

Evolutionäre Algorithmen: Techniken für spezifische Problemanforderungen

Karsten Weicker

HTWK Leipzig

5. Januar 2013

Überblick

- 1 Randbedingungen
- 2 Mehrzieloptimierung
- 3 Zeitabhängige Optimierungsprobleme
- 4 Approximative Bewertung

Überblick

- 1 Randbedingungen
- 2 Mehrzieloptimierung
- 3 Zeitabhängige Optimierungsprobleme
- 4 Approximative Bewertung

Randbedingungen

Definition

- Für Suchraum Ω ist eine *Randbedingung* eine Funktion

$$Rand : \Omega \rightarrow \{wahr, falsch\}.$$

- *harten Randbedingung*: muss zwingend erfüllt sein
man spricht auch von gültigen und ungültigen Individuen
- *weiche Randbedingung*: ist nur erwünscht

Charakteristika von Randbedingungen

Graduierbarkeit

boolesches Kriterium oder Grad der Verletzung
bestimmbar

Bewertbarkeit

für ein ungültiges Individuum berechenbar?

Schwierigkeit

... überhaupt ein gültiges Individuum zu finden

Charakteristika von Randbedingungen

Reparierbarkeit

kann ungültiges in gültiges überführt werden?

Bekanntheit

Grenze zwischen gültigen und ungültigen Individuen vorab bekannt?

Abgrenzung

Erfüllbarkeitsprobleme

- viele harte Randbedingungen
- hohe Schwierigkeit, nicht graduierbar und reparierbar, unbekannte Grenze
- Optimierung reduziert sich darauf, überhaupt ein gültiges Individuum zu finden

Zu schwache Randbedingungen

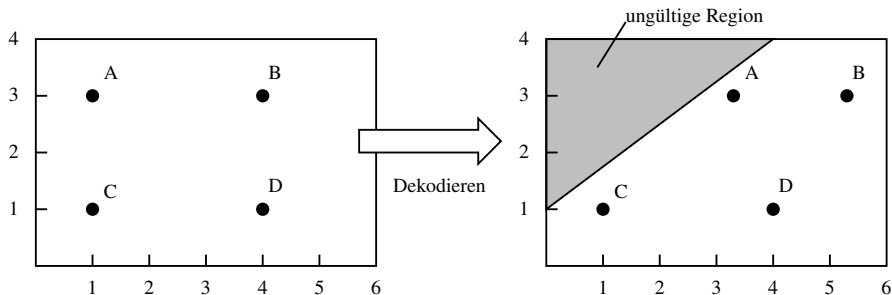
- viele weiche Randbedingungen
- hohe Graduierbarkeit und geringe Schwierigkeit
- besser: durch Mehrzieloptimierung angehen

Übersicht Algorithmen

Drei Klassen

- auf unbeschränktem Ω ungültige Individuen vermeiden
- ungültige Individuen zulassen, aber z.B. in der Selektion benachteiligen
- auf einem neuen Genotyp arbeiten, der immer gültige Lösungskandidaten garantiert

Der dritte Ansatz



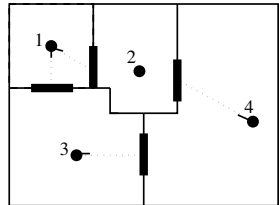
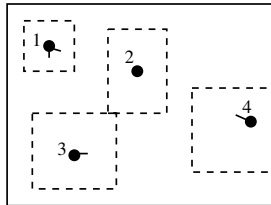
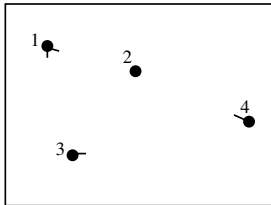
Dekodierer

$$dec(x, y) = \begin{cases} (x, y) & \text{falls } y < 1 \\ \left(\frac{4}{3} \cdot (y - 1) + \frac{7-y}{6} \cdot x, y \right) & \text{falls } y \geq 1 \end{cases}$$

Der dritte Ansatz

Konstruktion von Lösungskandidaten

- auch: aus einem Raum $\mathcal{G} \neq \Omega$ lassen sich durch eine Heuristik Lösungskandidaten erzeugen
- Beispiel: Grundrisse aus Punkten und Wachstumsregeln



Die weiteren Techniken

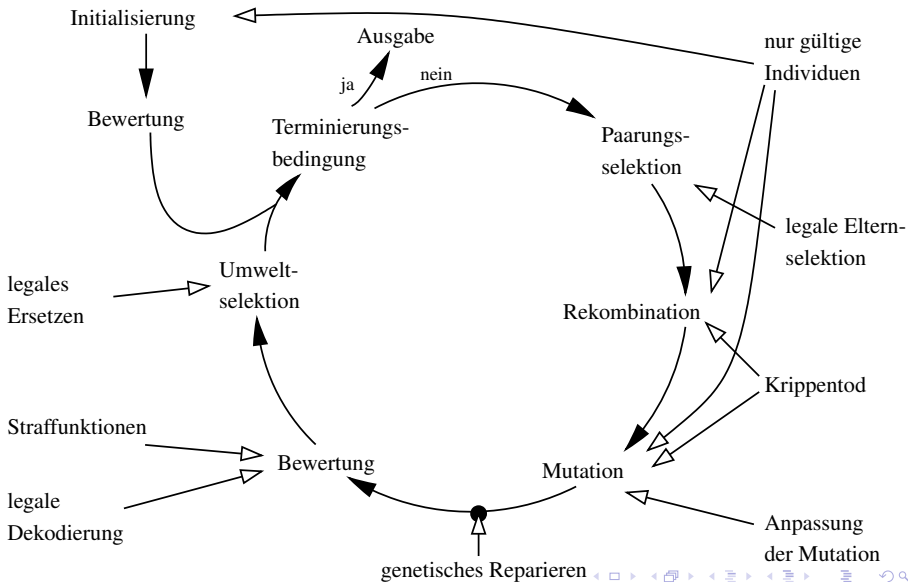
Ungültiges vermeiden

- Krippentod
- genetisches Reparieren
- Methode der gültigen Individuen

Ungültiges benachteiligen

- legale Elternselektion
- legales Ersetzen
- Anpassung der Mutation
- legale Dekodierung
- Straffunktionen

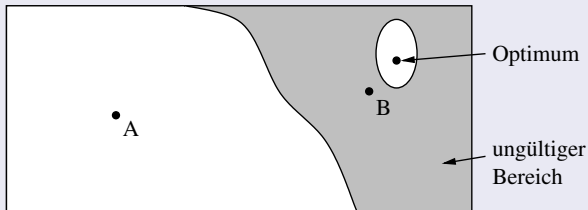
Überblick



Techniken

Krippentod

- einfachstes Verfahren
- ungültige Individuen werden sofort gelöscht
- häufig erstaunlich gut für einfach strukturierte Randbedingungen
- geringe Schwierigkeit; sonst: viele Iterationen notwendig



Techniken

Genetisches Reparieren

- falls Reparierbarkeit gilt
- ungültige Individuen werden mit einem Reparaturalgorithmus in gültige verwandelt
- effizienter bei geringer Schwierigkeit
- meist trotzdem Probleme bei unzusammenhängenden gültigen Bereichen
- Entwurf von Reparaturalgorithmen kann problematisch sein

