

Anpassung vs. Optimierung

- *Optimierung* -

Optimierung

Experimentelle Optimierung vs. mathematische (modell-basierte) Optimierung

- | | |
|--|---|
| <ul style="list-style-type: none">▪ keine explizite Gütefunktion▪ kein <i>mathematisches</i> Modell aber das Experiment kann wiederum ein Modell sein▪ Störungen sind inhärent▪ minimale Stabilitätsanforderungen müssen oftmals erfüllt sein | <ul style="list-style-type: none">▪ 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 or kombinatorisches Optimierungsproblem
 - sonst: gemischtes Optimierungsproblem
- optimaler Punkt im n-dimensionalen metrischen Raum (z.B. Euklidisch) wird gesucht

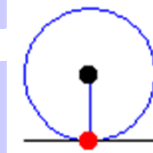
Bsp: Designoptimierung, Handlungsreisendenproblem, Maschinenbelegungspläne

Optimierung

Funktionsoptimierung

- optimale Trajektorien im Funktionenraum (Banach oder Hilbertraum) werden gesucht
- die Variablen x_i sind Funktionen, die 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 $F: M \subset \mathbb{R}^n \rightarrow \mathbb{R}$ und $M \neq \emptyset$. 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, so dass 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
 - $k = 1, \dots, n$: $\det^k H > 0$: lokales Minimum
 - $k = 1, \dots, n$: $\det^k H \cdot (-1)^k > 0$: lokales Maximum

 Lösen von linearen (nichtlinearen) Gleichungssystemen → *iterativ*

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:

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

Klassisch: Lagrange Multiplikatoren

- 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

die Parameter der Optimierungsaufgabe:

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

sind durch Randbedingungen eingeschränkt:

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

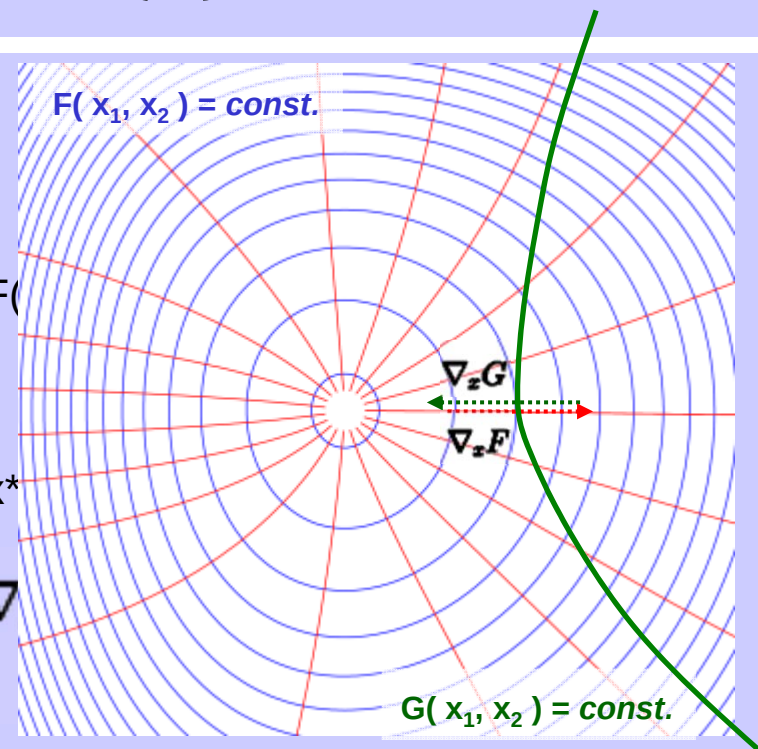
Klassisch: Lagrange Multiplikatoren

- Suche das Minimum der Funktion

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

- λ ist der Lagrange Multiplikator
- das Optimum (x^*, λ^*) erfüllt direkt

$$\nabla H(x, \lambda) = \begin{pmatrix} \nabla F(x) \\ \nabla G(x) - \lambda \end{pmatrix} = 0$$



Optimierung unter Randbedingungen

Randbedingungen in der Optimierung

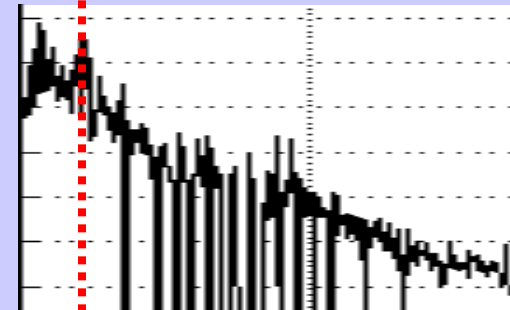
- *Bestrafungsterme (penalty term)*: Genügt eine Lösung einer der Randbedingungen nicht, so wird zur Qualität eine hohe Zahl hinzuaddiert (bei Minimierung)
 - statische Bestrafungsterme (konstante Zahl)
 - entfernungsbasierte Terme (wie weit liegt die Lösung von einer gültigen Lösung weg)
 - dynamische 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

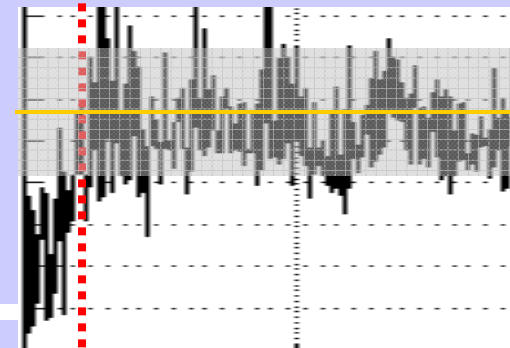
Beispiel: Bestrafungsterm in der
Designoptimierung

$$\begin{aligned} f(x) = & \eta_1 f_1(\omega) + \text{Druckverlust (Güte)} \\ & \eta_2 f_2(\alpha_2) + \text{Auslasswinkel (Randbedingung)} \\ & \eta_3 f_3(x_{\min}) + \eta_4 f_4(x_{\max}) \quad \text{Geometrische} \\ & \quad \quad \quad \text{Randbedingung} \end{aligned}$$

Druckverlust



Auslassswinkel

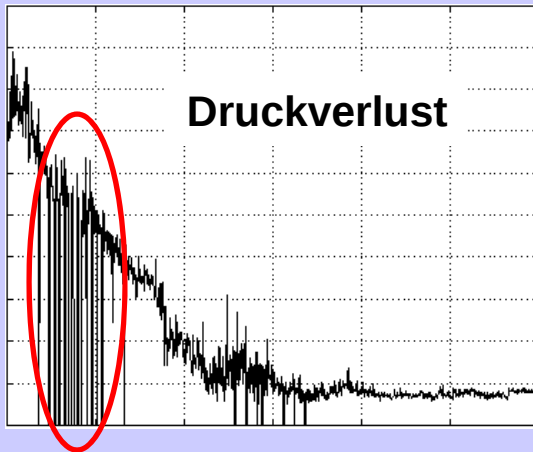


Randbedingungen in der Optimierung

- *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

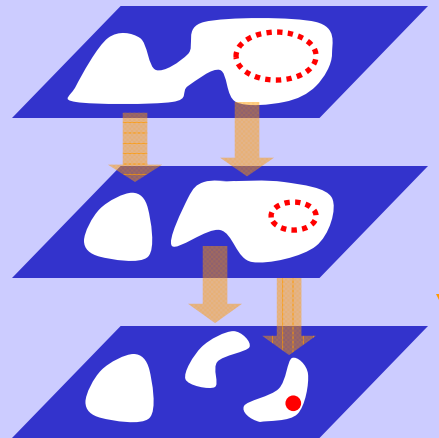


Das Model liefert für bestimmte Parameterwerte keine verlässlichen Aussagen

Problem, wenn

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

Inseln gültiger Lösungen



Hierarchisierung

- unterschiedliche Repräsentationen
- unterschiedliche Modelle
- unterschiedliche externe Parameter

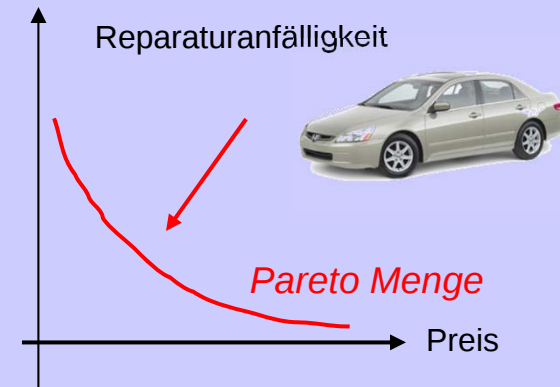
Multi-kriterielle Optimierung

- Multi-kriterielle Optimierung:
mehrere (unvereinbare) Optimierungskriterien
- Kriterien können gewichtet werden und zu einer neuen skalaren Gütefunktion zusammengefasst werden - z.B. Summe gewichteter Kriterien
- Soll eine explizite Gewichtung der Kriterien vermieden werden, so ist die Lösung des Optimierungsproblems nicht ein Parametervektor, sondern eine Menge von Vektoren (Pareto Menge)
- formale Definition eines multi-kriteriellen Optimierungsproblems:

$$\min \vec{f}(\vec{x}) = \min(f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x}))$$

$$\text{Randbedingungen: } \vec{g}(\vec{x}) > \vec{0}$$

$$\vec{h}(\vec{x}) = \vec{0}.$$



Multi-kriterielle Optimierung: Definitionen

Seien $\vec{a}$ und $\vec{b}$ zwei Parametervektoren

➤ Schwache Pareto Dominanz $\vec{a} \succeq \vec{b}$ or $\vec{b} \preceq \vec{a}$ if $\forall i; f_i(\vec{a}) \leq f_i(\vec{b})$.

➤ Pareto Dominanz

$\vec{a} \succ \vec{b}$ or $\vec{b} \prec \vec{a}$ if $\forall i; f_i(\vec{a}) \leq f_i(\vec{b})$ and $\exists i; f_i(\vec{a}) < f_i(\vec{b})$.

➤ Starke Pareto Dominanz $\vec{a} \succ\succ \vec{b}$ or $\vec{b} \prec\prec \vec{a}$ if $\forall i; f_i(\vec{a}) < f_i(\vec{b})$.

Die Paretomenge ist die Menge aller *Paretooptimalen* Parametervektoren

Ein Parametervektor $\vec{x} \in \Omega$ heisst *Paretooptimal*, wenn

$$\nexists \vec{\omega} \in \Omega ; \vec{\omega} \succ \vec{x}$$

wobei Ω die Menge der gültigen Lösungen darstellt

$$\Omega = \left\{ \vec{x} \in R^n ; \vec{g}(\vec{x}) > \vec{0}, \vec{h}(\vec{x}) = \vec{0} \right\}.$$

Multi-kriterielle Optimierung: Algorithmen

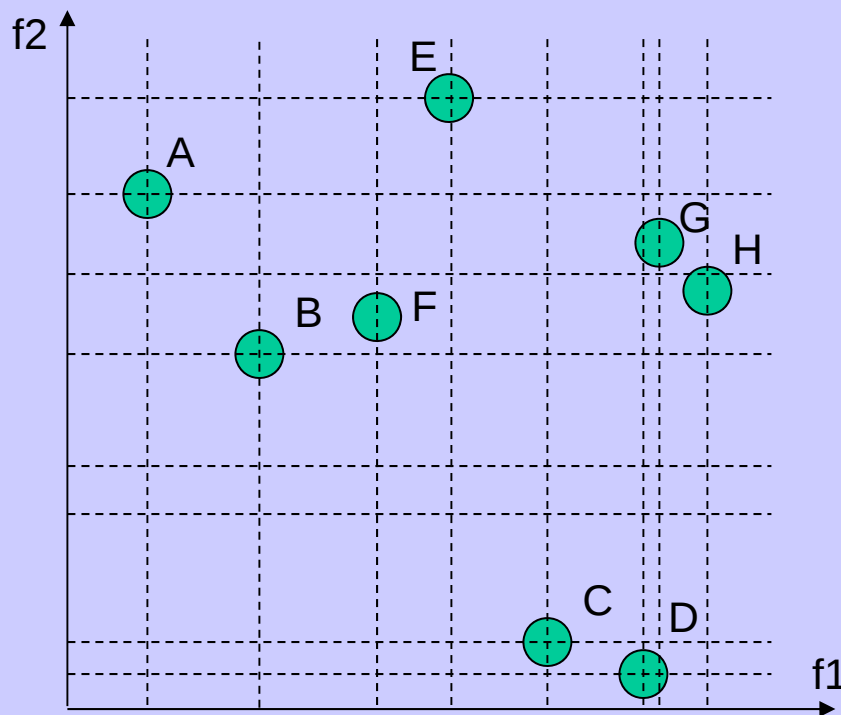
- Es gibt eine grosse Anzahl unterschiedlicher Algorithmen, die meisten basieren auf einer Rankingmethode

