FAKULTÄT FÜR INFORMATIK

Evolutionäre Algorithmen

Anwendung: Mehrkriterienoptimierung

Prof. Dr. Rudolf Kruse **Pascal Held**

{kruse,pheld}@iws.cs.uni-magdeburg.de

Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik

Institut für Wissens- und Sprachverarbeitung

Übersicht

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

Mehrkriterienoptimierung

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

$$f_i : \Omega \to \mathbb{R}, \qquad i = 1, \ldots, k$$

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 $f \to \max$, 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, $i = 1, \ldots, k$, wenn es kein Element $s' \in \Omega$ gibt, für das gilt

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

- **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

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

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

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

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:

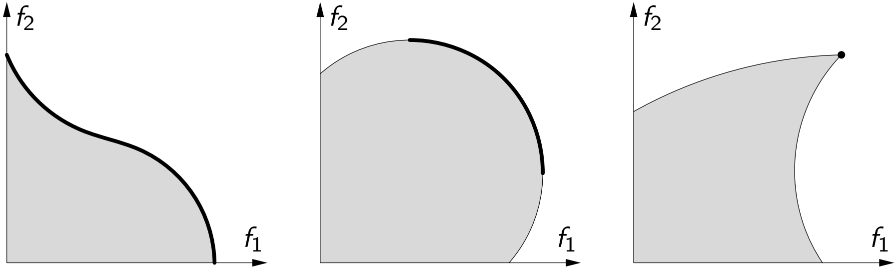
- Zielfunktionen müssen nicht zusammengefasst werden

  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

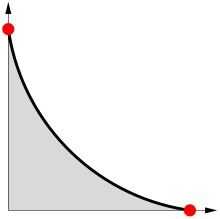
- **Einfachster Ansatz:** verwende gewichtete Summe der einzelnen Zielfunktionen als Fitnessfunktion

Lösung mit evolutionären Algorithmen

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

- gegeben k Kriterien, denen Zielfunktionen f_i, $1 \ldots, k$ zugeordnet sind
- $\forall i$, $1, \ldots, k$: 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

$\Rightarrow$ Gendrift: Pareto-Front konvergiert an beliebigem Punkt durch Zufallseffekte

Verhindern des Gendrifts

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

Algorithm 1 NSGA-SELEKTION

Input: Gütewerte $\langle A^{(i)}.F_j \rangle_{1 \le i \le r, 1 \le j \le k}$, Stichprobengröße N_{dom}

$I \leftarrow \{\}$
for $t \rightarrow 1, \ldots, s$ {
    $a \leftarrow U(\{1, \ldots, r\})$
    $b \leftarrow U(\{1, \ldots, r\})$
    $Q \leftarrow$ Teilmenge von $\{1, \ldots, r\}$ der Größe N_{dom}
    $d_a \leftarrow \exists i \in Q : A^{(i)} >_{\text{dom}} A^{(a)}$
    $d_b \leftarrow \exists i \in Q : A^{(i)} >_{\text{dom}} A^{(b)}$
    if d_a **and not** d_b {
        $I \leftarrow I \cup \{b\}$
    } **else** {
        if not d_a **and** d_b {
            $I \leftarrow I \cup \{a\}$
        } **else** {
            $n_a \leftarrow \left| \left\{ 1 \le i \le r \mid d(A^{(i)}, A^{(a)}) < \varepsilon \right\} \right|$
            $n_b \leftarrow \left| \left\{ 1 \le i \le r \mid d(A^{(i)}, A^{(b)}) < \varepsilon \right\} \right|$
            if $n_a > n_b$ {
                $I \leftarrow I \cup \{b\}$
            } **else** {
                $I \leftarrow I \cup \{a\}$
            }
        }
    }
}
return I

NSGA-Selektion

trotzdem mangelhafte Approximation der Pareto-Front

Gründe:

Parametereinstellung von ε

Population wird für zwei Zwecke genutzt

- als Speicher für nicht-dominierte Individuen (Pareto-Front)
- als lebendige Population (zur Durchforstung des Suchraums)

Abhilfe: Trennung des Archivs für nicht-dominierte Individuen von Population

- Archiv hat (meistens) endliche Größe
- Test aller Individuen auf Dominanz durch Archivindividuen
- bei Neuzugängen: dominierte Individuen aus Archiv entfernen

