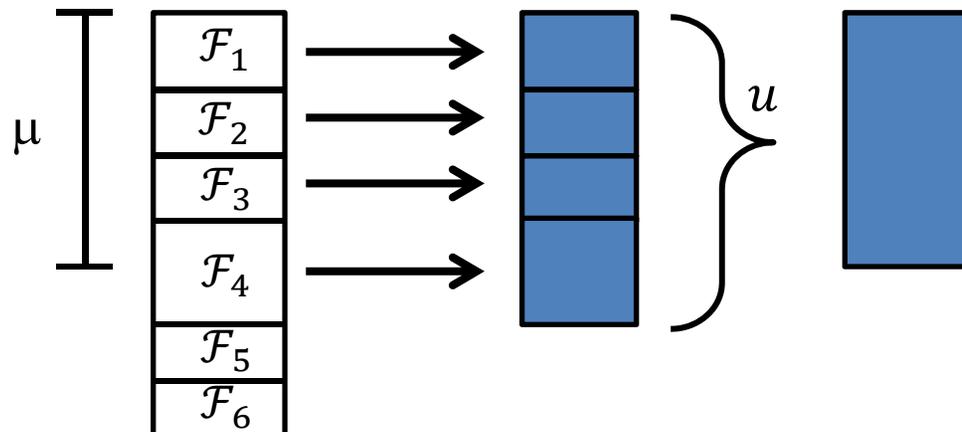


- Standard-ES zu Optimierung reellwertiger Probleme:
 - Rekombinationsselektion, Rekombination und Mutation bleiben erhalten
- Umweltselektion wird angepasst:
 - weiterhin Suche nach nicht-dominierten Lösungen → erster Schritt der NSGA-II Selektion bleibt somit erhalten
 - anstatt der Approximation der gesamten Paretofront wird die Suche auf solche Bereich fokussiert, die für den Benutzer von Interesse sind → cd Selektion wird durch u ersetzt
 - Benutzer trainiert u indem er Feedback gibt → interaktives Verfahren



Focussed Selection

- sortiere Individuen nach Dominanz
 - füge die Mengen \mathcal{F}_i aufsteigend nach i in eine temporale Menge T ein
 - breche ab, wenn die Anzahl der Individuen in T die Populationsgröße μ übersteigt
- Individuen in T sind gleichberechtigt
- wende Nutzenwerte $u(l)$ an um nächste Population aus T zu selektieren

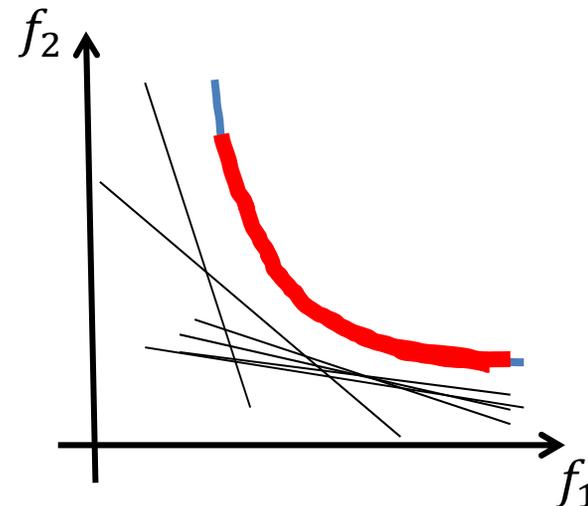
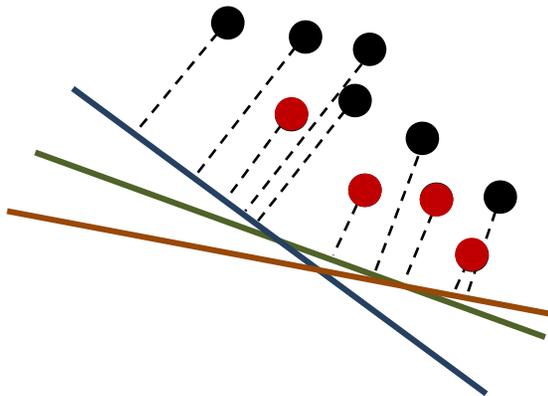


Nutzenfunktion: Verwendetes Modell

- Nutzen u wird repräsentiert durch Aggregation k linearer Funktionen u_i :

$$u(f(x)) = \max_{1 \leq i \leq k} u_i(f(x))$$

- Unsicherheit in der Schätzung korrespondiert mit Diversität der Funktionen
- Vorhandensein mehrerer Funktionen hilft bei der Aktualisierung → active learning