Beispiel: NSGA II (Non-dominated Sorting GA II)

K. Deb, et al. , 2000.

Schritt 1

Crowded Tournament Selection



ordne nach dem Rang:

A : Rank 1
B : Rank 1
C : Rank 1
D : Rank 1
E : Rank 3
F : Rank 2
G : Rank 3
H : Rank 3



A
B
C
D
F
E
G
H

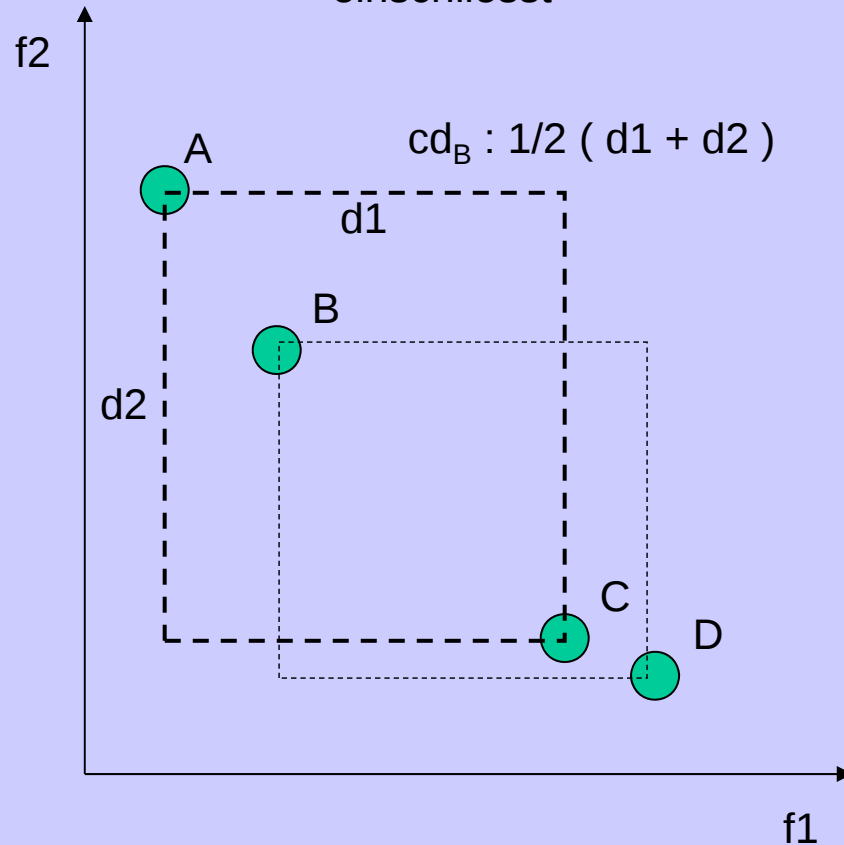
Multi-kriterielle Optimierung: Algorithmen

Beispiel: NSGA II (Non-dominated Sorting GA II)

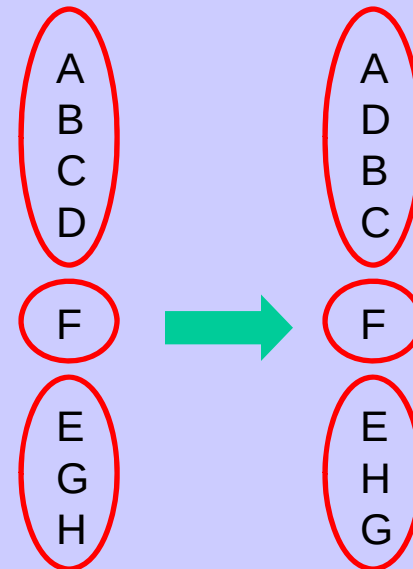
K. Deb, et al. , 2000.