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 $F_1, \ldots, F_k$, Populationsgröße μ, Archivgröße $\tilde{\mu}$

```
t ← 0
P(t) ← erzeuge Population mit μ Individuen
R(t) ← ∅
while Terminierungsbedingung nicht erfüllt {
    bewerte P(t) durch F_1, ..., F_k
    for each A ∈ P(t) ∪ R(t) {
        AnzDom(A) ← |{B ∈ P(t) ∪ R(t) | A >_dom B}|
    }
    for each A ∈ P(t) ∪ R(t) {
        d ← Distanz von A und seinen √(μ + μ̃) nächsten Individuen in P(t) ∪ R(t)
        A.F ← 1/(d+2) + Σ_{B∈P(t)∪R(t), B>_dom A} AnzDom(B)
    }
    R(t + 1) ← {A ∈ P(t) ∪ R(t) | Aist nicht-dominiert}
    while |R(t + 1)| > μ̃ {
        entferne dasjenige Individuum aus R(t + 1) mit dem kürzesten/zweitkürzesten Abstand
    }
    if |R(t + 1)| < μ̃ {
        fülle R(t + 1) mit den gütebesten dominierten Individuen aus P(t) ∪ R(t)
    }
    if Terminierungsbedingung nicht erfüllt {
        Selektion aus P(t) mittels TURNIER-SELEKTION
        P(t + 1) ← wende Rekombination und Mutation an
        t ← t + 1
    }
}
return nicht-dominierte Individuen aus R(t + 1)
```

Pareto-Archived ES (PAES)

- $(1+1)$-Evolutionsstrategie
- Akzeptanzbedingung: Archivindividuum wird dominiert oder Funktionswertebereich is wenig frequentiert
- Nischen: ergeben sich aus Organisation des Archivs als mehrdimensionale Hash-Tabelle

Algorithm 3 PAES

Input: Zielfunktionen $F_1, \ldots, F_k$, Archivgröße $\tilde{\mu}$

$t \leftarrow 0$
$A \leftarrow$ erzeuge ein zufälliges Individuen
$R(t) \leftarrow \{A\}$ als mehrdimensionale Hash-Tabelle organisiert
while Terminierungsbedingung nicht erfüllt $\{$
    $B \leftarrow$ Mutation auf A
    bewerte B durch $F_1, \ldots, F_k$
    if $\forall C \in R(t) \cup \{A\}$: **not** $(C >_{\text{dom}} B)$ $\{$
        if $\exists C \in R(t) : (B >_{\text{dom}} C)$ $\{$
            entferne alle durch B dominierten Individuen aus $R(t)$
            $R(t) \leftarrow R(t) \cup \{B\}$
            $A \leftarrow B$
        $\}$ **else** $\{$
            if $|R(t)| = \tilde{\mu}$ $\{$
                $g^* \leftarrow$ Hash-Eintrag mit meisten Einträgen
                $g \leftarrow$ Hash-Eintrag für B
                if Einträge in $g <$ Einträge in g^* $\{$
                    entferne einen Eintrag aus g^*
                    $R(t) \leftarrow$ füge B in $R(t)$ ein
                $\}$
            $\}$ **else** $\{$
                $R(t) \leftarrow$ füge B in $R(t)$ ein
                $g_A \leftarrow$ Hash-Eintrag für A
                $g_B \leftarrow$ Hash-Eintrag für B
                if Einträge in $g_B <$ Einträge in g_A $\{$
                    $A \leftarrow B$
                $\}$
            $\}$
        $\}$
    $\}$
    $t \leftarrow t + 1$
$\}$
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

Übersicht

Aufgabenstellung: Antennenplatzierung

- Basisantennen für Mobilfunknetze
- erstes Ziel: hohe Netzverfügbarkeit
- zweites Ziel: geringe Kosten
- übliche Vorgehensweise:
  - Basisantennen platzieren und Größe/Reichweite konfigurieren $\Rightarrow$ Bedarf abdecken
  - Frequenzen zuweisen $\Rightarrow$ 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

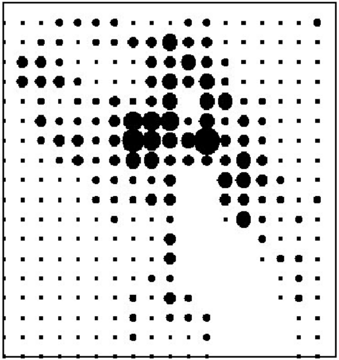
Rechteckiges Gebiet $(x_{\text{min}}, y_{\text{min}})$ und $(x_{\text{max}}, y_{\text{max}})$ mit Rasterung *res*