- $u_i(I) = u_i(f(x)) = -\langle w, f(x) \rangle = -\sum_{i=1}^m w_i \cdot f_i(x)$

- totale Ordnung auf dem Wertebereich:

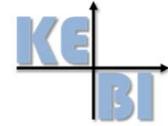
$$f(x) \succ_u f(x') \stackrel{df}{\Leftrightarrow} u_i(f(x)) \geq u_i(f(x'))$$

- monotone Funktion u ist wünschenswert

- $w = (w_1, \dots, w_m)^T \geq 0$

- Optimum von u ist auch ein Pareto-optimaler Zielvektor

Nutzenfunktion: Trainingsdaten



- vorausgesetzt wird „ \succ_u “
 - qualitative Bewertung
 - c Paare von Zielvektoren, beide Elemente eines Pairs aus der selben Menge \mathcal{F}_i
 - Bagging
- Lernproblem wird in ein binäres Klassifikationsproblem überführt
 - $u_i(f(x)) > u_i(f(x'))$ wird durch $f(x) \succ_u f(x')$ induziert
 - äquivalent zu $\langle w, f(x) - f(x') \rangle < 0$ und $\langle w, f(x') - f(x) \rangle > 0$
 - somit ist $f(x) - f(x')$ ein positives und damit $f(x') - f(x)$ ein negatives Beispiel

	f_1	f_2
I_1	3	3
I_2	2	2
I_3	1	4
I_4	4	0

$$w^* = (0.8, 0.2)$$

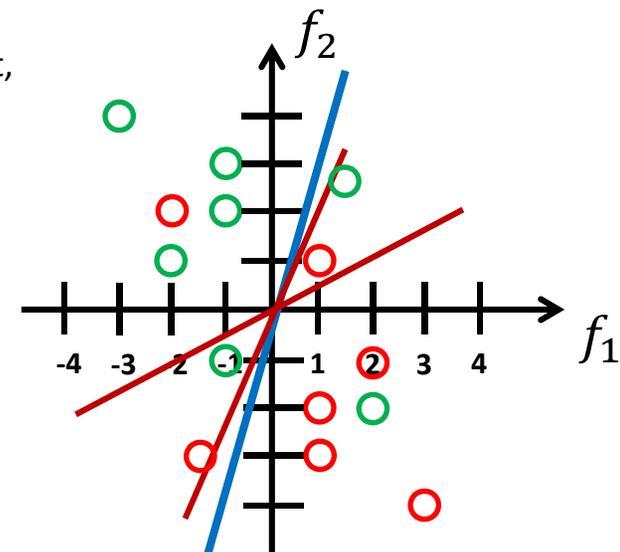
$$\rightarrow I_4 \prec_u I_1 \prec_u I_2 \prec_u I_3$$

positives Beispiel	negatives Beispiel
$I_3 - I_2 = (-1, 2)$	$I_2 - I_3 = (1, -2)$
$I_3 - I_1 = (-2, 1)$	$I_1 - I_3 = (2, -1)$
$I_3 - I_4 = (-3, 4)$	$I_4 - I_3 = (3, -4)$
$I_2 - I_1 = (-1, -1)$	$I_1 - I_2 = (1, 1)$
$I_2 - I_4 = (-2, 2)$	$I_4 - I_2 = (2, -2)$
$I_1 - I_4 = (-1, 3)$	$I_4 - I_1 = (1, -3)$

- iteratives Verfahren: Betrachte nacheinander die Beispiele

- $f(x) - f(x')$ positiven Beispiel $\rightarrow \langle w, f(x) - f(x') \rangle < 0$
- $f(x') - f(x)$ negatives Beispiel $\rightarrow \langle w, f(x') - f(x) \rangle > 0$
- ist dies verletzt, setze:
 - $w := w - \alpha(f(x) - f(x'))$ falls das Beispiel positiv ist,
 - $w := w + \alpha(f(x') - f(x))$ ansonsten
 - inkrementiere Zähler dieses Beispiels

- fehlertolerant
- inkrementelles Lernen

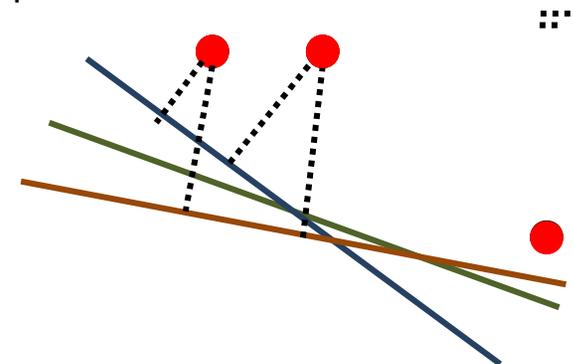


- Monotonie wird durch einfache Modifikation erreicht:

- wird ein Element in w auf einen negativen Wert gesetzt, wird dieser Wert nachfolgend auf 0 „korrigiert“ \rightarrow thresholded adaptation

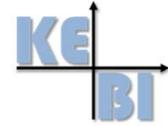
Nutzenfunktion: Active Learning

- mit W'keit p_{update} wird in einer Generation eine Aktualisierung der Funktionen durchgeführt → dazu sind neue Trainingsdaten notwendig
 - selektiere ein Paar, das möglichst viel Information enthält → „active learning“
 - suche das Paar von Funktionen (u_i, u_j) und ein Paar von Zielvektoren $(f(x), f(x'))$, die am stärksten im Widerspruch stehen
 - $W = \{w^1, \dots, w^k\}$ Gewichtsvektoren der Modelle
 - $\operatorname{argmax}_{(i,j) \in \{1,k\}^2} \|w^i - w^j\|$
 - $\operatorname{argmax}_{(f(x), f(x')) \in P^2} |u_i(f(x)) - u_i(f(x')) - (u_j(f(x)) - u_j(f(x')))|$
 - Benutzer gibt seine Präferenz \succ_u ab
 - neue Information wird verwendet um u_i und u_j anzupassen

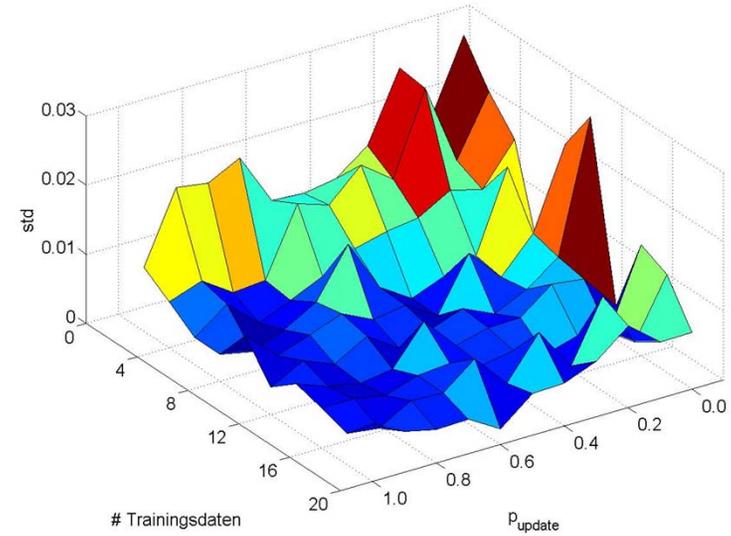
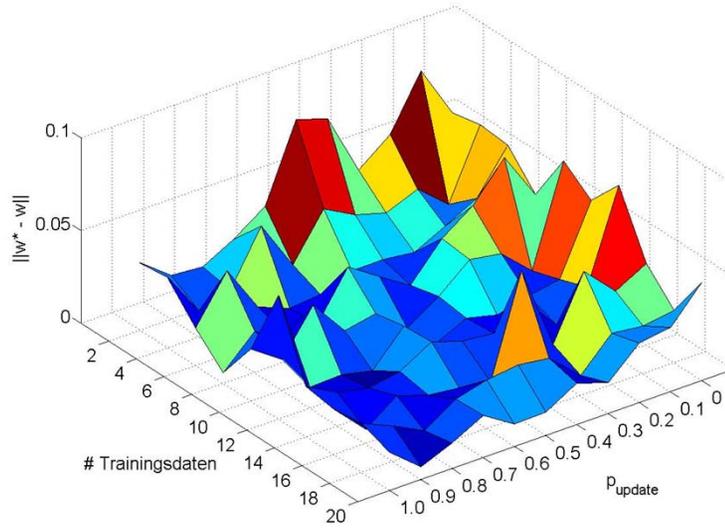


- CEC'07 Datensatz
 - beinhaltet 13 multikriterielle, reellwertige Testfunktionen
 - unterschiedliche Eigenschaften:
 - Dimensionen n und m
 - Separierbarkeit
 - Geometrie
 - Auswertungskriterien → jedoch ist das Ziel dort die Approximation der gesamten Paretofront
- Testdatensatz \subset CEC'07 Datensatz
 - OKA2 ($n = 3, m = 2$)
 - SYM-PART ($n = 10, m = 2$)
 - S_ZDT1 ($n = 10, m = 2$)
 - S_DTLZ ($n = 10, m = 8$)
 - weiterhin unterschiedliche Eigenschaften
 - wahrer Nutzen u^* wird zufällig bestimmt → $w^* \in \mathcal{U}_{[0,1]}^m$
 - ES: $\mu = 25, \nu = 2$, intermediäre Rekombination, Selbstadaptation, +Selektion
 - Vergleichsverfahren: NSGA-II

