

Anpassung vs. Optimierung

- *Optimierung* -

Optimierung

Experimentelle Optimierung vs. mathematische (modell-basierte) Optimierung

- keine explizite Gütefunktion
- kein *mathematisches* Modell aber das Experiment kann wiederum ein Modell sein
- Störungen sind inhärent
- minimale Stabilitätsanforderungen müssen oftmals erfüllt sein
- Gütefunktion ist über das Modell gegeben, muss nicht deterministisch sein:
 $F(x) = F(x_1, x_2, \dots, x_n) \rightarrow \text{Extremum}$
- Minimierung ist prinzipiell analog zu Maximierung:
 $\max\{F(x)\} = \min\{-F(x)\}$

Parameteroptimierung

- alle x_i sind Skalare
 - gilt $x_i \in \mathbb{R}$, $\forall i$: kontinuierliches Optimierungsproblem
 - gilt $x_i \in \mathbb{N}$, $\forall i$: diskretes oder kombinatorisches Optimierungsproblem
 - sonst: gemischtes Optimierungsproblem
- optimaler Punkt im n -dimensionalen metrischen Raum (z.B. Euklidisch) wird gesucht

Bsp: Designoptimierung, Handlungsreisendenproblem, Maschinenbelegungspläne

Optimierung

Funktionsoptimierung

- optimale Trajektorien im Funktionenraum (Banach oder Hilbertraum) werden gesucht
- die Variablen x_i sind Funktionen, diese selbst wieder von mehreren Parametern abhängen, F ist somit ein *Gütefunktional*
- Optimierung nutzt die Variationsrechnung

Bsp: Finde die Kurve, die eine Punktmasse zwischen zwei Punkten unter Einfluss der Gravitation in kürzester Zeit beschreibt; Optimale Kontrollfunktionen

Strukturoptimierung

- *es gibt keine einheitliche Definition*
- gesucht werden optimale Strukturen, d.h. topologische Anordnungen von Teilelementen zu einem Ganzen
- auch wenn Strukturen oftmals *parametrisiert* werden können, operiert die Gütefunktion auf dem Strukturraum
- ähnlich wie bei der Funktionsoptimierung sind die Strukturen oftmals selbst wieder von Parametern abhängig

Bsp: Optimierung neuronaler Netze

Parameteroptimierung: Optima

Globales Minimum

Sei $F: M \subset \mathbb{R}^n \rightarrow \mathbb{R}$ mit $M \neq \emptyset$. Für $x^* \in M$ heisst der Wert $F^* := F(x^*) > -\infty$ das globale Minimum der Gütefunktion F , wenn für alle $x \in M$ gilt $F(x^*) \leq F(x)$.

Lokales Minimum

Für $x^+ \in M$ heisst $F^+ := F(x^+)$ lokales Minimum der Gütefunktion F , wenn es eine ε -Umgebung $U_\varepsilon(x^+) := \{x \in M \mid |x - x^+| < \varepsilon\}$ gibt, sodass für alle $x \in U_\varepsilon(x^+)$ gilt: $F(x^+) \leq F(x)$.

Unimodal

Eine Gütefunktion heisst unimodal, wenn sie genau ein lokales Minimum besitzt, sonst heisst sie multi-modal.

Optimierungsverfahren

Direkte (numerische) Optimierung

- direkte oder numerische Methoden sind solche die das Optimum iterativ (schrittweise) approximieren
- in jedem Schritt wird der Funktionswert verbessert, sonst spricht man von *trial and error* Methoden
- die zu optimierende Funktion muss nicht analytisch vorliegen, ein Simulationsmodell oder ein variabler experimenteller Aufbau sind ausreichend

Indirekte (analytische) Optimierung

- bei indirekten bzw. analytischen Methoden wird das Optimum in einem Schritt erreicht
- die Funktion muss in analytischer Form vorliegen, dann gilt als
 - notwendiges Kriterium: $\nabla F(x^*) = 0$
 - hinreichendes Kriterium: betrachte alle n Determinanten der Hesse Matrix H
 - $k=1, \dots, n: \det H_k > 0$: lokales Minimum
 - $k=1, \dots, n: \det H_k \cdot (-1)^k > 0$: lokales Maximum

Globale Zufallssuche

$t:=1$

wähle eine Dichtefunktion p_1 auf M , solange kein Abbruch wiederhole:

erzeuge Zufallspunkte $\{x_t^{(1)}, \dots, x_t^{(N(t))}\}$

berechne $\{F(x_t^{(1)}), \dots, F(x_t^{(N(t))})\}$

erzeuge p_{t+1} gemäss einer definierten Regel

$t=t+1$

- die globale Zufallssuche garantiert *nicht* die globale Konvergenz mit Wahrscheinlichkeit 1, da optimale Lösungen immer wieder verlassen werden können → Elitist
- evolutionäre Algorithmen gehören (auch wenn keine Dichtefunktion explizit angepasst wird) zu dieser Klasse von Algorithmen
- die direkte Anpassung der Dichtefunktion aus der Historie der Suche hat in den letzten Jahren im Bereich der EA großes Interesse gefunden
- die Annahme Gausscher Wahrscheinlichkeitsdichten hat sich dabei (unter Einschränkungen) als auch theoretisch sehr gut herausgestellt
- Monte-Carlo Verfahren: Gleichverteilung

Gradientenverfahren

- iterative Suche mit Richtung $\vec{s}(t)$ und Schrittweite $\sigma(t)$

$$\vec{x}(t+1) = \vec{x}(t) + \sigma(t) \vec{s}(t)$$

- Methode des steilsten Abstieges (steepest decent)

$$\vec{s}(t) = -\nabla f(\vec{x}(t))$$

- Newton Verfahren :Hinzunahme der Information der zweiten Ableitung

$$\vec{s}(t) = -\mathcal{H}^{-1}(f(\vec{x}(t))) \nabla f(\vec{x}(t))$$

- das Newton Verfahren ist sehr schnell, jedoch oftmals (numerisch) instabil aufgrund der Berechnung der Inversen der Hesse-Matrix
- Algorithmen, die die Inverse der Hesse-Matrix iterativ berechnen nennt man quasi-Newton Verfahren, z.B. BFGS-Algorithmus
- bei konjugierten Gradientenverfahren setzt sich die aktuelle Suchrichtung \vec{s}_t aus einer exponentiell gedämpften Summe vorheriger Suchschritte zusammen
- Gradientenverfahren sind nur bei unimodalen Gütefunktionen globale Suchverfahren