Menge aller (mögliche) Positionen:

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

Gesprächsbedarf Zürich

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



Formalisierung: Antenne

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

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

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

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

alle möglichen Antennenkonfigurationen:

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

Genotyp

Problemnaher Genotyp

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

Variable Länge

Randbedingungen

Netzverfügbarkeit hat oberste Priorität $\Rightarrow$
als harte Randbedinung formuliert

Erreichbare Positionen gemäß Wellenverbreitungsmodell:
$wp : Pos \times [MinPow, MaxPow] \to \mathcal{P}(Pos)$

$A.G = (t_1, \ldots, t_k)$ 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 $t_i = (pow, cap, pos, frq)$

Bewertungsfunktionen

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

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

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 Bedarfszellen

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 uberlappende Regionen haben ⇒ eine Antenne entfernen
DM4	falls Antennen große uberlappende 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

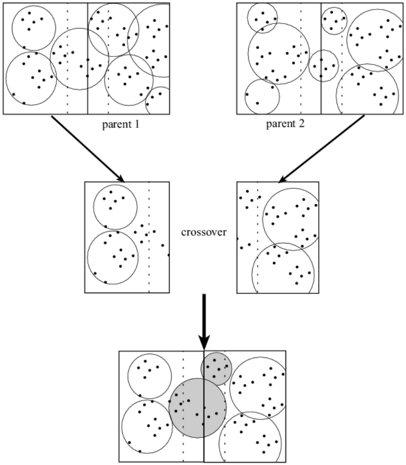
Zufällige Mutationsoperatoren

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 $\Rightarrow$ Ausgleich zu *DM4*
RM4	Kapazität einer Antenne zufällig verändern $\Rightarrow$ Ausgleich zu *DM1*
RM5	zugeordnete Frequenzen einer Antenne ändern $\Rightarrow$ Ausgleich zu *DM5*

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

- $\mathit{Dominiert}(A) =$ Menge der von A dominierten Individuen in Population
- $\mathit{WirdDominiert}(A) =$ Menge der Individuen, die A dominieren

Rang zuweisen

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

Einziges Problem: Gendrift, wenn alle Individuen gleichwertig

Selektion

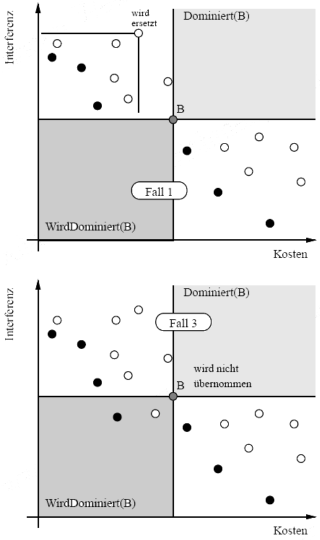
Vier Möglichkeiten

Wird das neue Individuum übernommen?

Welches wird ersetzt?

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

Selektion


Interferenz
wird ersetzt
Dominiert(B)
B
Fall 1
WirdDominiert(B)
Kosten
Interferenz
Dominiert(B)
wird ersetzt
B
Fall 2
WirdDominiert(B)
Kosten
Interferenz
Dominiert(B)
Fall 3
wird nicht übernommen
B
WirdDominiert(B)
Kosten
Interferenz
Dominiert(B)
Fall 4
B
ein Individuum mit vielen Nachbarn wird ersetzt
WirdDominiert(B)
Kosten

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

```
t ← 0
P(t) ← initialisiere μ Individuen mit Reparaturfunktion
berechne Rang für Individuen in P(t)
while t ≤ G {                          /* maximale Generationenzahl G */
    A, B ← selektiere aus P(t) gemäß Rang und TURNIER-SELEKTION
    C ← wende Operator auf A (und bei Rekombination auf B) an
    berechne Mengen Dominiert(C) und WirdDominiert(C)
    P(t + 1) ← integriere C in P(t) und aktualisiere Ränge
    t ← t + 1
}
return nicht-dominierte Individuen aus P(t)
```

Konkrete Problemdaten

- $9 \times 9\ km^2$ 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:

