

Evolutionäre Algorithmen

Kapitel 8: Mehrkriterienoptimierung

Sanaz Mostaghim
Intelligente Systeme

SS 2016

Outline

Mehrkriterienoptimierung

Einfachster Lösungsansatz

Pareto-optimale Lösungen

Lösung mit evolutionären Algorithmen

Beispiel: Antennenplatzierung

Mehrkriterienoptimierung

In vielen Alltagsproblemen: nicht nur eine Größe zu optimieren

Verschiedene Ziele zu möglichst hohem Grad erreichen

Beispiel: Wünsche beim Autokauf

- ▶ niedriger Preis,
- ▶ geringer Kraftstoffverbrauch,
- ▶ möglichst viel Komfort (elektr. Fensterheber, Klimaanlage)

verschiedene, zu erreichende Ziele sind oft nicht unabhängig, sondern **gegensätzlich**

Beispiel: Autokauf

- ▶ Aufpreis für viele Ausstattungsmerkmale
- ▶ Klimaanlage oder geräumigeres Auto bedingen oft größeren Motor und damit höheren Preis und Kraftstoffverbrauch

Introduction

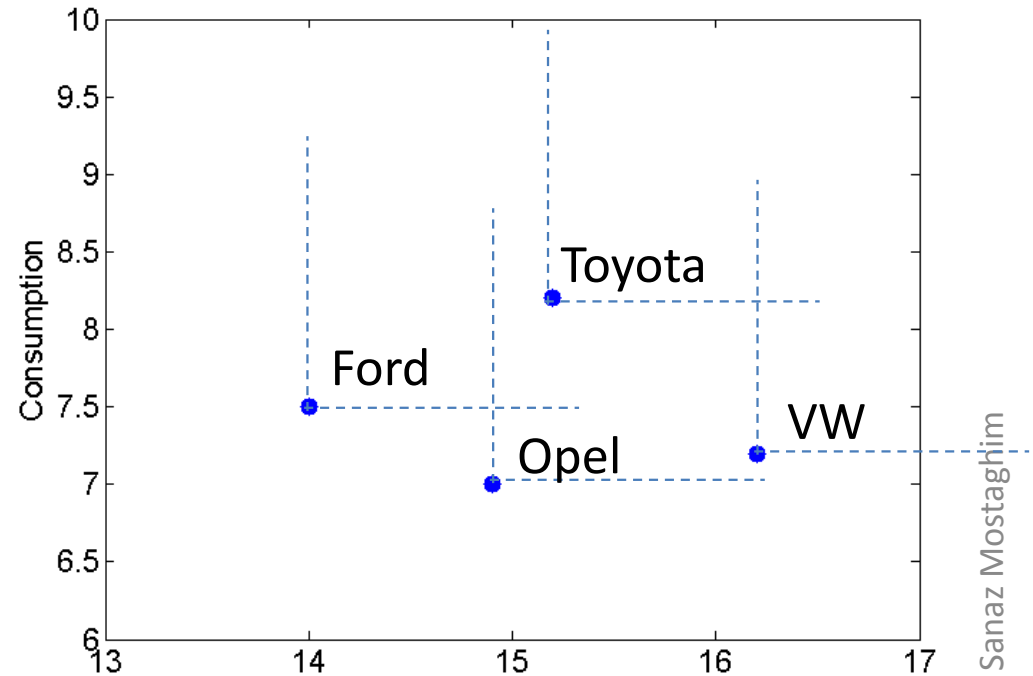
Multiple objectives instead of just one.

Example: Buying a new car (objectives/ alternatives)

Alternatives

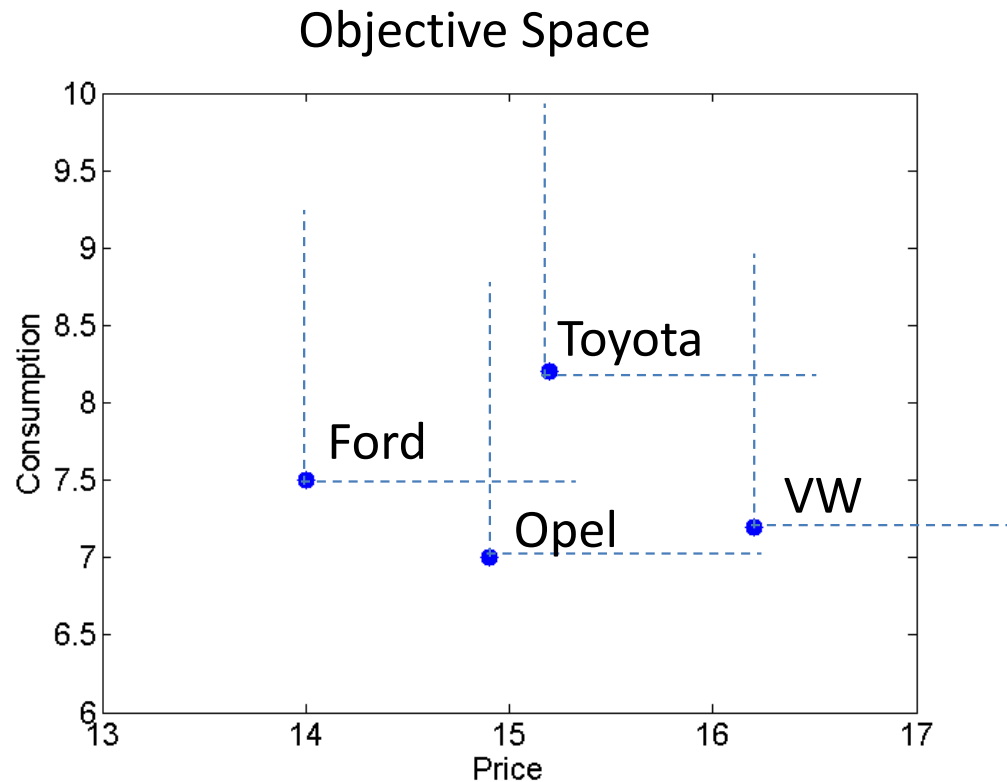
	VW	Opel	Ford	Toyota
Price (1,000 €)	16.2	14.9	14.0	15.2
Consumption (litres/100 km)	7.2	7.0	7.5	8.2
Power (kW)	66.0	62.0	55.0	71.0

Objective Space



Difference to having one objective

- If *price* is our only objective
 - Ford, Opel, Toyota, VW (best to worst)
- If *price* and *consumption* are our only objectives
 - Ford is better than Toyota
 - Opel is better than VW, Opel is better than Toyota



MOP is *different*

- We *cannot* compare all pair of cars- considering all the objectives
 - Partial ordering
- We *can* compare all the cars- considering just one objective
 - Complete ordering

		Alternatives			
		VW	Opel	Ford	Toyota
Objectives	Price (1,000 €)	16.2	14.9	14.0	15.2
	Consumption (litres/100 km)	7.2	7.0	7.5	8.2
	Power (kW)	66.0	62.0	55.0	71.0

MOP is *difficult*

- All the cars are **optimal**, if we consider all the three objectives
- Just *one* is **optimal** if we consider any one of the objectives
- Larger *optimal* set

		Alternatives			
		VW	Opel	Ford	Toyota
Objectives	Price (1,000 €)	16.2	14.9	14.0	15.2
	Consumption (litres/100 km)	7.2	7.0	7.5	8.2
	Power (kW)	66.0	62.0	55.0	71.0