Techniken

Methode der gültigen Individuen

- Anfangspopulation wird mit gültigen Individuen initialisiert
- Operatoren erzeugen aus gültigen Individuen ausschließlich wieder gültige Lösungskandidaten
- große Herausforderung für den Entwickler
- Beispiel: Maschinenbelegungspläne
Operatoren nutzen Lücken
- oft einfach für die Mutation, kritisch für die Rekombination
- weiteres Problem: Erreichbarkeit aller Punkte

Techniken

Legale Elternselektion

- falls in Population: gültige Individuen selektieren
- sonst: bevorzugt Individuen mit wenig Verletzungen der Randbedingungen
- geeignet für hohe Schwierigkeit mit Graduierbarkeit
- Problematisch: zu starke Konzentration bei Populationen mit wenig gültigen Individuen
- restriktivstes Verfahren, das ungültige Individuen zulässt

Techniken

Legales Ersetzen

- Voraussetzung: überlappende Populationen
- In der Ersetzungsstrategie werden diejenigen ersetzt, welche die meisten Randbedingungen verletzen
- falls nur gültige Individuen: zufällig oder schlechtestes ersetzen
- zusätzlicher Selektionsdruck in Elternselektion notwendig
- relativ universelles Verfahren

Techniken

Legale Dekodierung

- Variante des genetischen Reparierens
- Einsatz der Reparaturfunktion in der Dekodierung
- Genotyp bleibt unverändert
- bessere Erforschung ungültiger Gebiete
- Voraussetzung: Reparierbarkeit

Straffunktionen

Definition

Berücksichtigung der Randbedingung in der Bewertungsfunktion – meist als Strafterm.

Szenario 1

- Bewertungsfunktion nicht berechenbar für ungültige Individuen
- Bsp.: technische Systeme
- $$\tilde{f}(x) = \begin{cases} f(x) & \text{falls } x \text{ gültig} \\ f'(x) & \text{falls } x \text{ ungültig} \end{cases}$$
- Straffunktion f'

Straffunktionen

Szenario 2

- Bewertungsfunktion wird nicht durch verletzte Randbedingungen beeinflusst
- Bsp.: Minimierung von Produktkosten, Randbedingung: Mindestqualität
- Strafterm $\textit{Straf} : \Omega \rightarrow \mathbb{R}$ (mit $0 \succ \textit{Straf}(x)$ für alle ungültigen $x \in \Omega$)
- $$\tilde{f}(x) = \begin{cases} f(x) & \text{falls } x \text{ gültig} \\ f(x) + \textit{Straf}(x) & \text{falls } x \text{ ungültig} \end{cases}$$

Straffunktionen

Szenario 3

- Bewertungsfunktion liefert evtl. falsche Werte
- beide Ansätze möglich

Straffunktion

Einstellung

- Es muss gelten:

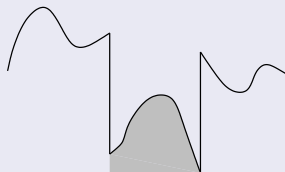
$$\forall \text{ ungültiges } x \in \Omega : \tilde{f}(x) \prec \max_{\text{gültiges } y \in \Omega} \tilde{f}(y)$$

- Fragwürdige Empfehlung:

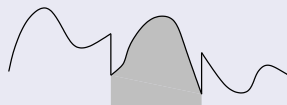
$$\forall \text{ ungültiges } x \in \Omega \forall \text{ gültiges } y \in \Omega : \tilde{f}(y) \succ \tilde{f}(x)$$



Gütelandschaft ohne Strafterm



mit großem Strafterm



mit moderatem Strafterm

Straffunktionen

Weitere Möglichkeiten

- falls graduierbar: Höhe des Strafterms am Grad der Verletzung orientieren
- falls reparierbar: Strafterm als Schätzung der Reparaturkosten
- vorbestimmte Anpassung:

$$\widetilde{Straf}(x) = \left(\frac{t}{MaxGen}\right)^2 \cdot Straf(x)$$

Straffunktionen

Adaptives Verfahren

- großer Strafterm bei vielen ungültigen Lösungskandidaten
- sonst: kleiner Strafterm
- $\widetilde{Straf}(x) = \eta^{(t)} \cdot Straf(x)$, wobei

$$\eta^{(t+1)} = \begin{cases} \frac{1}{\alpha_1} \cdot \eta^{(t)}, & \text{falls beste Individuen der} \\ & \text{letzten } k \text{ Generationen gültig} \\ \alpha_2 \cdot \eta^{(t)}, & \text{falls beste Individuen der} \\ & \text{letzten } k \text{ Generationen ungültig} \\ \eta^{(t)}, & \text{sonst} \end{cases}$$
 mit $\alpha_1, \alpha_2 > 1$

Strafterme

Mehrere Randbedingungen

- üblich: gewichtete Summe

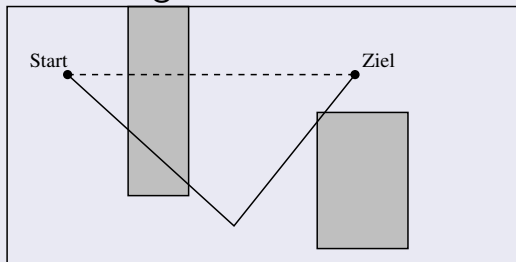
$$\textit{Straf}(x) = \sum_{i=1, \dots, n_c} \eta_i \cdot \textit{Straf}_i(x)$$

- dies ist nicht immer eine gute Idee – siehe nächster Abschnitt!

Schwierigkeiten bei der Bewertung

Beispiel: Pfadplanung

- Bewertung zweier Pfade



- direkter Pfad im Vorteil bei:
 - die Anzahl der Kollisionen,
 - die Länge des Pfads in den Hindernissen oder
 - der prozentuale Anteil des Pfads in Hindernissen

Überblick

- 1 Randbedingungen
- 2 Mehrzieloptimierung**
- 3 Zeitabhängige Optimierungsprobleme
- 4 Approximative Bewertung

