Experimentelle Analyse von Evolutionären Algorithmen

Mike Preuss

Algorithm Engineering/Computational Intelligence
Technische Universität Dortmund

Vorlesung: Praktische Optimierung

Wintersemester 2008/2009, 5. Februar 2009

Übersicht

Teil I: Experimente?

Experimente mit (evolutionären) Algorithmen?

Experiment bedeutet:

- Eine Implementierung beschaffen/selbst machen
- An Optimierungsproblem anschliessen, Parameter einstellen
- Laufen lassen (mehrfach), generierte Ergebnisse aufnehmen und bewerten

Warum tut man das?

- Zeigen, dass es funktioniert
- Überprüfen, ob Algorithmus den Anforderungen entspricht
- Bei komplexen/randomisierten/heuristischen Verfahren Vorhersagen schwer

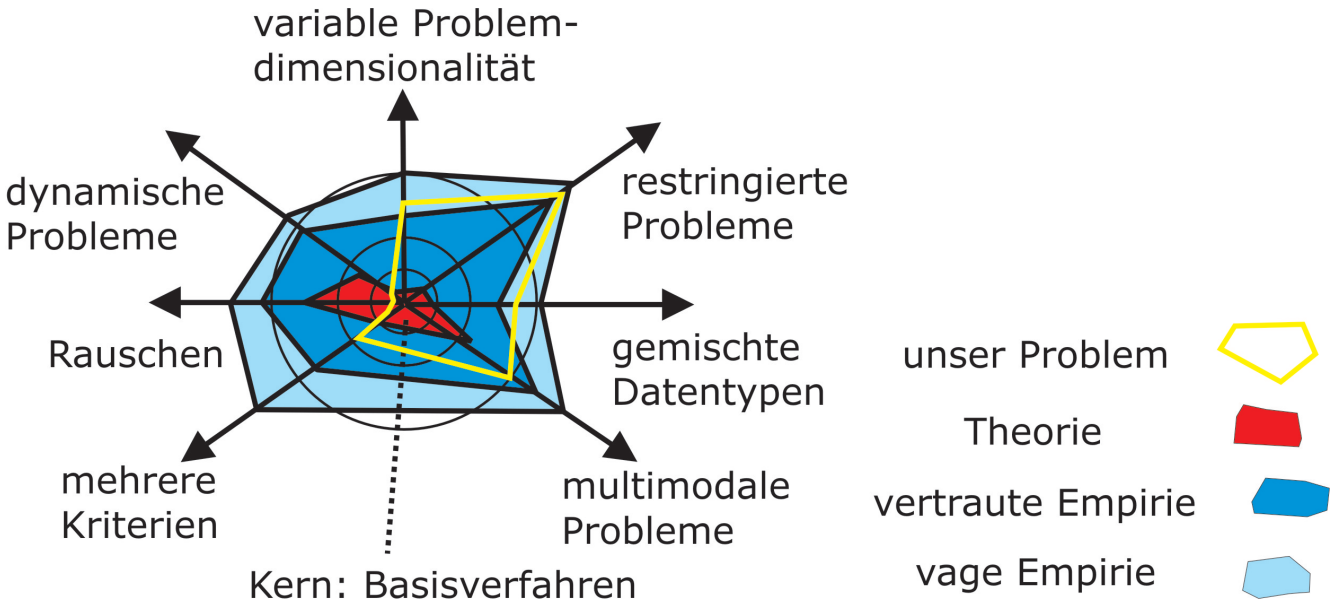
Wann genügt das?

- Ingenieurssicht: Funktioniert gut genug für gegebenes Problem
- Genauer: Anforderungen sind hinreichend gut bekannt
- Wissenschaftlich: Die Anwendung wird durch Theorie abgedeckt

Was genau ist ein konkreter EA?

Unsere 3 Probleme: I

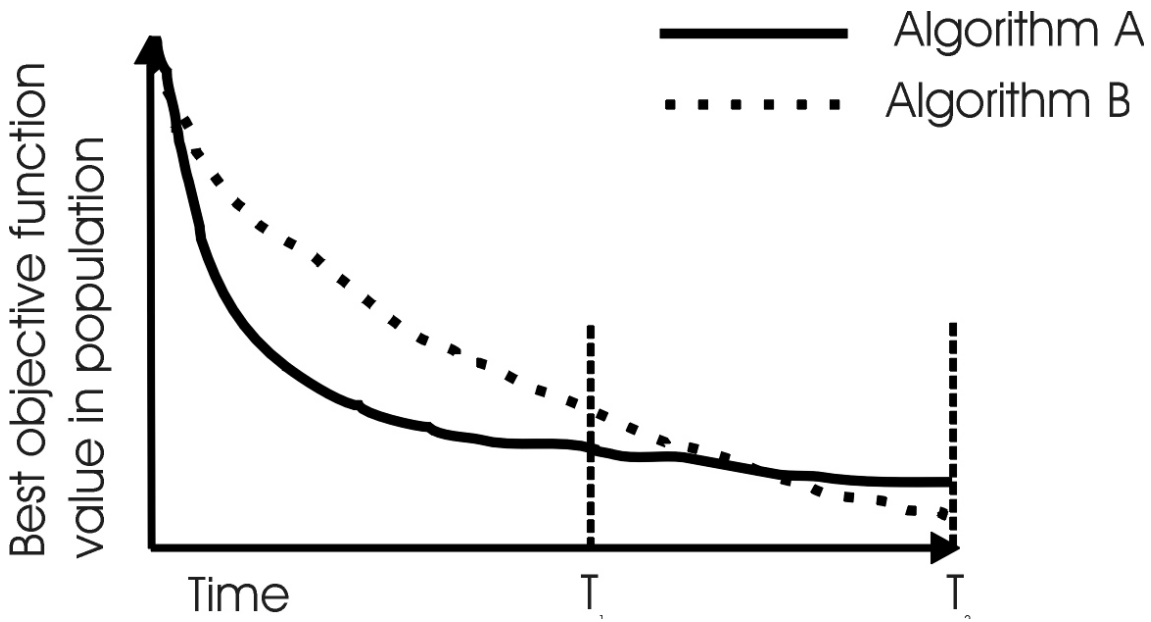
- Evolutionäre Algorithmen sind ’unscharfe Verfahren’.
- Fix sind nur (analog zur biol. Evolution) 2 Mechanismen:
  - Variation – fortgesetztes Erzeugen neuer Information
  - Selektion – bevorzugte Auswahl der besseren Varianten
- EA anwendbar auf *alle Optimierungsprobleme*, aber wie?



Was genau ist ein Optimierungsproblem?

Unsere 3 Probleme: II

Verschiedene Bedingungen müssen definiert/festgestellt werden:

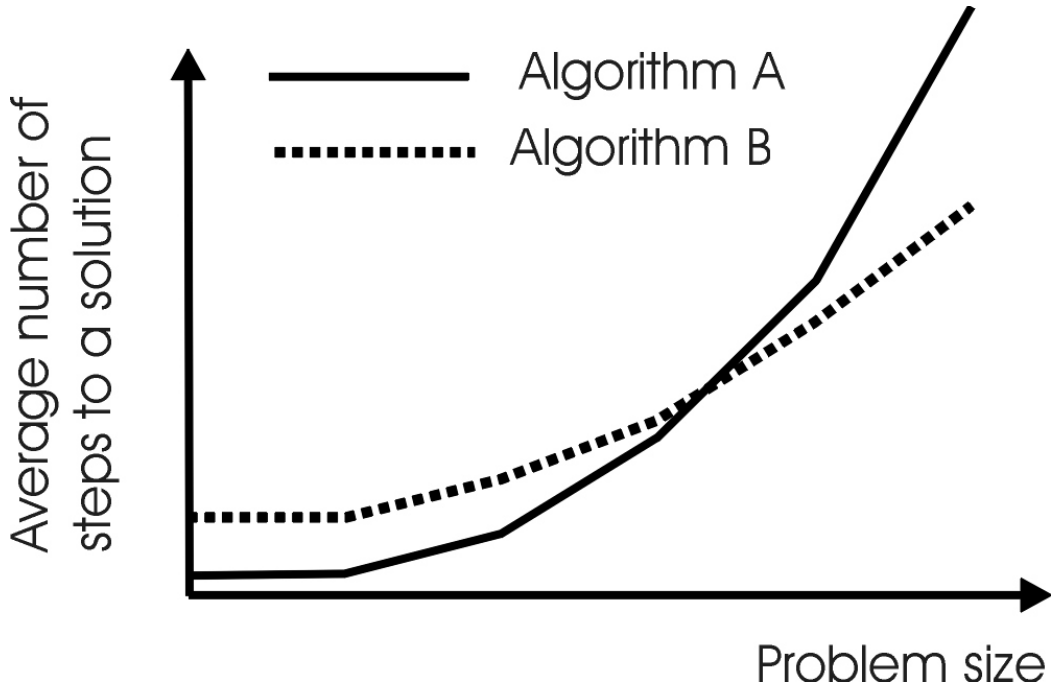


Laufzeit (Anzahl der Funktionsauswertungen)

Was genau ist ein Optimierungsproblem?

