Mehrkriterielle Optimierung
6.3.2012

Carsten Franke

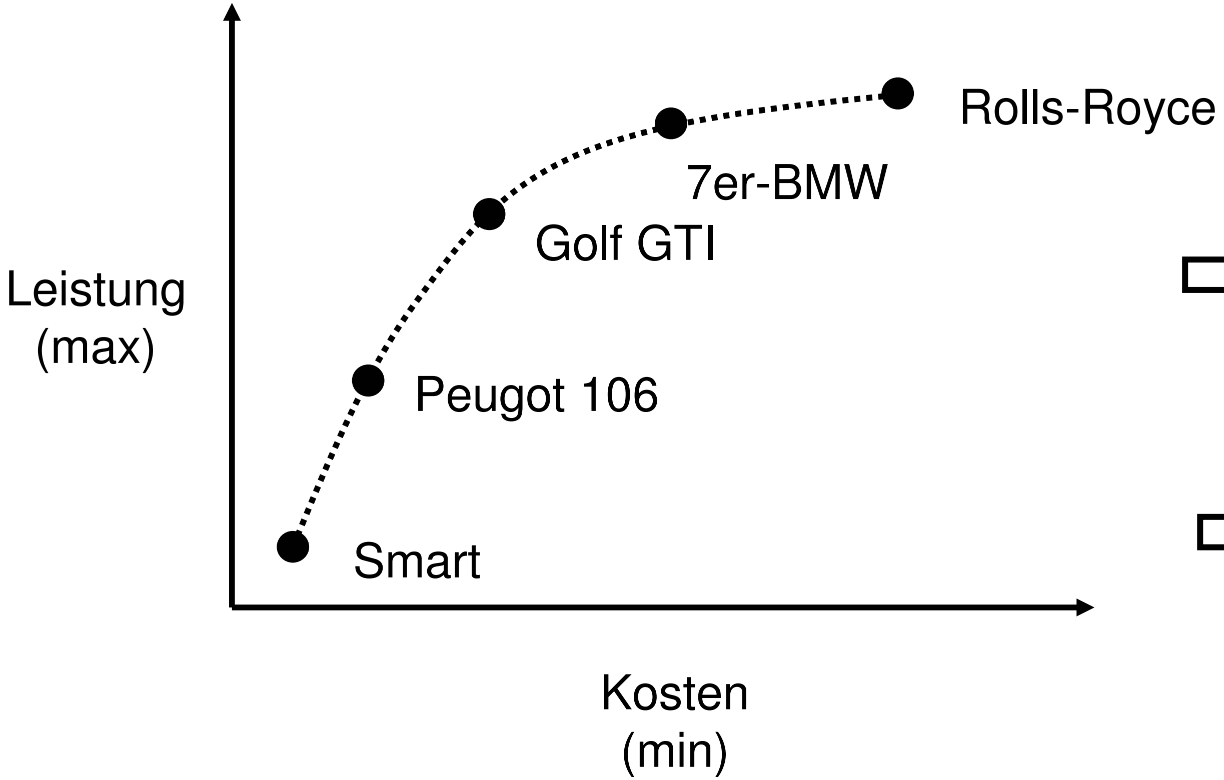
Lernziele dieser Vorlesung

- Entwicklung des Grundverständnis für mehrkriterielle Optimierungsprobleme
- Erlernen der Pareto-Begrifflichkeiten
- Verständnis der klassischen Optimierungsmethoden
- Erlernen des ersten evolutionären mehrkriteriellen Verfahrens

Einleitung

- Bisher nur Optimierungen mit einem Zielkriterium
- Nun, Zielkritierien mit Zielkonflikten für ein gemeinsames Optimierungsproblem
- Beispiele:
  - Autokauf (Preis versus Leistung)
  - Motorenleistung (Hubraum versus Verbrauch)
  - Prüfungen (Arbeitszeit versus Freizeit)