Motivation

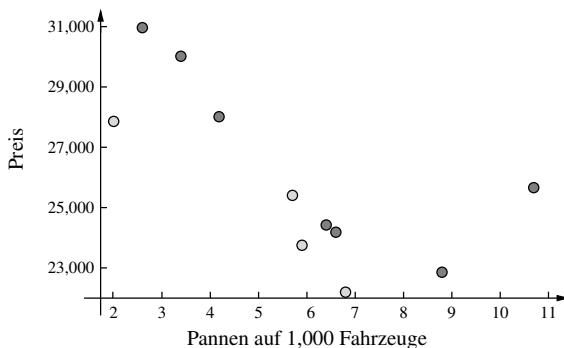
Bsp.: Ein Mittelklassefahrzeug

- möglichst günstiger Preis
- möglichst positive Pannenstatistik

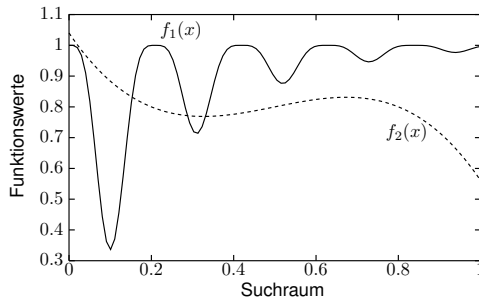
Motivation

Bsp.: Ein Mittelklassefahrzeug

- möglichst günstiger Preis
- möglichst positive Pannenstatistik



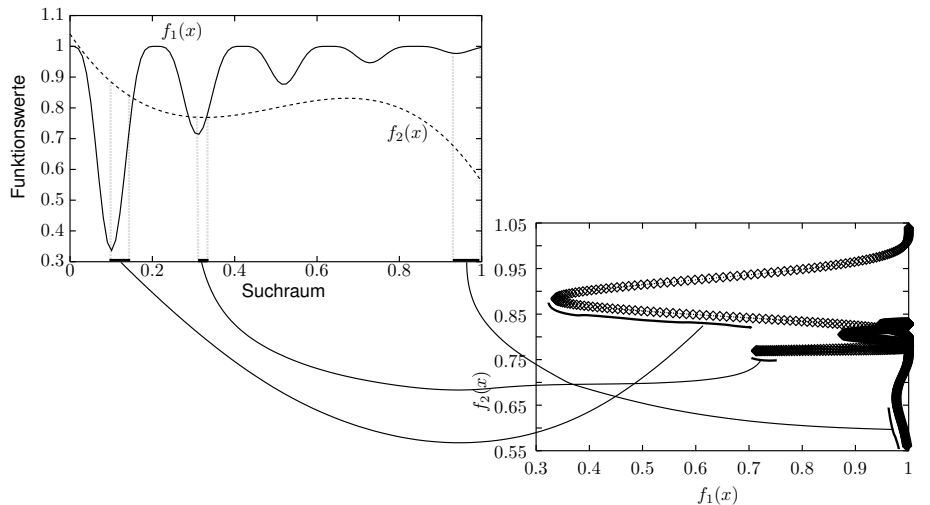
Welche Werte sind optimal?



f_1 und f_2 sind
zu minimieren

**Welche Lösungskandidaten sind wie gut?
Was ist ein sinnvoller Kompromiss?**

Welche Werte sind optimal?



Erste Beobachtungen

- große Teile im Raum der Bewertungsfunktionswerte sind nicht durch Lösungskandidaten abgedeckt
- im Raum der Bewertungsfunktionswerte nahe beieinanderliegende Individuen können Suchraum weit voneinander entfernt sein
- möglichen Kompromisslösungen liegen oft weit auseinander
- Wege im Suchraum führen nicht zwingend auf das ideale „Optimum“ zu

Wie sieht jetzt ein Optimum aus?

Definition

- Bewertungsfunktionen F_i ($1 \leq i \leq k$)
- B dominiert A , wenn

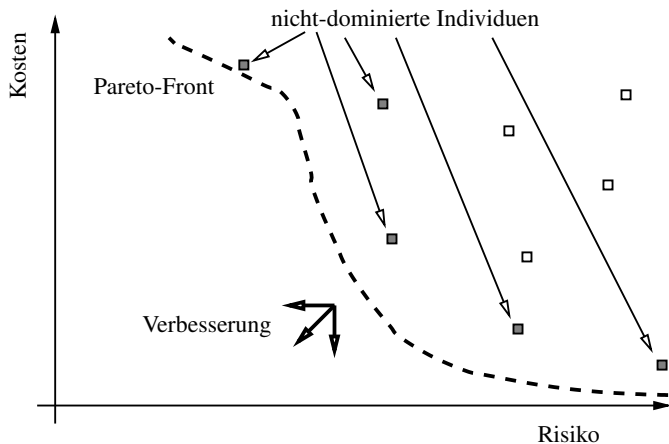
$$B >_{dom} A := \forall 1 \leq i \leq k : F_i(B.G) \succeq F_i(A.G) \\ \wedge \exists 1 \leq i \leq k : F_i(B.G) \succ F_i(A.G)$$

- nicht-dominierte Individuen in P :

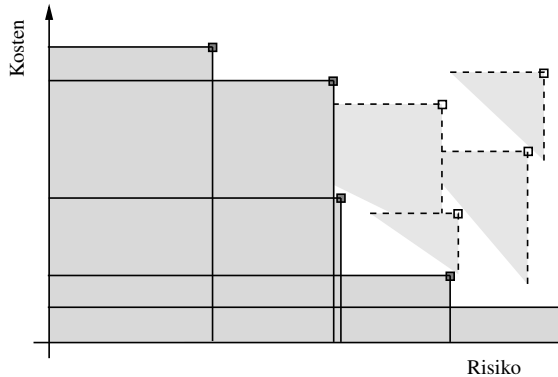
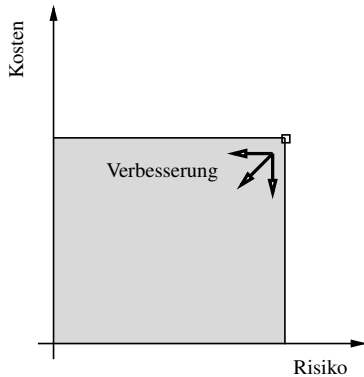
$$nichtdom(P) := \left\{ A \in P \mid \forall B \in P : \neg (B >_{dom} A) \right\}$$

- Pareto-Front: $nichtdom(\Omega)$

Veranschaulichung der Pareto-Front



Veranschaulichung der Pareto-Front



Nicht-funktionierende intuitive Ideen

Separate Optimierung

- erst eine Bewertungsfunktion, dann eine andere benutzen
- entweder: nur die erste bestimmt Ergebnis
- oder: zweiter Teil führt vom Zwischenergebnis weg
- fast immer: suboptimales Ergebnis

als Randbedingung

- Schwellwert einer Bewertung berücksichtigen
- Bewertungsfunktionen sind nicht mehr gleichberechtigt

Überblick über Verfahren

Entscheiden vor Optimieren

- erst bestimmen, wie stark welches Kriterium berücksichtigt werden soll
- dann demgemäß optimieren

Optimieren vor Entscheiden

- zunächst optimieren mit dem Ziel, ein breites Spektrum an Alternativen zu bekommen
- dann wird die passende Alternative gewählt

Interaktiv

- Optimierungs- und Entscheidungsphasen wechseln

Modifikation der Bewertungsfunktion

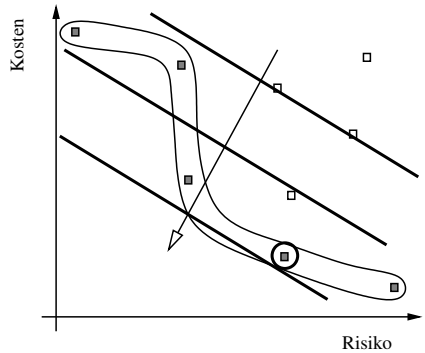
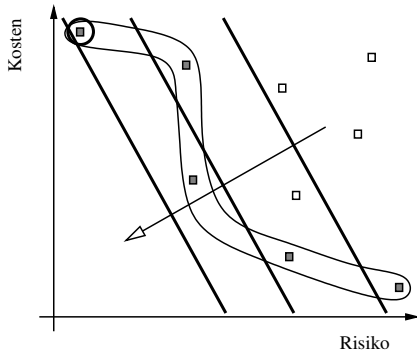
Aggregierende Verfahren

- die Gütwerte mehrerer Bewertungsfunktionen werden auf einen Wert abgebildet
- Linearkombination: zu minimierendes

$$f(x) = \sum_{i=1}^k \eta_i \cdot f_i(x)$$

mit den Gewichtungsfaktoren η_i . Wird $\eta_i > 0$ für zu minimierende $f_i(x)$ bzw. $\eta_i < 0$ für zu maximierende $f_i(x)$

Aggregierend: Linearkombination

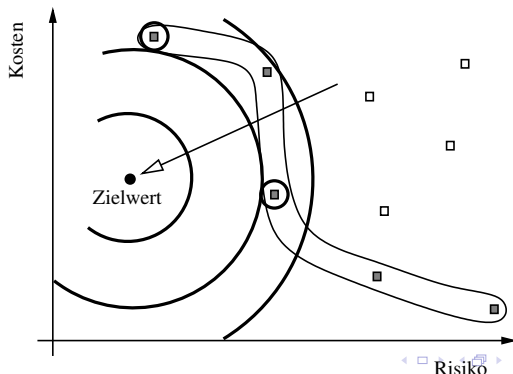


Mögliche Probleme?