Schritt 2 *crowded distance*

mittlere Seitenlänge des max. Rechteckes, welches nur die Lösung einschliesst

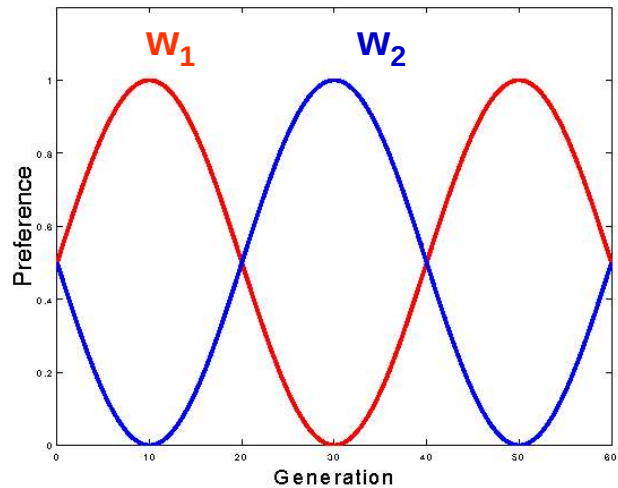


ordne innerhalb desselben Ranges
gemäß crowded distance

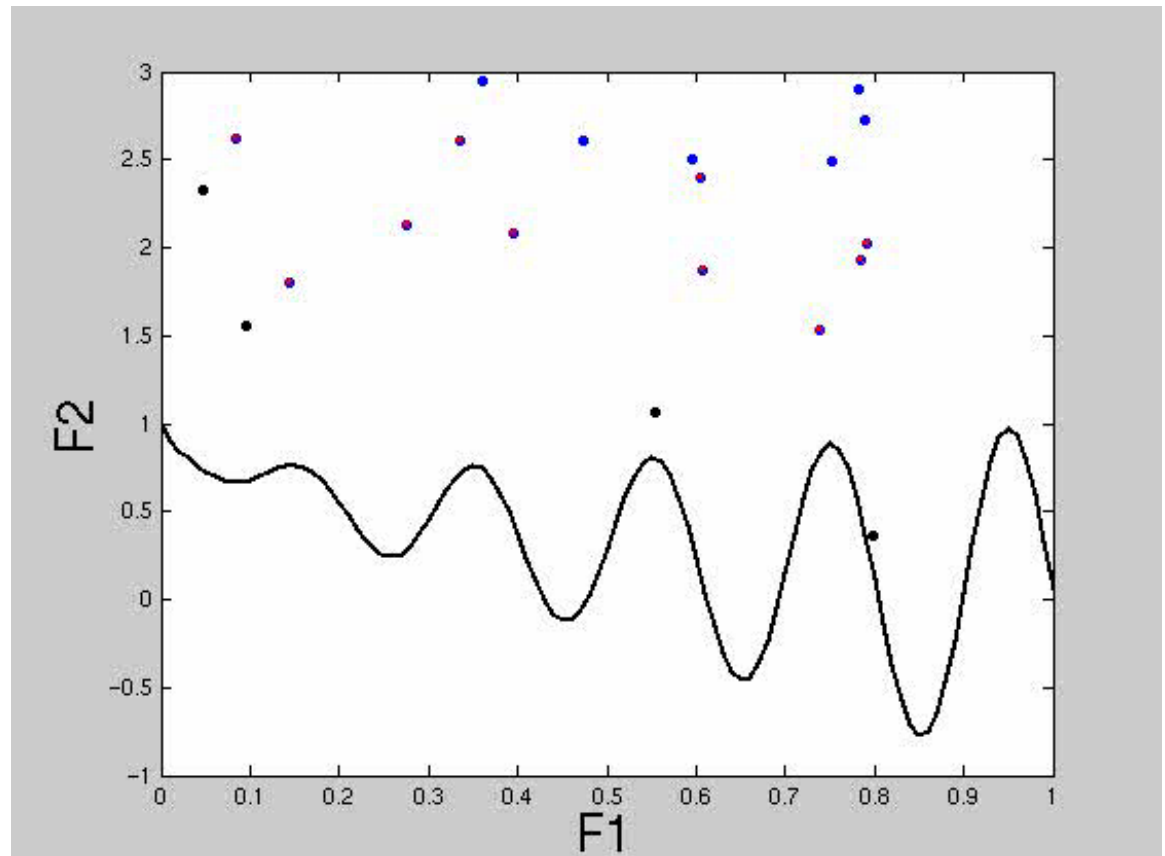


Multi-kriterielle Optimierung - Dynamische Gewichtung

$$f = f_1 w_1 + f_2 w_2$$



Dynamik der Gewichte
während der Optimierung

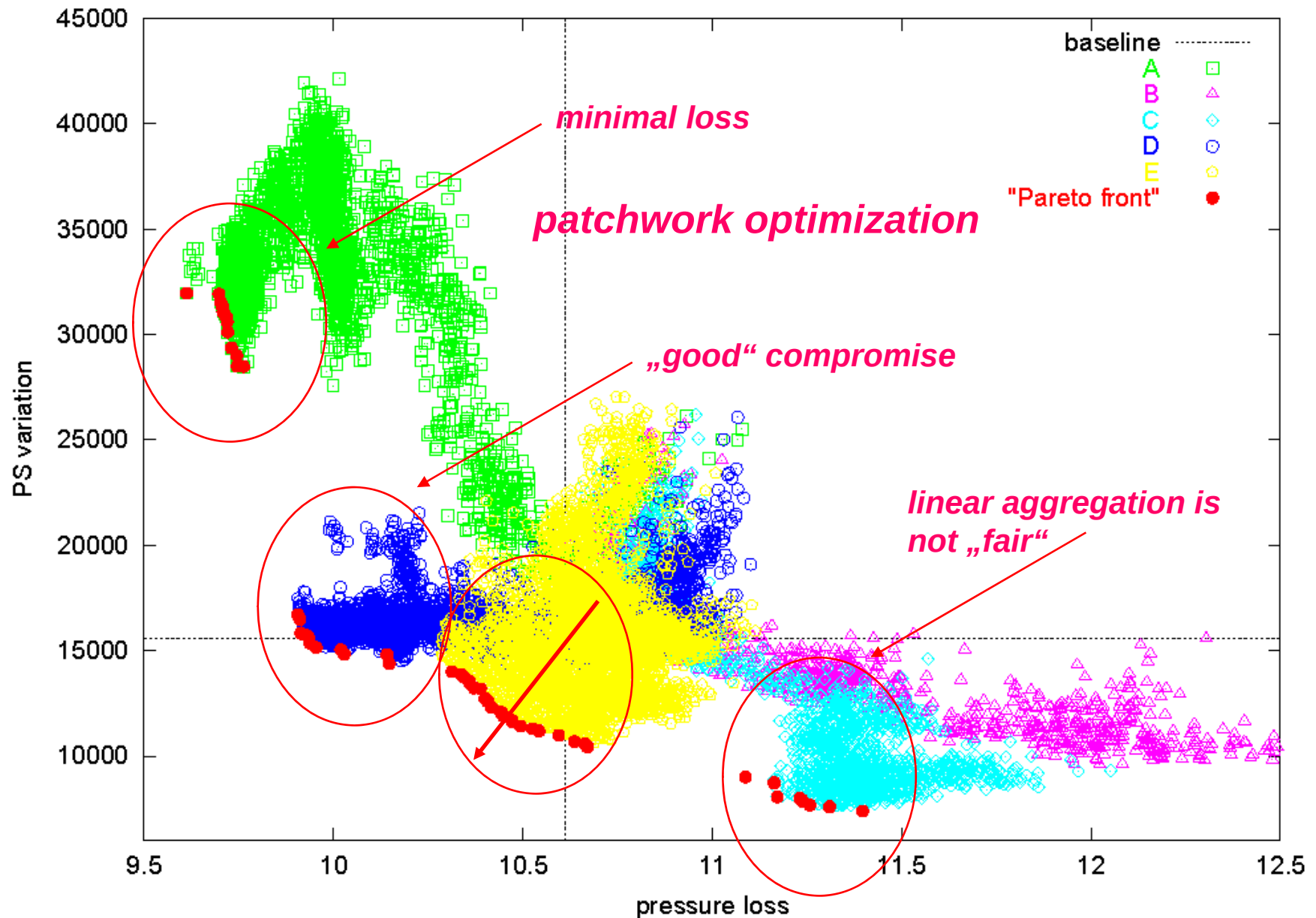


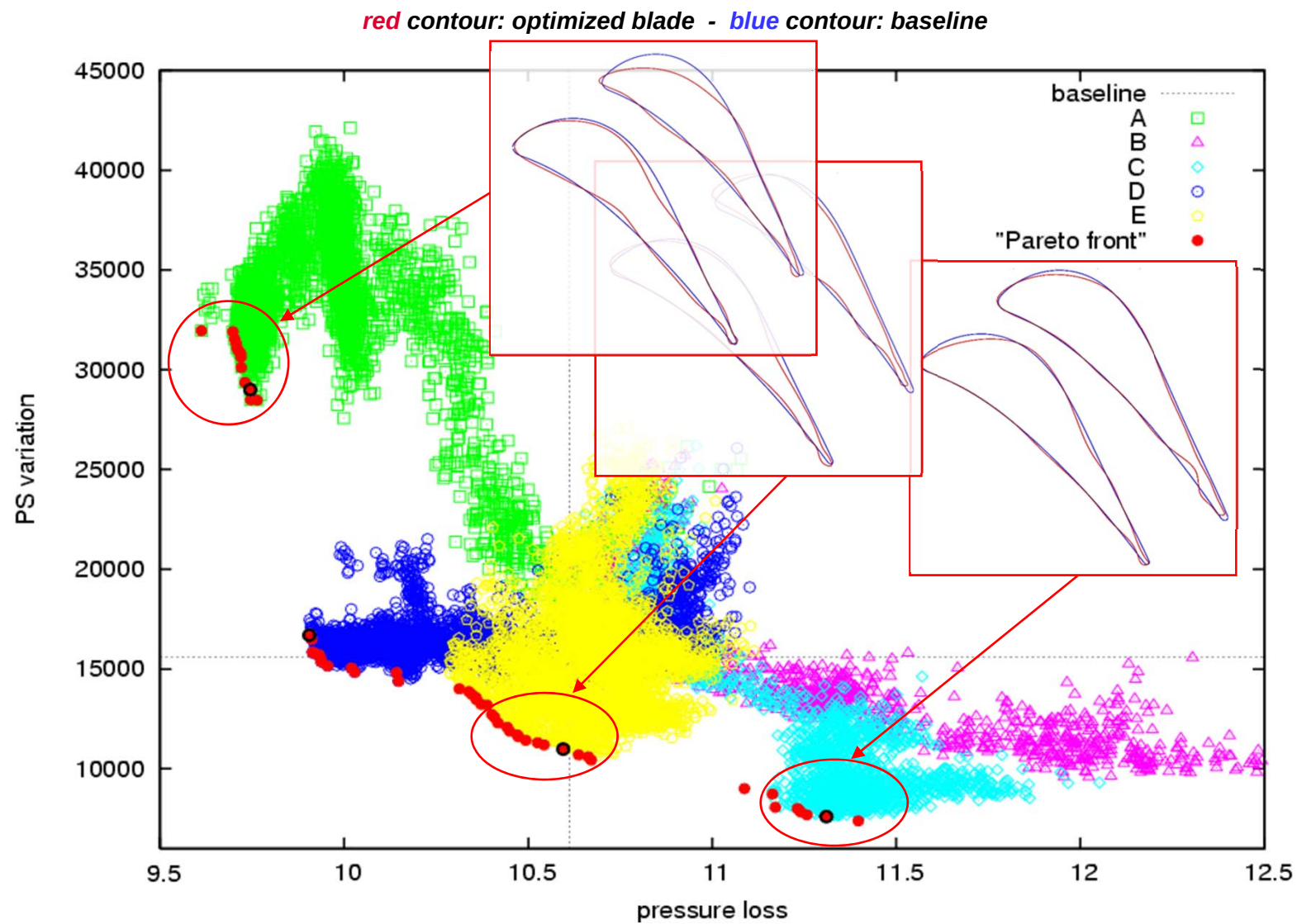
● Pareto Lösungen

● Elterpopulation

● Nachkommenpopulation

Patchwork Optimisation – 3D Turbinenblattoptimierung





Optimierung unter Störungen

Optimierungsprobleme in der Anwendung sind oftmals (zumeist) verrauscht

- *Implizite Störungen* in der Evaluation des Systems
 - Genauigkeit bei Simulation, z.B. computational fluiddynamics
 - 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

Ansätze zum algorithmischen Entwurf robuster Systeme

- (1) Methoden, die die Robustheitsmaße explizit berechnen und dann das resultierende Optimierungsproblem lösen, z.B. nichtlineares Programmieren

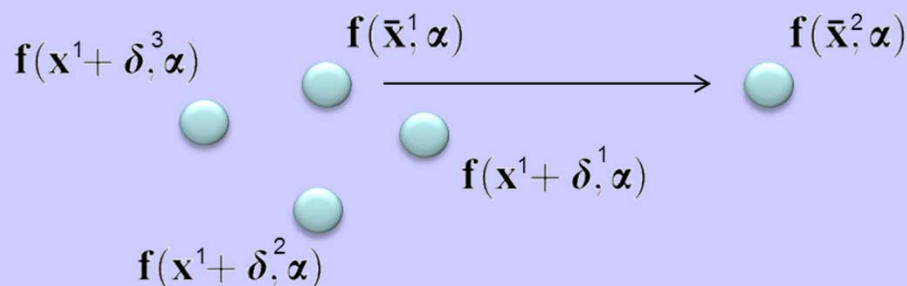
$$F_1(\mathbf{x}) = \int f(\mathbf{x} + \boldsymbol{\delta}, \boldsymbol{\alpha}) p(\boldsymbol{\delta}, \boldsymbol{\alpha}) d\boldsymbol{\delta} d\boldsymbol{\alpha} \quad \text{Fast nie berechenbar!}$$



- (2) Optimierungsverfahren, die direkt auf der verrauschten Funktion operieren

- Methoden, die Robustheitsmaße approximieren
- Methoden die $\mathbf{f} = \mathbf{f}(\mathbf{x}, \boldsymbol{\alpha})$ direkt für eine verrauschte Optimierung nutzen

Bewertung durch Sampling

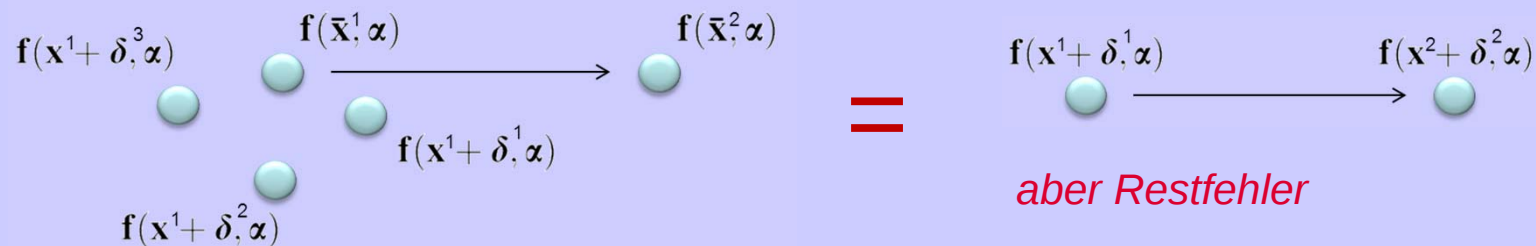


Direkte verrauschte Bewertung



Robustheit mit Evolutionären Algorithmen

Evolutionäre Algorithmen maximieren (minimieren) den Erwartungswert bei direkter verrauschter Optimierung - es bleibt ein Restfehler bei Annäherung an das verrauschte Optimum proportional zur Varianz des Rauschens und zur Suchraumdimension



- ➡ Resultat empirischer Untersuchungen für verschiedenen evolutionäre Algorithmen
- ➡ bei Evolutionsstrategien kann man theoretisch zeigen, dass die Abnahme des Restfehlers bei einer **quadratischen N-dimensionalen** Funktion mit $\kappa \mu^2$ skaliert
(μ : Populationsgröße und κ : Stichprobengröße)
- ➡ Es ist effizienter mit großen Populationen zu arbeiten als zu sampeln

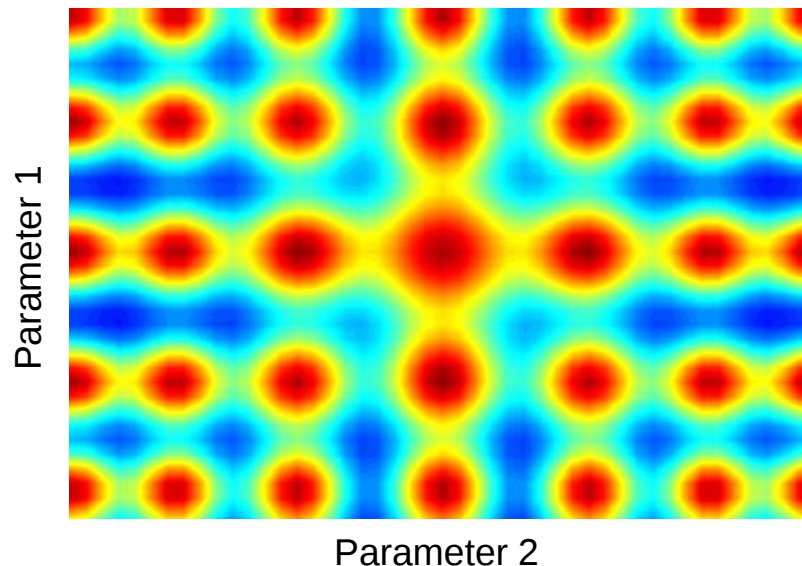
Optimierung unter Störungen - Multi-modale Funktion

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

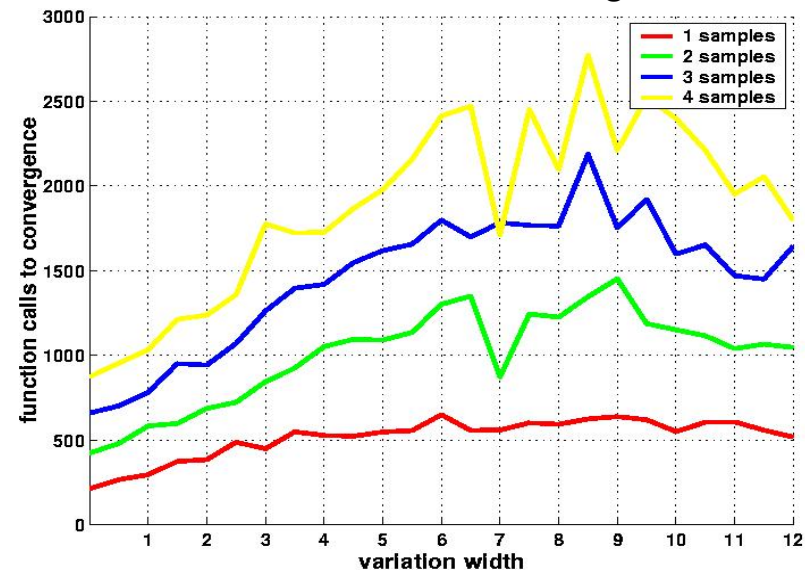
➡ *explizites Mitteln ist nicht notwendig*