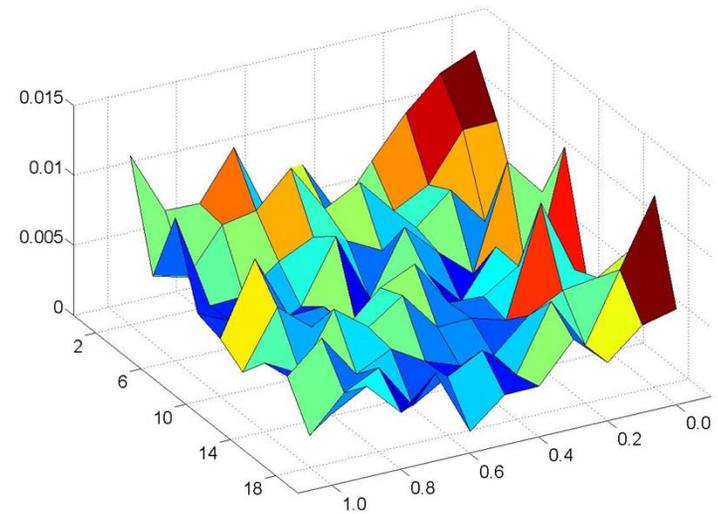
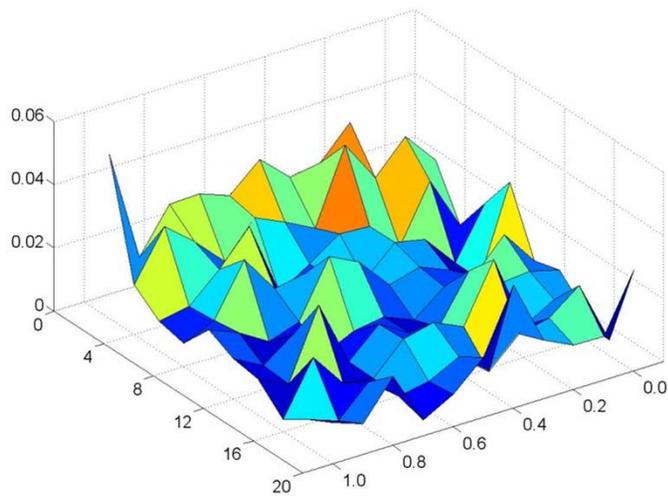
Experimente: OAK 2



k = 3

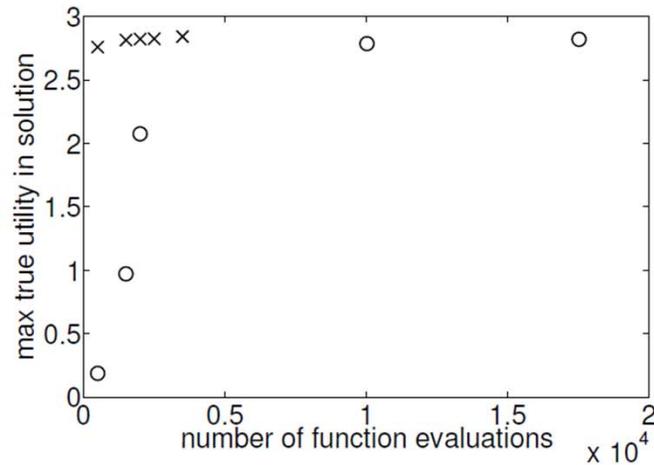
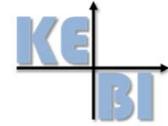