Aggregierend: Zielvektor

Distanz zu einem Zielvektor minimieren

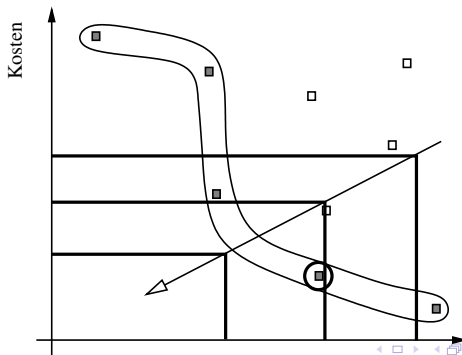
$$f(x) = \sqrt{\|f_1(x) - y_1^*\|^2 + \dots + \|f_k(x) - y_n^*\|^2}$$



Aggregierend: Minimax-Methode

Maximale Bewertungsfunktion minimieren

$$f(x) = \max_{i=1,\dots,n} \eta_i \cdot \|f_i(x) - y_i^*\|$$



Berechnung der Pareto-Front

Nutzen der Aggregation

- z.B. Linearkombination mit verschiedenen Kombinationen
- Beispiel für zwei Bewertungsfunktionen:

$$f(x) = \eta \cdot f_1(x) + (1 - \eta) \cdot f_2(x)$$

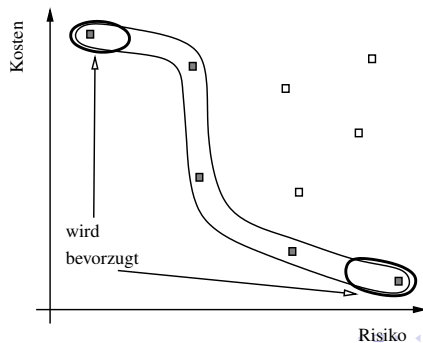
mit $\eta \in [0, 1]$

- Vorsicht: Probleme der aggregierenden Verfahren gelten auch hier!

Einfache Ideen

VEGA-Verfahren

- gleichberechtigtes Nutzen von k Bewertungsfunktionen in der Selektion
- jeweils k -ter Teil bei der Elternselektion



Nutzen der Pareto-Dominanz

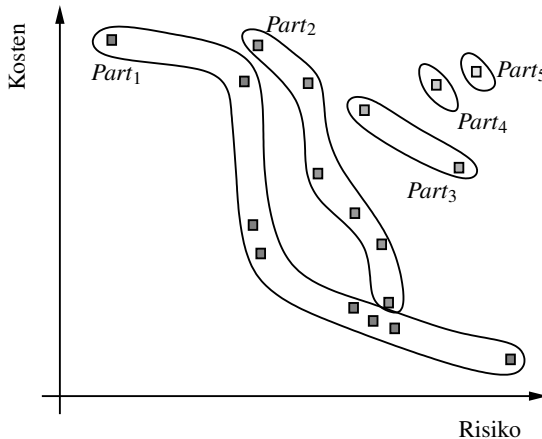
„Zwiebelschalentechnik“

$$Part_1 := \{A \in P \mid \forall B \in P : \neg(B >_{dom} A)\}$$

$$Part_i := \left\{ A \in P \setminus \bigcup_{1 \leq j < i} Part_j \mid \forall B \in P \setminus \bigcup_{1 \leq j < i} Part_j : \neg(B >_{dom} A) \right\}$$

$$F(A.G) := \begin{cases} 1 & \text{falls das zugehörige } A \in Part_1 \\ \vdots & \vdots \\ n & \text{falls das zugehörige } A \in Part_n \end{cases}$$

Nutzen der Pareto-Dominanz



Pareto-Bewertung

Probleme

- alle Individuen der Pareto-Front werden gleich bewertet
- es setzt Gendrift ein und die Pareto-Front konvergiert an einem beliebigen Punkt

Maßnahmen

- Verteilung der Individuen entlang der Front
- insbesondere: nischenbildende Maßnahmen

Nischenbildung

Güteteilen

- Haufen von Individuen werden schlecht bewertet
- isolierte Individuen gut
- monoton fallende Funktion $Teile : \mathbb{R} \rightarrow [0, 1]$ mit $Teile(0) = 1$ und $\lim_{d \rightarrow \infty} Teile(d) = 0$
- Individuen in der Nische von A :

$$m_{A,P} = \sum_{B \in P} Teile(\hat{d}(A, B))$$
 mit $\hat{d}$ als Entfernung im Raum der Funktionswerte
- Modifikation der Güte: $A.F = \frac{F(A)}{m_{A,P}}$

NSGA-Selektion

Idee

- aufwändige Berechnung der Partitionen vermeiden
- stattdessen: Turnierselektion welche sowohl Pareto-Optimalität und Nischen benutzt

Technik

- Referenzindividuen werden gewählt
- nichtdominiertes Individuum wird selektiert
- sonst: dasjenige mit weniger Individuen in der Nische

NSGA-Selektion

NSGA-SELEKTION(Güterwerte $\langle A^{(i)}.F_j \rangle_{1 \leq i \leq r, 1 \leq j \leq k}$)

```

1   $I \leftarrow \langle \rangle$ 
2  for  $i \leftarrow 1, \dots, s$ 
3  do  $\lceil indexA \leftarrow U(\{1, \dots, r\})$ 
4       $indexB \leftarrow U(\{1, \dots, r\})$ 
5       $Q \leftarrow$  Teilmenge von  $\{1, \dots, r\}$  der Größe  $N_{dom}$  (Stichprobengröße)
6       $dominatedA \leftarrow \exists index \in Q : index >_{dom} indexA$ 
7       $dominatedB \leftarrow \exists index \in Q : index >_{dom} indexB$ 
8      if  $dominatedA \wedge \neg dominatedB$ 
9      then  $\lceil I \leftarrow I \circ \langle indexB \rangle$ 
10     else  $\lceil$  if  $\neg dominatedA \wedge dominatedB$ 
11         then  $\lceil I \leftarrow I \circ \langle indexA \rangle$ 
12         else  $\lceil nischeA \leftarrow \#\{1 \leq index \leq r \mid \widehat{d}(A^{(index)}, A^{(indexA)}) < \varepsilon$  (N
13              $nischeB \leftarrow \#\{1 \leq index \leq r \mid \widehat{d}(A^{(index)}, A^{(indexB)}) < \varepsilon\}$ 
14             if  $nischeA > nischeB$ 
15             then  $\lceil I \leftarrow I \circ \langle indexB \rangle$ 
16             else  $\lceil I \leftarrow I \circ \langle indexA \rangle$ 
17 return  $I$ 
    
```

Immernoch Probleme...

nämlich

- immernoch mangelhafte Approximation der Pareto-Front
- Grund: u.a. wird die Population für zwei Zwecke genutzt:
 - als Speicher für nicht-dominierte Individuen
 - als lebendige Population

Immernoch Probleme...