Unsere 3 Probleme: II

Verschiedene Bedingungen müssen definiert/festgestellt werden:

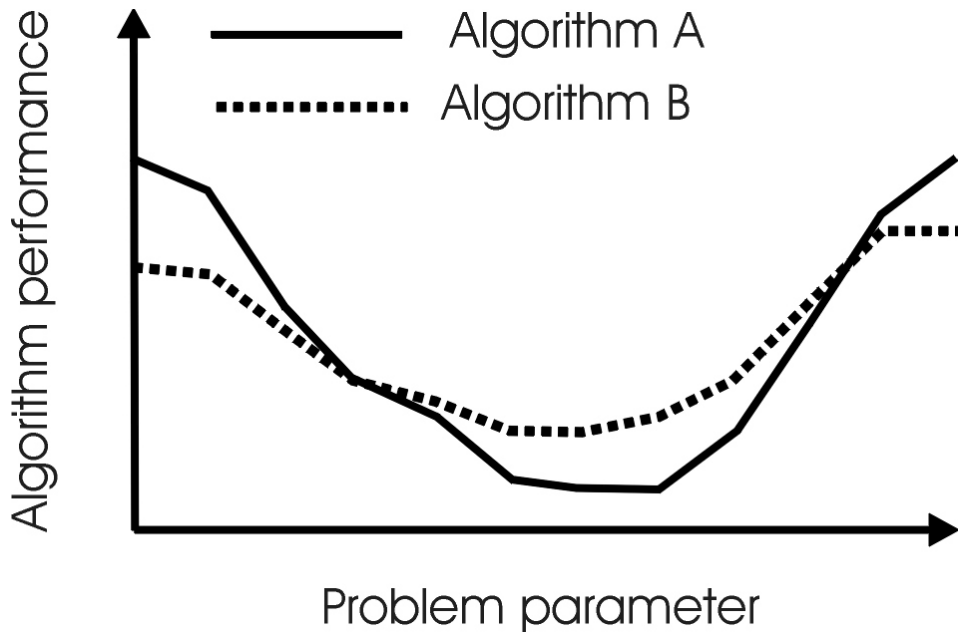


Problemgröße (Anzahl der Dimensionen)

Was genau ist ein Optimierungsproblem?

Unsere 3 Probleme: II

Verschiedene Bedingungen müssen definiert/festgestellt werden:



Probleminstanz (hier gesteuert über einen Parameter)

Was genau ist ein Ergebnis?

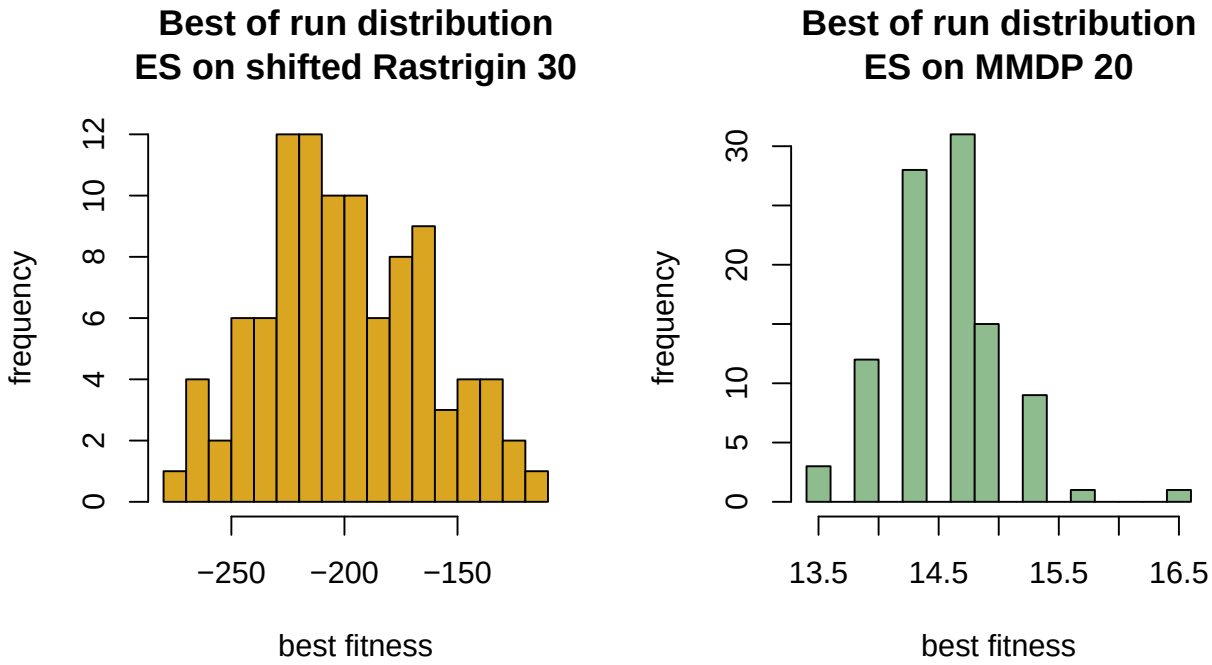
Unsere 3 Probleme: III

Bei deterministischen Algorithmen:

$$f(a, p) = \text{const für } a \in \text{Algorithmen}, p \in \text{Probleme}$$

Bei nichtdeterministischen Algorithmen (z.B. EA):

$$f(a, p) = X \text{ (Zufallsvariable) für } a \in \text{Algorithmen}, p \in \text{Probleme}$$



Was genau ist ein Ergebnis?

Unsere 3 Probleme: III

Bei deterministischen Algorithmen:

$$f(a, p) = \text{const für } a \in \text{Algorithmen}, p \in \text{Probleme}$$

Bei nichtdeterministischen Algorithmen (z.B. EA):

$$f(a, p) = X \text{ (Zufallsvariable) für } a \in \text{Algorithmen}, p \in \text{Probleme}$$

Folgerung

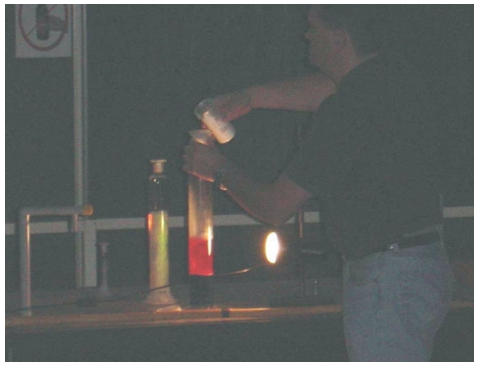
Zu allgemeine Schlüsse sind gefährlich, Verallgemeinerung ist oft genauso unmöglich wie die Vorhersage über eine Theorie

Was nun ???

Sind wir allein (mit diesem Problem)?

In den Naturwissenschaften sind Experimente selbstverständlich

- Viele Erfindungen experimentell gemacht, oft 'zufällig' (Batterien, Röntgenstrahlung, ...)
- Experimente führen zu Theorie, Theorie muss *nützlich sein* (Vorhersagen möglich?)
- Theorie idealisiert (Abstraktion der 'realen Welt')



Ein Experiment

In der Informatik erscheint die Situation anders

- 2 weitverbreitete Stereotypen beinflussen unsere Sicht auf Computer-Experimente:

a) Programme tun genau das, was Algorithmen spezifizieren

b) Computer (Programme) sind deterministisch, also warum Statistik?

Ein Experiment?

Der Blick über den Tellerrand

In den Wirtschaftswissenschaften werden experimentelle Techniken noch nicht lange eingesetzt

- Modellierung menschlichen Verhaltens mithilfe der Rationalitätsannahme war gescheitert
- Bisher kein neues akzeptiertes Modell, daher Experimente als 'Ersatz'

Nichtlineares Verhalten

In der (Evolutions-) Biologie sind sowohl Experimente als auch die Theoriebildung problematisch

- Aktives Experimentieren ist nur selten möglich (*drosophila et al.*)
- Sonst nur Beobachtung (passiv)
- Konzepte anstatt von Theorien: Es gibt immer Ausnahmen

$\Rightarrow$ Stochastische Verteilungen, *population thinking*

Ernst Mayr

Der momentane Stand der experimentellen Technik

Über 40 Jahre empirische Tradition in EC, aber:

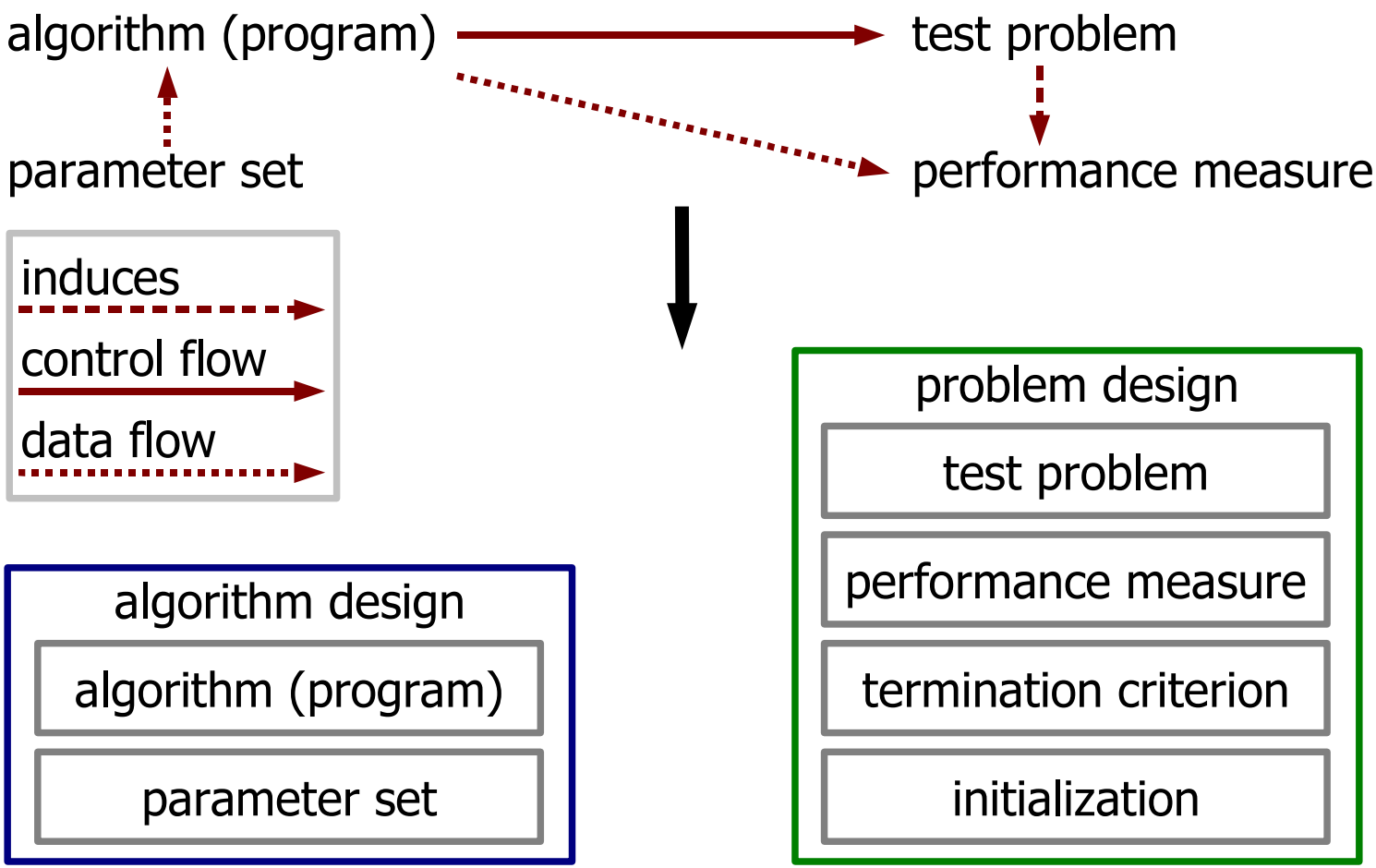
- Kein Standardschema für Präsentation experimenteller Ergebnisse
- Erkenntnisgewinn und Reproduzierbarkeit zumeist problematisch
- Experimentelle Methodologie gerade im Umbau, vor allem durch Integration statistischer Techniken
- Dies ist wissenschaftliche Sicht, im 'realen' Einsatz Bedingungen oft noch viel schwerer (Laufzeit!)

Andere Disziplinen haben Standards für die Aufzeichnung experimenteller Ergebnisse. Warum?

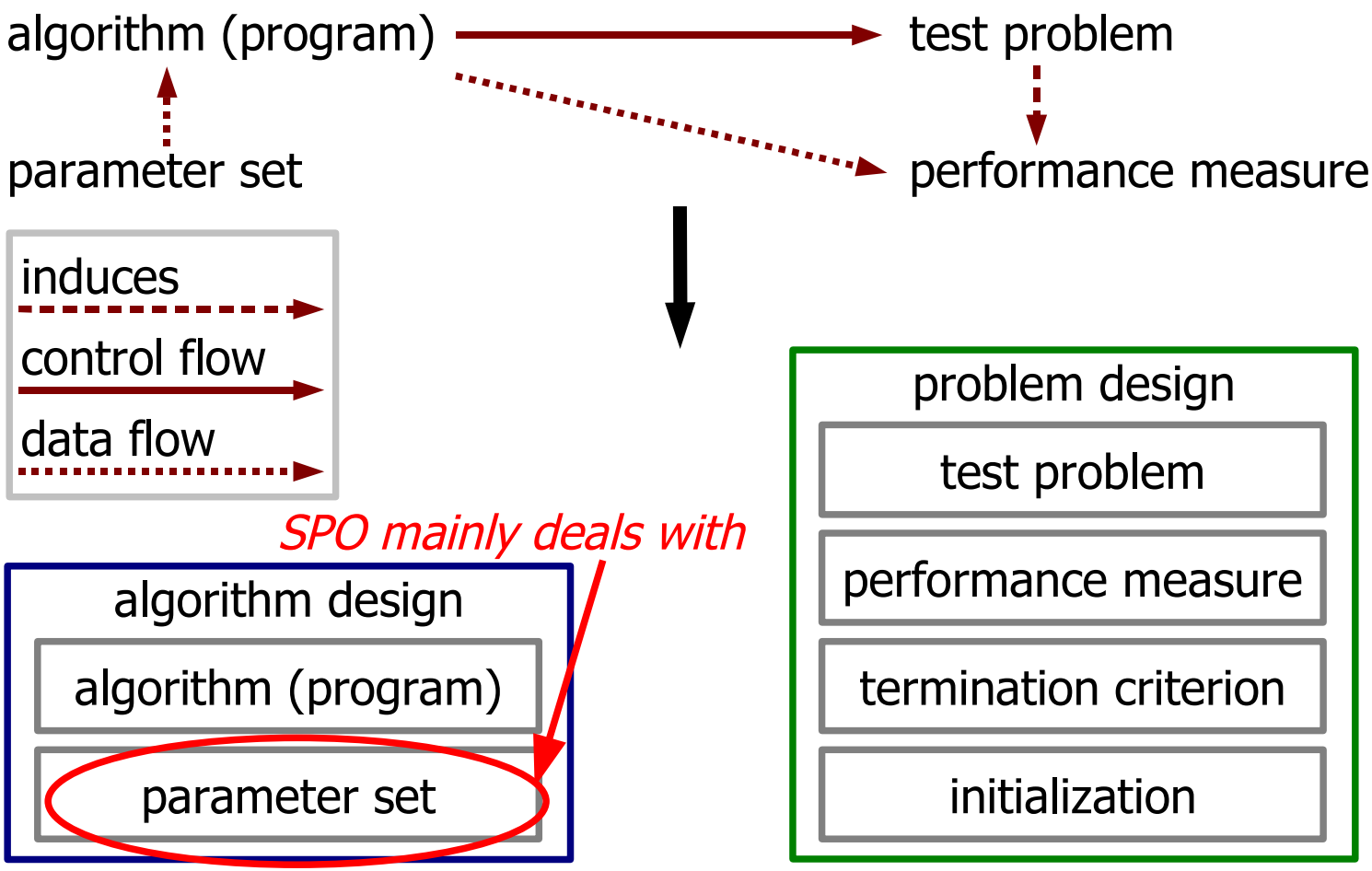
- Naturwissenschaften: Lange Tradition, Aufbau oft relativ schnell, Experiment selbst langsam ($\Rightarrow$ Ergebnisse sind wertvoll)
- Informatik: Kurze Tradition, Aufbau (Implementierung) dauert, Experiment selbst relativ schnell ($\Rightarrow$ Ergebnisse flüchtig)

Teil II: Die *Sequentielle Parameter-Optimierung* (SPO)

Komponenten von Experimenten in EC



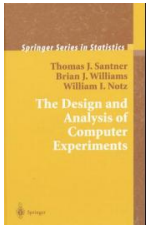
Komponenten von Experimenten in EC



Wurzeln und Definitionen

SPO integriert Elemente von

Design of Experiments (DOE)



Design and Analysis of Computer Experiments (DACE)

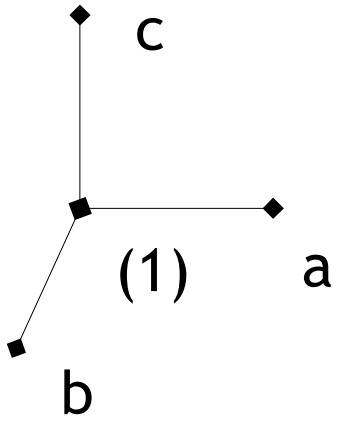
- 1 Experiment := 1 Lauf des Optimierers
- Designvariablen / Faktoren := (Algorithmen-)Parameter

- Endogene Faktoren: Werden im Lauf modifiziert
- Exogene Faktoren: Werden im Lauf konstant gehalten
  - Problemspezifische Faktoren $\Rightarrow$ Problemdesign
  - Algorithmenspezifische Faktoren $\Rightarrow$ Algorithmendesign

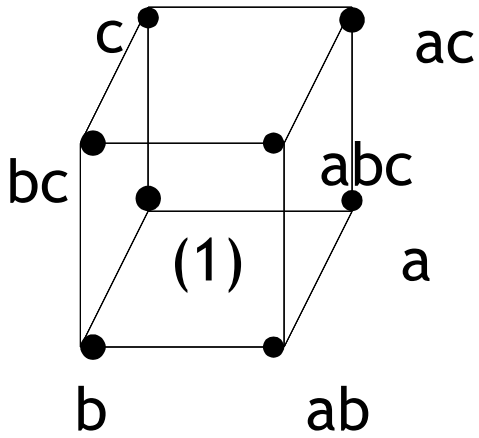
Design of Experiments

Verschiedene klassische (2 Level) Designs:

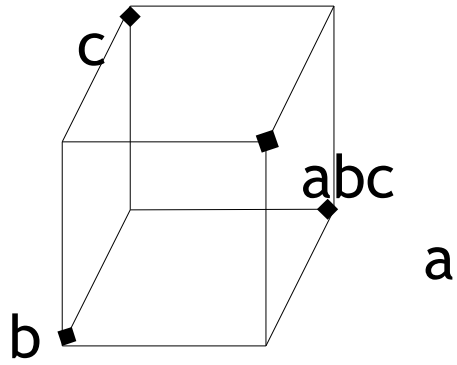
- One-factor Designs (i)
- Full factorial Designs (ii)
- Fractional factorial Designs (iii)



i)



ii)



iii)

Raumfüllende Designs

- Beliebt, da einfach: Latin Hypercube Designs (LHD)
- Klassisch: Designpunkte an den Rändern (niedrige und hohe Level)
- DACE: Designpunkte (fast) zufällig im Inneren des Suchraums