Optimierung unter Randbedingungen

die Parameter der Optimierungsaufgabe:

$$F(x) = F(x_1, x_2, \dots, x_n) \rightarrow \text{Extremum}$$

sind durch Randbedingungen eingeschränkt:

$$G_j(x_1, x_2, \dots, x_n) \begin{cases} \leq \\ = \\ \geq \end{cases} 0; \quad j=1, \dots, m$$

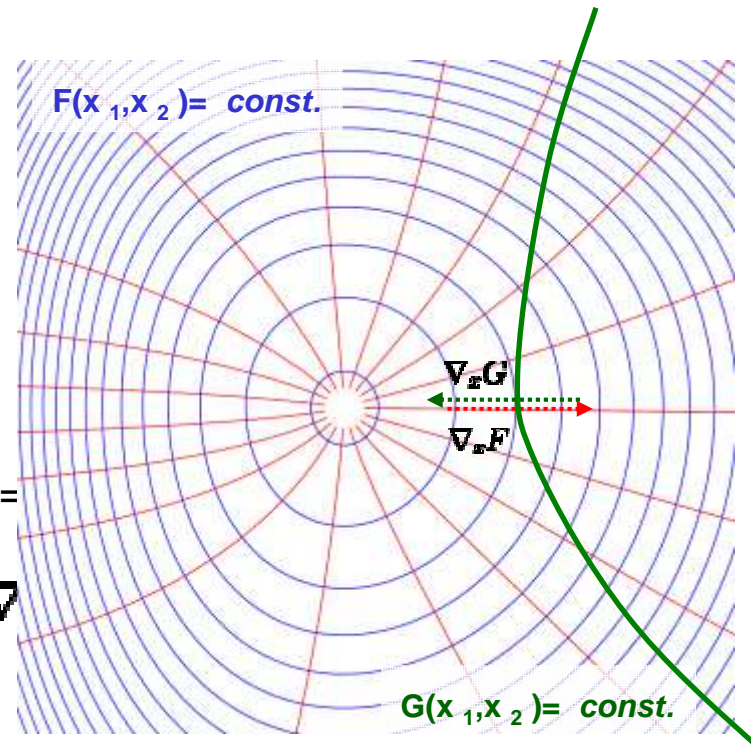
- Suche das Minimum der Funktion

$$H(x_1, \dots, x_n, \lambda) = F(x)$$

- λ ist der Lagrange Multiplikator

- das Optimum (x^*, λ^*) erfüllt direkt $G(x^*) =$

$$\nabla H(x, \lambda) = \begin{pmatrix} \nabla F \\ \nabla G \end{pmatrix}$$



Optimierung unter Randbedingungen

die Parameter der Optimierungsaufgabe:

$$F(x) = F(x_1, x_2, \dots, x_n) \rightarrow \text{Extremum}$$

sind durch Randbedingungen eingeschränkt:

$$G_j(x_1, x_2, \dots, x_n) \begin{cases} \leq \\ = \\ \geq \end{cases} 0; \quad j=1, \dots, m$$

- Suche das Minimum der Funktion

$$H(x_1, \dots, x_n, \lambda) = F(x_1, \dots, x_n) + \lambda G(x_1, \dots, x_n)$$

- λ ist der Lagrange Multiplikator
- das Optimum (x^*, λ^*) erfüllt direkt $G(x^*) = 0$, denn

$$\nabla H(x, \lambda) = \begin{pmatrix} \nabla_x F + \lambda \nabla_x G \\ G(x) \end{pmatrix} \stackrel{!}{=}_{(x, \lambda) = (x^*, \lambda^*)} 0$$

Optimierung unter Randbedingungen

Randbedingungen in der Optimierung

- *Bestrafungsterme (penaltyterm)*: Genügt eine Lösung einer der Randbedingungen nicht, so wird zur Qualität eine hohe Zahl hinzuaddiert (bei Minimierung)
 - statische Bestrafungsterme (konstante Zahl)
 - entfernungsbasierte Terme (wie weit liegt die Lösung von einer gültigen Lösung weg)
 - dynamische Terme (die Stärke der Bestrafung steigt mit Länge der Optimierung)
 - adaptive Terme (die Stärke der Bestrafung hängt vom Zustand der Optimierung ab, z.B. von der Güte der Lösung (oder bei populationsbasierten Suchverfahren von der mittleren Güte oder der Diversität))

- Bestrafungsterme sind besonders effektiv, wenn F und G_j entkoppelt sind

Optimierung unter Randbedingungen

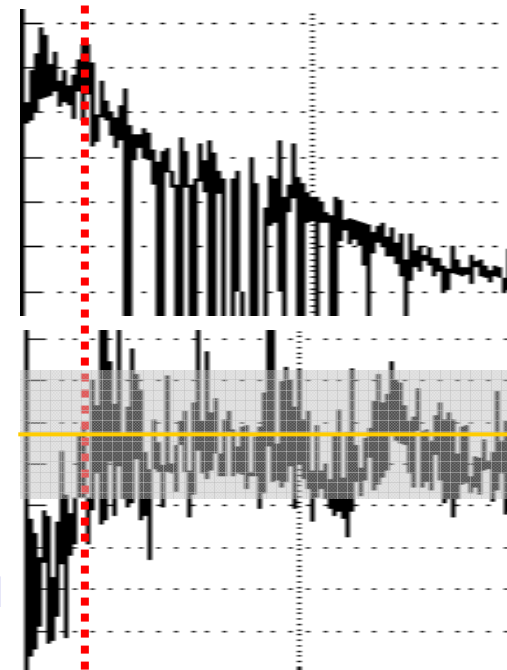
Beispiel: Bestrafungsterm in der Designoptimierung

$$f(x) = \eta_1 f_1(\omega) + \eta_2 f_2(\alpha_2) + \eta_3 f_3(x_{\min}) + \eta_4 f_4(x_{\max})$$

Druckverlust (Güte)
Auslasswinkel (Randbedingung)
Geometrische Randbedingung

Druckverlust

Auslasswinkel

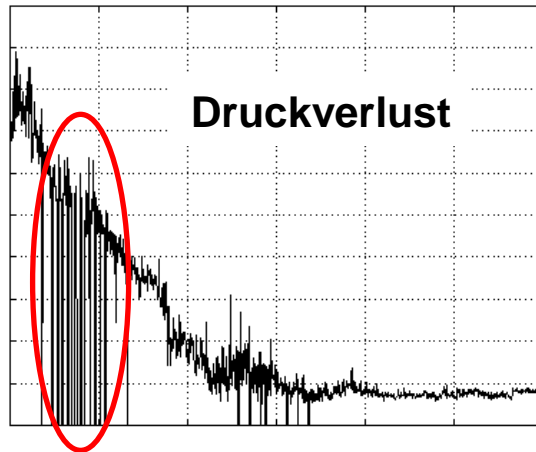


Randbedingungen in der Optimierung

- *Reparaturalgorithmen*: ungültige Lösungen werden auf gültige Lösungen abgebildet, z.B.: Belegungsprobleme (entfernen von Komponenten); Spiegelung von Lösungen bei kontinuierlichen Problemen (siehe auch Zusammenhang zum Lernen)
- *Multi-kriterielle Optimierung - Paretomenge*