- Multi-modale Funktion: Optima 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

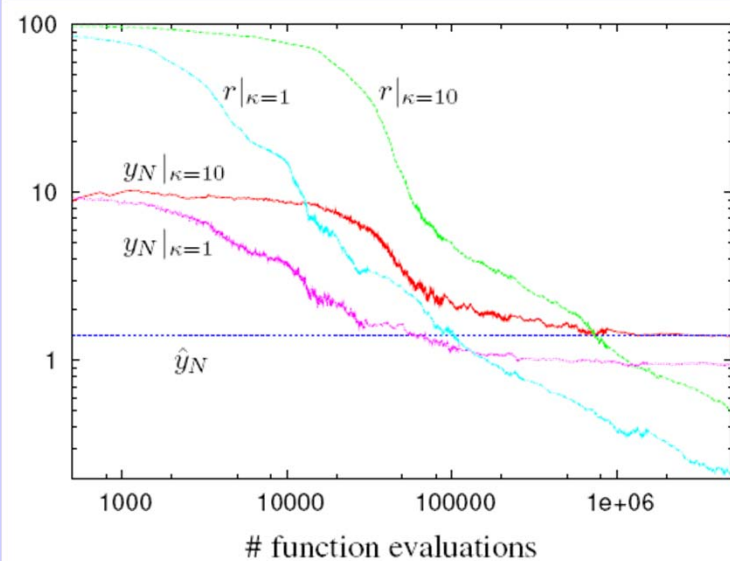
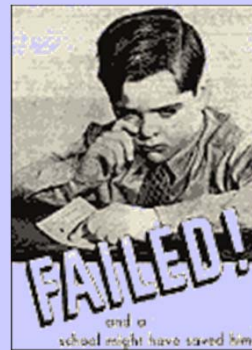
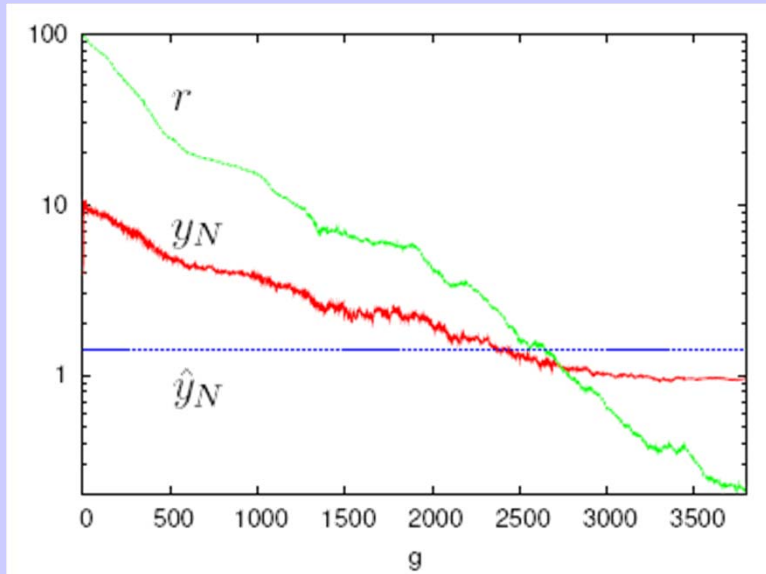
Contour Plot der 2D-Testfunktion



Funktionsevaluierungen



Sampling oder nicht Sampling - das ist die Frage



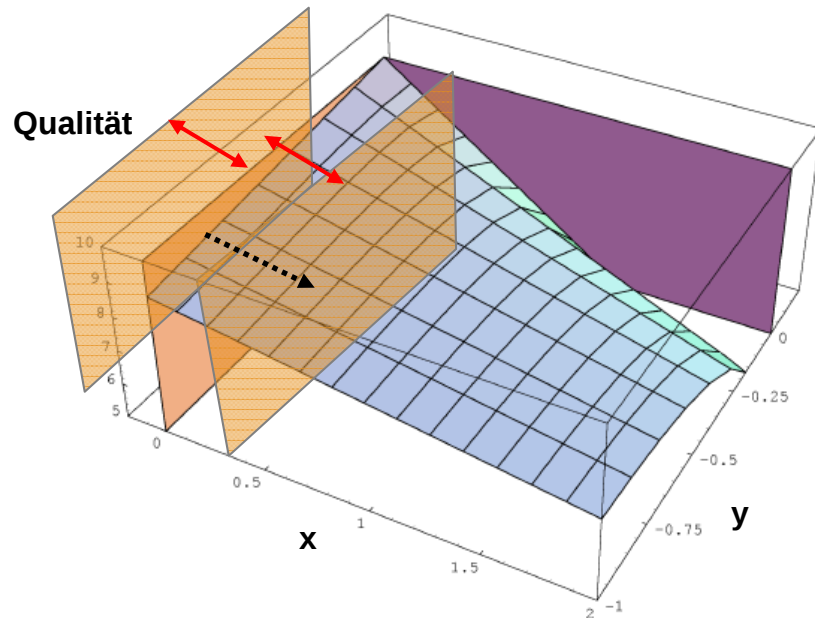
bei einigen Funktionen sind
explizite Stichproben notwendig

warum manche Funktionen schwer
robust zu optimieren sind, ist
gegenwärtig nicht geklärt

explizite Stichproben reduzieren die Effizienz, der
optimale Zustand wird jedoch besser approximiert

Systeminterne Störung

- Erweiterung des additiven Störmodells auf system-interne Störungen



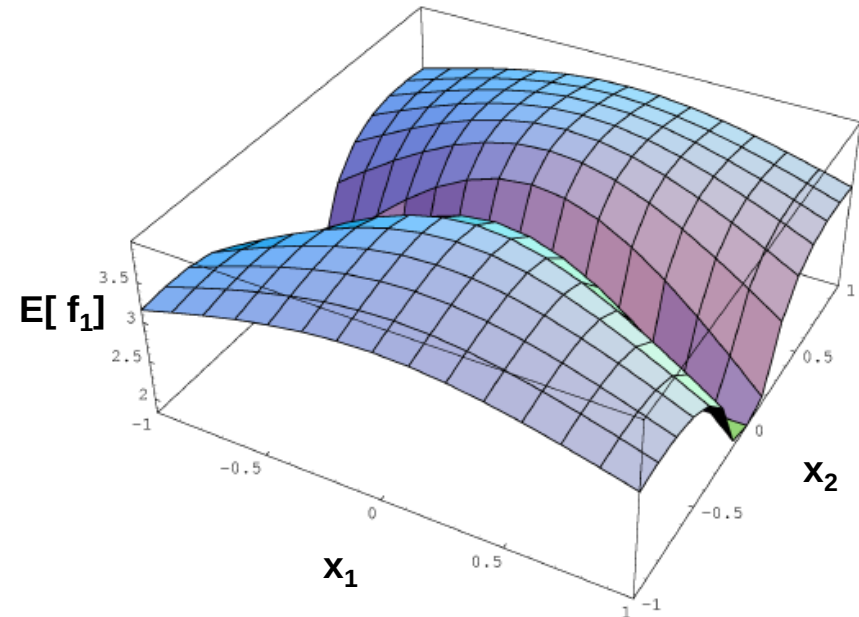
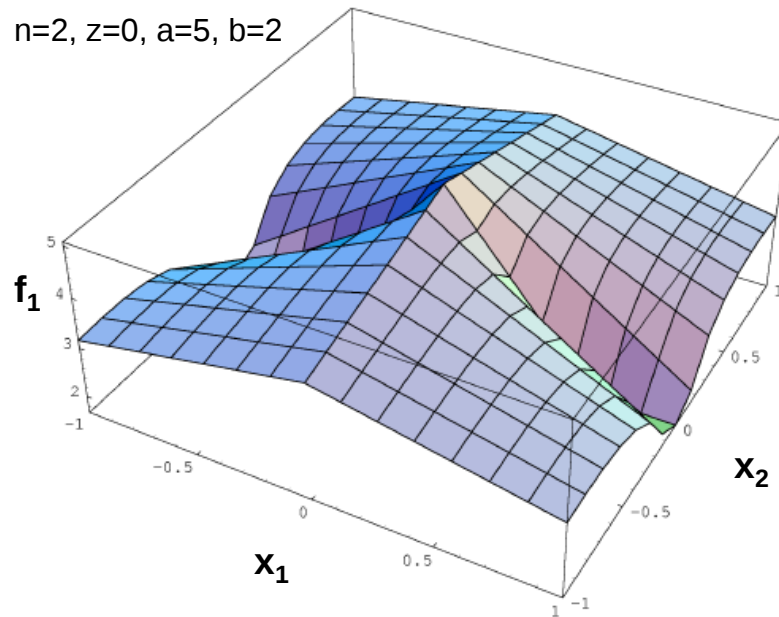
- die Grenze gültiger Lösungen ist nicht fest und kann gegenüber dem Grad variieren

- Qualitatives lokales Fitnessmodell für das Verhalten der Evolutionstrategie bei Designoptimierung unter bestimmten Randbedingungen
- Gradient in x-Richtung wird mit zunehmenden y-Werten steiler
- Ziel ist Robustheit gegenüber Parametervariation senkrecht zum Grad, d.h.
 $x \rightarrow x + \delta, \delta \sim N(0, \sigma_\delta^2)$
- Mittelwert dient als Qualitätskriterium

Systeminterne Störung - Model

$$f_1(x_1, \dots, x_n) = a - \frac{|x_{n-1} + z| + \sum_{i=1}^{n-2} x_i^2}{|x_n| + b} - |x_n|$$
$$z \sim N(0, \varepsilon^2), b > 0, \mathbf{x} \in \mathbb{R}^n$$

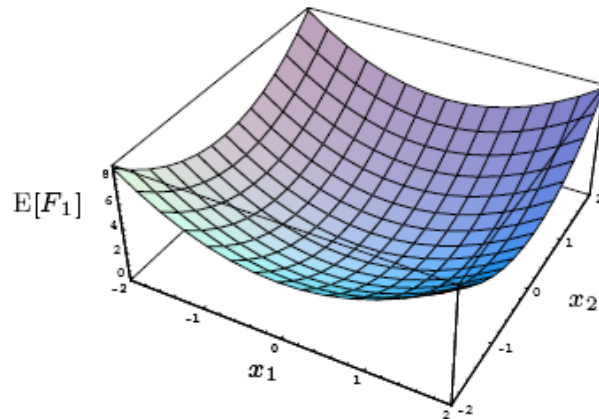
➤ Berechnung des Mittelwertes führt zu
störungsinduzierter Multimodalität
Bifurkation



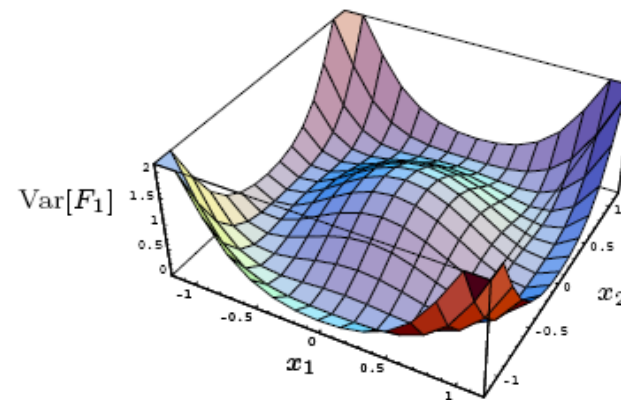
- Residuumwert für R nach Konvergenz der Evolutionstrategie
- Wechsel (*switching*) zwischen Optima in Abhängigkeit der gewählten Evolutionsstrategie