Glasfenster in Cambridge (UK) in Erinnerung an R. A. Fisher

Gütemaße

MBF (mean best fitness):

- Stabil für viele Wiederholungen
- Ausreißer / Asymmetrische Verteilungen stören stark
- Hängt von gegebenem Zeitlimit ab

SR (success rates):

- Gewünschte Qualität muss angegeben werden (oft nicht-trivial)
- Hängt von gegebenem Zeitlimit ab

AES (average evaluation count):

- Was passiert mit fehlgeschlagenen Läufen?
- Gewünschte Qualität muss angegeben werden, gleiche Probleme wie für SR

Beste Fitness aus N Läufen:

- Hängt von der Anzahl der Läufe (Ressourcen) ab
- Sinnvoll für viele praktische Probleme

Zielvorgabe

Cohens Untersuchung von 1990 (alle Artikel der AAAI Konferenz):

- Kein wesentlicher Zusammenhang zwischen Experiment und Theorie
- 60% haben nur auf einer Probleminstanz getestet
- 80% gaben keine Erklärung für das ermittelte Ergebnis
- 16% haben eine Hypothese oder Zielvorgabe angegeben

Paul R. Cohen

SPO Übersicht

Phase I	Konstruktion des Experimentes
Phase II	SPO core: Parameteroptimierung
Phase III	Auswertung

- Die Phasen I und III definieren die experimentelle Methodologie (wie man Experimente durchführt)
- Phase II ist das Parameter-Tuning Verfahren, hier der SPO core, aber andere Verfahren möglich
- SPO ist *per se* kein Meta-Algorithmus: Wir sind hauptsächlich an guten Parameterkombinationen interessiert, nicht an Lösungen des ursprünglichen Problems

SPO Ablauf

1. *Vor-experimentelle* Planung
2. *Wissenschaftliche* These
3. *Statistische* Hypothese
4. Experimentelles *Design*: Problem, Nebenbedingungen, Start-/Endbedingungen, Gütemaße, Parameter des Algorithmus

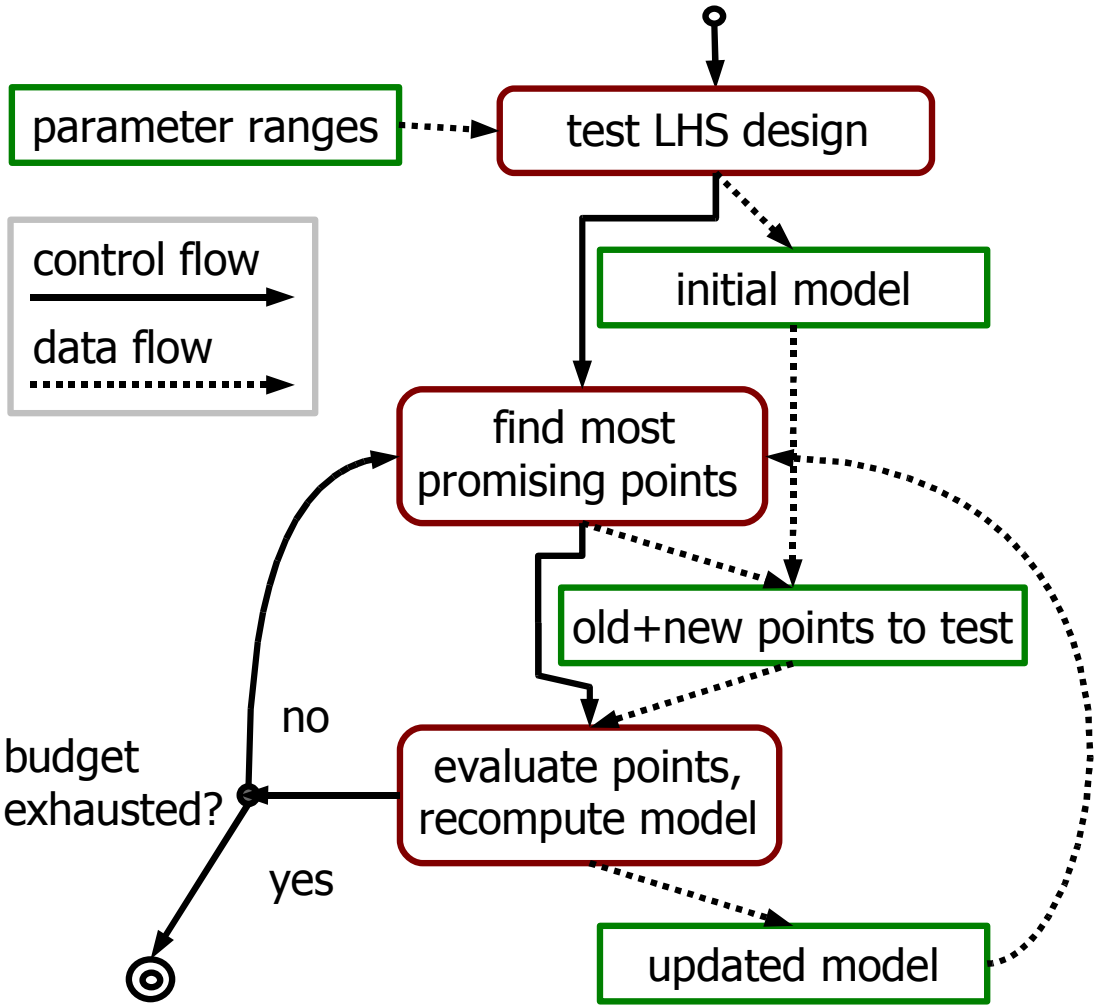
---

5. *Experimente*
6. Statistisches *Model* und Vorhersage (DACE). Auswertung und Visualisierung
7. Lösung gut genug?
   - Yes: Gehe nach Schritt 8
   - No: Verbessere das Design (Optimierung). Gehe nach Schritt 5

---

8. *Annehmen/verwerfen* der statistischen Hypothese
9. Objektive *Interpretation* der Ergebnisse aus dem vorherigen Schritt

SPO Core: Ablauf



SPO Core: Unsere Standardmethode

Heuristik für stochastisch gestörte Funktionswerte

- Startet mit Latin Hypercube Design (LHD): Gute Verteilung der Startpunkte, kleine Anzahl von Läufen
- Sequentielle Verbesserung, unterstützt von erlerntem Model
- Erwartete Verbesserung: Kompromiss zwischen Optimierung und Modelgüte
- Budget Konzept: Die besten Konfigurationen werden erneut ausgewertet
- Fairness: Neu vorgeschlagene Konfigurationen werden ebenso häufig ausgewertet wie die bisher beste

Tabelle: Momentan beste Konfigurationen, initiales LHD

$\frac{\lambda}{\mu}$	τ_0	*restart threshold*	*#eval best*	*config ID*	*result*	*std. deviation*
10.075	0.4180	22	4	42	0.0034	0.0058
5.675	0.7562	2	4	72	0.0042	0.0035
10.625	0.0796	5	4	57	0.0042	0.0054
4.905	0.1394	10	4	86	0.0047	0.0068
3.585	0.0398	13	4	81	0.0048	0.0056
3.145	0.0200	8	4	3	0.0050	0.0056
2.595	0.7960	4	4	83	0.0065	0.0048
2.375	1.8905	7	4	64	0.0113	0.0115

SPO Core: Unsere Standardmethode

Heuristik für stochastisch gestörte Funktionswerte

- Startet mit Latin Hypercube Design (LHD): Gute Verteilung der Startpunkte, kleine Anzahl von Läufen
- Sequentielle Verbesserung, unterstützt von erlerntem Model
- Erwartete Verbesserung: Kompromiss zwischen Optimierung und Modelgüte
- Budget Konzept: Die besten Konfigurationen werden erneut ausgewertet
- Fairness: Neu vorgeschlagene Konfigurationen werden ebenso häufig ausgewertet wie die bisher beste

Tabelle: Momentan beste Konfigurationen, Iteration 7

$\frac{\lambda}{\mu}$	τ_0	*restart threshold*	*#eval best*	*config ID*	*result*	*std. deviation*
5.675	0.7562	2	4	72	0.0042	0.0035
10.625	0.0796	5	4	57	0.0042	0.0054
4.905	0.1394	10	4	86	0.0047	0.0068
3.585	0.0398	13	4	81	0.0048	0.0056
3.145	0.0200	8	4	3	0.0050	0.0056
2.595	0.7960	4	4	83	0.0065	0.0048
3.866	0.0564	4	8	106	0.0096	0.0065
2.375	1.8905	7	4	64	0.0113	0.0115
...	...	...	...	...	...	...
10.075	0.4180	22	8	42	0.0177	0.0181

SPO Core: Unsere Standardmethode

Heuristik für stochastisch gestörte Funktionswerte

- Startet mit Latin Hypercube Design (LHD): Gute Verteilung der Startpunkte, kleine Anzahl von Läufen
- Sequentielle Verbesserung, unterstützt von erlerntem Model
- Erwartete Verbesserung: Kompromiss zwischen Optimierung und Modelgüte
- Budget Konzept: Die besten Konfigurationen werden erneut ausgewertet
- Fairness: Neu vorgeschlagene Konfigurationen werden ebenso häufig ausgewertet wie die bisher beste

Tabelle: Momentan beste Konfigurationen, Iteration 12

$\frac{\lambda}{\mu}$	τ_0	*restart threshold*	*#eval best*	*config ID*	*result*	*std. deviation*
10.625	0.0796	5	10	57	0.0024	0.0038
5.675	0.7562	2	5	72	0.0042	0.0031
4.905	0.1394	10	4	86	0.0047	0.0068
3.585	0.0398	13	4	81	0.0048	0.0056
3.145	0.0200	8	4	3	0.0050	0.0056
11.620	0.0205	2	10	111	0.0055	0.0052
2.595	0.7960	4	4	83	0.0065	0.0048
3.866	0.0564	4	8	106	0.0096	0.0065