Optimierung mit ungültigen Lösungen

Bsp: Designoptimierung

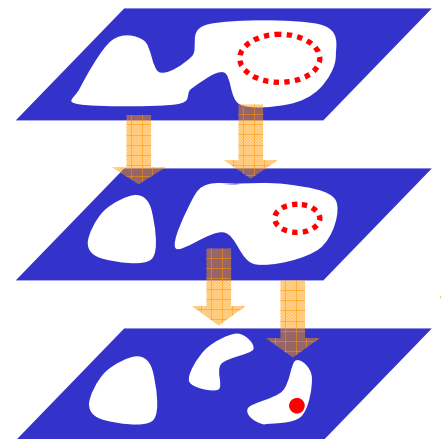


Das Modell liefert für bestimmte Parameterwerte keine verlässlichen Aussagen

Problem, wenn

- die tatsächliche Güte der Lösungen ist vom zugehörigen Modellwert unabhängig
- die kritischen Parameterwerte treten blockhaft auf
- die Verlässlichkeit der Modellaussagen ist nicht eindeutig zu bestimmen

Inseln gültiger Lösungen



Hierarchisierung

- unterschiedliche Repräsentationen
- unterschiedliche Modelle
- unterschiedliche externe Parameter

Optimierung unter Störungen

Optimierungsprobleme in der Anwendung sind oftmals (zumeist) verrauscht

- *Implizite Störungen* in der Evaluation des Systems
 - Genauigkeit bei Simulation, z.B. computational fluid dynamics
 - Reproduzierbarkeit bei Experimenten oder Simulationen (Neuronale Netze)
 - subjektive Bewertung, Mensch-Maschine Interaktion

- *Explizite Störungen* als Methoden um Robustheit zu erhöhen

Definition der Robustheit

- Minimierung der statistischen Momente
- Minimierung der Wahrscheinlichkeit, dass Qualitätswerte unter (über) einer definierten Schwelle liegen

- Robustheit gegenüber Variationen von Objektparametern
z.B. Fertigungstoleranzen
- Robustheit gegenüber Variationen von externen Randbedingungen,
z.B. Anströmungswinkel, Einströmgeschwindigkeit

Optimierung unter Störungen - additiver Störterm

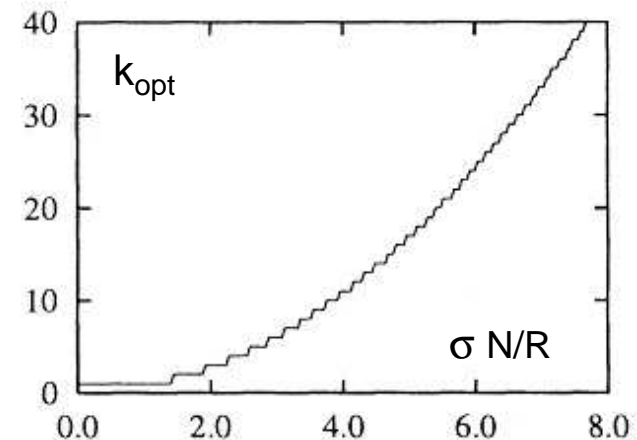
Einfachstes Modell für verrauschte Gütefunktionen: $F_{nsp}(\mathbf{x}) := Q_{sp}(\mathbf{x}) + \delta$,
additive Zufallsvariable mit Gaußscher Wahrscheinlichkeitsdichtefunktion

- alle theoretischen Untersuchungen zu ES mit additivem Störterm basieren auf linearen bzw. quadratischen Fitnessmodellen mit $N \rightarrow \infty$
- höhere Varianz in der Population (höhere Diversität) macht die Strategie robuster gegenüber Störungen
- globale *intermediate* Rekombination ($\mu/\mu, \lambda$)-ES zeigt signifikant bessere Resultate als andere Evolutionsstrategien
- es gibt einen Residuumsabstand vom Optimum, der von der Varianz der Störung abhängt und über das Stabilitätskriterium, welches lokale Konvergenz gewährleistet, berechnet werden kann:

Optimierung unter Störungen - additiver Störterm (cont'd)

Einfachstes Modell für verrauschte Gütefunktionen: $F(x) + \delta$, $\delta \sim N(0, \sigma_\delta^2)$
additive Zufallsvariable mit Gaußscher Wahrscheinlichkeitsdichtefunktion

- optimale Anzahl k_{opt} von Fitnessauswertungen, über die in jedem Zeitschritt gemittelt werden als Funktion der normalisierten Stärke der Störung für die (1+1)-ES (ohne Neuberechnung der Elter)



- für $(\mu/\mu, \lambda)$ -ES ist eine grössere Population ein effizienteres Mittel mit Störungen umzugehen, als Fitnessauswertungen über mehrere Samples zu mitteln (im Gegensatz zu $(1, \lambda)$ -ES)

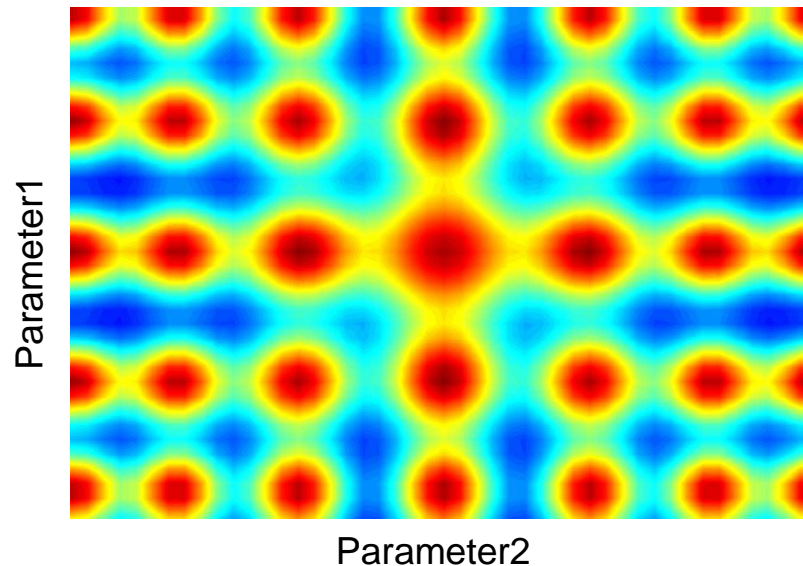
Optimierung unter Störungen - Multi-modale Funktion