- Trade-off zwischen Robustheit und Leistung

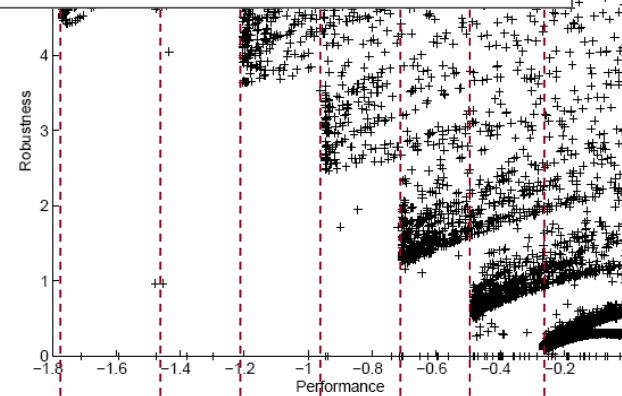
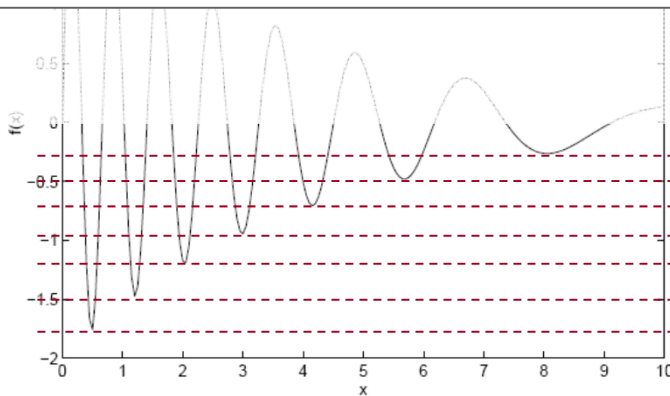
$$F_1(\mathbf{x}) = (\mathbf{x}^2 - 1)z + \mathbf{x}^2, \quad z \sim \mathcal{N}(0, \varepsilon^2), \mathbf{x} \in \mathbb{R}^N$$



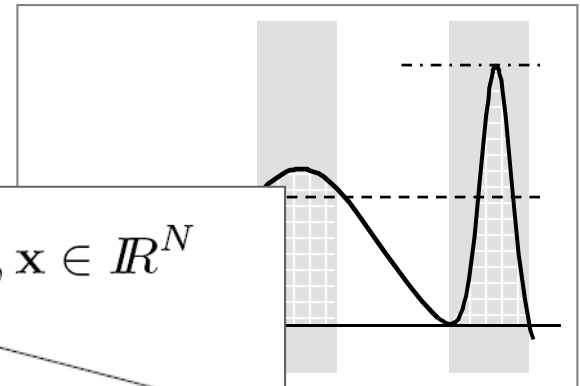
(a)



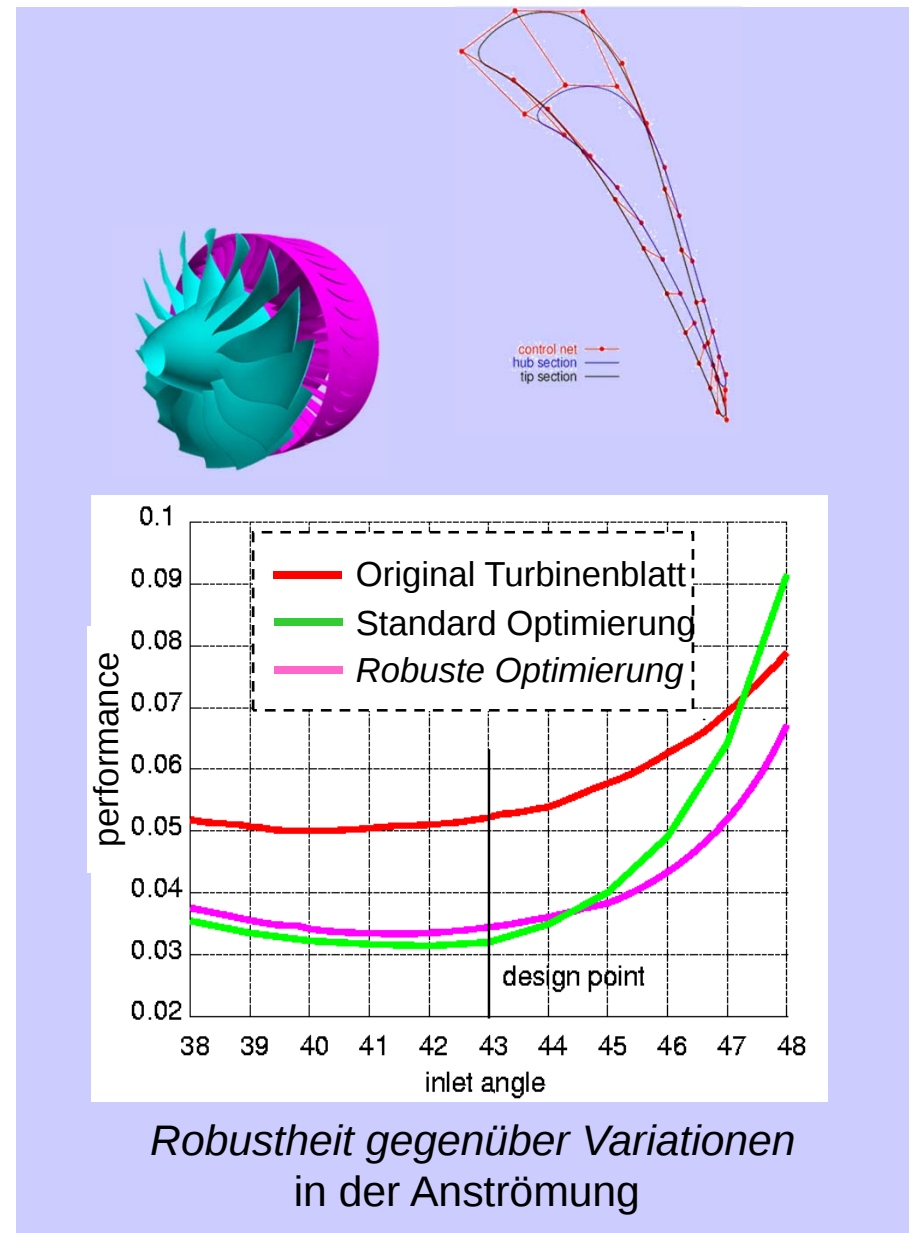
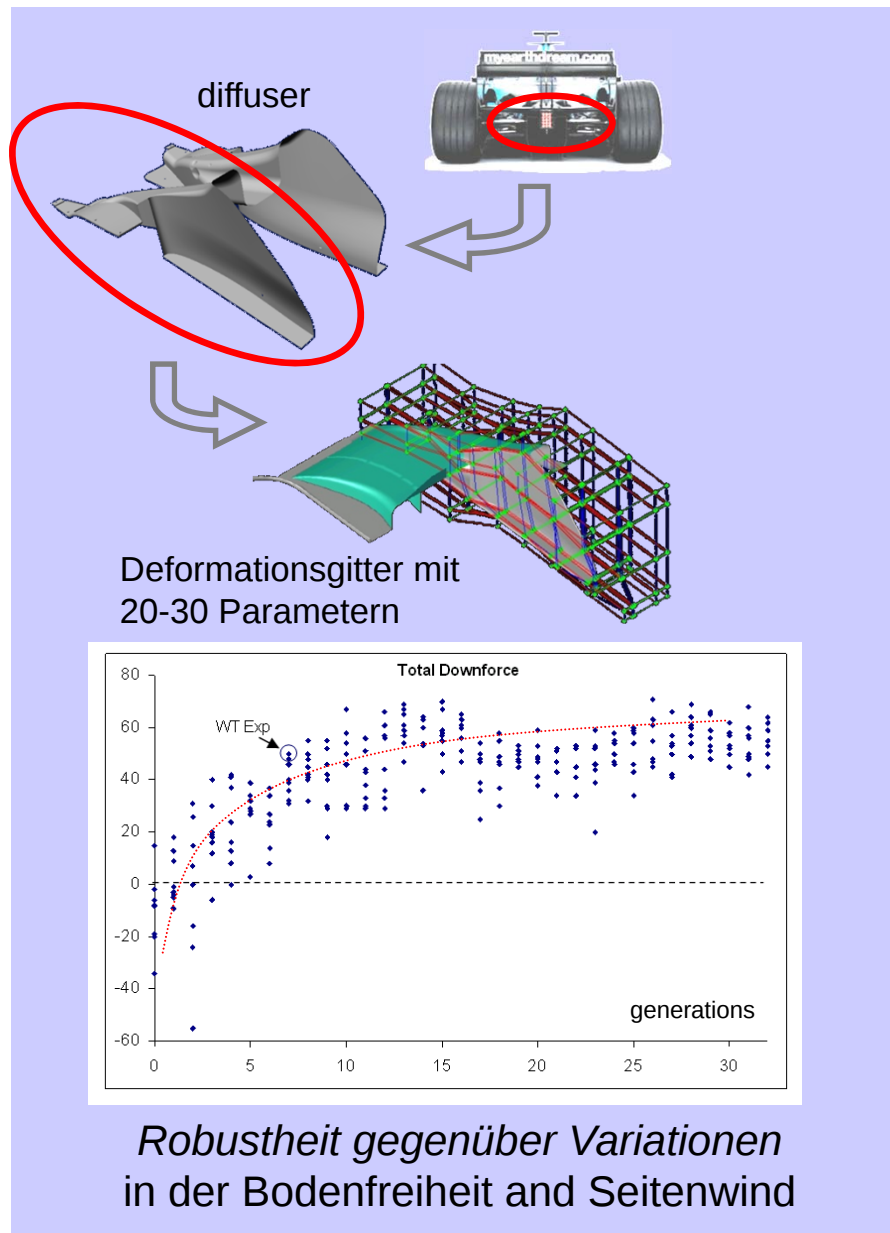
(b)



$$S \left(\frac{f_j - f_k}{x_{ji} - x_{ki}} \right)^2$$



Anwendungen



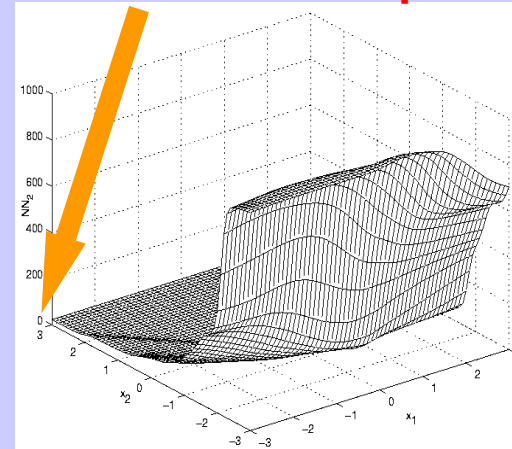
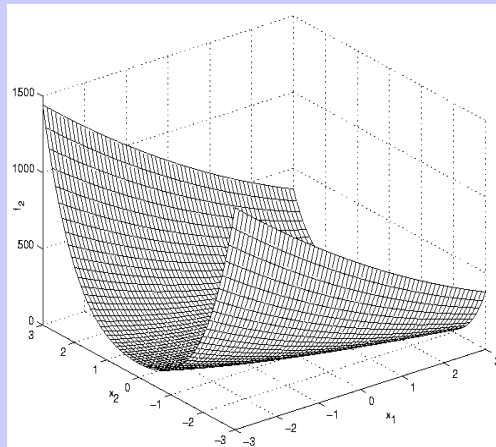
Metamodelle in der Optimierung

■ Motivation

- die Gütefunktion liegt nicht in analytischer Form vor und Fitnesssevaluierungen 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

Metamodelle in der Optimierung

■ Motivation

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

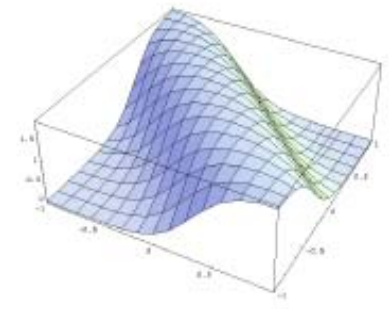
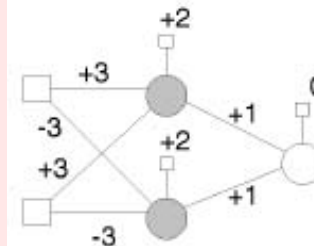
■ Probleme