SPO Core: Unsere Standardmethode

Heuristik für stochastisch gestörte Funktionswerte

- Startet mit Latin Hypercube Design (LHD): Gute Verteilung der Startpunkte, kleine Anzahl von Läufen
- Sequentielle Verbesserung, unterstützt von erlerntem Model
- Erwartete Verbesserung: Kompromiss zwischen Optimierung und Modelgüte
- Budget Konzept: Die besten Konfigurationen werden erneut ausgewertet
- Fairness: Neu vorgeschlagene Konfigurationen werden ebenso häufig ausgewertet wie die bisher beste

Tabelle: Momentan beste Konfigurationen, Iteration 17

$\frac{\lambda}{\mu}$	τ_0	*restart threshold*	*#eval best*	*config ID*	*result*	*std. deviation*
10.625	0.0796	5	20	57	0.0023	0.0034
4.881	0.0118	8	20	116	0.0028	0.0029
5.675	0.7562	2	5	72	0.0042	0.0031
4.905	0.1394	10	4	86	0.0047	0.0068
3.585	0.0398	13	4	81	0.0048	0.0056
3.145	0.0200	8	4	3	0.0050	0.0056
11.620	0.0205	2	10	111	0.0055	0.0052
7.953	0.0213	2	10	114	0.0065	0.0055

SPO Core: Unsere Standardmethode

Heuristik für stochastisch gestörte Funktionswerte

- Startet mit Latin Hypercube Design (LHD): Gute Verteilung der Startpunkte, kleine Anzahl von Läufen
- Sequentielle Verbesserung, unterstützt von erlerntem Model
- Erwartete Verbesserung: Kompromiss zwischen Optimierung und Modelgüte
- Budget Konzept: Die besten Konfigurationen werden erneut ausgewertet
- Fairness: Neu vorgeschlagene Konfigurationen werden ebenso häufig ausgewertet wie die bisher beste

Tabelle: Momentan beste Konfigurationen, Ende (Iteration 49)

$\frac{\lambda}{\mu}$	τ_0	*restart threshold*	*#eval best*	*config ID*	*result*	*std. deviation*
7.486	0.0329	13	50	140	0.0014	0.0022
6.367	0.0452	8	50	121	0.0015	0.0021
9.572	0.0536	11	50	134	0.0018	0.0031
6.024	0.0158	10	50	119	0.0019	0.0033
10.294	0.0229	8	50	133	0.0021	0.0036
6.798	0.0679	6	50	120	0.0021	0.0030
10.625	0.0796	5	50	57	0.0022	0.0032
4.8819	0.0118	8	20	116	0.0028	0.0029

Berichte

Vorgeschlagene Struktur:

ER-1: **Grundfrage** Was ist die Hauptfrage, die wir klären wollen?
ER-2: **Vor-experimentelle Planung** Erste Tests, die zur Zielvorgabe führen
ER-3: **Zielvorgabe** Genaue (wissenschaftliche) These und abgeleitete statistische Hypothesen
ER-4: **Setup** Problemdesign, Algorithmendesign, Gütemaß, Start- und Endbedingung u.s.w.
ER-5: **Ergebnis/Visualisierung** Die (wichtigsten) erhaltenen Daten und erste grafische Darstellungen
ER-6: **Beobachtungen** Abweichungen vom erwarteten Ergebnis, oder auffällige Muster, aber keine Bewertung
ER-7: **Diskussion** Resultate der statistischen Tests und die (subjektive) Einordnung/Bewertung von Ergebnissen und Beobachtungen

Was ist die Bedeutung von Parametern?

Sind Parameter "schlecht"?

Dagegen:

- Viele Parameter verwirren den Benutzer
- Es ist oft nicht trivial, die Interaktionen Parameter vs. Problem und Parameter vs. Parameter zu verstehen

  $\Rightarrow$ Parameter machen es schwieriger, Algorithmen zu bewerten

Dafür:

- Parameter sind eine einfache Möglichkeit, Algorithmen anzupassen
- Viele der erfolgreichsten EA haben etliche Parameter

Mögliche Alternativen?

Parameterlose EAs:

- Einfach anzuwenden, aber was ist mit Güte und Robustheit?
- Wohin sind die Parameter verschwunden?

Üblicherweise ein Mix aus:

- Default Werten, die maximale Güte für gute Robustheit opfern
- Heuristische Regeln, anwendbar auf *viele* aber nicht *alle* Probleme; funktionieren wahrscheinlich nicht für ganz neue Probleme
- (Selbst-)Adaptation, kann aber nur wenige Parameter gleichzeitig lernen (meist 1), und reduziert nicht unbedingt die Parameterzahl

$\Rightarrow$ Wir können Parameter loswerden, aber zumeist auf Kosten von Güte oder Robustheit (oder beiden)

$\Rightarrow$ In den meisten Fällen bleiben Parameter übrig

Argumente gegen das Tunen von Parametern

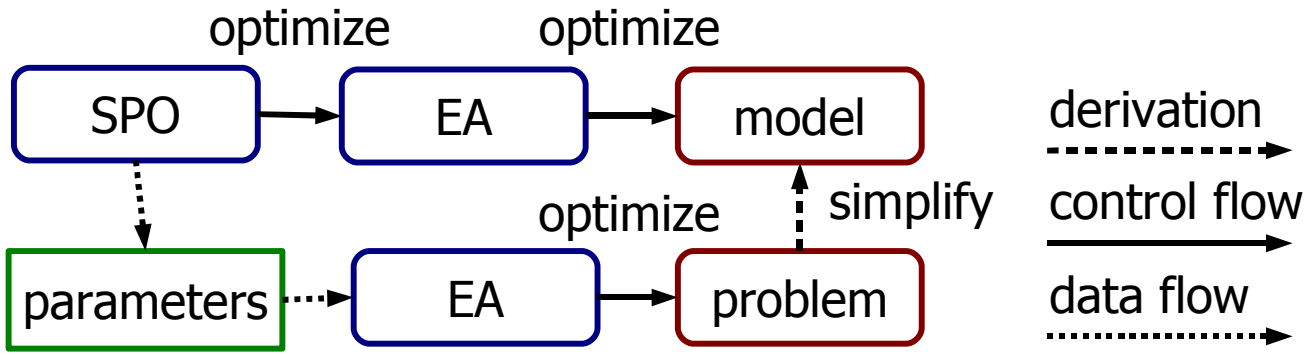
. . . und wie man ihnen (hoffentlich) begegnen kann

a) Der Meta-Algorithmus (1. Optimiere die Parameter eines Verfahrens, dass dann 2. benutzt wird um das originale Problem zu lösen) ist ein Optimierverfahren und fällt unter das NFL[1]

b) Parameter-Optimierung ist zu teuer

Mögliche Lösungen für b):

- Schon ein kleines Sample über den Parameterraum kann hilfreich sein
- Für wiederholend auftretende Probleme zahlt sich der Aufwand aus
- Parameter können (hoffentlich) mithilfe einfacher Modellprobleme optimiert werden



[1] No Free Lunch Theorem

Sinn und Unsinn des Vergleichens

Das NFL erklärt uns, was wir schon vermutet haben:

- Das universal beste Optimierverfahren gibt es nicht
- Die Güte eines Verfahrens ist stark abhängig vom zu lösenden Problem und den Randbedingungen (Ressourcen etc.)

Das bedeutet:

- Die gestellte Frage/Aufgabe beeinflusst die Bedeutung der erhaltenen Ergebnisse maßgeblich
- Der Fokus von Vergleichen zwischen Verfahren sollte sich verändern von:

  Welcher Algorithmus ist besser?

  zu

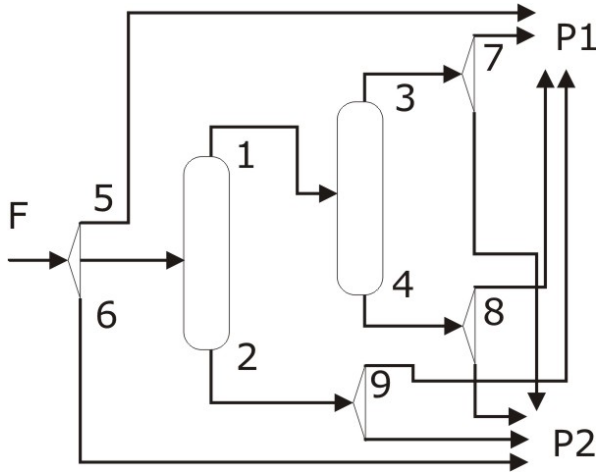
  Für was genau ist der Algorithmus gut?

Teil III: Beispiele

Beispiel (Real-World): Trennprozessoptimierung

Aufgabe: Design eines unscharfen Trennprozesses

- Separiere 3-Komponenten Zufluss in 2 verschiedene Mischungen
- 9 Real-Variablen steuern Kolonnen und Stomteiler
- 18 (17 versteckte) Nebenbedingungen, diskretisierte Bestrafungen



- Shortcut-Simulator prüft physikalische Gültigkeit der generierten Prozesse
- Kommerzieller Simulator (langsam) bewertet gültige Layouts
- Auch mit Shortcut-Simulator, nur wenige ($\approx 10^5$) Auswertungen möglich

Vor-experimentelle Planung

- Erste Tests mit Standard (μ, κ, λ)-ES ergeben:
  Es ist schwer, gültige Lösungen zu finden
- ρ Maß $< 10^{-5}$
- Weitere Tests geben Hinweis auf nicht-konvexen gültigen Suchraum
- Manuelles Tuning ergibt Success Rates p(gültig) < 0.1.