Maßnahme

- Trennung des Archivs für nicht-dominierte Individuen von der Population
- Archiv hat meist endliche Größe, da
 - alle Individuen auf Dominanz durch Archivindividuen zu testen sind
 - bei Neuzugängen dann dominierte Individuen zu entfernen sind

SPEA2

Idee

- normaler evolutionärer Algorithmus
- Bewertungsfunktion besteht aus zwei Komponenten
 - wieviele Individuen dominieren die Individuen, die dieses Individuum dominieren
 - Distanz zum $\sqrt{n}$ -nächsten Individuum
- Archiv geht in Güteberechnung mit ein
- Archiv enthält nicht-dominierte Individuen
- falls zu wenig: zusätzlich gütebeste Individuen
- Ersetzen im Archiv aufgrund der Entfernung zu anderen archivierten Individuen

SPEA2(Zielfunktionen $F_1, \dots, F_m$)

```

1   $t \leftarrow 0$ 
2   $P(t) \leftarrow$  erzeuge Population mit  $\mu$  (Populationsgröße) Individuen
3   $R(t) \leftarrow \emptyset$  (Archiv der Größe  $\tilde{\mu}$  (Archivgröße) )
4  while Terminierungsbedingung nicht erfüllt
5  do  $\lceil$  bewerte  $P(t)$  durch  $F_1, \dots, F_m$ 
6      for each  $A \in P(t) \cup R(t)$ 
7          do  $\lfloor \text{AnzDom}(A) \leftarrow \#\{B \in P(t) \cup R(t) \mid A >_{\text{dom}} B\}$ 
8      for each  $A \in P(t) \cup R(t)$ 
9          do  $\lceil \text{dist} \leftarrow$  Distanz von  $A$  und seinem  $\sqrt{\mu + \tilde{\mu}}$ -nächsten Individuum in  $P(t) \cup R(t)$ 
10          $\lfloor A.F \leftarrow (\sum_{B \in P(t) \cup R(t) \text{ mit } B >_{\text{dom}} A} \text{AnzDom}(B)) + \frac{1}{\text{dist}+2}$ 
11      $R(t+1) \leftarrow \{A \in P(t) \cup R(t) \mid A \text{ ist nicht-dominiert}\}$ 
12     while  $\#R(t+1) > \tilde{\mu}$ 
13     do  $\lfloor$  entferne dasjenige Individuum aus  $R(t+1)$  mit dem kürzesten/zweitkürzesten Abstand
14     if  $\#R(t+1) < \tilde{\mu}$ 
15     then  $\lfloor$  fülle  $R(t+1)$  mit den gütebesten dominierten Individuen aus  $P(t) \cup R(t)$ 
16     if Terminierungsbedingung nicht erfüllt
17     then  $\lceil$  Selektion aus  $P(t)$  mittels TURNIER-SELEKTION
18          $P(t+1) \leftarrow$  wende Rekombination und Mutation an
19          $u \leftarrow$  wähle Zufallszahl gemäß  $U([0,1))$ 
20      $\lfloor \lfloor t \leftarrow t + 1$ 
21 return nicht-dominierte Individuen aus  $R(t+1)$ 
    
```

PAES

Idee

- $(1 + 1)$ -Evolutionsstrategie
- Akzeptanzbedingung:
 - ein Archivindividuum wird dominiert oder
 - der Funktionswertbereich ist wenig frequentiert
- Nischen ergeben sich aus der Organisation des Archivs als Gridfile (mehrdimensionale Hash-Tabelle)

PAES(Zielfunktionen $F_1, \dots, F_m$)

```

1   $t \leftarrow 0$ 
2   $A \leftarrow$  erzeuge ein zufälliges Individuum
3  bewerte  $A$  durch  $F_1, \dots, F_m$ 
4   $R(t) \leftarrow \langle A \rangle$  als Gridfile organisiert
5  while Terminierungsbedingung nicht erfüllt
6  do  $B \leftarrow$  Mutation auf  $A$ 
7      bewerte  $B$  durch  $F_1, \dots, F_m$ 
8      if  $\forall C \in R(t) \circ \langle A \rangle : \neg(C >_{dom} B)$ 
9      then  $\lceil$  if  $\exists C \in R(t) : B >_{dom} C$ 
10         then  $\lceil R(t) \leftarrow$  entferne alle durch  $B$  Individuen aus  $R(t)$ 
11              $R(t) \leftarrow$  füge  $B$  in  $R(t)$  ein
12          $\lceil A \leftarrow B$ 
13     else  $\lceil$  if  $\#R(t) = \tilde{\mu}(\text{Archivgröße})$ 
14         then  $\lceil g^* \leftarrow$  Grid-Zelle mit den meisten Einträgen
15              $g \leftarrow$  Grid-Zelle für  $B$ 
16             if Einträge in  $g <$  Einträge in  $g^*$ 
17             then  $\lceil R(t) \leftarrow$  entferne einen Eintrag aus  $g^*$ 
18                  $\lceil R(t) \leftarrow$  füge  $B$  in  $R(t)$  ein
19         else  $\lceil R(t) \leftarrow$  füge  $B$  in  $R(t)$  ein
20          $g_A \leftarrow$  Grid-Zelle für  $A$ 
21          $g_B \leftarrow$  Grid-Zelle für  $B$ 
22         if Einträge in  $g_B <$  Einträge in  $g_A$ 
23         then  $\lceil A \leftarrow B$ 
24      $\lceil t \leftarrow t + 1$ 
25 return nicht-dominierte Individuen aus  $R(t + 1)$ 
    
```

Überblick

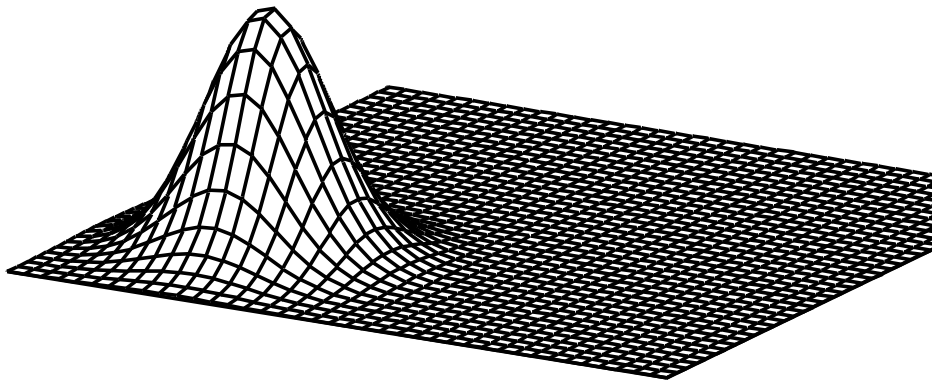
- 1 Randbedingungen
- 2 Mehrzieloptimierung
- 3 Zeitabhängige Optimierungsprobleme
- 4 Approximative Bewertung