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

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

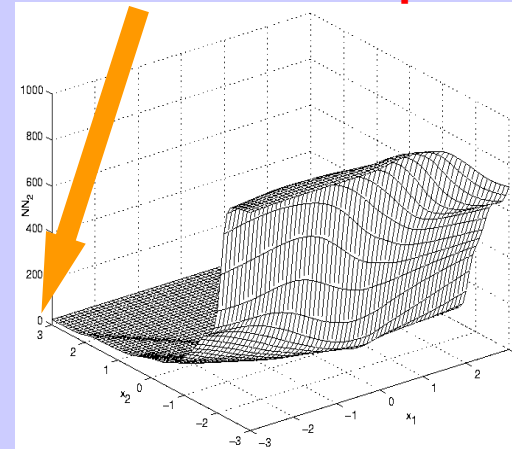
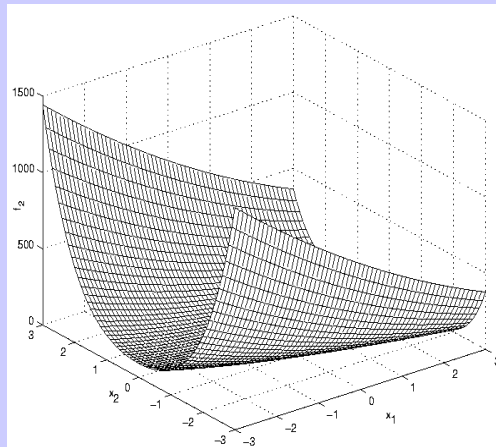
Metamodelle in der Optimierung

■ Motivation

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

■ Probleme

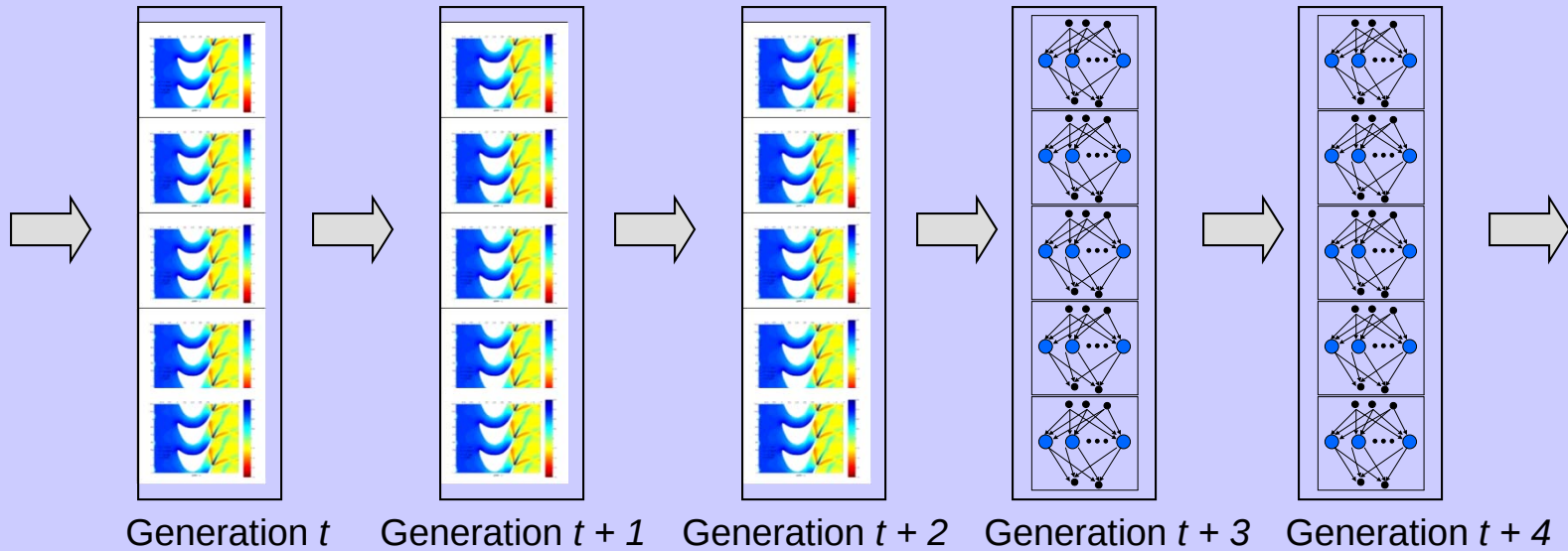
- Metamodelle können falsche Optima haben **Falsches Optimum**



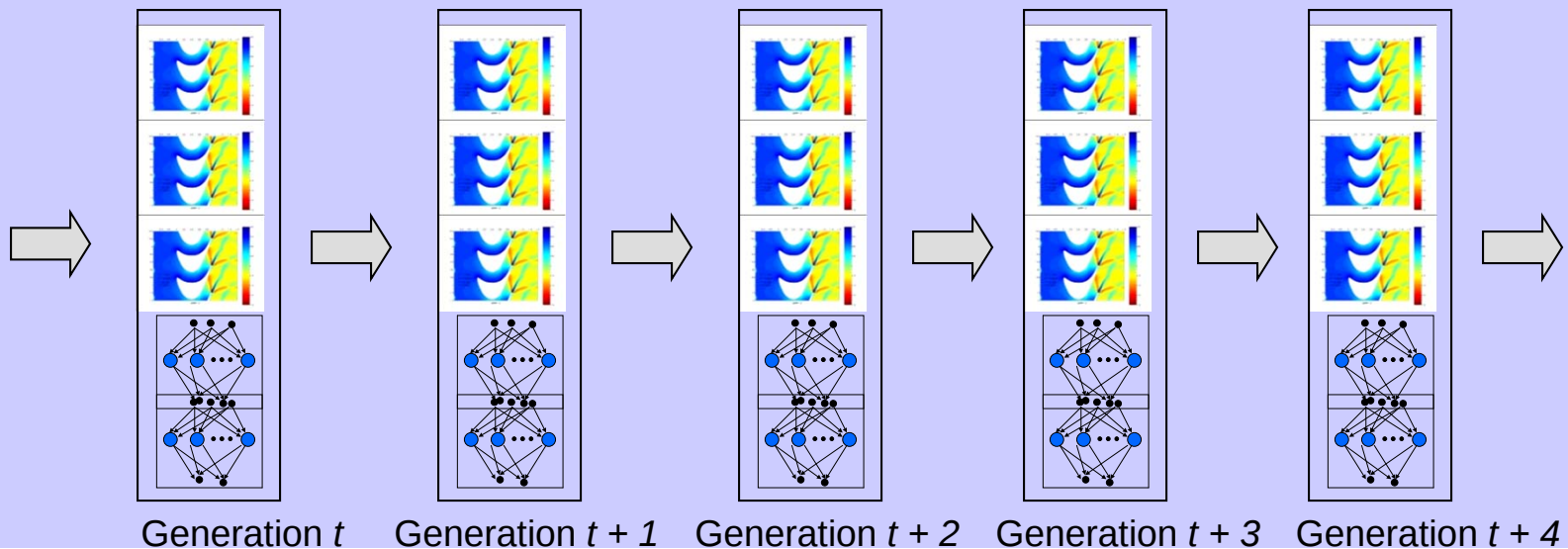
- Lösung: evolutionäre Kontrolle, d.h. Metamodelle werden mit der originalen Gütefunktion kombiniert

Generationen- vs. Individuell-basierte Anpassung

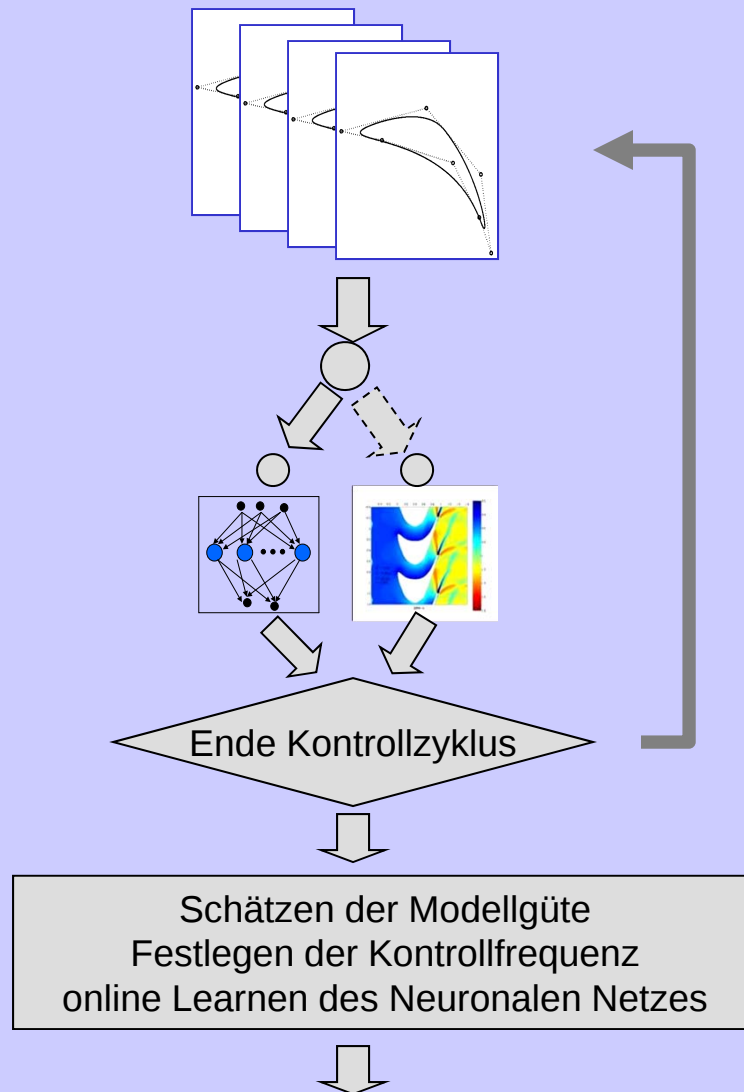
■ Generationen-basierte Anpassung



■ Individuell-basierte Anpassung



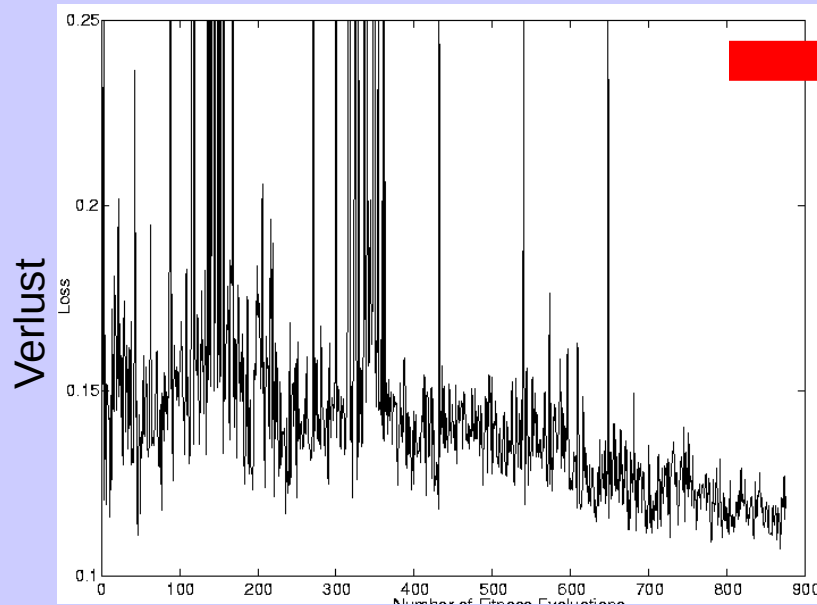
Neuronale Netze als Metamodelle in EAs



- on-line Lernen der neuronalen Netze muss schnell und effizient sein, d.h. Kombination mit offline Struktur-optimierung ist sinnvoll
- die Adaptation der Kontrollfrequenz ist entscheidend für das richtige Gleichgewicht zwischen korrekter Konvergenz und Rechenaufwand
- Möglichkeit der Ensemblebildung zur Güteschätzung