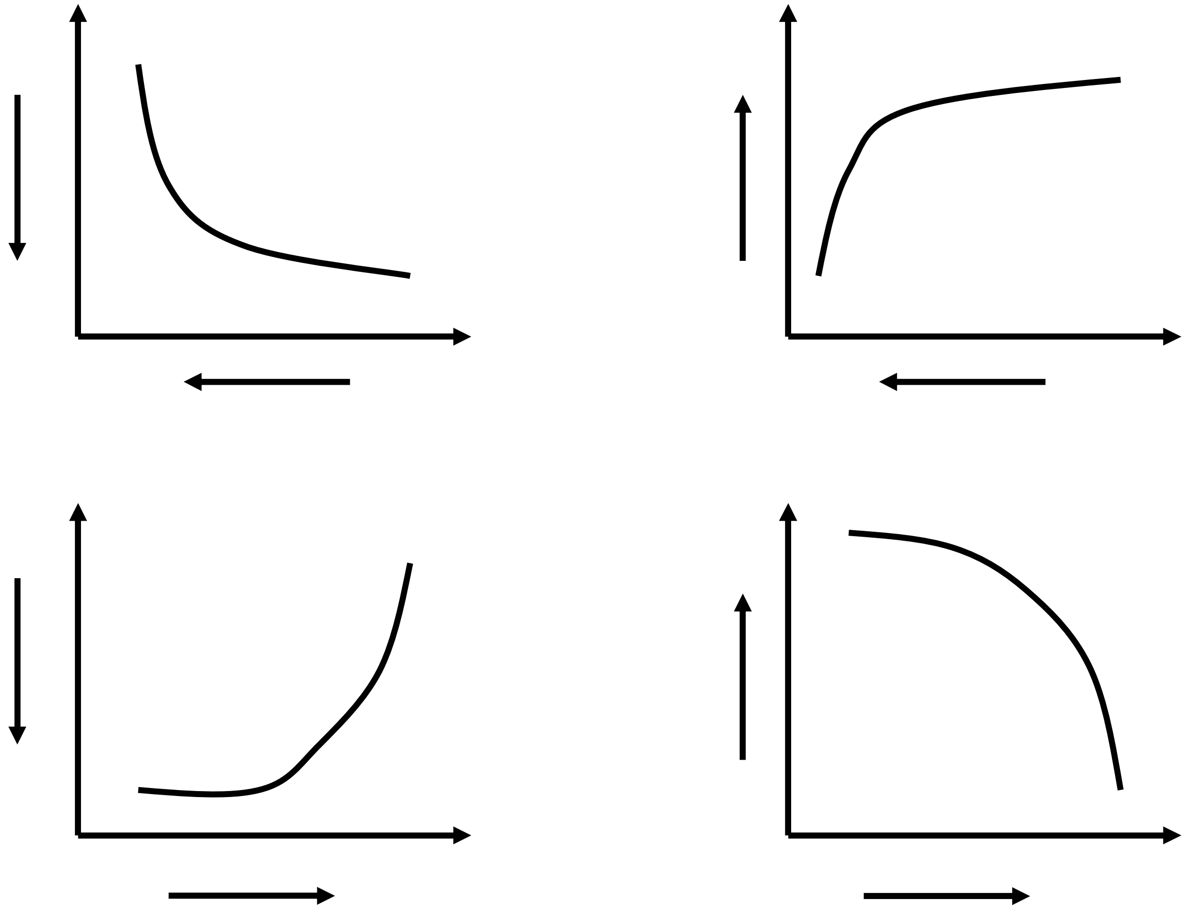
Beispiel Autokauf



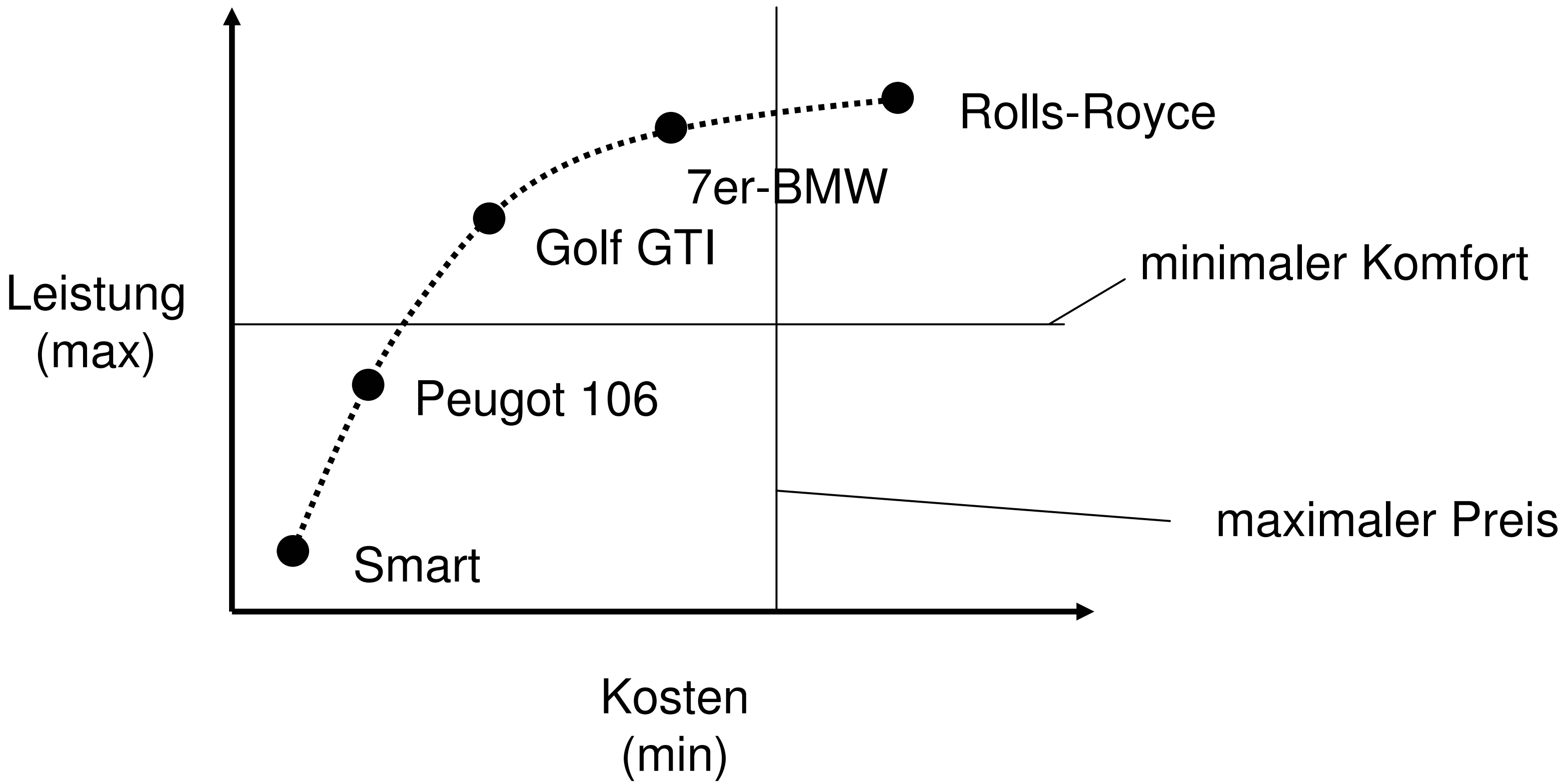
⇨ keine klare Präferenz

⇨ unterschiedliche Käufer gewichten die Kriterien unterschiedlich

Darstellung unterschiedlicher Optimierungskombinationen



In realen Problemen - Nebenbedingungen



Aber: Was wird dazwischen präferiert?

Methode 1: Präferenz-basierte Mehrkriterielle Optimierung

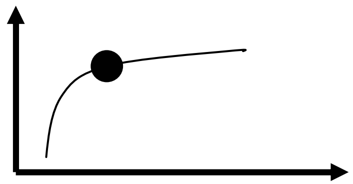
$$f_1 \to \min$$
$$f_2 \to \min$$
$$\vdots$$
$$f_N \to \min$$