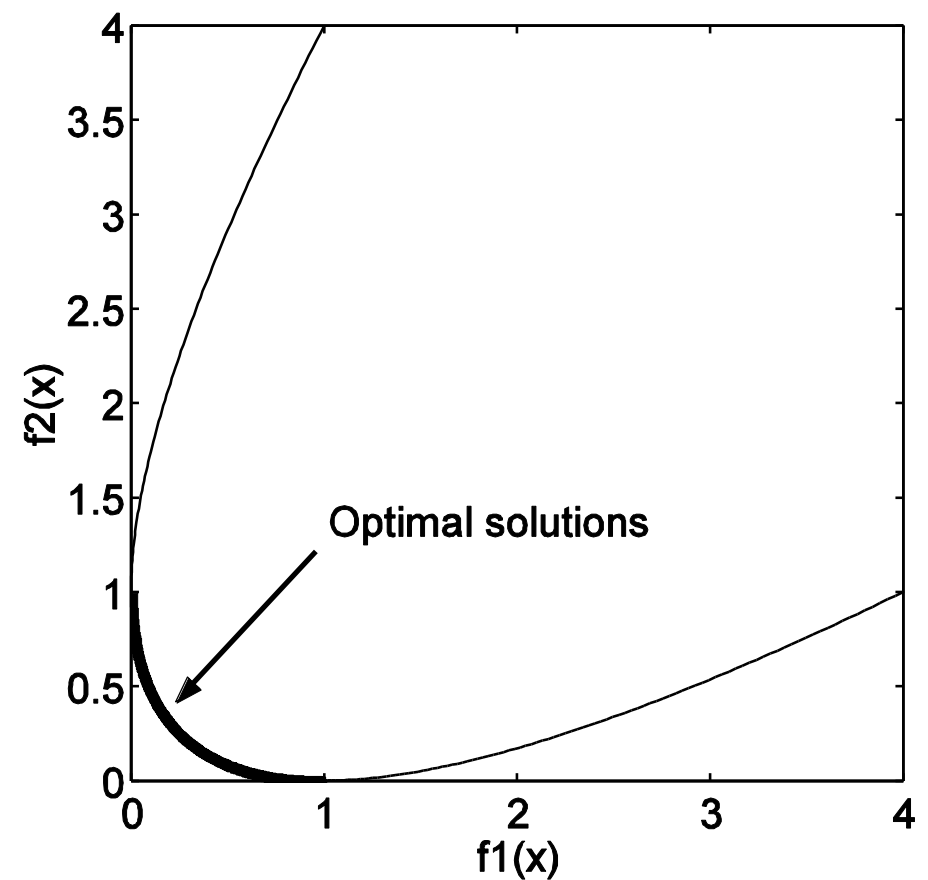
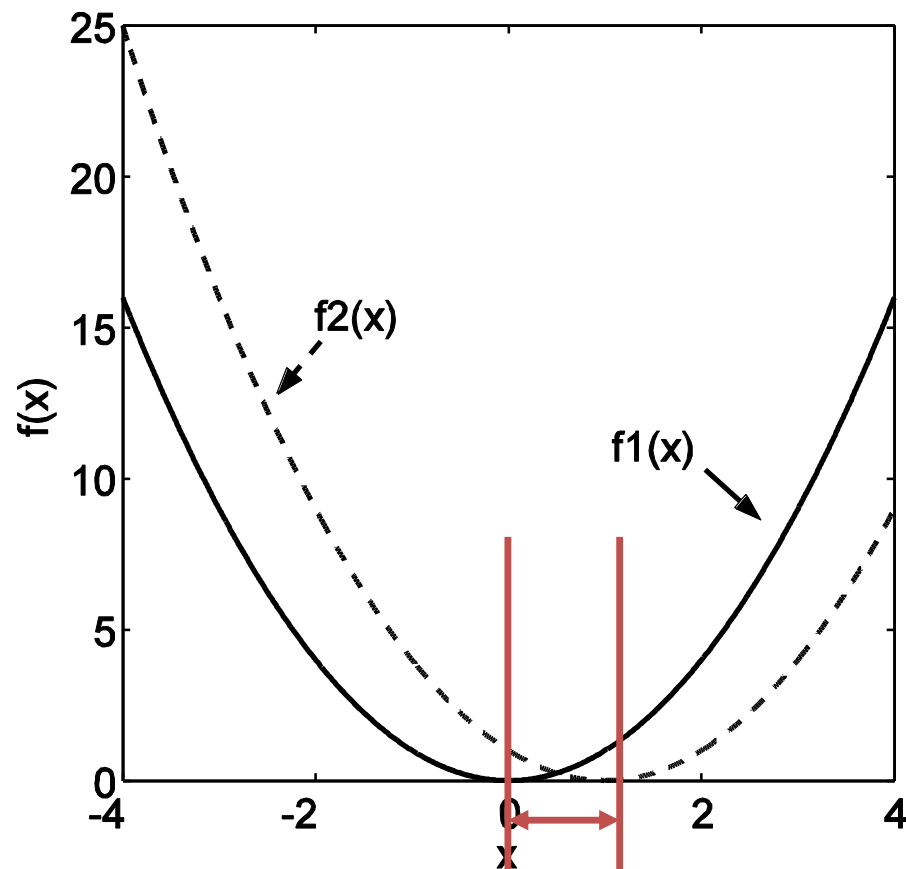
Mehrkriterienoptimierung

Formale Beschreibung: k Kriterien gegeben, denen jeweils eine zu optimierende Zielfunktion zugeordnet ist:

$$f_i : \Omega \rightarrow \mathbb{R}, \quad i = 1, \dots, k$$

$$\min \vec{f}(x) = (f_1(x), f_2(x)) = (x^2, (x-1)^2)$$

$$s.t. x \in [-4, 4]$$



Mehrkriterienoptimierung

Einfachster Lösungsansatz: fasse k Zielfunktionen zu einer Gesamtzielfunktion zusammen, z.B. durch

$$f(s) = \sum_{i=1}^k w_i \cdot f_i(s)$$

Wahl der Gewichte:

- ▶ *Vorzeichen:* falls $f \rightarrow \max$, dann alle $w_i > 0$ von zu maximierenden f_i , anderen $w_i < 0$
- ▶ *Absolutwert:* relative Wichtigkeit der Kriterien (Schwankungsbreite berücksichtigen!)

Mehrkriterienoptimierung

Probleme dieses Ansatzes:

- ▶ relative Wichtigkeit verschiedenen Kriterien bereits muss vor Suche festlegt werden
- ▶ Wahl der Gewichte nicht immer einfach, sodass Präferenzen zwischen Kriterien angemessen

Probleme, die mit Linearkombination der f_i auftreten, sind noch viel fundamentaler:

- ▶ allgemein: Problem der **Aggregation von Präferenzordnungen**
- ▶ tritt auch bei Personenwahlen auf (Kandidatenpräferenzen der Wähler müssen zusammengefasst werden)
- ▶ **Arrowsches Paradoxon** [Arrow, 1951]: es gibt keine Wahlfunktion, die alle wünschenswerten Eigenschaften hat

Mehrkriterienoptimierung

- ▶ Arrowschen Unmöglichkeitssätze [Arrow, 1951] lassen sich im Prinzip durch **skalierte Präferenzordnungen** umgehen
- ▶ **Aber:** Skalierung der Präferenzordnung ist weiterer Freiheitsgrad
- ▶ es ist u.U. noch schwieriger, eine passende Skalierung zu finden, als Gewichte einer Linearkombination angemessen zu bestimmen

Pareto-optimale Lösungen

- ▶ **alternativer Ansatz:** versuche, alle/möglichst viele **Pareto-optimale** Lösungen zu finden

Definition

Ein Element $s \in \Omega$ heißt **Pareto-optimal** bezüglich der Zielfunktionen f_i , $i = 1, \dots, k$, wenn es kein Element $s' \in \Omega$ gibt, für das gilt (Minimierung)

$$\begin{aligned} \forall i, 1 \leq i \leq k : \quad & f_i(s') \leq f_i(s) \quad \text{und} \\ \exists i, 1 \leq i \leq k : \quad & f_i(s') < f_i(s). \end{aligned}$$

- ▶ **Anschaulich:** Wert keiner Zielfunktion kann verbessert werden, ohne Wert einer anderen zu verschlechtern

Definition des Begriffs „Pareto-optimal“

Element $s_1 \in \Omega$ **dominiert** Element $s_2 \in \Omega$, wenn gilt

$$\forall i, 1 \leq i \leq k : f_i(s_1) \leq f_i(s_2)$$

Element $s_1 \in \Omega$ **dominiert** Element $s_2 \in \Omega$ **echt**, wenn s_1 s_2 dominiert und außerdem gilt

$$\exists i, 1 \leq i \leq k : f_i(s_1) < f_i(s_2)$$

Element $s_1 \in S$ heißt **Pareto-optimal**, wenn es von keinem Element $s_2 \in \Omega$ echt dominiert wird