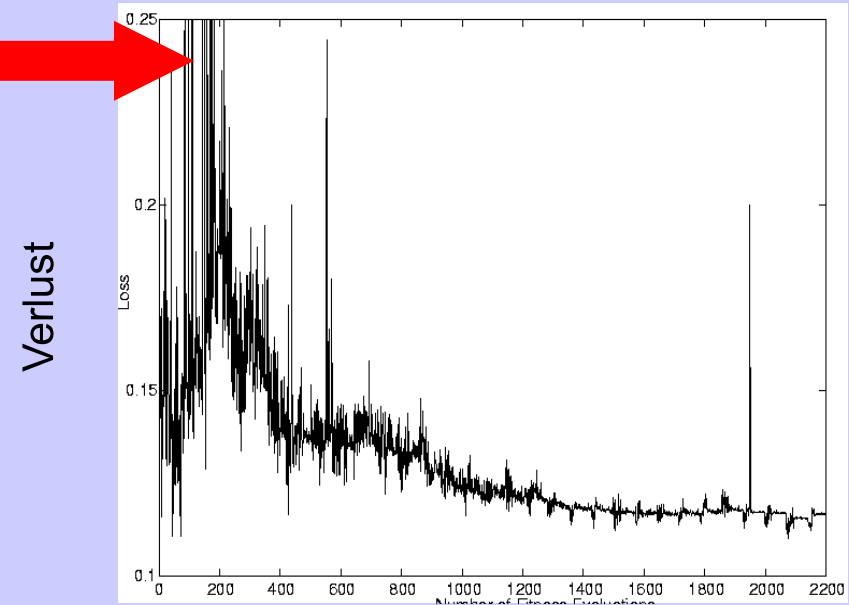
Anwendungsbeispiel: Optimierung einer Turbinenschaufel

Optimierung ohne NN Metamodell



CFD Berechnungen

Optimierung mit NN Metamodell

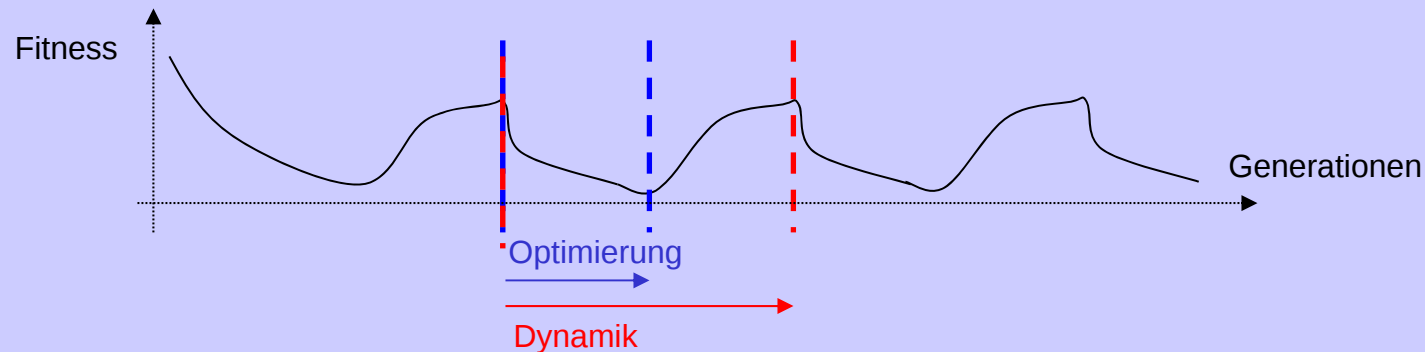


CFD + NN Berechnungen

- 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

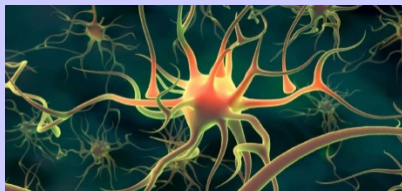
- hohe Diversität innerhalb der Population, z.B. Prinzip des Fitnesssharing
Maximierung der Entropie als Randbedingung
- untere Grenze für die Varianz bei Selbstadaptation von Schrittweiten

Organisation eines Gedächtnisses

- Individuumebene: redundante Repräsentationen, diploide 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 Lernen



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
- ...

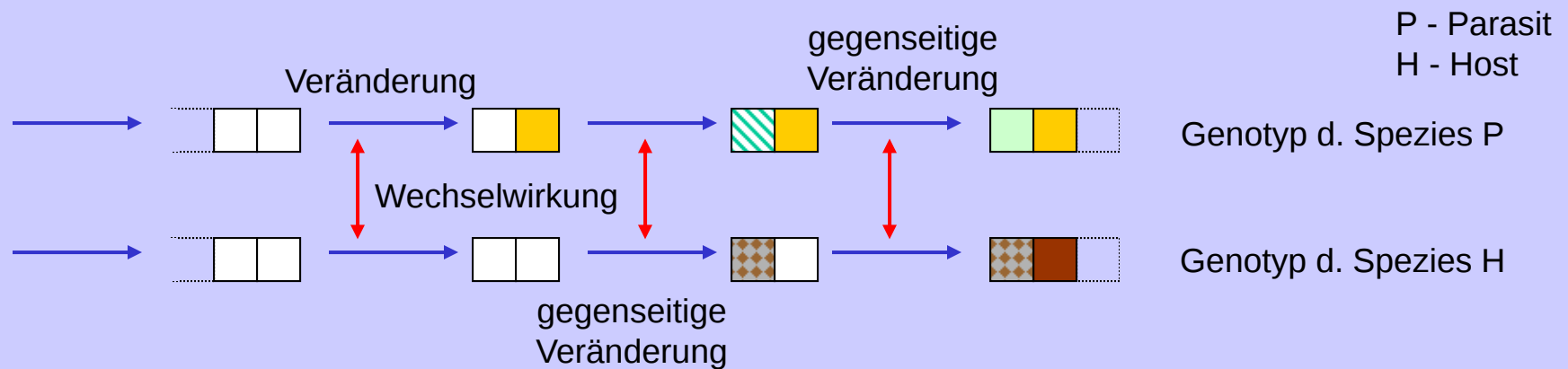
 produziert die Evolution "*optimal*" an die Randbedingungen und die Umwelt adaptierte Lösungen

Anpassung vs. Optimierung

- *Co-evolution* -

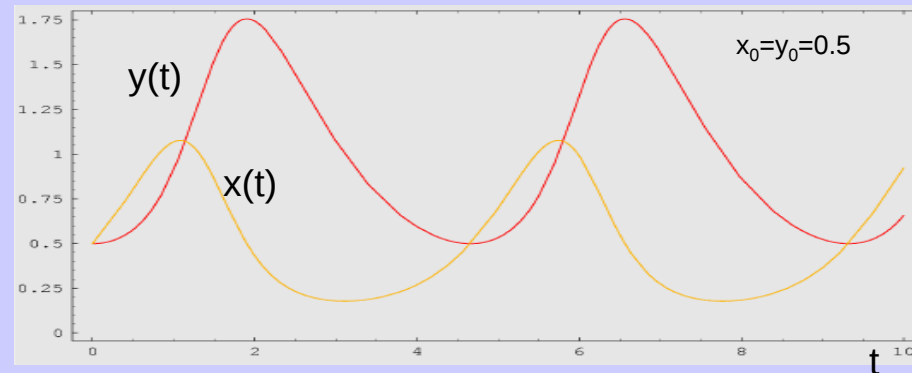
Coevolution

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

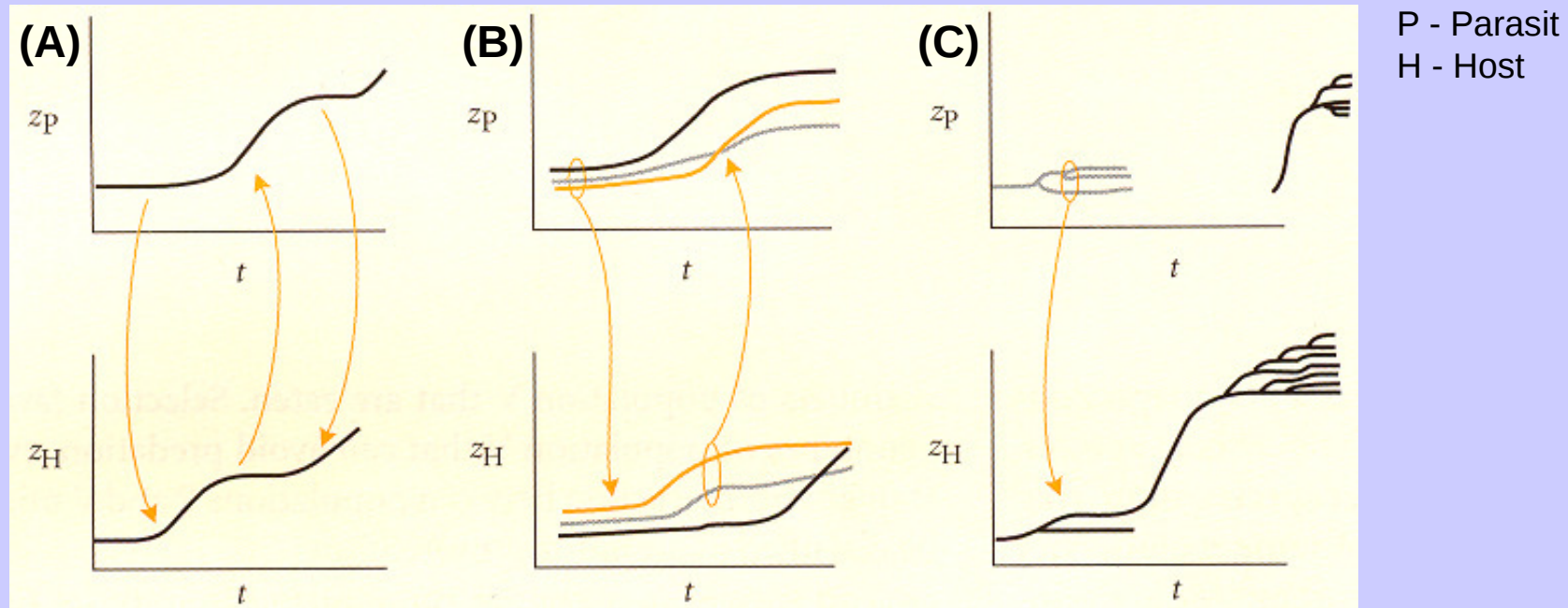


Coevolution ist dynamische Optimierung (Anpassung) mit Rückkopplung, d.h. die eigenen Veränderungen beeinflussen die Veränderungen der Fitnesslandschaft

Bsp: Räuber-Beute Model



Coevolution - Unterscheidungen



(aus: D.J. Futuyama, *Evolutionary Biology*)

- (A) *specific coevolution*** - beide Spezies üben gegenseitigen Selektionsdruck aus
- (B) *guild coevolution*** - mehrere Typ P Spezies interagieren mit mehreren Typ H Spezies
jeder Character evolviert ähnlich aber unterschiedlich schnell
- (C) *escape & radiate*** - Spezies vom Typ P (H spezialisiert) werden ausgelöscht,
Typ H diversifiziert;
später können Spezies Typ P, die auf andere Host spezialisiert
waren, wieder auf Typ H übergehen