Zeitabhängige Optimierungsprobleme

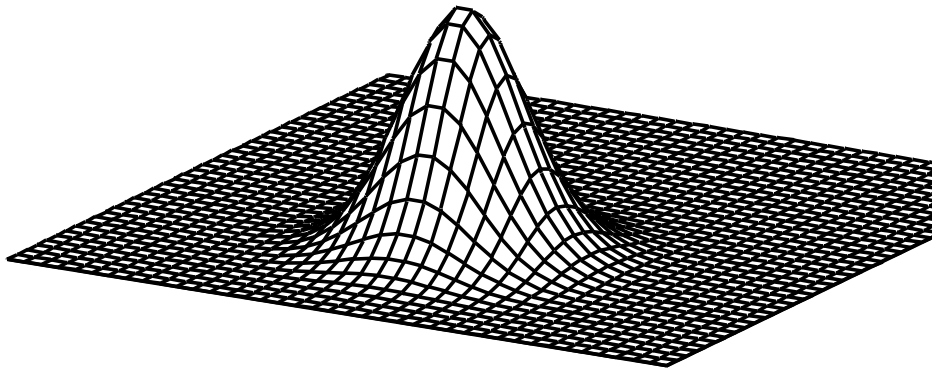
Definition

- Ein *zeitabhängiges Optimierungsproblem* besteht aus einer Folge $Opt^{(t)}$ ($t \in \mathbb{N}$) von statischen Optimierungsproblemen $Opt^{(t)} = (\Omega, f^{(t)}, \succ)$.
- Ges. für jedes $t \in \mathbb{N}$: eine Approximation der globalen Optima $\mathcal{X}^{(t)}$.

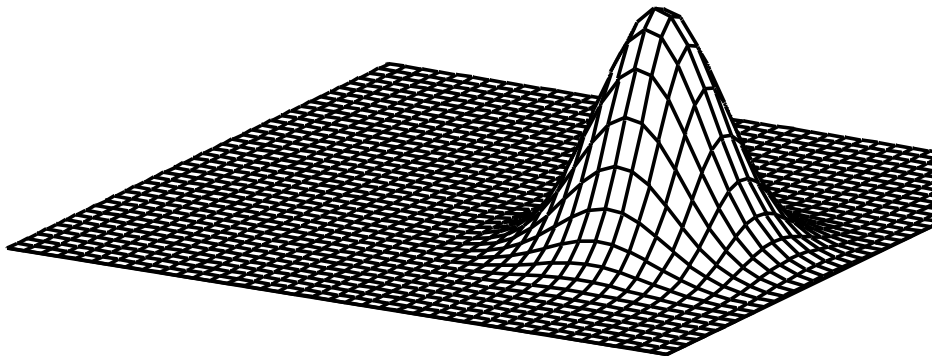
Einfaches Beispiel



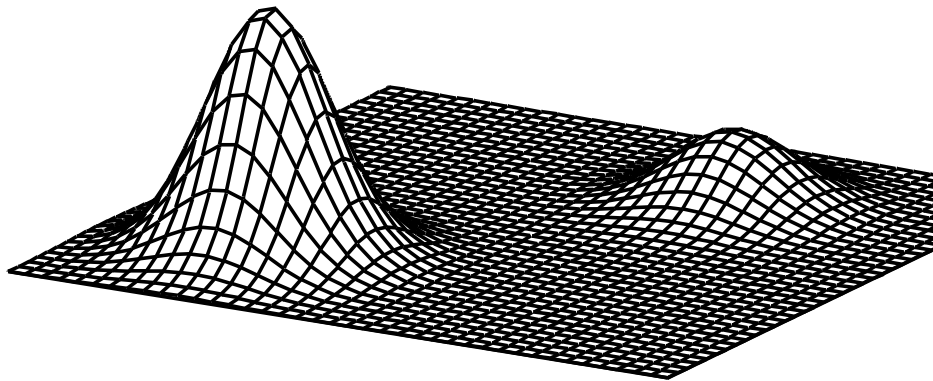
Einfaches Beispiel



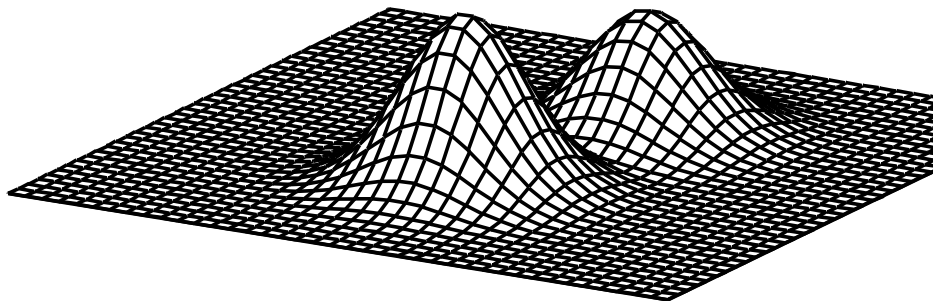
Einfaches Beispiel



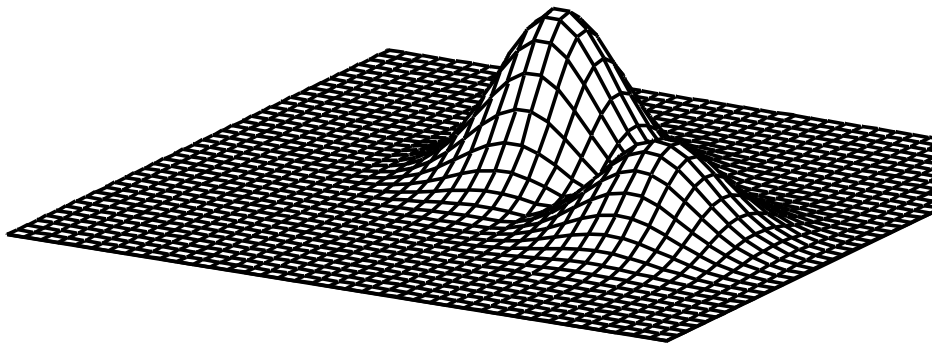
Schwierigeres Beispiel



Schwierigeres Beispiel



Schwierigeres Beispiel



Motivation

Neue Anforderungen

- Sich bewegende lokale Optima müssen verfolgt werden
- Neu entstehende lokale Optima müssen entdeckt werden
- Bei zu starken Veränderungen: komplett neue Optimierung notwendig.