k = 9

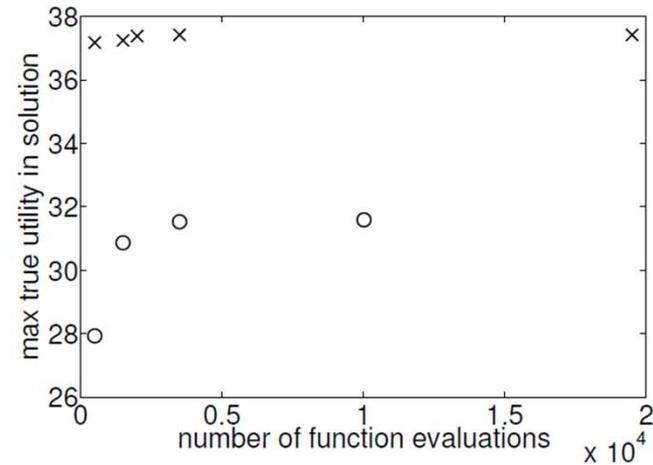


- fixiere die Anzahl der Funktionsauswertungen
- wende NSGA-II an
 - unterschiedliche Populationsgrößen: 10 – 100 in 10er Schritten
- wende FES an
 - unterschiedliche Menge an Trainingsdaten: 5 – 50 in 5er Schritten
 - unterschiedliche Update-Wahrscheinlichkeiten: 0, 0.3, 0.5
 - unterschiedliche Anzahl von Nutzen-Funktionen: 1, 3, 7, 9
- evaluiere Individuen der letzten Population $\rightarrow \langle f(x), w^* \rangle$
und wähle die beste Lösung

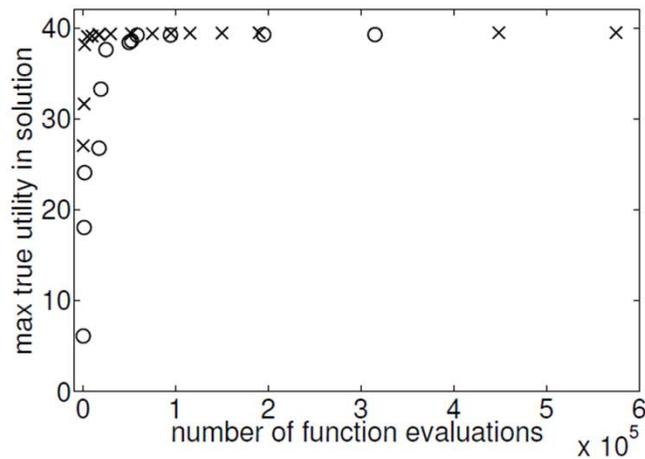
Experimente



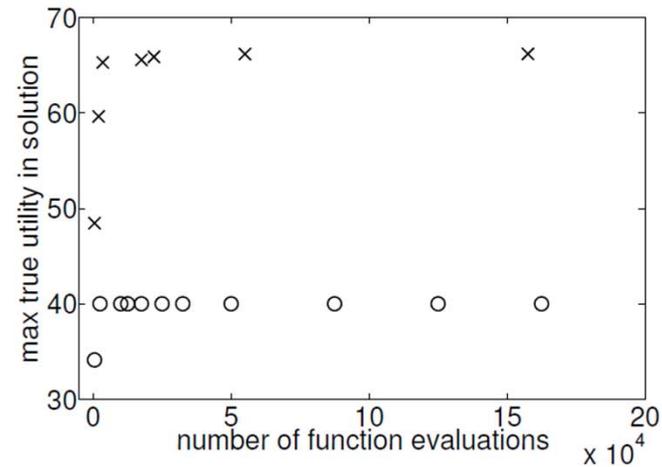
(a) OAK2 ($n = 3, m = 2$)



(b) SYM-PART ($n = 10, m = 2$)



(c) S_ZDT1 ($n = 10, m = 2$)



(d) S_DTLZ2 ($n = 10, m = 8$)

- ✓ interaktives Verfahren
 - ✓ Techniken aus dem maschinellen Lernen um wahre Präferenz zu lernen
 - qualitatives Feedback
 - ✓ Verwendung einer einkriterieller Präferenz-Funktion, aber weiterhin Selektion basierend auf nicht-Dominanz
-
- Verwendung eines Kernels anstelle der Aggregation linearer Modelle
 - Untersuchung der Auswirkungen fehlerhafter Präferenzinformation
 - ausführliche Betrachtung der Parameter
 - differenziertere Untersuchung der Güte in Abhängigkeit von Trainingsmenge und Aktualisierungswahrscheinlichkeit
 - Betrachtung weiterer (insbesondere interaktiver) Verfahren

Selektion im NSGA-II Algorithmus (Backup)

- Sortierung der Individuen nach Dominanz
 - \mathcal{F}_1 Menge der Individuen, die nicht dominiert werden
 - \mathcal{F}_2 Menge der Individuen, die nur durch die Menge \mathcal{F}_1 dominiert werden
 - \mathcal{F}_3 Menge der Individuen, die nur durch die Menge \mathcal{F}_2 dominiert werden
 - ...
- Crowding-Distanz (cd)
 - Individuen, die sortiert bezüglich einer Funktion, erste oder letzte Position haben, bekommen $cd = \infty$ zugewiesen
 - Ansonsten ergibt sich die cd aus der Summe der Kantenlängen der umschließenden Individuen