Menge der Pareto-optimalen Elemente heißt **Pareto-Front**

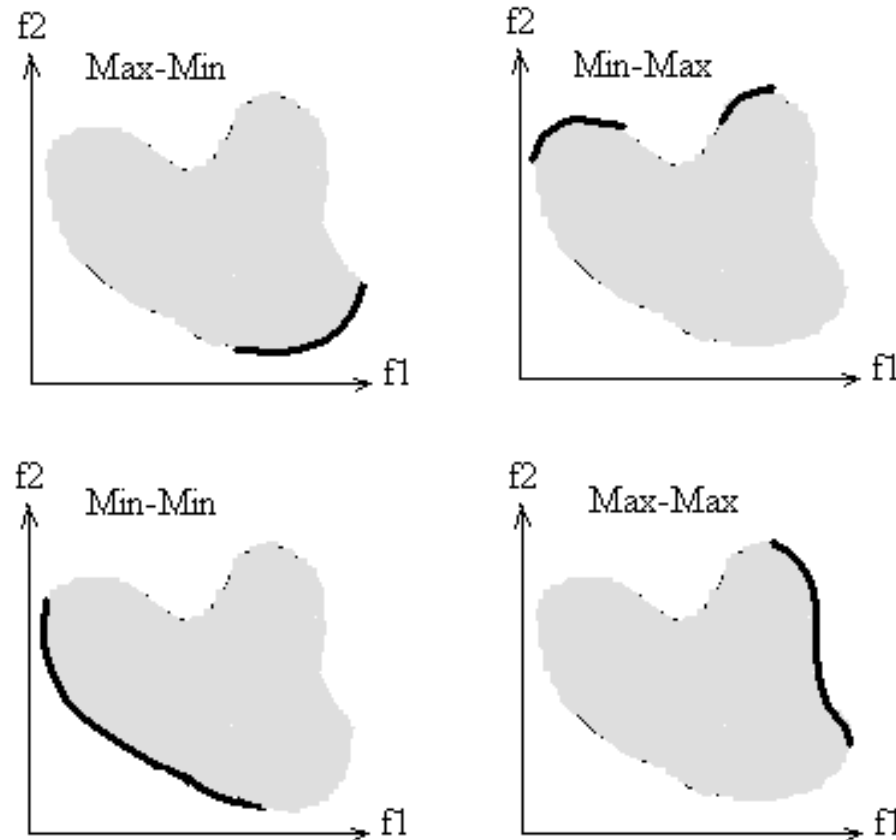
Mehrkriterienoptimierung

Vorteile der Suche nach Pareto-optimalen Lösungen:

- ▶ Zielfunktionen müssen nicht zusammengefasst werden
d.h. Bestimmung von Gewichten entfällt

- ▶ Suche muss auch für verschiedene Präferenzen nur einmal durchgeführt werden
erst anschließend wird aus gefundenen Lösungen gewählt

Pareto-optimaler Lösungen / Pareto-Front



- ▶ alle Punkte von Ω liegen im grau gezeichneten Bereich
- ▶ Pareto-optimale Lösungen = fett gezeichneten Teil des Randes
- ▶ beachte: je nach Lage der Lösungskandidaten kann Pareto-optimale Lösung auch eindeutig sein

Lösung mit evolutionären Algorithmen

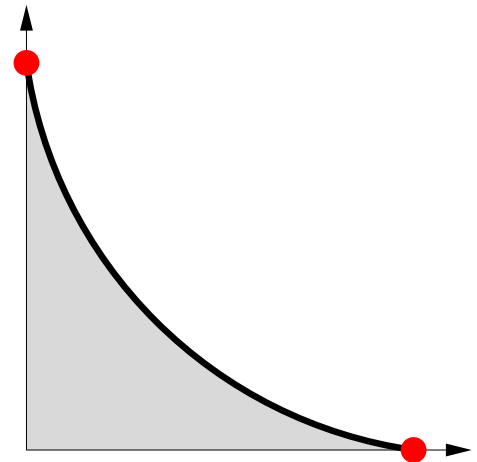
- ▶ Ziel: möglichst breite Verteilung der Population entlang Pareto-Front
- ▶ Herausforderung: ohne vorab bestimmte Gewichtung viele verschiedene, gleichwertige Lösungen

- ▶ **Einfachster Ansatz:** verwende gewichtete Summe der einzelnen Zielfunktionen als Fitnessfunktion

Lösung mit evolutionären Algorithmen

naheliegende Alternative: sog. **VEGA-Verfahren**

- ▶ gegeben k Kriterien, denen Zielfunktionen $f_i, 1 \dots, k$ zugeordnet sind
- ▶ $\forall i, 1, \dots, k$: wähle $\frac{|P|}{k}$ Individuen basierend auf Fitnessfunktion f_i
- ▶ **Vorteil:** einfach, geringer Rechenaufwand
- ▶ **Nachteil:** Lösungen, die alle Kriterien recht gut, aber keines maximal erfüllen, haben deutlichen Selektionsnachteil
- ▶ **Folge:** Suche konzentriert sich auf Randlösungen



Lösung mit evolutionären Algorithmen

Besserer Ansatz: nutze Dominanzbegriff zur Selektion

Aufbau einer **Rangskala** der Individuen einer Population:

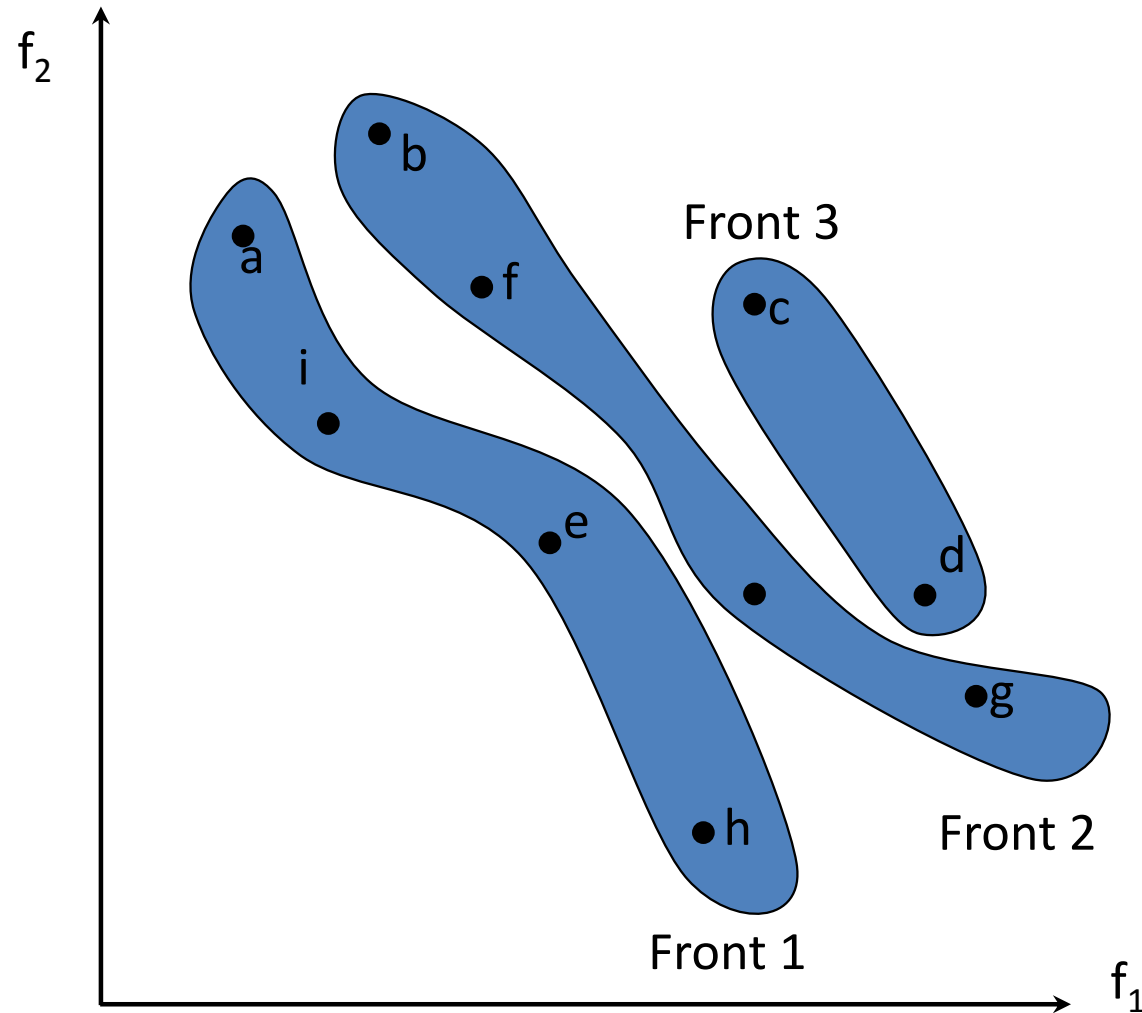
- ▶ finde alle nicht dominierten Lösungskandidaten der Population
- ▶ ordne Lösungskandidaten höchsten Rang zu und entferne sie aus Population
- ▶ wiederhole Bestimmen und Entfernen der nicht dominierten Lösungskandidaten für weiteren Ränge, bis Population leer