Versuch der Einordnung Technik/Problem

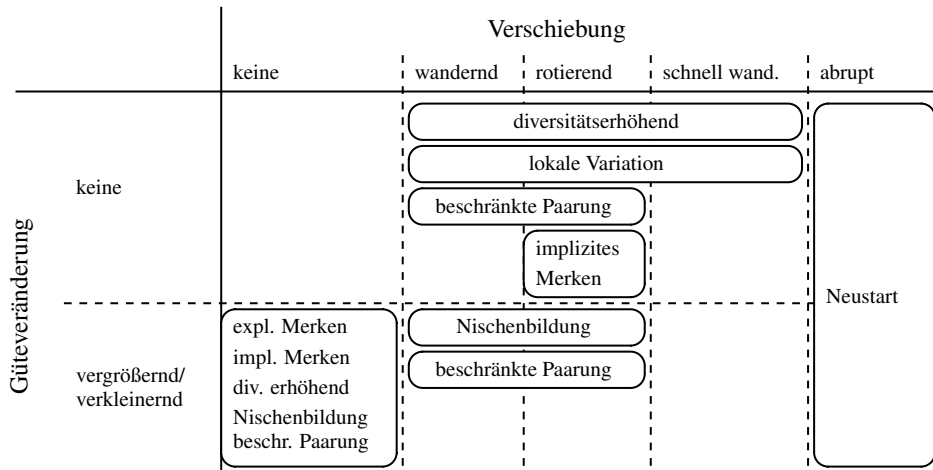
Kategorisierung von Verschiebungen

- keine Veränderung,
- eine langsam wandernde Bewegung,
- eine drehende Bewegung,
- eine schnell wandernde Bewegung und
- eine einzelne abrupte, große Veränderung.

Kategorisierung der Güteveränderung

- keine Veränderung oder
- meist in kleinen Schritten Vergrößerung oder Verkleinerung.

Versuch der Einordnung Technik/Problem



Techniken

Neustart

- bei abrupten Änderungen mit hinreichend langen, stabilen Phasen
- vollständig neue Optimierung
- wichtig: Mechanismus zum Detektieren einer abrupten Änderung

Techniken

Mehr Diversität

- bei kontinuierlichen oder sich wiederholenden Änderungen
- mehrere Einzeltechniken:
 - Diversität künstlich erhöhen
 - Nischenbildung im Rahmen der Selektion
 - beschränkte Paarung

Techniken

Diversitätserhöhende Technik

- zufällige Einwanderer
 - jede Generation: 10-30% neue zufällige Individuen
 - schwierig auszubalancieren
- Hypermutation
 - erhöhte Mutationsrate unter Stress
 - bei Abfall der durchschnittlichen Güte in der Population: Mutationsrate kurzzeitig drastisch erhöhen
 - Problem: Konvergenz im ehemaligen, unveränderten globalen Optimum

Techniken

Einnischung

- ähnlich zur Einnischung bei der Mehrzieloptimierung
- nur: Abstand im Genotyp/Phänotyp
- Variante: theormodynamischer GA
 - GA erzeugt $2 \cdot \mu$ Kindindividuen
 - Umweltselektion wählt μ Individuen
 - jeweils das Individuum, das mit den bereits gewählten Individuen in P' die Funktion $\hat{F}(P' \circ \langle A \rangle)$ minimiert.

$$\hat{F}(P) = \overline{F}(P) - \eta \cdot \text{Divers}_{\text{Shannon}}(P)$$

Techniken

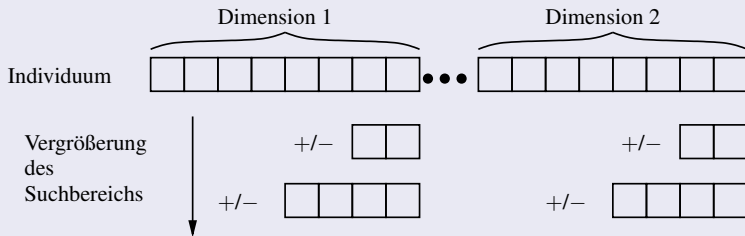
Einnischung

- nur die Paarung zwischen bestimmten Individuen ist für die Rekombination erlaubt
- erste Möglichkeit: zufällige Markierungen an Individuen, nur gleich markierte sind erlaubt
- zweite Möglichkeit: Zuordnung zu Teilpopulationen aufgrund von Entfernungsberechnungen
- dritte Möglichkeit: tatsächlich getrennte Populationen z.B. bei parallelen Algorithmen

Techniken

lokale Variation

- reines Verfolgen von lokalen Optima
- ideal: selbstadaptive Mutationen aus ES und EP
- Variante auf standardbinär kodiertem Genotyp:
variable lokale Suche
 - ähnlich getriggert wie Hypermutation
 - Suchbereich wird von lokal zu global inkrementiert



Techniken

Memorisierende Techniken

- explizit
- implizit durch internes Speichern von Teillösungen in Individuen
- gut bei: Problemen die zu früheren Zuständen zurückkehren

Techniken

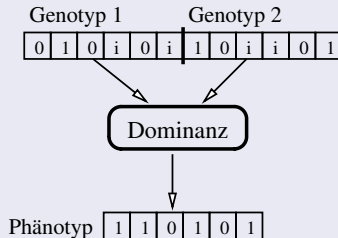
explizites Gedächtnis/Merken

- Speicher für gute Lösungen früherer Generationen
- Zugriff wie bei
 - zufälligen Einwanderern oder
 - bei Abfall der Güte (vgl. Hypermutation)
- notwendig: Mechanismus zur Verwaltung des Speichers

Techniken

Implizites Gedächtnis/Merken

- ahmt Konzept der Diploidität nach
- alte Informationen können wieder später reaktiviert werden
- einfachstes Beispiel: rezessive 1 (Notation: i)



Dominanzregel:

	0	i	1
0	0	0	1
i	0	1	1
1	1	1	1

Überblick

- 1 Randbedingungen
- 2 Mehrzieloptimierung
- 3 Zeitabhängige Optimierungsprobleme
- 4 **Approximative Bewertung**

Approximative Bewertung

Motivation

- Bewertungsfunktion kann nur genäherte Gütwerte liefern
- Ursachen:
 - Toleranzen bei Messungen
→ Verrauschte Bewertung
 - Schwierigkeit, Werte exakt einzustellen
→ Stabile Lösungen
 - zu zeitaufwändige Bewertung
 - Bewertung ist nur über viele Testfälle möglich
 - Strategien (z.B. für Spiele)

Verrauschte Bewertung

Beispiel

- Einstellung eines Motors am Motorprüfstand
- Verbrauch wird durch Differenz des Kraftstoffgewichts vor/nach Testzeit ermittelt
- CO₂-Gehalt: Absorption von Infrarotstrahlen

Probleme

- gute Individuen können versehentlich schlecht bewertet werden und umgekehrt
- Daher: kann man sich auf eine Bewertung verlassen?
- Aber auch: Weg aus lokalen Optima ist leichter!