Tabelle: Parameterbereiche für manuelles Tuning

Parameter	Bereich
Populationsgröße μ	10-20
Maximales Alter κ	1-20
Selektionsdruck λ/μ	1-5
Lernrate τ	0.05-0.2

Task

- Wissenschaftliche These: $\exists$ eine Parameterkonfiguration die zu hohen Erfolgsraten (für das Erreichen gültiger Lösungen) führt
- Statistische Hypothese: $SR(\text{SPO-tuned}) - SR(\text{man-tuned}) > 0$
- Im Folgenden: Kommerzieller Simulator ausgeschaltet, somit Auffinden der besten Lösungen verschoben auf einen 2. Schritt

Setup

- Problemdesign:
  - Lauflänge in Auswertungen $\Leftarrow$ 10000 (5-10 mins).
  - Gütemaß $\Leftarrow$ MBF
    (ungültige Lösungen werden bestraft und sind immer schlechter als gültige)
- Algorithmendesign:

Tabelle: Parameterbereiche für SPO (vergrößerte Bereiche).

Parameter	min	max
Populationsgröße μ	10	100
Maximales Alter κ	1	50
Selektionsdruck λ/μ	1	10
Lernrate τ	0	1

- Experimentelles Ziel: Finde Parameterbereich, der das MBF minimiert (Fitness ungültiger Punkte $\geq 10^6$).

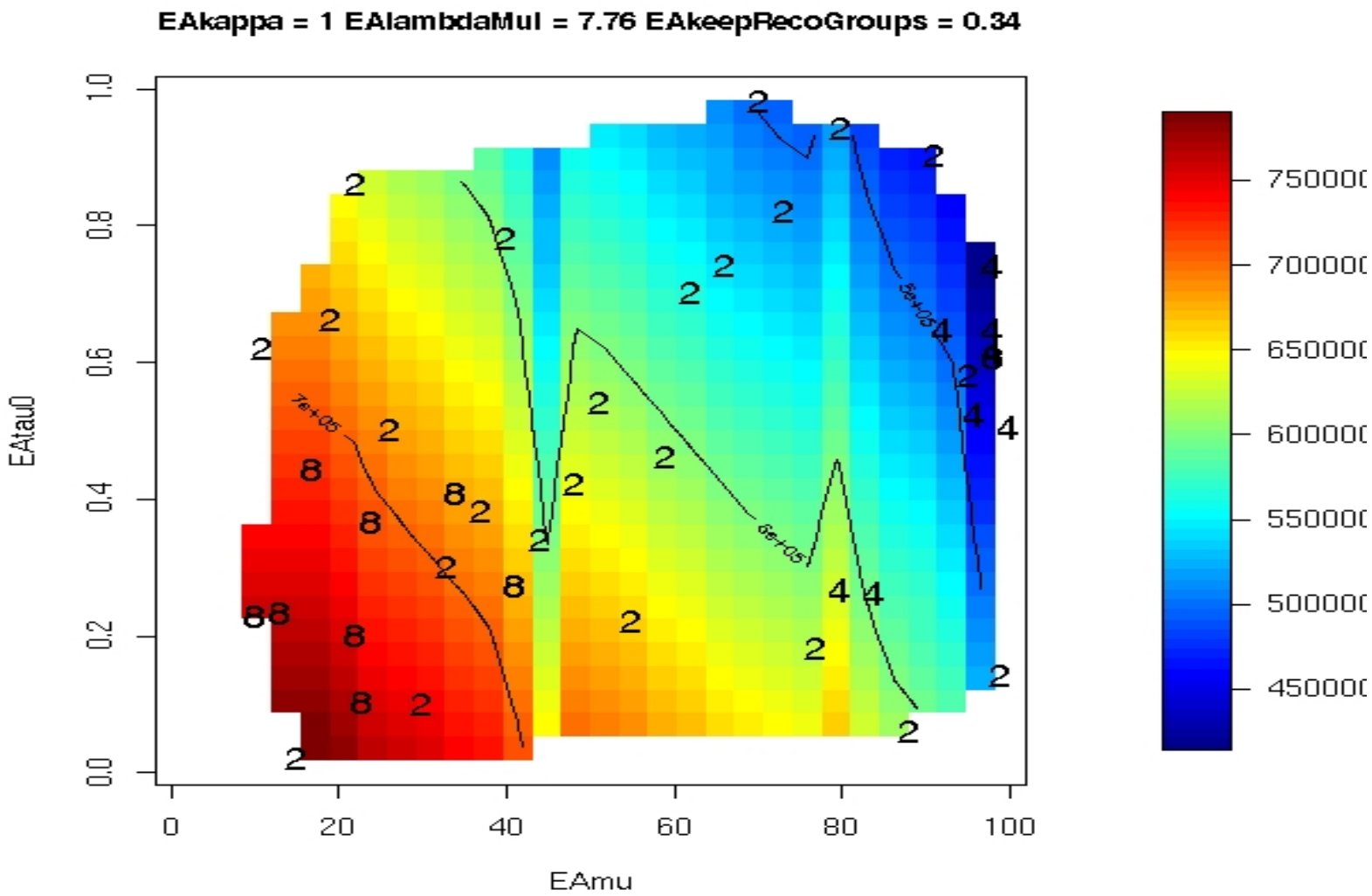
Ergebnisse

- Zielfunktion ist teuer, versuche die Anzahl der Läufe gering zu halten
- 25 initiale Designpunkte (LHD).
- Initiale Konfigurationen $r = 2$ mal wiederholt
- Modell erweitert mit 1 bestem, 4 erwarteten besten, 4 das Modell verbessernden Punkten pro Iteration.

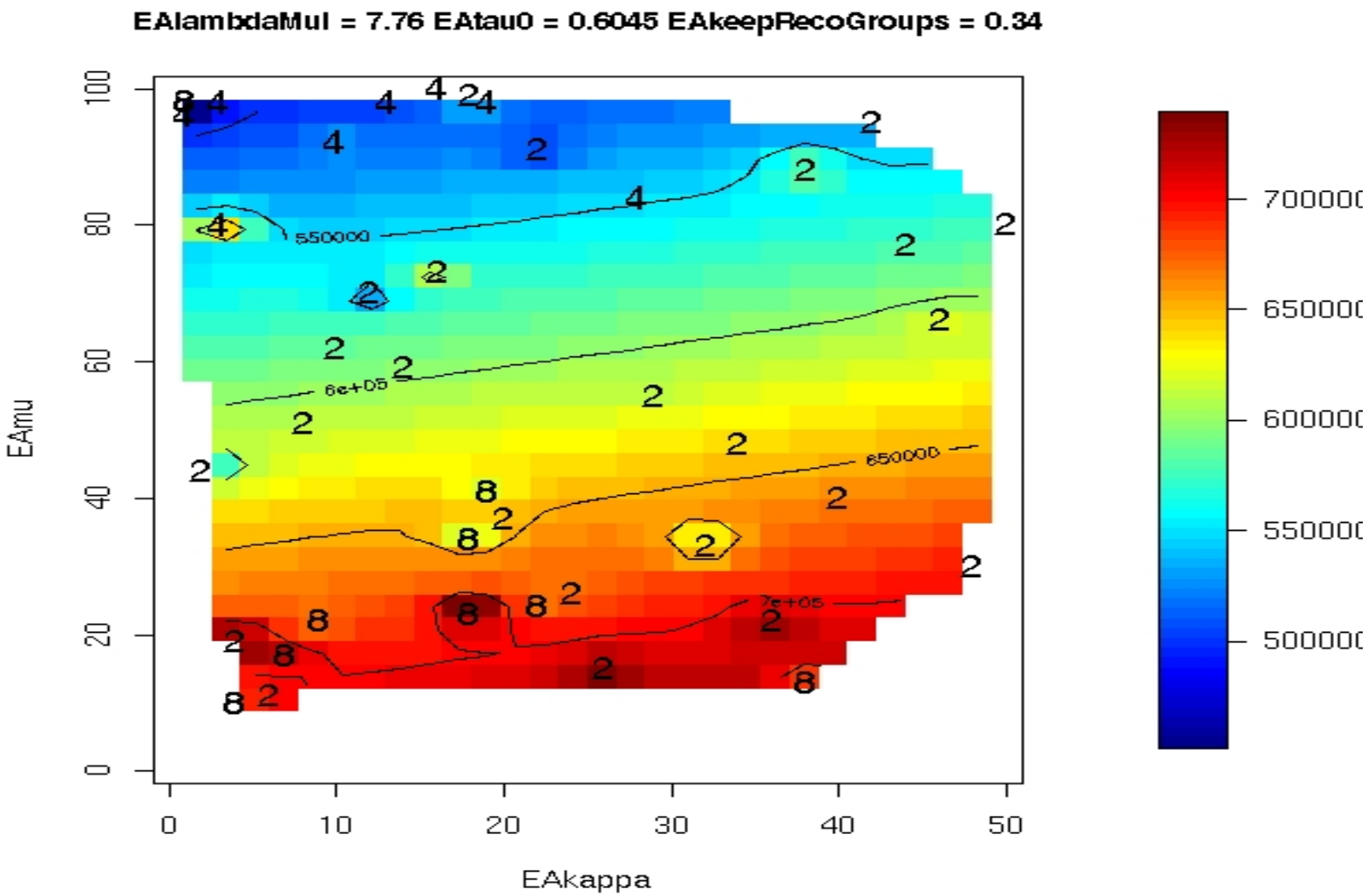
Tabelle: Beste Konfigurationen nach initialem Design + 3 Iterationen, altes MBF $\approx 1E6$.