Führe mithilfe der Rangskala **Rangauswahl** durch

Problem: alle Individuen der Pareto-Front werden gleich bewertet

⇒ Gendrift: Pareto-Front konvergiert an beliebigem Punkt durch Zufallseffekte

Non-dominated sorting



Verhindern des Gendriffs

Ziel: möglichst gleichmäßige Verteilung entlang Pareto-Front

Lösung: **Nischentechniken** um zwischen Individuen mit gleichem Rang zu unterscheiden

- ▶ z.B. *power law sharing*: Individuen mit häufiger Kombination von Funktionswerten erhalten geringere Fitness
isoliert auftretende Kombinationen gleich wahrscheinlich wie Lösungskandidaten der gehäuft vorkommenden Kombination
- ▶ Sharing wie für eine Bewertungsfunktion, nur mit Abstandsmaß für Funktionswerte

Problem: aufwändige Berechnung der Rangskala

NSGA-Selektion

Non-dominated Sorted Genetic Algorithm

Alternative: **Turnierauswahl**, wobei Turniersieger über Dominanzbegriff und ggf. Nischentechniken bestimmt

Vorgehensweise:

- ▶ wähle Referenzindividuen
- ▶ selektiere nichtdominiertes Individuum
- ▶ ansonsten: Individuum mit weniger Individuen in Nische

hier: Nische durch Radius ε bestimmt

Strength Pareto EA (SPEA2)

gewöhnlicher EA

Bewertungsfunktion: zwei Komponenten

1. wie viele Individuen dominieren Individuen, die dieses Individuum dominieren
2. Distanz zum \sqrt{n} -nächsten Individuum

Archiv geht in Güteberechnung mit ein und enthält nicht-dominierte Individuen

- ▶ falls zu wenig: zusätzlich gütebeste Individuen
- ▶ Ersetzen im Archiv aufgrund der Entfernung zu anderen archivierten Individuen

Algorithm 2 SPEA2

Input: Zielfunktionen F_1, \dots, F_k , Populationsgröße μ , Archivgröße $\tilde{\mu}$

```
1:  $t \leftarrow 0$ 
2:  $P(t) \leftarrow$  erzeuge Population mit  $\mu$  Individuen
3:  $R(t) \leftarrow \emptyset$ 
4: while Terminierungsbedingung nicht erfüllt {
5:   bewerte  $P(t)$  durch  $F_1, \dots, F_k$ 
6:   for each  $A \in P(t) \cup R(t)$  {
7:      $\text{AnzDom}(A) \leftarrow |\{B \in P(t) \cup R(t) \mid A >_{\text{dom}} B\}|$ 
8:   }
9:   for each  $A \in P(t) \cup R(t)$  {
10:     $d \leftarrow$  Distanz von  $A$  und seinen  $\sqrt{\mu + \tilde{\mu}}$  nächsten Individuen in  $P(t) \cup R(t)$ 
11:     $A.F \leftarrow \frac{1}{d+2} + \sum_{B \in P(t) \cup R(t), B >_{\text{dom}} A} \text{AnzDom}(B)$ 
12:   }
13:    $R(t+1) \leftarrow \{A \in P(t) \cup R(t) \mid A \text{ ist nicht-dominiert}\}$ 
14:   while  $|R(t+1)| > \tilde{\mu}$  {
15:     entferne dasjenige Individuum aus  $R(t+1)$  mit dem kürzesten/zweitkürzesten Abstand
16:   }
17:   if  $|R(t+1)| < \tilde{\mu}$  {
18:     fülle  $R(t+1)$  mit den gütebesten dominierten Individuen aus  $P(t) \cup R(t)$ 
19:   }
20:   if Terminierungsbedingung nicht erfüllt {
21:     Selektion aus  $P(t)$  mittels TURNIER-SELEKTION
22:      $P(t+1) \leftarrow$  wende Rekombination und Mutation an
23:      $t \leftarrow t + 1$ 
24:   }
25: }
26: return nicht-dominierte Individuen aus  $R(t+1)$ 
```

Zusammenfassung

selbst modernste Verfahren haben bei mehr als 3 Kriterien Probleme, Pareto-Front anzunähern

Grund: Rechenzeit zur Detektion ist riesig

Abhilfe: iterative Präsentation der bisherigen Lösungen

Nutzer fällt Entscheidungen über Konzentration der Suche auf Teilbereich

Outline

Mehrkriterienoptimierung

Beispiel: Antennenplatzierung

Einführung

Formalisierung

Entwurfsmuster

Selektion

Algorithmus

Konkretes Problem

Aufgabenstellung: Antennenplatzierung

- ▶ Basisantennen für Mobilfunknetze
- ▶ erstes Ziel: hohe Netzverfügbarkeit
- ▶ zweites Ziel: geringe Kosten
- ▶ übliche Vorgehensweise:
 - ▶ Basisantennen platzieren und Größe/Reichweite konfigurieren
⇒ Bedarf abdecken
 - ▶ Frequenzen zuweisen ⇒ Interferenzen minimal halten

Ausgangssituation

Beide Probleme sind \mathcal{NP} -hart

Platzierung kann Frequenzzuweisung stark einschränken

In einer Iteration können Ergebnisse der Frequenzzuweisung nur bedingt in Platzierung wieder einfließen

Grundsatzentscheidung:

Beide Probleme werden gleichzeitig bearbeitet

Formalisierung

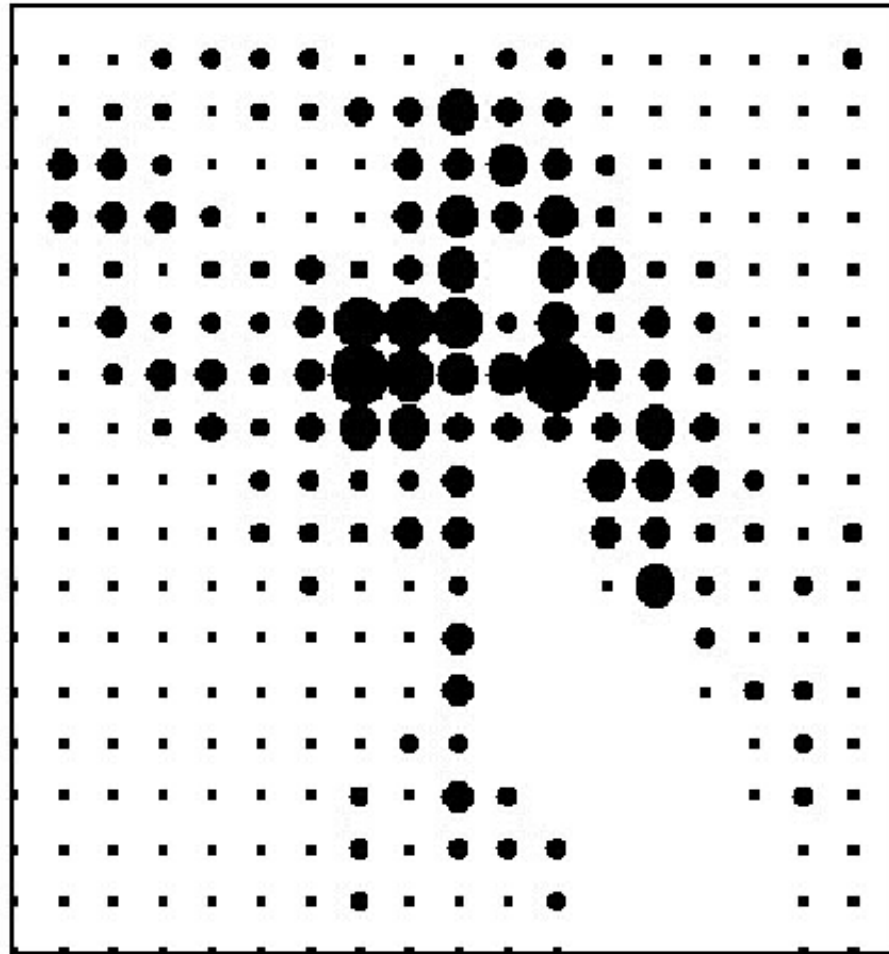
Rechteckiges Gebiet (x_{\min}, y_{\min}) und (x_{\max}, y_{\max}) mit
Rasterung res