- Unter der Annahme von linearer (proportionaler) Selektion kann man zeigen, dass im Schematheorem die effektive Fitnessfunktion relevant ist:

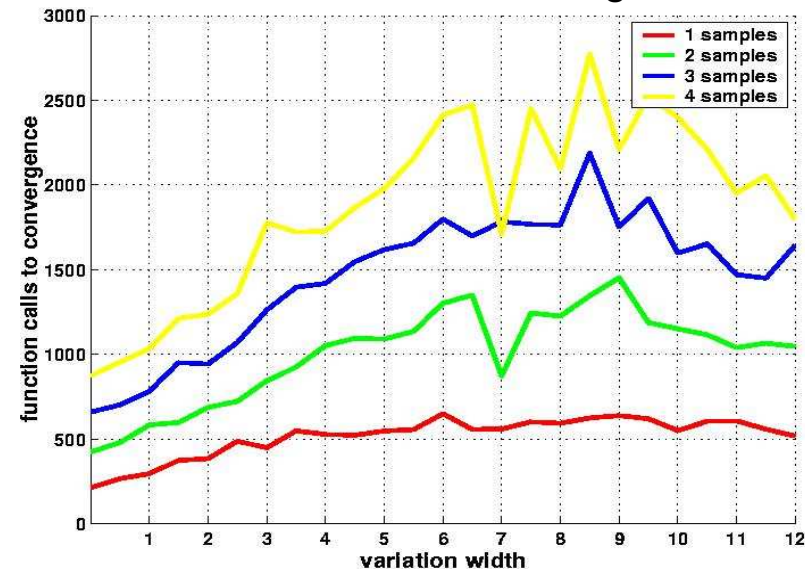
➔ *explizites Mitteln ist nicht notwendig*

- Multi-modale Funktion: Optimierer haben unterschiedliche Robustheit (Varianz)
- Explizites Mitteln ist bezüglich der Anzahl der Funktionsevaluierungen ineffizient
- Inhärente Mittelung in der Population (Rekombination) bzw. implizit über der Zeit erscheint effizienter

ContourPlot der 2D-Testfunktion



Funktionsevaluierungen



Metamodelle in der Optimierung

- Motivation

- die Gütefunktion liegt nicht in analytischer Form vor und Fitnessbewertungen sind sehr zeitintensiv, Bsp.: CFD bzw. Experimente
- Beispiele für Metamodelle: Response Surface Methoden (Polynome erster/zweiter Ordnung), Neuronale Netze, et c.

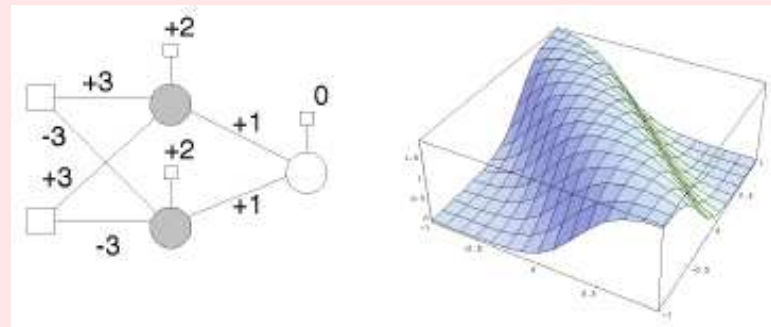
- Probleme

- Metamodelle können falsche Optima haben **Falsches Optimum**

einfaches neuronales Netzwerk besteht aus

- Verarbeitungseinheiten, den Neuronen
- gewichteten Verbindungen zwischen den Neuronen, den Synapsen

die Funktionalität hängt neben den Gewichten insbesondere von der Struktur der neuronalen Systeme ab



➔ evolutionäre Optimierung der Verbindungsstruktur

- Lösung: evolutionäre Kontrolle, d.h. Metamodelle werden mit der originalen Gütefunktion kombiniert

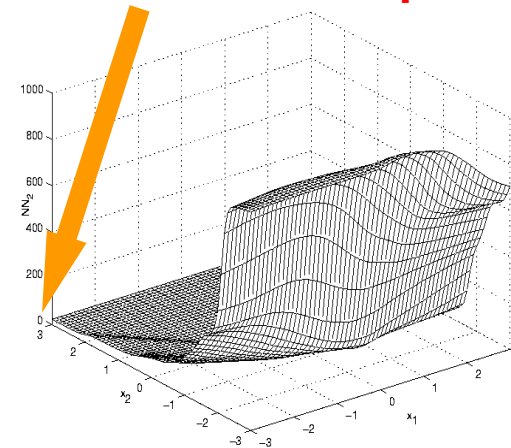
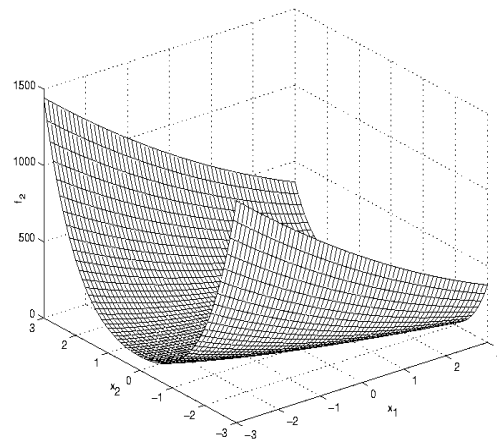
Metamodelle in der Optimierung

- Motivation

- die Gütefunktion liegt nicht in analytischer Form vor und Fitnessbewertungen sind sehr zeitintensiv, Bsp.: CFD bzw. Experimente
- Beispiele für Metamodelle: Response Surface Methoden (Polynome erster/zweiter Ordnung), Neuronale Netze, etc.