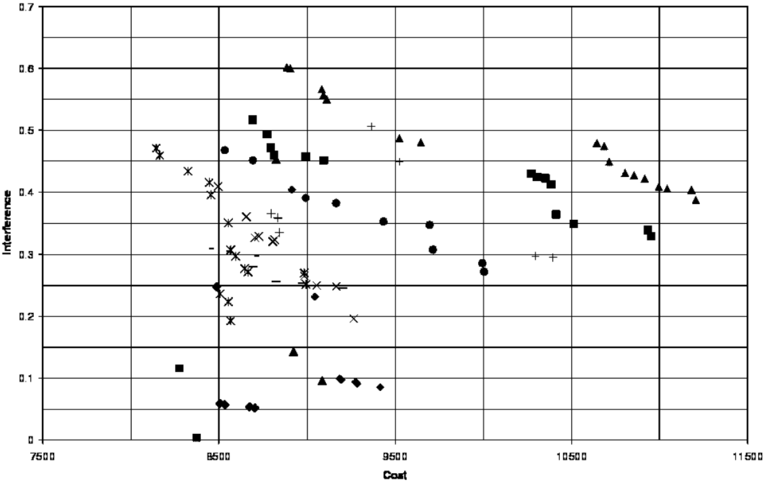
$$kosten(pow_i, cap_i) = 10 \cdot pow_i + cap_i$$

Parametereinstellungen

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

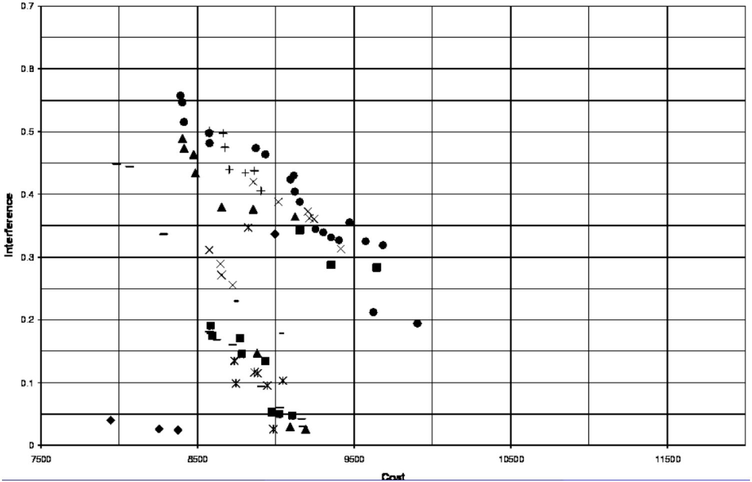
Pareto-Front

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:

$$\begin{aligned}
\widehat{f_{inferenz}}(A) &= \frac{f_{inferenz}}{0.7} \\
\widehat{f_{kosten}}(A) &= \frac{f_{kosten} - 7500}{4500} \\
Qual(P) &= \min_{A \in P} \left(\alpha \cdot \widehat{f_{inferenz}}(A) + (1-\alpha) \cdot \widehat{f_{kosten}}(A) \right)
\end{aligned}$$

t-Test auf Werte von je 16 Experimenten

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

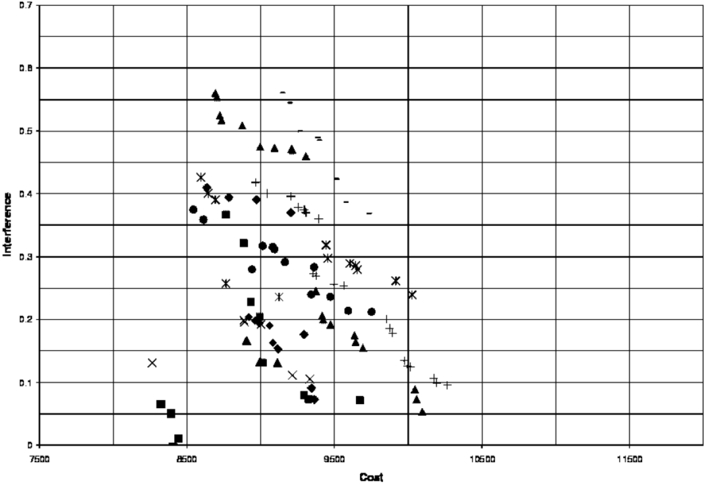
Signifikant: Kombination besser als rein zufällig

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.