Menge aller (mögliche) Positionen:

$$Pos = \left\{ (x_{\min} + i \cdot res, y_{\min} + j \cdot res) \mid 0 \leq i \leq \frac{x_{\max} - x_{\min}}{res} \right. \\ \left. \text{und } 0 \leq j \leq \frac{y_{\max} - y_{\min}}{res} \right\}$$

Gesprächsbedarf Zürich

statistisch ermitteltes Gesprächsaufkommen $bedarf(zelle)$
 $\in \mathbb{N}$ für einige $zelle \in Pos$



Formalisierung: Antenne

Antenne $t = (pow, cap, pos, frq)$

Sende-/Empfangsstärke $pow \in [MinPow, MaxPow] \subset \mathbb{IN}$

Gesprächskapazität $cap \subset [0, MaxCap] \subset \mathbb{IN}$

Frequenzen/Kanäle $frq \subset Frequ = \{f_1, \dots, f_k\}$ mit $|frq| \leq cap$

alle möglichen Antennenkonfigurationen:

$$T = [MinPow, MaxPow] \times [0, MaxCap] \times Pos \times Frequ$$

Genotyp

Problemnaher Genotyp

$$\Omega = \mathcal{G} = \{ \{t_1, \dots, t_k\} \mid k \in \mathbb{N} \text{ und } \forall i \in \{1, \dots, k\} : t_i \in T \}$$

Variable Länge

Randbedingungen

Netzverfügbarkeit hat oberste Priorität \Rightarrow
als harte Randbedingung formuliert

Erreichbare Positionen gemäß Wellenverbreitungsmodell:

$$wp : Pos \times [MinPow, MaxPow] \rightarrow \mathcal{P}(Pos)$$

$A.G = (t_1, \dots, t_k)$ heißt legal, wenn für jedes t_i eine Zuordnung $bedient(t_i, zelle) \in \mathbb{N}$ (mit $zelle \in Pos$) existiert, sodass

- ▶ $bedient(t_i, zelle) > 0 \Rightarrow zelle \in wp(t_i)$
- ▶ $\sum_{i=1}^k bedient(t_i, zelle) \geq bedarf(zelle)$
- ▶ $\sum_{zelle \in Pos} bedient(t_i, pos) \leq cap$ mit $t_i = (pow, cap, pos, freq)$

Bewertungsfunktionen

Störungen durch Antennen mit gleichen oder eng beieinander liegenden Frequenzen in einer Zelle

$$f_{interferenz}(A) = \frac{\sum_{i=1}^k \#gestörteGespräche(t_i)}{\sum_{zelle \in Pos} bedarf(zelle)}$$

Kosten $kosten(pow_i, cap_i)$ pro Antenne

$$f_{kosten}(A) = \sum_{i=1}^k kosten(t_i)$$

„Entwurfsmuster“

Nur legale Individuen, daher Reparaturfunktion notwendig

Jede Antennenkonfiguration muss noch erreichbar sein

Verlängernde und verkürzende Operatoren halten sich die Waage

Feinabstimmung und Erforschung sind ausgeglichen:
problemspezifische und zufällige Operatoren

Reparaturfunktion

Zellen in einer zufälligen Reihenfolge besuchen

Falls deren Bedarf nicht gedeckt ist:

1. bei Existenz mindestens einer Antennen mit freier Kapazität:
die stärkste Antenne wählen und Frequenzen zuweisen
2. ggf. diejenige Antenne ermitteln, die kostenminimal durch
Erhöhung der Stärke den Bedarf decken kann
3. ggf. prüfen, welche Kosten durch eine neue Antenne
unmittelbar bei der Zelle entstehen
4. ggf. Lösung (2) oder (3) umsetzen

Reparaturfunktion

Einsatz:

Auf jedes neu erzeugte Individuum

Zur Initialisierung der Anfangspopulation

- ▶ Reparaturfunktion auf leeres Individuum
- ▶ max. $2^{|Pos|}$ Individuen durch mögliche zufällige Reihenfolge der Bedarfzellen

Mutationsoperatoren

- ▶ 6 „gerichtete“ Mutationen, die spezieller Idee folgen
- ▶ 5 „zufällige“ Mutationen

Gerichtete Mutationsoperatoren

Name	Wirkung
DM1	falls Antenne unbenutzte Frequenzen hat ⇒ Kapazität entsprechend reduzieren
DM2	falls Antenne maximale Kapazität nutzt ⇒ weitere Antenne mit Standardeinstellungen in der Nähe platzieren
DM3	falls Antennen große überlappende Regionen haben ⇒ eine Antenne entfernen
DM4	falls Antennen große überlappende Regionen haben ⇒ Stärke einer Antenne reduzieren, so dass dennoch alle Anrufe bedient
DM5	falls Interferenzen vorkommen ⇒ involvierende Frequenzen verändern
DM6	falls Antenne nur kleine Anzahl von Anrufen hat ⇒ Antenne entfernen

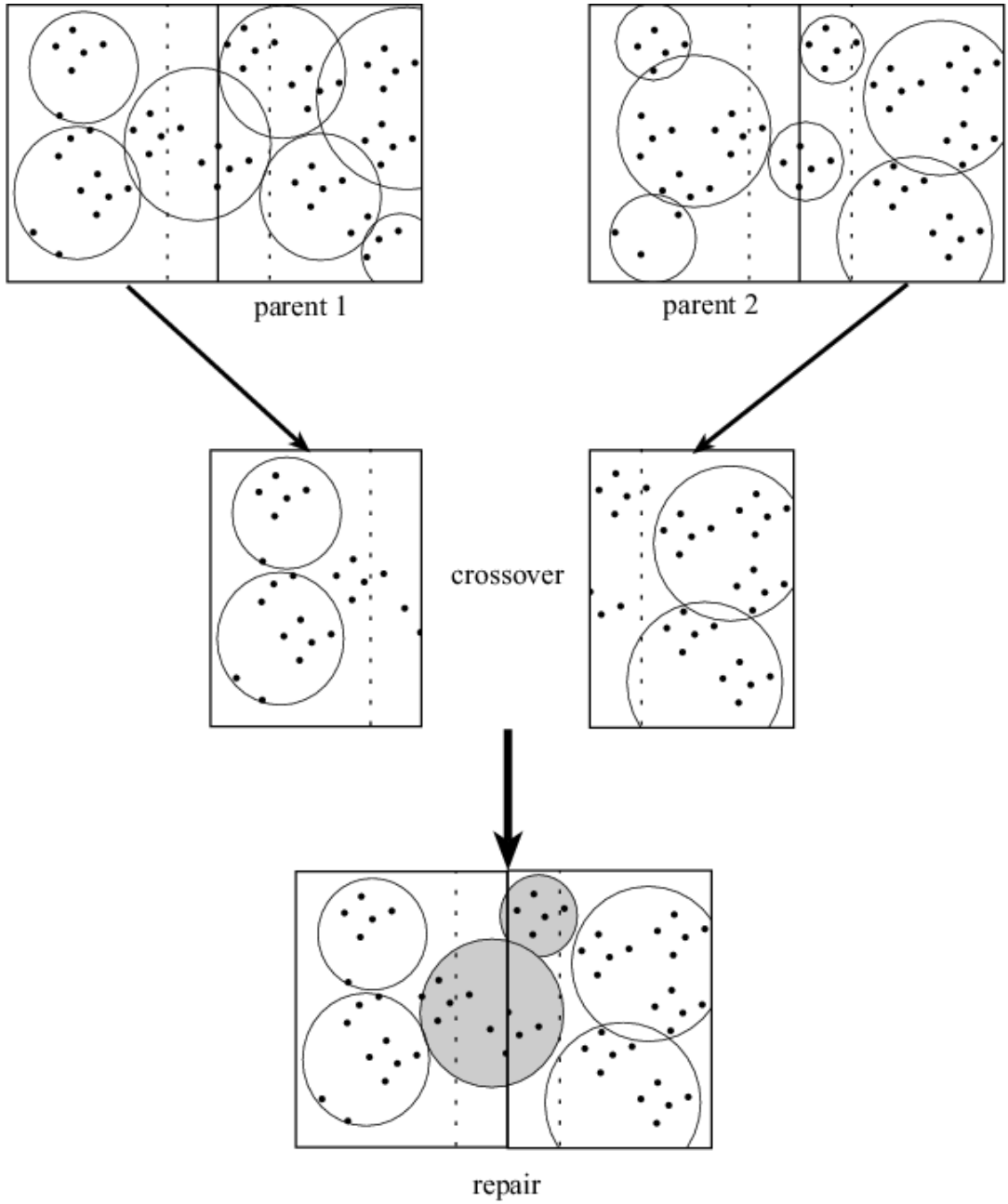
Zufällige Mutationsoperatoren

Name	Wirkung
RM1	Position einer Antenne ändern (Stärke und Kapazität unverändert, Frequenzen neu durch Reparaturfunktion)
RM2	komplett zufälliges Individuum (wie in Initialisierung)
RM3	Stärke einer Antenne zufällig ändern ⇒ Ausgleich zu <i>DM4</i>
RM4	Kapazität einer Antenne zufällig verändern ⇒ Ausgleich zu <i>DM1</i>
RM5	zugeordnete Frequenzen einer Antenne ändern ⇒ Ausgleich zu <i>DM5</i>