- Probleme

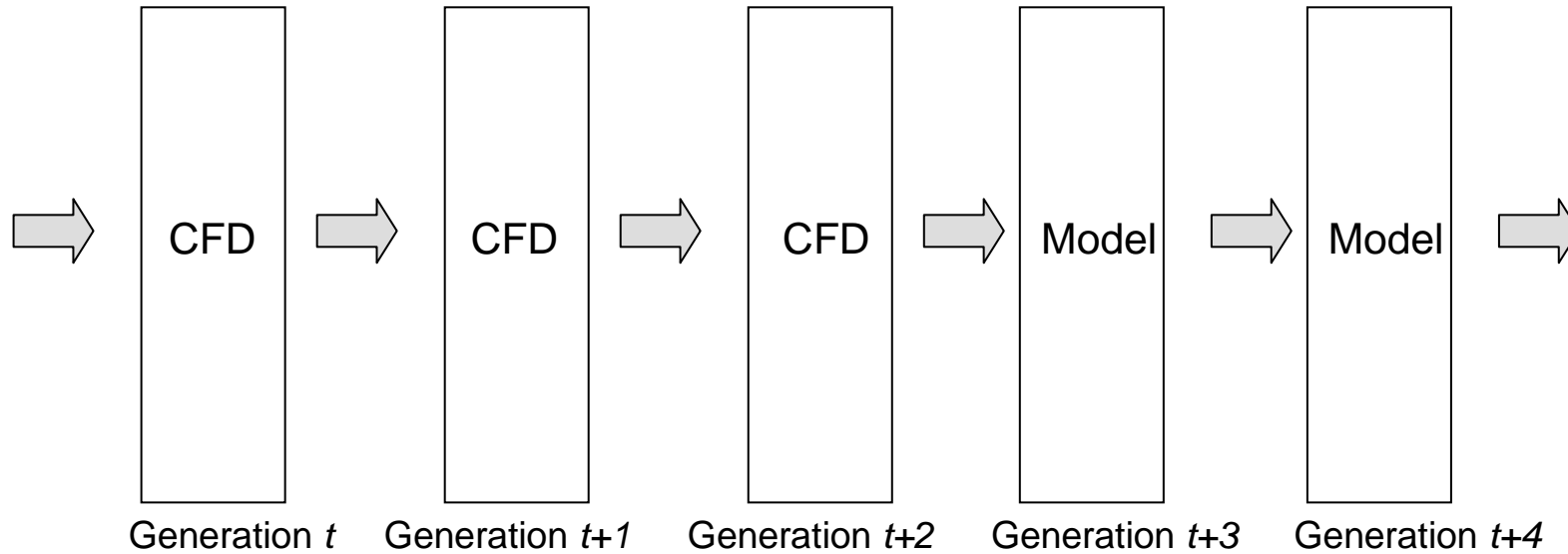
- Metamodelle können falsche Optima haben **Falsches Optimum**



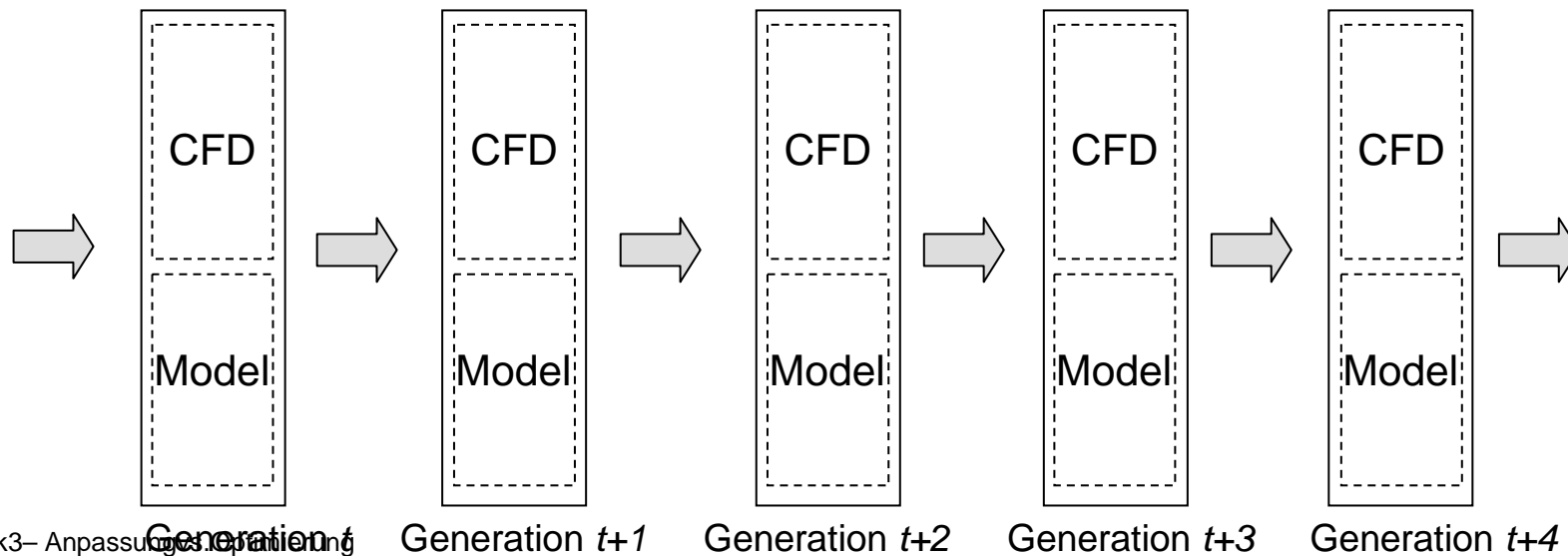
- Lösung: evolutionäre Kontrolle, d.h. Metamodelle werden mit der originalen Gütefunktion kombiniert