Verrauschte Bewertung

Standardtechnik

- mit zunehmender Anzahl an wiederholten Bewertungen nähert sich der Mittelwert der objektiven Güte an
- bei K Bewertungen und Abweichung σ bei einer Bewertung: neue Abweichung $\frac{\sigma}{\sqrt{K}}$

Verrauschte Bewertung

Involvierte Kosten

- $Kosten = (KostVerw + KostEval \cdot K) \cdot \mu \cdot MaxGen$
- K groß und μ klein, oder anders rum?
- Falls $KostVerw > KostEval$: großes K
- GA mit $KostVerw \ll KostEval$: großes μ
 - Grund: Rekombination und verteilte Information in Schemata
⇒ einzelne Fehlbewertungen fallen nicht in's Gewicht
 - Auch: bei ES mit globaler arithmetischer Rekombination

Verrauschte Bewertung

Selektive Mehrfachauswertung

- häufig: es ist nicht genügend Zeit vorhanden
- Idee: für (μ, λ) -Selektion nur „notwendige“ Bewertungen durchführen
- zunächst wenige Bewertungen für alle Individuen
- dann paarweiser Vergleich der Individuen durch statistische Tests
- aus der resultierenden partiellen Ordnung ergeben sich die noch kritischen Individuen

Rauschen: Selbstanpassung

Problem

- auch der Selbstanpassungsmechanismus kann durch verrauschte Bewertung gestört werden

Vergleich der Techniken

- ES: Multiplikativ mit Exponentialfunktion
- EP: Additiv
- sonst: ES ist überlegen
- im Verrauschten: EP-Regel kann besser (d.h. weniger anfällig) sein

Rauschen: Selbstanpassung

Kappa-Ka-Methode

- Idee: Änderungen an Strategie- und Objektvariablen nur gedämpft weitergeben
- Annahme: die Problemlandschaft ist eher glatt

$$\widehat{B}.S \leftarrow A.S \cdot \left(\exp \left(\frac{1}{\sqrt{I}} \cdot u \right) \right)^{\kappa}$$

$$\widehat{B}_i \leftarrow A_i + \widehat{B}.S \cdot k \cdot u_i$$

$$B.S \leftarrow A.S \cdot \exp \left(\frac{1}{\sqrt{I}} \cdot u \right)$$

$$B_i \leftarrow A_i + \widehat{B}.S \cdot u_i.$$

Stabile Lösungen

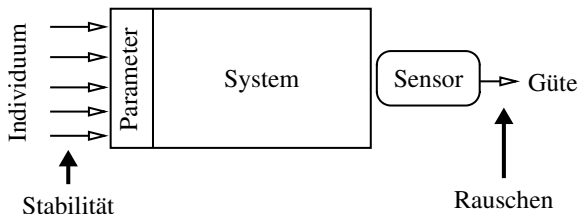
Beispiel

- Maschinenbelegungsplan in einer (manuell operierenden) Stuhlfabrik
- leichte Variaten in Abarbeitungsreihenfolge sind üblich
- Stabil: dies hat keinen Einfluss auf den Durchsatz

Unterschied zum Rauschen

- Rauschen: Abweichungen im beobachtbaren Ergebnis
- Stabilität: Abweichungen bei der Einstellung der Parameterwerte

Stabile Lösungen



Technik

- künstliches Rauschen
- Werte des Individuums werden leicht variiert
- Bewertung durch Mittelung der beobachteten Gütwerte

Stabile Lösungen

Problem

- gemittelte Gütwerte sind nur eine bedingte Lösung
- große positive und negative Abweichungen sind ergebnisgleich zu kleinen Abweichungen

Weitere Verbesserung

- zusätzlich: Varianz der Gütwerte berücksichtigen
- Optimierung mit Mehrzielverfahren

Zeitaufwändige Bewertung

Beispiel

- Kalibrierung eines Motorsteuergeräts am Prüfstand
- Motor muss einschwingen vor der Messung
- für aussagekräftige Messwerte muss er einige Zeit laufen

Problem

- Widerspruch zwischen verfügbarer Zeit und benötigter Anzahl an Auswertungen

Zeitaufwändige Bewertung

Technik

- einige Bewertungen nur schätzen
- Modellierung der Gütelandschaft mit neuronalen Netzen, RBF-Netzen, polynomielle Regression...

Vorgehensweise

- manuell iterieren: Optimierung auf dem Modell, später am realen System überprüfen
- besser:
 - grundsätzlich: auf Modell
 - „echte“ Bewertung nur an kritischen Punkten
 - Modell wird langsam verbessert

Bewertung durch Testfälle

Beispiel

- Ampelschaltung soll optimiert werden
- Daten für verschiedene Verkehrssituationen stehen in Simulationen zur Verfügung

Intuitiver Ansatz

- alle Testfälle heranziehen
- Mittel-/Maximalwert als Güte benutzen
- aber: kostspielig

Bewertung durch Testfälle

Koevolution

- Population an Lösungskandidaten vs. Population an Testfällen
- Güte $x \in [0, 1]$ wird als $1 - x$ für den Testfälle gewertet
- akkumulierende Bewertung in beiden Populationen
- jeweils fitnessproportionale Selektion
- Resultat: schwierige Testfälle werden bevorzugt
- optional: Evolution der Testfälle
- erwünscht: Wettrüsten „arms race“, aber: oft nur Schweinezyklus

KOEVLUTIONÄRER-ALGORITHMUS(Zielfunktion F)

```

1   $t \leftarrow 0$ 
2   $P^{(L)}(t) \leftarrow$  erzeuge Population mit Lösungskandidaten
3   $P^{(T)}(t) \leftarrow$  erzeuge Population mit Testfällen
4  bewerte alle Individuen in  $P^{(L)}(t)$  wie unten  $\nu$  Mal
5  bewerte alle Individuen in  $P^{(T)}(t)$  wie unten  $\nu$  Mal
6  while Terminierungsbedingung nicht erfüllt
7  do  $\ulcorner$  for  $i \leftarrow 1, \dots, \nu$  (Stichprobengröße)
8      do  $\ulcorner A^{(L)} \leftarrow$  selektiere ein Individuum aus  $P^{(L)}(t)$ 
9           $A^{(T)} \leftarrow$  selektiere einen Testfall aus  $P^{(T)}(t)$ 
10          $x \leftarrow F(A^{(L)}, A^{(T)})$ 
11         beziehe  $x$  in Güte von  $A^{(L)}$  mit ein
12          $\perp$  beziehe  $1 - x$  in Güte von  $A^{(T)}$  mit ein
13          $B^{(L)} \leftarrow$  erzeuge neues Individuum aus  $P^{(L)}(t)$ 
14         bewerte  $B^{(L)}$  wie oben  $\nu$  Mal
15          $t \leftarrow t + 1$ 
16          $P^{(L)}(t) \leftarrow$  ersetze schlechtestes Individuum in  $P^{(L)}(t - 1)$  durch  $B^{(L)}$ 
17         ( $B^{(T)} \leftarrow$  erzeuge Individuum aus  $P^{(T)}(t - 1)$ )
18         (bewerte  $B^{(T)}$  wie oben  $\nu$  Mal)
19          $\perp$  ( $P^{(T)}(t) \leftarrow$  ersetze schlechtestes Individuum in  $P^{(T)}(t - 1)$  durch  $B^{(T)}$ )
20 return bestes Individuum aus  $P^{(L)}(t)$ 

```

Bewertung von Spielstrategien

Beispiel

Schach, Dame, Go, ...

Darstellung von Spielstrategien

- Bewertung der aktuellen Spielsituation durch
 - Funktionen als Syntaxbäume
 - neuronale Netze
 - Regeln
- im Rahmen einer Minimax-Suche

Bewertung von Spielstrategien

Erste Idee

- gegen einen guten Spieler bewerten
- aber: konzentriert sich zu stark auf dessen Schwächen
- abweichendes Spielverhalten ist unbekannt

Turnierselektion

- Bewertung durch gegeneinander spielen im Rahmen der Umwelt-Turnierselektion
- „single population coevolution“
- grundsätzlich: Strategien sind schwer objektiv bewertbar