Rekombination

- ▶ Gesamtheit in zwei Hälften teilen (vertikal oder horizontal)
- ▶ pro Hälfte Antennen eines Individuums übernehmen
- ▶ Korridor um Grenze durch Reparaturalgorithmus füllen

Rekombination: Beispiel



Selektion

Moderne Mehrzielselektion notwendig

Problem bestehender Algorithmen (z.B. SPEA):

- ▶ Individuum wird mit $\mathcal{O}(\mu^2)$ in Archiv der Größe μ integriert
- ▶ schlecht für „steady state“-Ansatz (siehe Grundsatzentscheidung!)

Elternselektion als Turnierselektion basierend auf

- ▶ $Dominiert(A)$ = Menge der von A dominierten Individuen in Population
- ▶ $WirdDominiert(A)$ = Menge der Individuen, die A dominieren

Rang zuweisen

$$Rang(A) = \#WirdDominiert(A) \cdot \mu + \#Dominiert(A)$$

Einziges Problem: Gendrift, wenn alle Individuen gleichwertig

Selektion

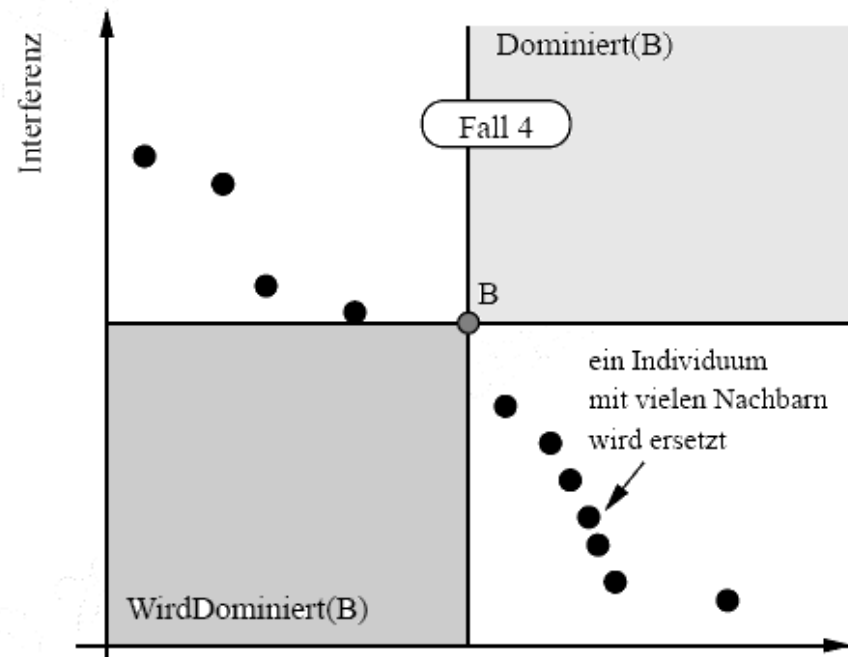
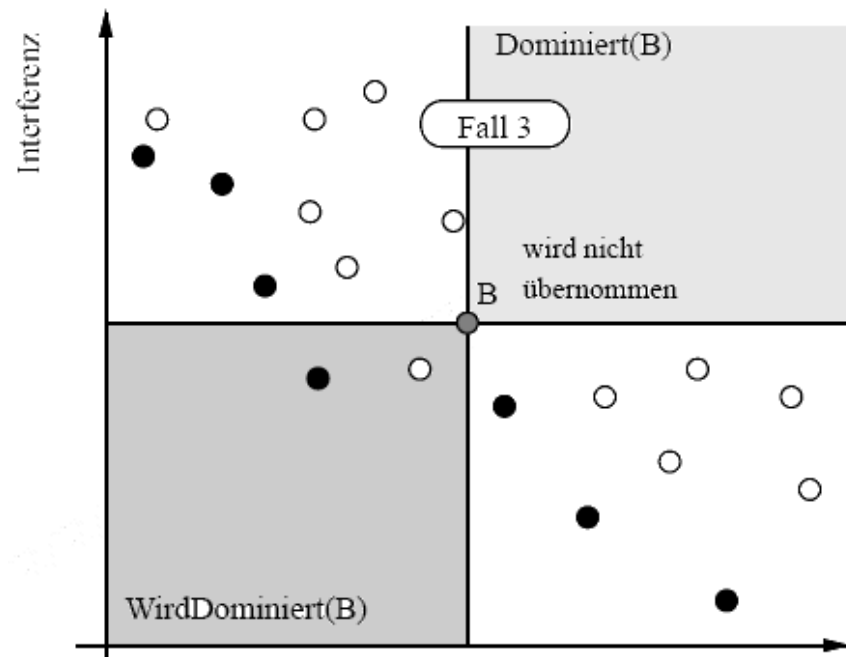
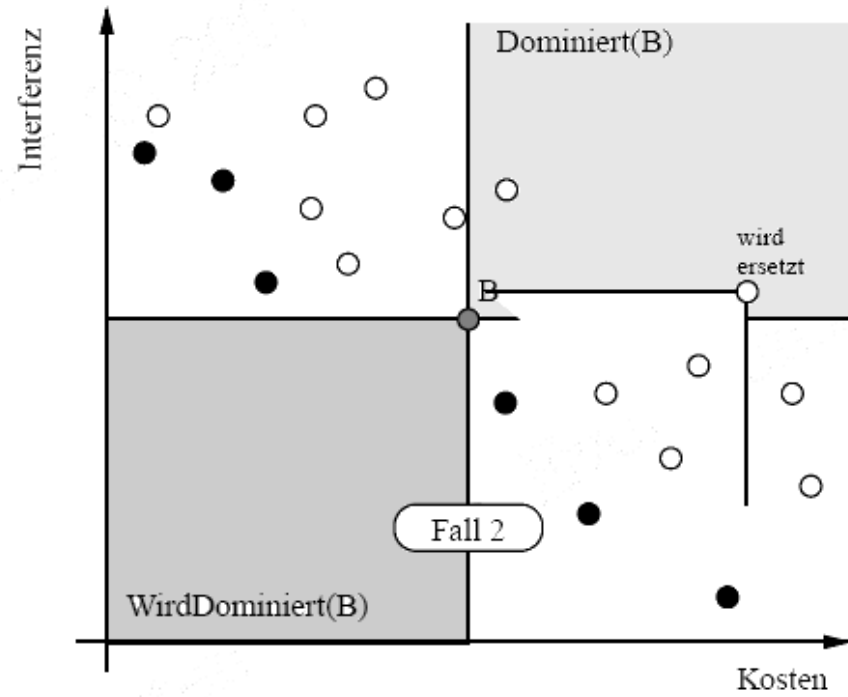
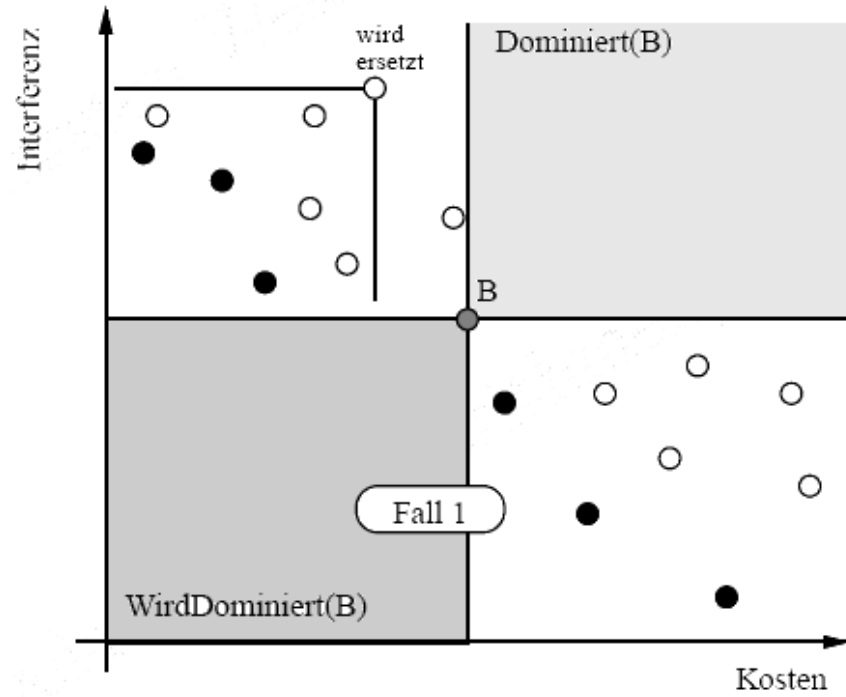
Vier Möglichkeiten

Wird das neue Individuum übernommen?