Generationen- vs. Individuell-basierte Anpassung

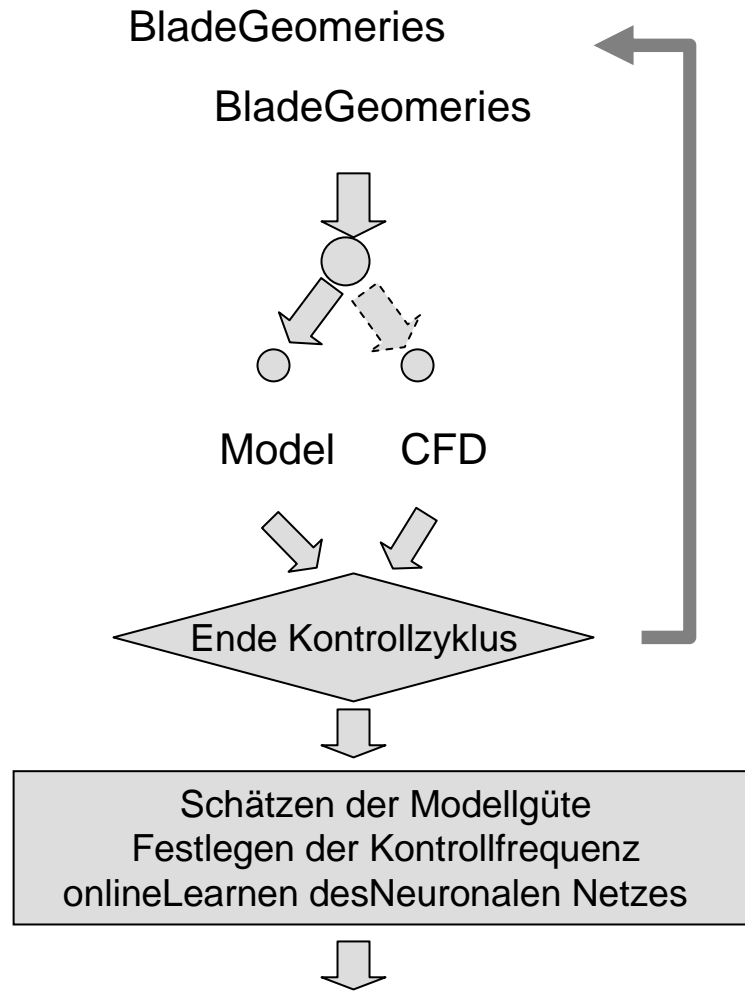
- Generationen-basierte Anpassung



- Individuell-basierte Anpassung



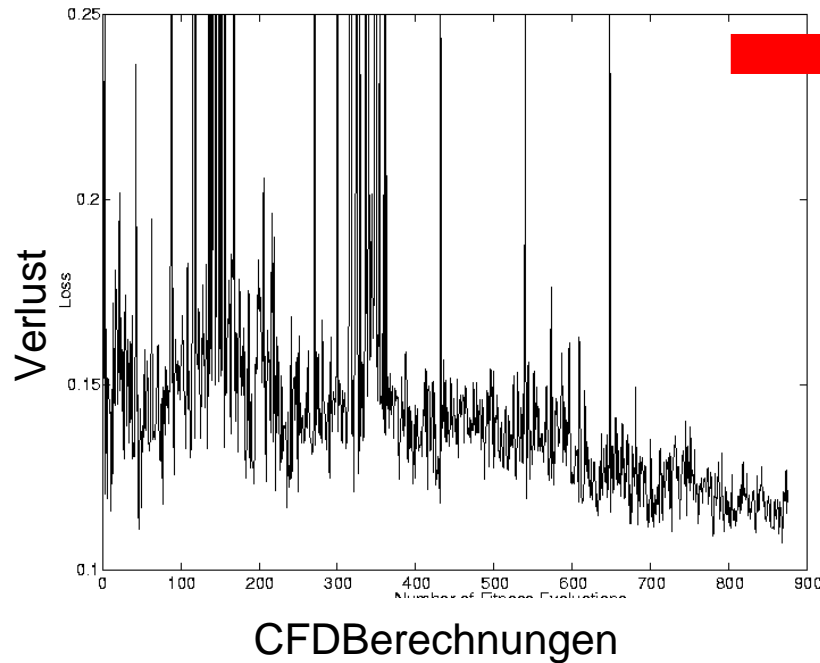
Neuronale Netze als Metamodelle inEAs



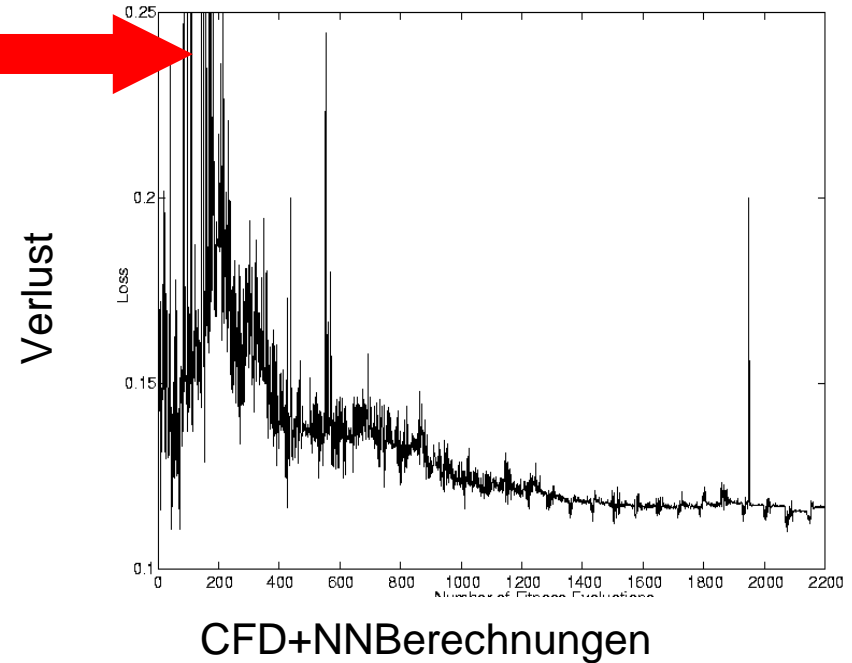
- on-lineLernen der neuronalen Netze muss schnell und effizient sein, d.h. Kombination mit offlineStruktur-optimierung ist sinnvoll
- dieAdaptation der Kontrollfrequenz ist entscheidend für das richtige Gleichgewicht zwischen korrekter Konvergenz und Rechenaufwand
- Möglichkeit der Ensemblebildung zur Güteschätzung

Anwendungsbeispiel: Optimierung einer Turbinenschaufe I

Optimierung ohne NNMetamodell



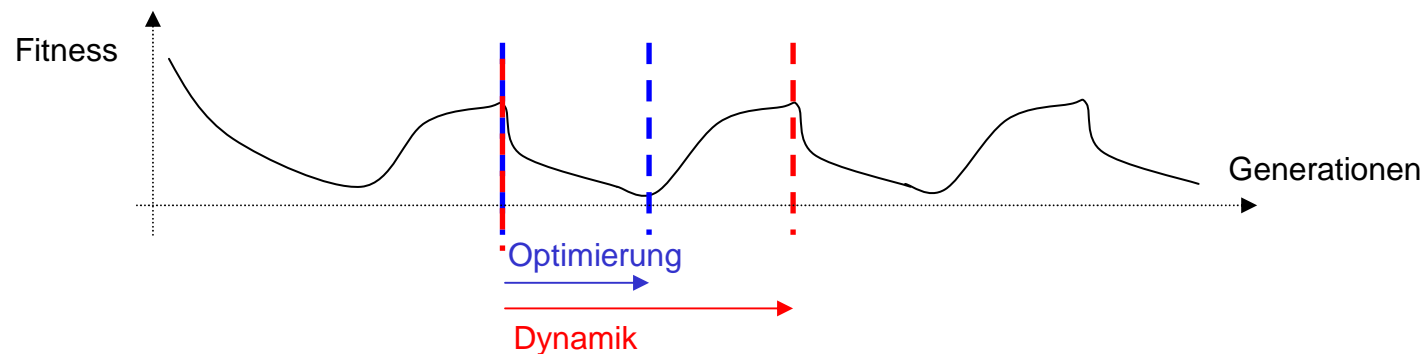
Optimierung mit NNMetamodell



- Besseres Optimierungsergebnis
- Weniger Aufrufe des rechenintensiven Computational Fluid Dynamics Prg.