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

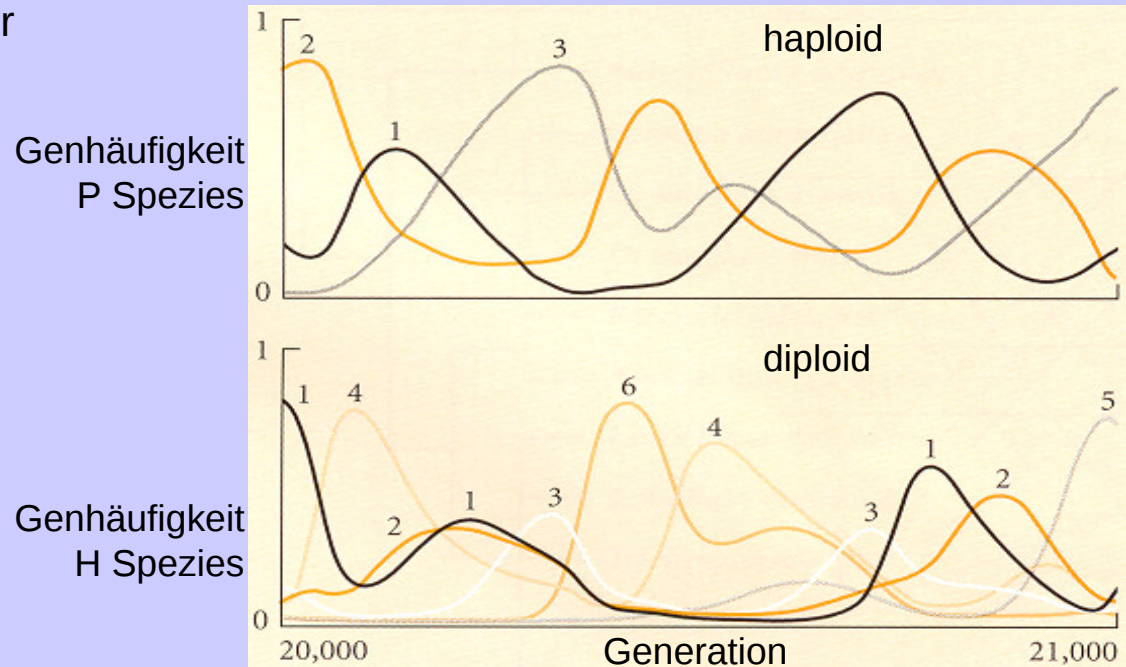
Blume

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

Coevolution

- Evolutionäres “Wettrüsten” wäre typisches Beispiel für Coevolution
- Die Muster der Zeitverläufe der gegenseitigen genetischen Veränderungen können sehr komplex sein, mit lokal stabilen Fixpunkten, Perioden und chaotischem Verhalten

Bsp: Computersimulation der Genhäufigkeiten an einem *resistance locus* und einem *virulence locus*



(aus: D.J. Futuyama, *Evolutionary Biology*)

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	defect
cooperate	3, 3	0, 5
defect	5, 0	1, 1

- der Vergleich kann indirekt sein

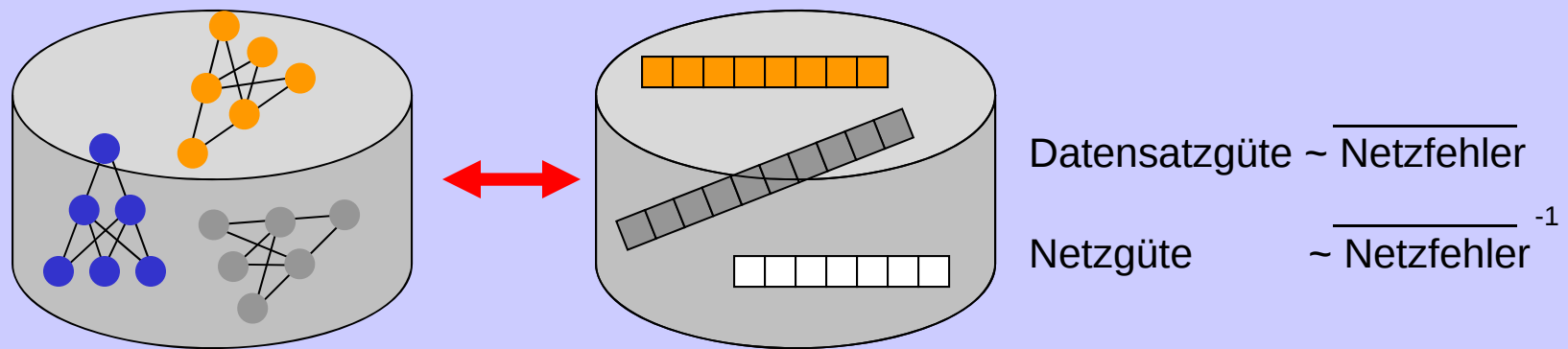
Bsp: Tierra System: Bei der Bestimmung der Güte spielen limitierte Ressourcen eine Rolle, die Interaktion kann sogar *parasitär* sein

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

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

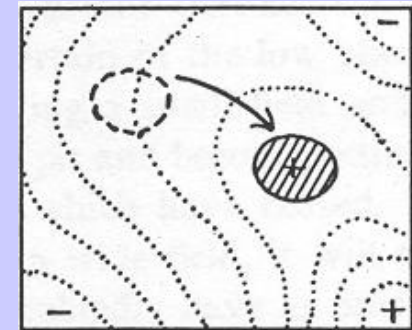
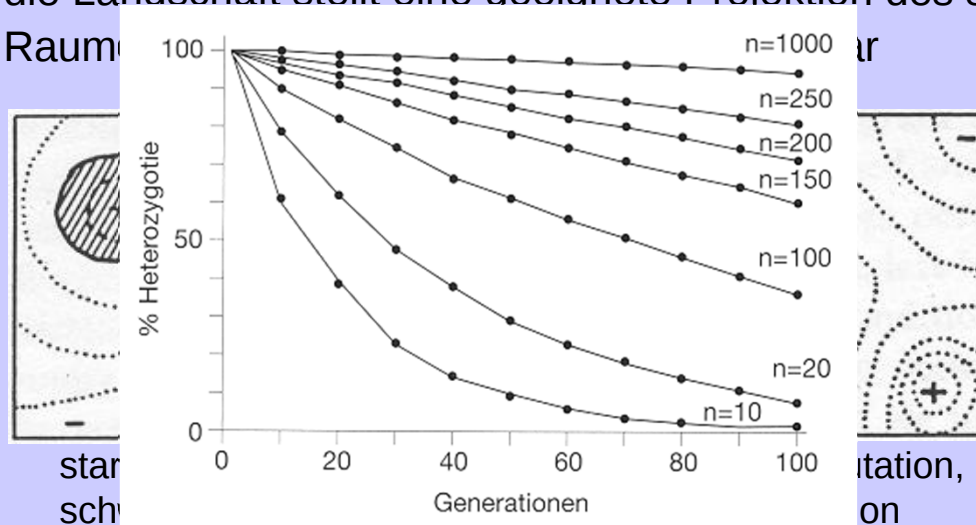
Anpassung vs. Optimierung

- *Fitnesslandschaften* -

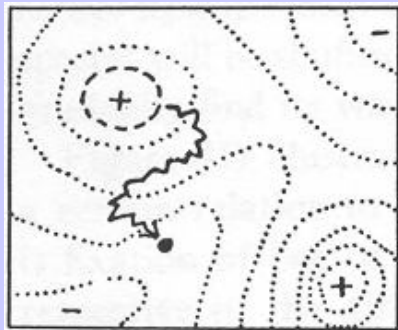
Konzept der Fitnesslandschaft (Wright, 1932)

Landschaft ist eine Abbildung von einem metrischen Raum in den Raum der reellen Zahlen

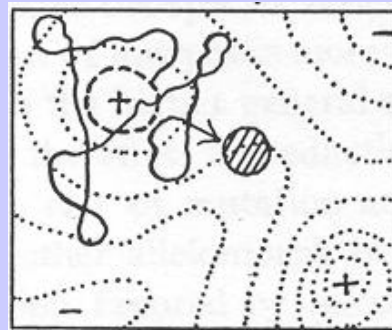
- Wright (1932): Betrachte die Bewegung von Populationen auf einer Landschaft, deren Gipfel hohe Anpassung repräsentieren
- die Landschaft stellt eine geeignete Projektion des sehr hochdimensionalen und diskreten Raums dar



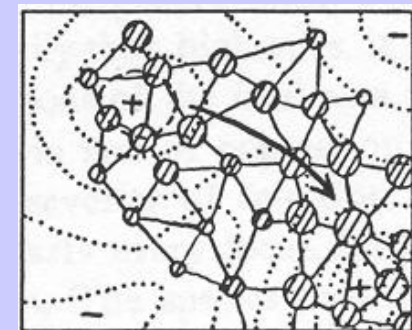
qualitative Änderung der Umwelt



starke Inzucht - *inbreeding*



leichte Inzucht



Aufteilung in lokale Subpopulationen

Fitnesslandschaftsanalyse

Korrelationsbasierte Analyse von Fitnesslandschaften

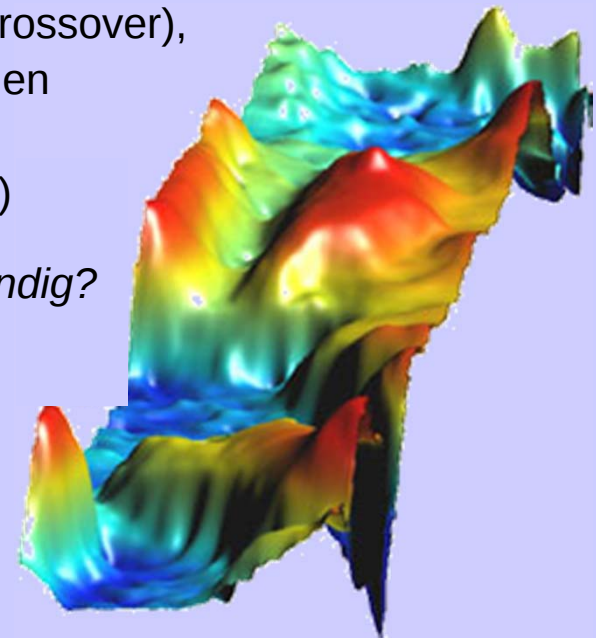
- Autokorrelationsfunktion von einer Abfolge von Schritten auf einer gegebenen Fitnesslandschaft wird berechnet:

$$\text{Corr}(F_t, F_{t+i}) = \frac{E[F_t F_{t+i}] - E[F_t] E[F_{t+i}]}{\text{Var}[F_t]} \approx \frac{\sum_{t=1}^T (F_t - \bar{F})(F_{t+i} - \bar{F})}{\sum_{t=1}^T (F_t - \bar{F})^2}$$

- Annahme: Fitnesslandschaft ist statistisch isotrop
- Optimierung von Variationsoperatoren (Mutation, Crossover), um die Korrelation zwischen Eltern und Nachkommen zu maximieren
(Grundlage ist das Prinzip der graduellen Evolution)

aber: Wieviel Korrelation ist wann(!) wirklich notwendig?

- Ansatz: Analyse des Schwierigkeitsgrades von Problemen



Kritik am Konzept der Fitnesslandschaft und der Korrelation

- Biologische Fitnesslandschaft ist dynamisch, stochastisch und von den eigenen Veränderungen abhängig - *statistische Analyse ist nicht sinnvoll*
 - Phänotypebene wird vernachlässigt
 - Niedrigdimensionale Vorstellung lädt zu falschen Verallgemeinerungen für hochdimensionale Räume ein
-
- Autokorrelation misst nur lineare Zusammenhänge zwischen Schritten; Informationstheoretische Erweiterung: Transinformation
 - Isotropieeigenschaft ist zumeist nicht gewährleistet
 - Korrelation ist schwerlich quantifizierbar
 - Statistische Analyse einzelner Probleme in technischen Systemen nicht effizient
 - als Mass für die Schwierigkeit von Problemen nur bei sehr speziellen Problemklassen erfolgreich

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. Residuumabstand skaliert bei quadratischen N-dimensionalen Funktionen mit $(\kappa \mu^2)^{-1}$
- Metamodelle approximieren die tatsächliche Fitnesslandschaft und erlauben (besonders mit online Anpassung) eine effizientere Suche
- gegenseitige genetische Veränderungen in wechselwirkenden Spezies aufgrund natürlicher Selektion, die 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 I and II. (*Cook Book*)
- [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