Welches wird ersetzt?

1. beide Mengen leer \Rightarrow übernehmen und Individuum mit schlechtesten Rang löschen
2. $Dominiert(B) \neq \emptyset \Rightarrow$ übernehmen und schlechtestes Individuum aus $Dominiert(B)$ löschen
3. $Dominiert(B) = \emptyset \wedge WirdDominiert(B) \neq \emptyset \Rightarrow B$ bleibt unberücksichtigt
4. beide Mengen leer und kein Individuum von einem anderen dominiert \Rightarrow übernehmen und gemäß Maß für Nischenbildung löschen

Selektion



Selektion

Datenstruktur für Population: 2D Bereichsbaum

Bereiche entsprechen beiden Zielfunktionswerten

Suchen, Einfügen und Löschen in $\mathcal{O}(\log^2 \mu)$

2D Bereichsanfragen (alle Individuen in diesem Bereich) in $\mathcal{O}(k + \log^2 \mu)$ mit Anzahl k der gefundenen Individuen

Algorithmus

Algorithm 4 Antennen-Optimierung

Input: Antennenproblem

- 1: $t \leftarrow 0$
 - 2: $P(t) \leftarrow$ initialisiere μ Individuen mit Reparaturfunktion
 - 3: berechne Rang für Individuen in $P(t)$
 - 4: **while** $t \leq G$ { /* maximale Generationenzahl G */
 - 5: $A, B \leftarrow$ selektiere aus $P(t)$ gemäß Rang und TURNIER-SELEKTION
 - 6: $C \leftarrow$ wende Operator auf A (und bei Rekombination auf B) an
 - 7: berechne Mengen $Dominiert(C)$ und $WirdDominiert(C)$
 - 8: $P(t + 1) \leftarrow$ integriere C in $P(t)$ und aktualisiere Ränge
 - 9: $t \leftarrow t + 1$
 - 10: }
 - 11: **return** nicht-dominierte Individuen aus $P(t)$
-

Konkrete Problemdaten

- ▶ $9 \times 9 \text{ km}^2$ Gebiet in Zürich
- ▶ Rasterung
 - ▶ Bedarf $500m$
 - ▶ Platzierung von Antennen $100m$
- ▶ insgesamt 505 Anrufe
- ▶ $\#Frequ = 128$
- ▶ maximale Kapazität $MaxCap = 64$
- ▶ Stärke zwischen $MinPow = 10dBmW$ und $MaxPow = 130dBmW$

Kostenfunktion und Parameter

Kostenfunktion

- ▶ Kosten einer Antenne:

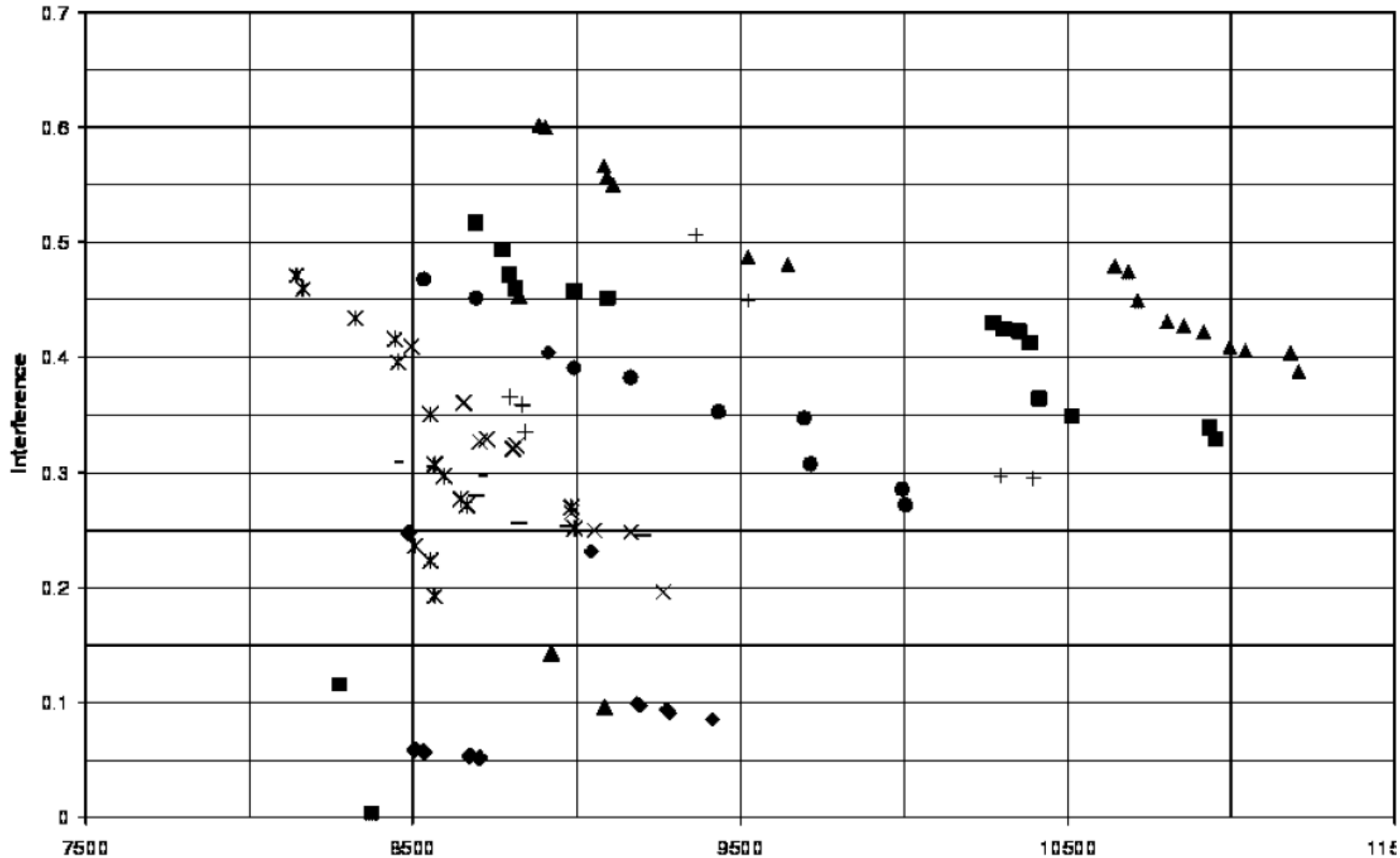
$$\textit{kosten}(\textit{pow}_i, \textit{cap}_i) = 10 \cdot \textit{pow}_i + \textit{cap}_i$$

Parametereinstellungen

- ▶ Populationsgröße $\mu = 80$
- ▶ 64000 Bewertungen
- ▶ Archivgröße von 80 Individuen (SPEA)