Optimierung von dynamischen Gütefunktionen

- dynamische Gütefunktion heisst die Funktion ändert sich mit der Zeit bzw. der Generationenzahl (deterministisch oder stochastisch),
Bsp: Veränderung von Präzision (Mechanik), Verbrauchsdurchschnitten, etc.
- Optimierung mit Störungen ist ein Spezialfall dynamischer Gütefunktionen, bei denen im Allgemeinen trotz der Veränderlichkeit nur ein Optimum gesucht wird
- *Annahme:* Zeitskala der Dynamik ist grösser (langsamer) als Zeitskala der Optimierung sonst repräsentiert die Dynamik eine Störung - Rauschen



- Veränderungen können kontinuierlich akkumulierend sein oder punktuell (Phasenübergang)
- Zusätzlich zu schneller Konvergenz auf das globale Optimum ist hohe "Reaktionsgeschwindigkeit" auf die dynamische Gütefunktion notwendig

Optimierung von dynamischen Gütefunktionen: Ansätze

Erhalten hoher Flexibilität

- hohe Diversität innerhalb der Population, z.B. Prinzip des Fitnesssharing
Maximierung der Entropie als Randbedingung
- untere Grenze für die Varianz bei Selbstadaptation von Schrittweiten

Organisation eines Gedächtnisses

- Individuumebene: redundante Repräsentationen, duplierte Repräsentationen
- Populationsebene: Dynamische Subpopulationen, "Aufbewahren" von besten Lösungen für eine gewisse Generationenzahl

Einführen von Variabilität auf kürzerer Zeitskala als die Evolution

- Evolution von variablen Strukturen, die sich auf kürzerer Zeitskala anpassen können
- Evolution von Regelsystemen
- Evolution von neuronalen Systemen - Evolution von Lernenden



Ist die Evolution ein Optimierer?

Evolution findet in dynamischen und stochastischen Umwelten statt

Evolution ist inherent iterativ

➔ Evolution ist kein Optimierungsverfahren im Standardsinne

Aber:

unter Beachtung der mannigfaltigen Randbedingungen der Lösungen

- Baubarkeit (im Sinne von aufeinander aufbauen)
- Robustheit
- Evolvierbarkeit
- ...

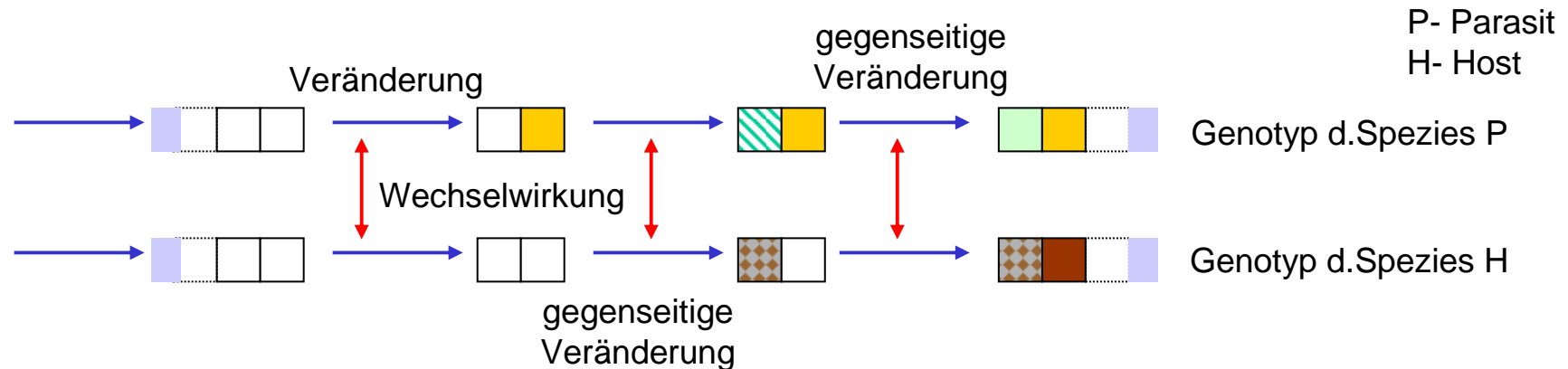
➔ produziert die Evolution "optimal" an die Randbedingungen und die Umwelt angepasste Lösungen

Anpassung vs. Optimierung

- *Co-evolution* -

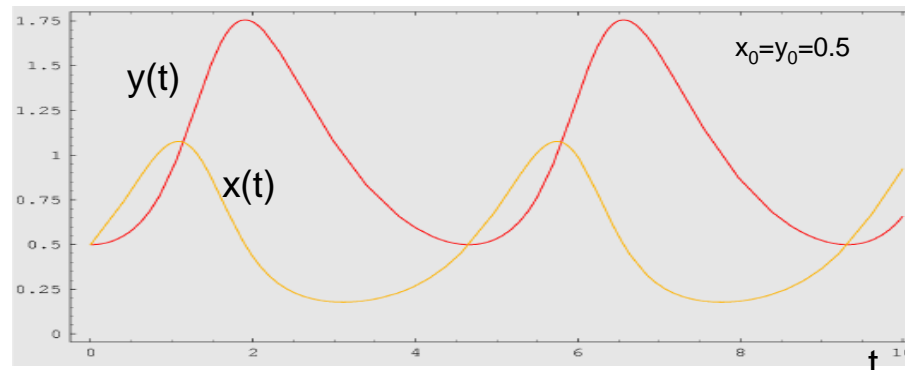
Coevolution

Gegenseitige genetische Veränderungen inwechselwirken den Spezies aufgrund natürlicher Selektion, die eine Spezies auf die andere ausübt, bezeichnet man als *Coevolution*



Coevolution ist dynamische Optimierung (Anpassung) mit Rückkopplung, d.h. die eigenen Veränderungen beeinflussen die Veränderungen der Fitnesslandschaft

Bsp: Räuber-Beute Modell



Coevolution - Beispiel aus der Biologie



Kolibri

(ca. 320 Spezies in Nord- und Südamerika)

- stammt vom Mauersegler ab, die einen kurzen Schnabel haben und nicht schweben können