κ	μ	λ/μ	τ	*recGrp*	*r*	*conf*	MBF	std.dev.
2	44	7.03	0.34	0.02	2	14	3.2306E+05	1.7635E+04
1	98	7.7576	0.6045	0.3425	8	27	3.2516E+05	3.7345E+04
32	33	9.406	0.3	0.94	2	16	3.2704E+05	301
22	91	6.238	0.9	0.42	2	13	3.3018E+05	3.0601E+04
16	100	5.342	0.5035	0.1695	4	32	3.3048E+05	1.8736E+04
42	95	3.466	0.58	0.22	2	21	3.3644E+05	3.2716E+04
29	55	3.862	0.22	0.26	2	10	3.3916E+05	3.668E+04
12	70	8.614	0.98	0.46	2	22	3.6124E+05	2.9507E+04
1	96	5.8865	0.5215	0.4075	4	26	3.7467E+05	2.6731E+04
28	84	1.09	0.26	0.14	4	23	4.8457E+05	3.1373E+05
19	41	9.8763	0.2724	0.5165	8	39	4.969E+05	3.0772E+05
18	34	9.1337	0.4074	0.4735	8	38	4.9819E+05	3.0607E+05
3	98	7.0845	0.6455	0.1925	4	29	5.0438E+05	3.0314E+05

Visualisierung



Visualisierung



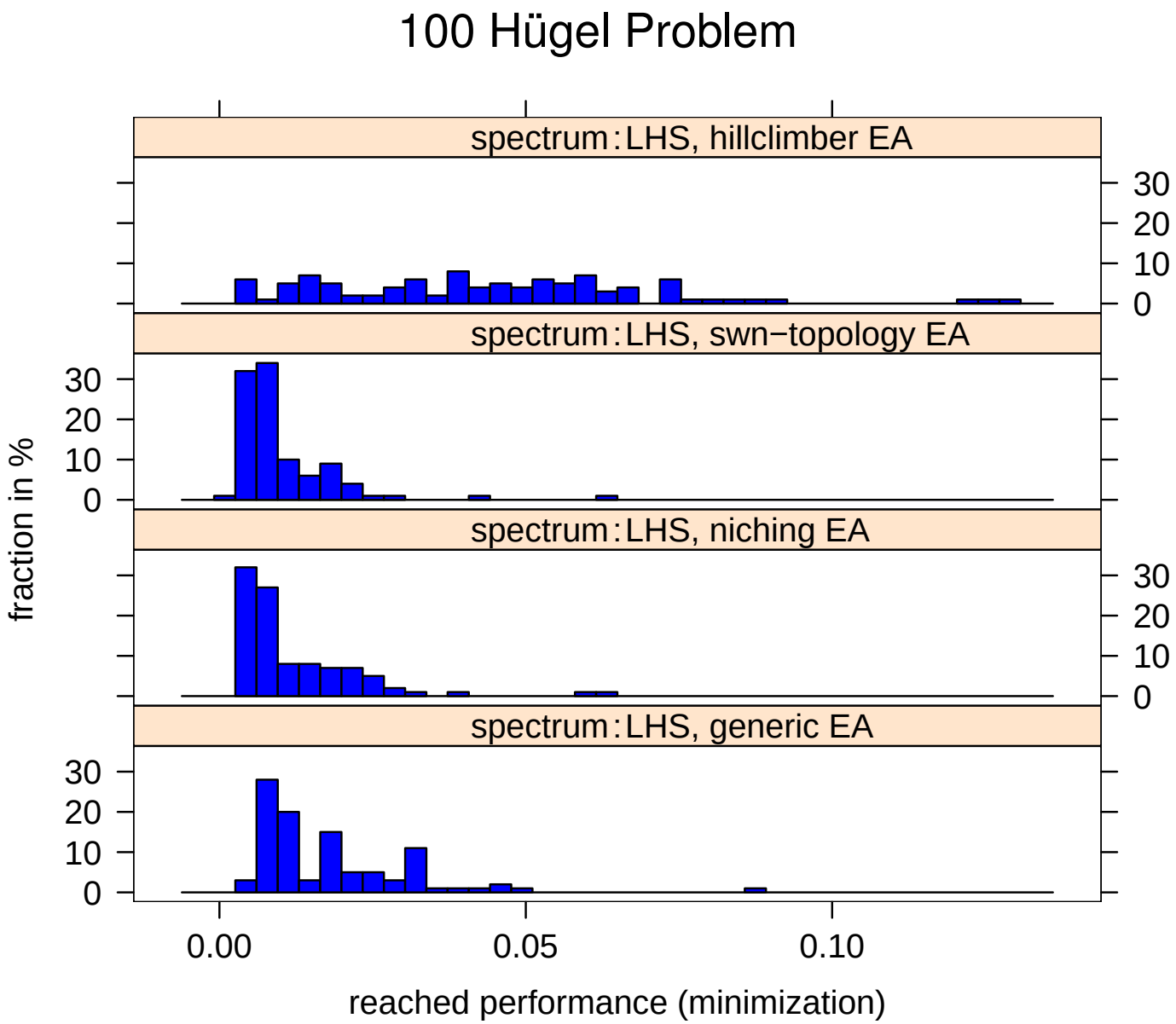
Annahme/Ablehnen der statistischen Hypothese

- Wir wählen Konfiguration 27 ($\mu = 98, \kappa = 1, \lambda/\mu = 7.76, \tau = 0.6$), da die Güte gut ist (2nd) und die Zahl der Wiederholungen hoch ist (8 $\Rightarrow$ Stabilität
- Validierung des Ergebnisses: 40 neue Läufe, Maß SR.
- SR $\approx$ 65%, signifikant besser als 10%.

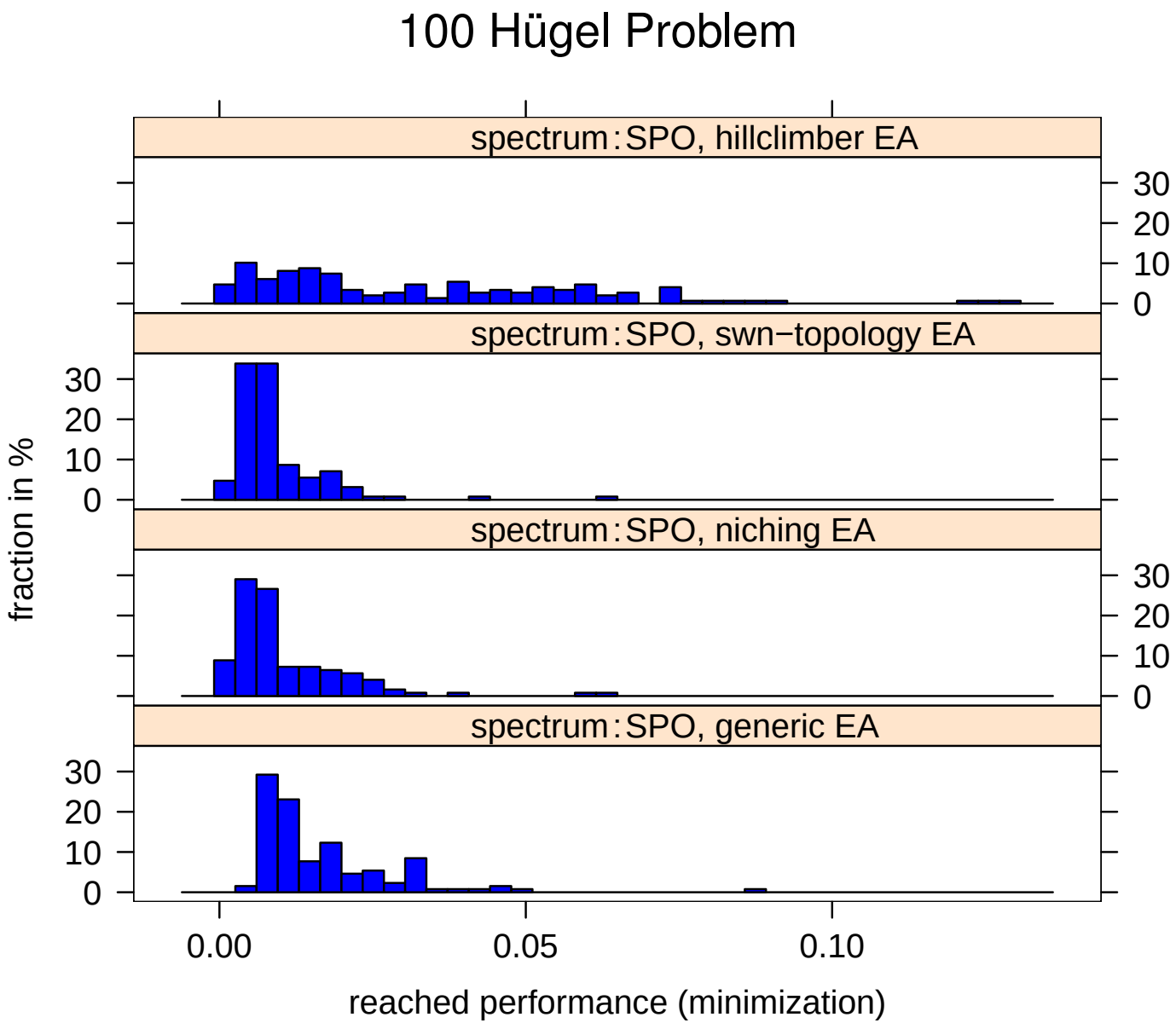
Interpretation der Ergebnisse

- (Erstes) Ziel erfüllt: ES Parameter für hohe SR gefunden
- Bessere Güte vermutlich möglich: μ Wert am Bereichsrand
- Parameter κ und *recGrp* haben nur kleinen Einfluss
- Mögliche Erklärungen:
  - Erhöhte Populationsgröße induziert grössere (benötigte?) Diversität
  - Großer Selektionsdruck und hohe Lernrate führen zu schneller Reaktion des EA wenn Bereiche gefunden werden, die weniger Nebenbedingungen verletzen