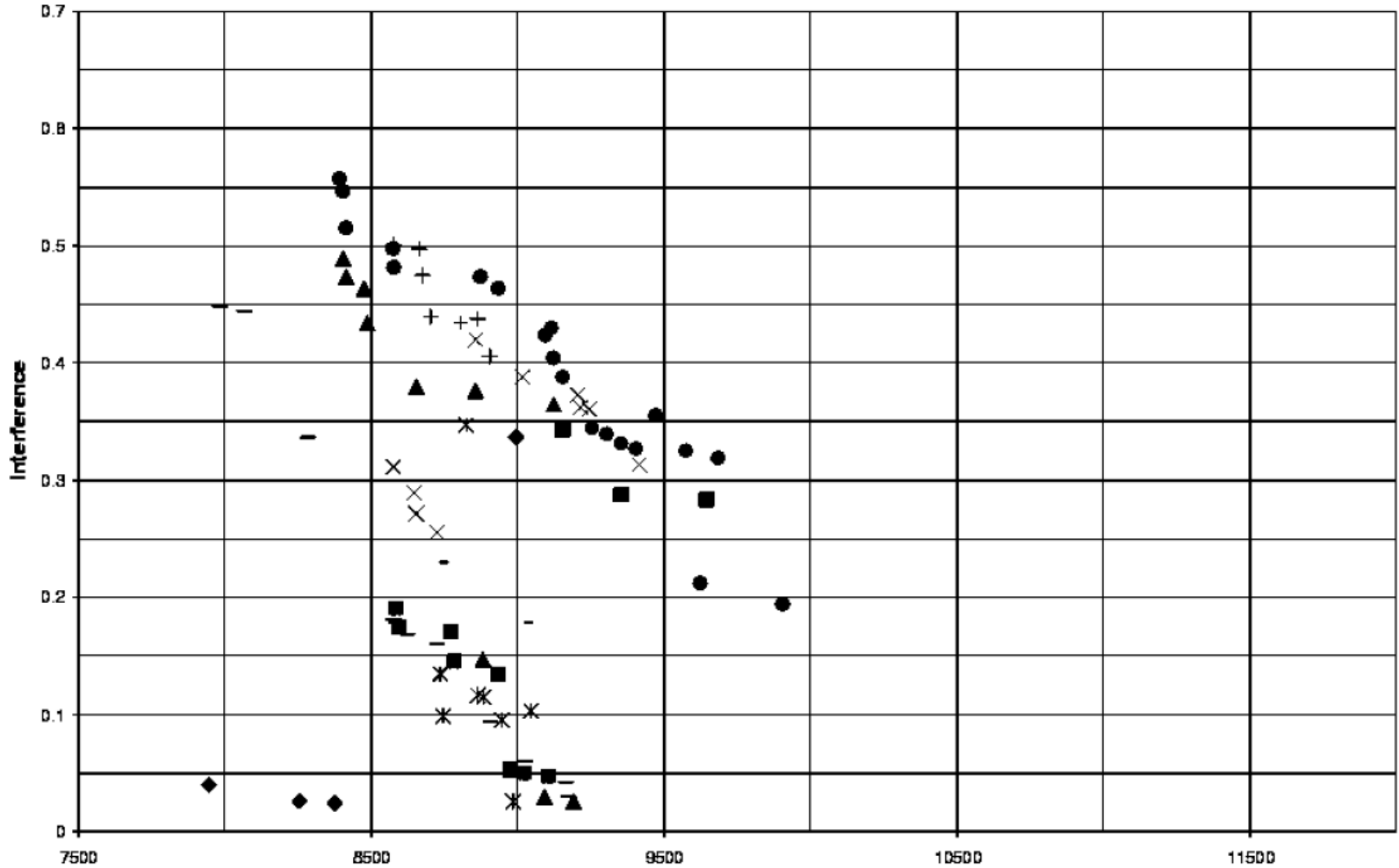
Pareto-Front

SPEA2, $p_{RM} = p_{DM} = 0.5$ und $p_{Rek} = 0$



Pareto-Front

eigene Selektion, $p_{RM} = p_{DM} = 0.5$ und $p_{Rek} = 0$



Mehrziel-Hypothesentest

Sa Fronten annähernd konvex:

$$\widehat{f}_{inferenz}(A) = \frac{f_{inferenz}}{0.7}$$

$$\widehat{f}_{kosten}(A) = \frac{f_{kosten} - 7500}{4500}$$

$$Qual(P) = \min_{A \in P} \left(\alpha \cdot \widehat{f}_{inferenz}(A) + (1 - \alpha) \cdot \widehat{f}_{kosten}(A) \right)$$

t-Test auf Werte von je 16 Experimenten

Positiv nur, wenn signifikant für alle $\alpha \in \{0.1, 0.2, \dots, 0.9\}$

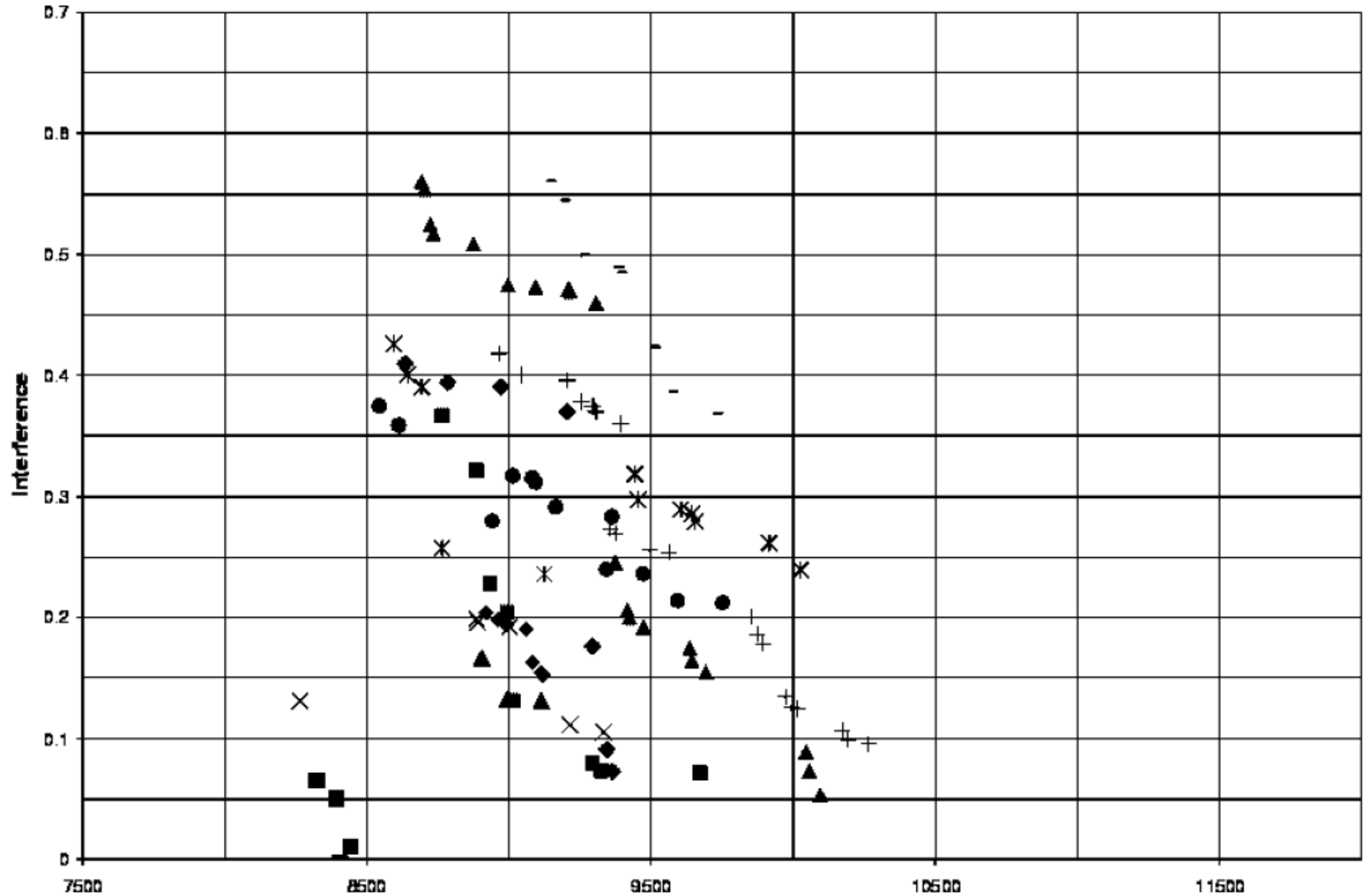
Signifikant: Kombination besser als rein zufällig

Kein Unterschied: vorherige Bilder

Bestes Ergebnis: nächste Seite

Pareto-Front

eigene Selektion, $p_{RM} = p_{DM} = 0.3$ und $p_{Rek} = 0.4$



Literatur zur Lehrveranstaltung I



Arrow, K. J. (1951).

Social Choice and Individual Values.

PhD thesis, Wiley, New York, USA.