Mimulus cardinalis



Mimulus whitneyi



Polemoniaceae



Polemonium viscosum

Hypothese zur coevolutionären Entwicklung von Eigenschaften

Kolibri

- lernt Nektarvolumen mit Blumenfarbe zu assoziieren
- langer Schnabel entwickelt sich
- Wechsel von Insekten zu Nektar
- schweben entwickelt sich

Blume

- rote Farbe entwickelt sich (schwieriger für Bienen zu lokalisieren)
- lange runde Krone entwickelt sich
- höhere Nektarproduktion
- "Landeplattform" degeneriert

Coevolution in Evolutionären Algorithmen

- co-evolutionäre Algorithmen benötigen im allgemeinen eine *relative* bzw. *kompetitive* Fitnessfunktion und häufig mehrere Population bzw. mehrere Spezies
- bei einer relativen (kompetitiven) Fitnessfunktion wird die Güte einzelner Individuen durch Vergleich (Kompetition) mit anderen Lösungen bestimmt
- der Vergleich kann direkt sein

Bsp: Tournaments, evolvieren von Spielstrategien (Dame, Backgammon)

Iterated Prisoners Dilemma

- beste Strategie: tit-for-tat
- Erweiterung: stochastisch
- Erweiterung: multi-player

	A	B	
	cooperate	cooperate	3,3
	cooperate	defect	0,5
	defect	cooperate	5,0
	defect	defect	1,1

- der Vergleich kann indirekt sein

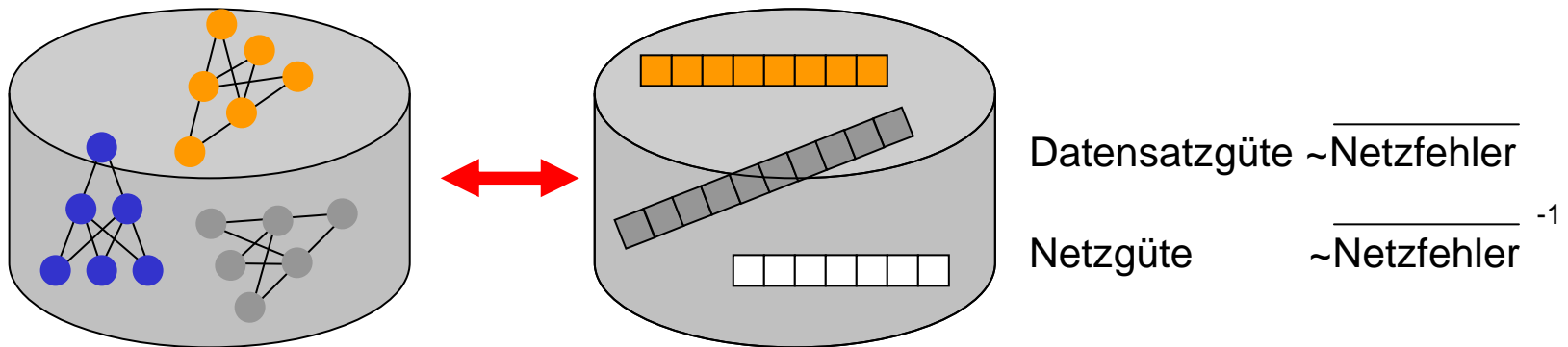
Bsp: Tierra System: Bei der Bestimmung der Güte spielen limitierte Ressourcen eine Rolle, die Interaktion kann sogar *parasitär* sein

- unterschiedliche Vergleichsmuster führen zu unterschiedlichen Dynamiken, z.B. *jeder-mit-jedem, jeder-mit-besten, jeder-mit-Zufall spool*

Coevolution in Evolutionären Algorithmen (cont'd)

- Multi-Spezies coevolutionäre Algorithmen CoEA können bei *test-solution* (Paredis, 1996) Problemen genutzt werden

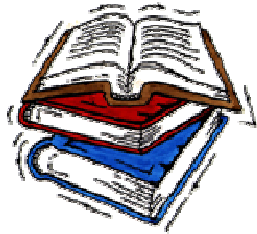
Bsp: Evolution von neuronalen Netzen zur Klassifikation bzw. Zeitreihenvorhersage



- das Prinzip der relativen Fitness kann auch zur Erhöhung der Robustheit bei stochastischen Fitnessauswertungen genutzt werden
- weitere Anwendungsgebiete sind multi-kriterielle Optimierungen (siehe Räuber/Beute Algorithmus), bzw. Optimierung unter Randbedingungen, *credit-assignment problem*
- CoEAs sind aufgrund der Bedingungen an die Gütefunktion bzw. an eine geeignete Aufteilung in Spezies zumeist schwieriger zu formulieren, aufgrund der "offenen" Evolution jedoch oftmals auch sehr erfolgreich
- aufgrund der komplexeren Dynamik gibt es sehr wenig Theorie zu CoEAs

Zusammenfassung

- Unterscheidung in experimentelle und mathematische Optimierung und Parameter-, Funktions- und Strukturoptimierung
- Evolutionäre Algorithmen gehören zur Klasse der globalen Zufallssuche - der direkten Optimierungsverfahren, (indirekte Verfahren berechnen das Optimum analytisch)
- Randbedingungen in der Optimierung werden klassisch durch Lagrangeparameter und bei EAs durch Bestrafungsterme berücksichtigt
- bei der Multikriteriellen Optimierung ist die Menge aller Pareto-Lösungen das Ziel der Optimierung, daher eignen sich populationsbasierte Suchverfahren besonders gut
- additive Störungen bewirken einen Residuumabstand zum Optimum, Störungen auf den Objektparametern können bei bestimmten Fitnessfunktionen zu rauschinduzierter Multimodalität führen. Direkte Mittelung der Fitnesswerte ist nur bei (1+1)-ES effizient, sonst sind grössere Populationen besser
- Metamodelle approximieren die tatsächliche Fitnesslandschaft und erlauben (besonders mit online-Anpassung) eine effizientere Suche
- gegenseitige genetische Veränderungen in Wechselwirkung der Spezies aufgrund natürlicher Selektion, die eine Spezies auf die andere ausübt, bezeichnet man als *Coevolution*



Weiterführende Literatur

- [1] H.-P.Schwefel, Evolution and Optimum Seeking.
- [2] Z.Michalewicz and D.B.Fogel, How to Solve It: Modern Heuristics.
- [3] T.Bäck and D.B.Fogel, Evolutionary Computation Land II. (*CookBook*)
- [4] J.Branke, Evolutionary Optimization in Dynamic Environments
- [5] D.V.Arnold, Noisy Optimization with Evolution Strategies
- [6] S.Kauffman, The Origins of Order.
- [7] D.J.Futuyama, Evolutionary Biologie.
- [8] K.Deb, Multi-objective Optimization Using Evolutionary Algorithms