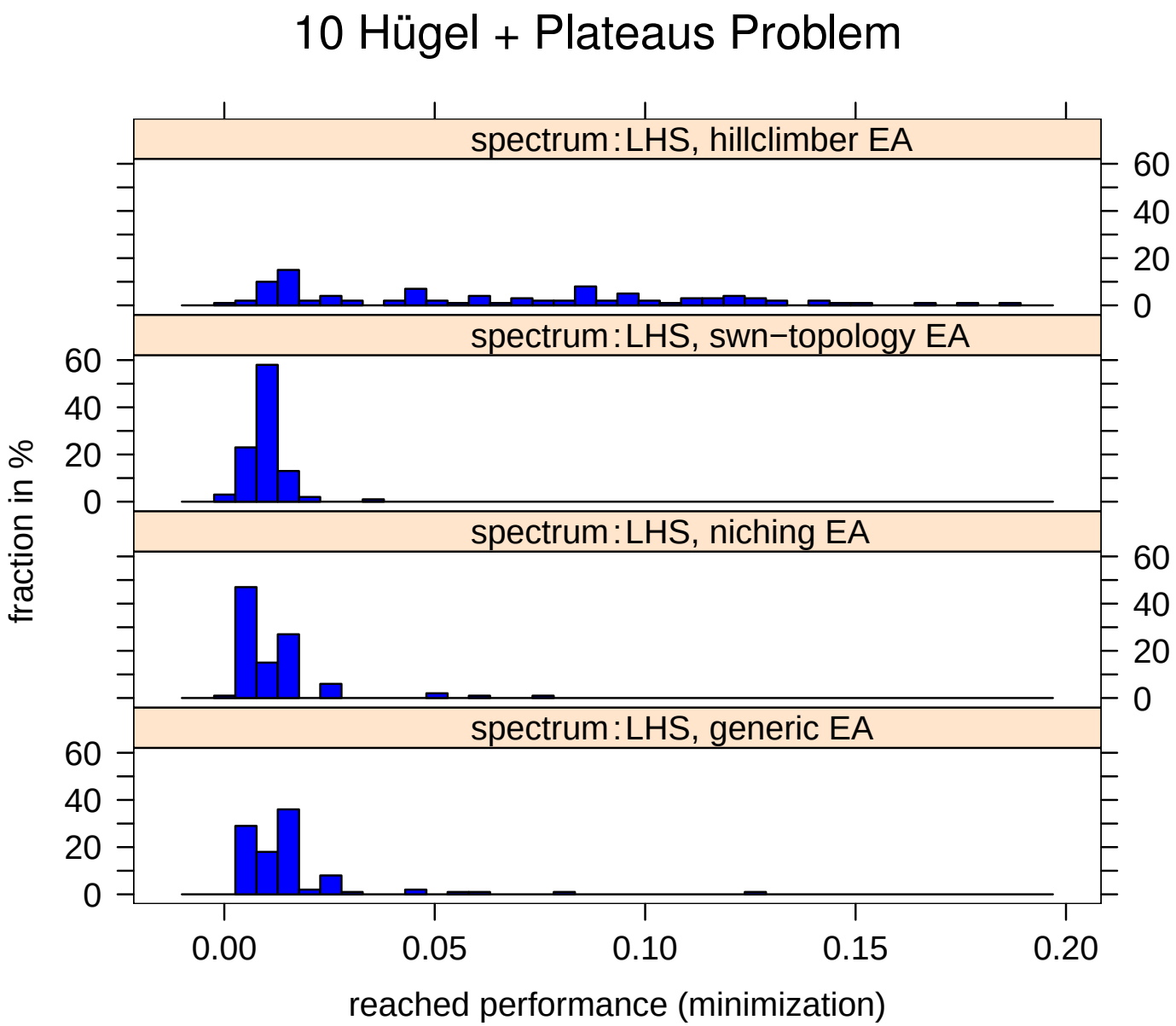
Anpassen von EA auf zwei ähnliche Probleme



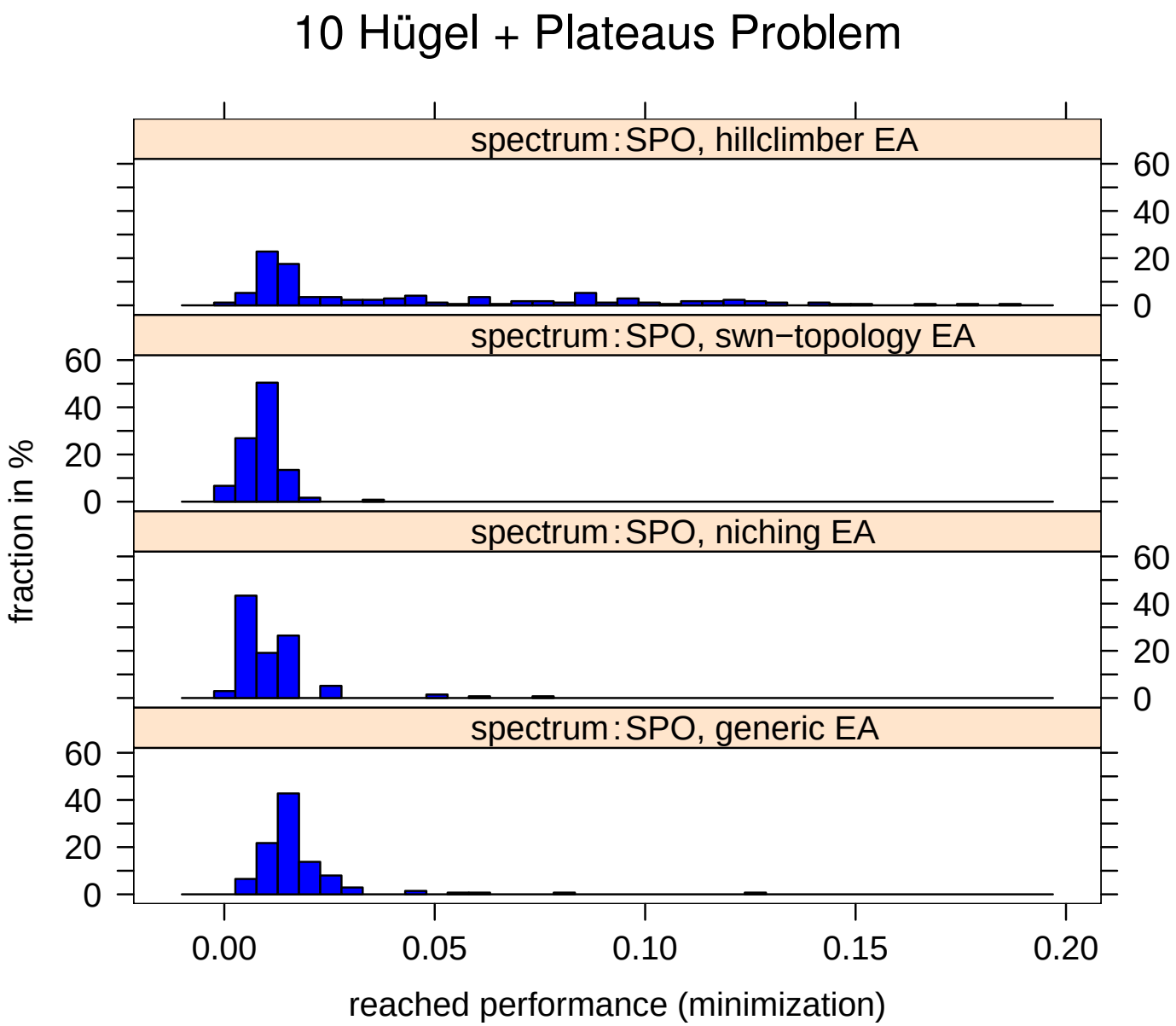
Anpassen von EA auf zwei ähnliche Probleme



Anpassen von EA auf zwei ähnliche Probleme



Anpassen von EA auf zwei ähnliche Probleme



Empirische Ergebnisse

Generell:

a) Einige Parameterkombinationen funktionieren gar nicht

b) Eine oft auftretende Situation:
   $\frac{1}{3}$ der Konfigurationen führen zu sehr schlechter Güte
   $\frac{1}{3}$ sind in der 'interessanten' Region (gut)
   $\frac{1}{3}$ liegen irgendwo dazwischen (nicht wirklich interessant)

c) Die von SPO auffindbare Güte hängt stark von der Anwendung ab, ist aber bei absoluten Distanzen besonders groß

Zusammenfassung

- Gerade für EA kann auf die experimentelle Überprüfung nicht verzichtet werden
- Strukturiertes Experimentieren kann zu völlig anderen (viel besseren) Ergebnissen führen
- Berichte und Visualisierungen helfen dabei, die Zusammenhänge zu verstehen und ermöglichen Wiederholbarkeit
- Durch mehrfache Wiederholung kann man das ’Rauschen’ in den Griff bekommen
- Die erhaltene Güte ist extrem abhängig von Gütemaß und Randbedingungen

Was wir durch Parameter-Optimierung erhalten:

- Eine (nahezu) optimale Konfiguration des EA, die Vergleiche zwischen Verfahren auf hohem Niveau ermöglicht
- Eine Qualitätsabschätzung für die vorher eingesetzten Konfigurationen
- *Ein erstes Bild ergibt sich aus einem relativ kleinen LHD, Gitterdesign oder randomisierten Design*