zusätzliches Wissen

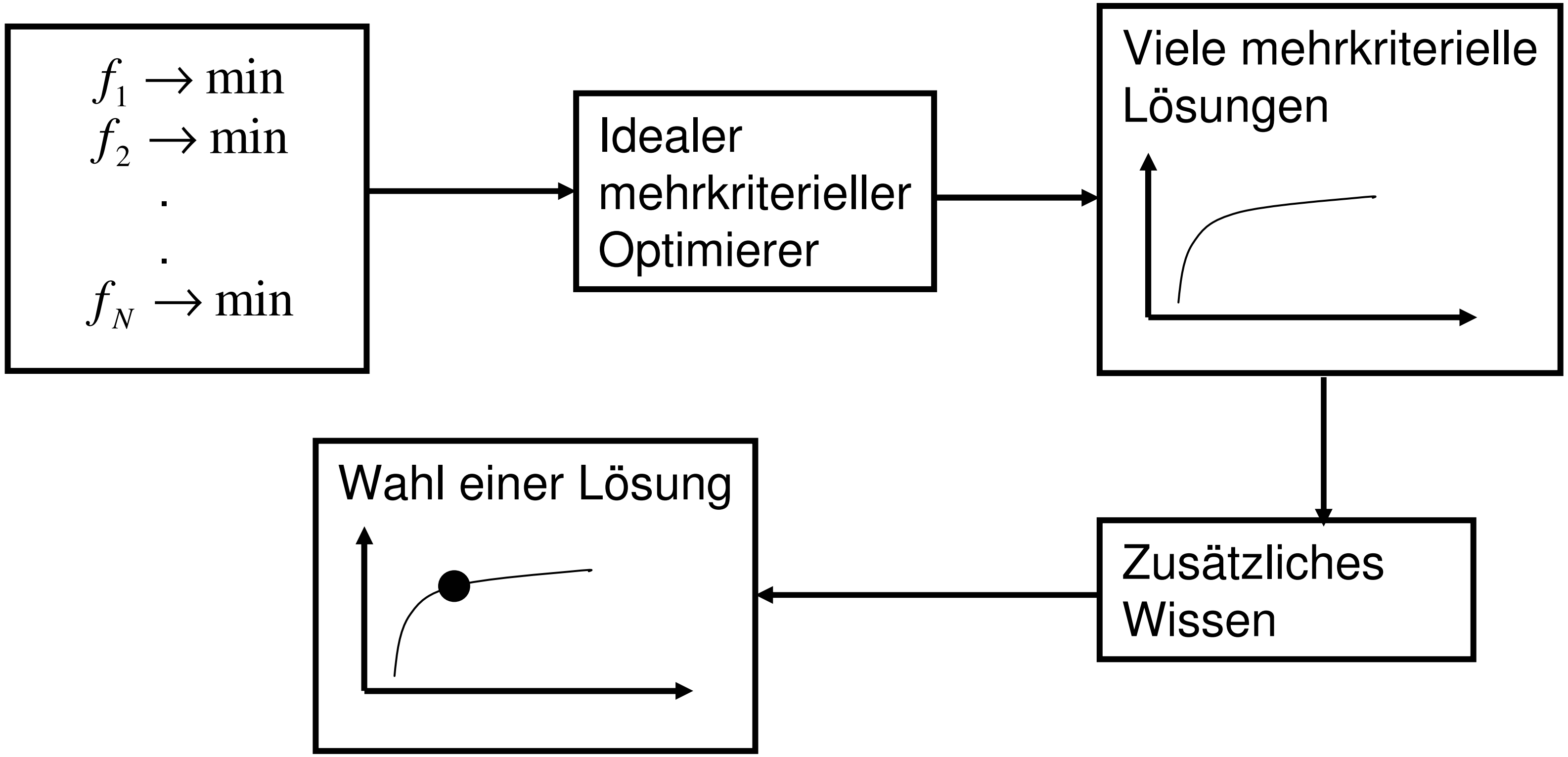
Schätzung der Gewichte $(w_1, w_2, \ldots, w_N)$

Einkriterielle Optimierung
$F = w_1 f_1, w_2 f_2, \ldots, w_N f_N$

Einkriterielle Lösung



Methode 2: Ideale Mehrkriterielle Optimierung



Vergleich

Präferenzbasierte mehr-kriterielle Optimierung

- Einkriterielle Lösung, die in Iterationen verbessert wird (klassische Verfahren)

⇒

Zur Erzeugung der Menge der Lösungen mit Ziel-konflikt sind viele parameterisierte Lösung nötig

Ideale mehrkriterielle Optimierung

- Ziel: viele Lösungen mit Zielkonflikt

⇒

Evolutionär geeignet, da pro Generation viele individuelle Lösungen existieren

Mehrkriterielle Optimierung – formaler Ansatz

$$f_m(x) \to \min, \; m = 1,2\ldots M$$

mit den Nebenbedingungen:

$$g_j(x) \geq 0, \quad \mathrm{j} = 1,2\ldots J;$$
$$h_k(x) = 0, \quad k = 1,2\ldots K;$$
$$x_i^{(U)} \leq \mathrm{x_i} \leq \mathrm{x_i^{(O)}}, \quad \mathrm{i} = 1,2\ldots n;$$

Konzept der Dominanz (Annahme: Ziel**minimierung**)

Eine Lösung $x^{(1)}$ wird als **dominant** bezeichnet gegenüber einer Lösung $x^{(2)}$ wenn beide Bedingungen erfüllt sind:

1) Die Lösung $x^{(1)}$ ist nicht schlechter als $x^{(2)}$ in allen Zielfunktionen, oder

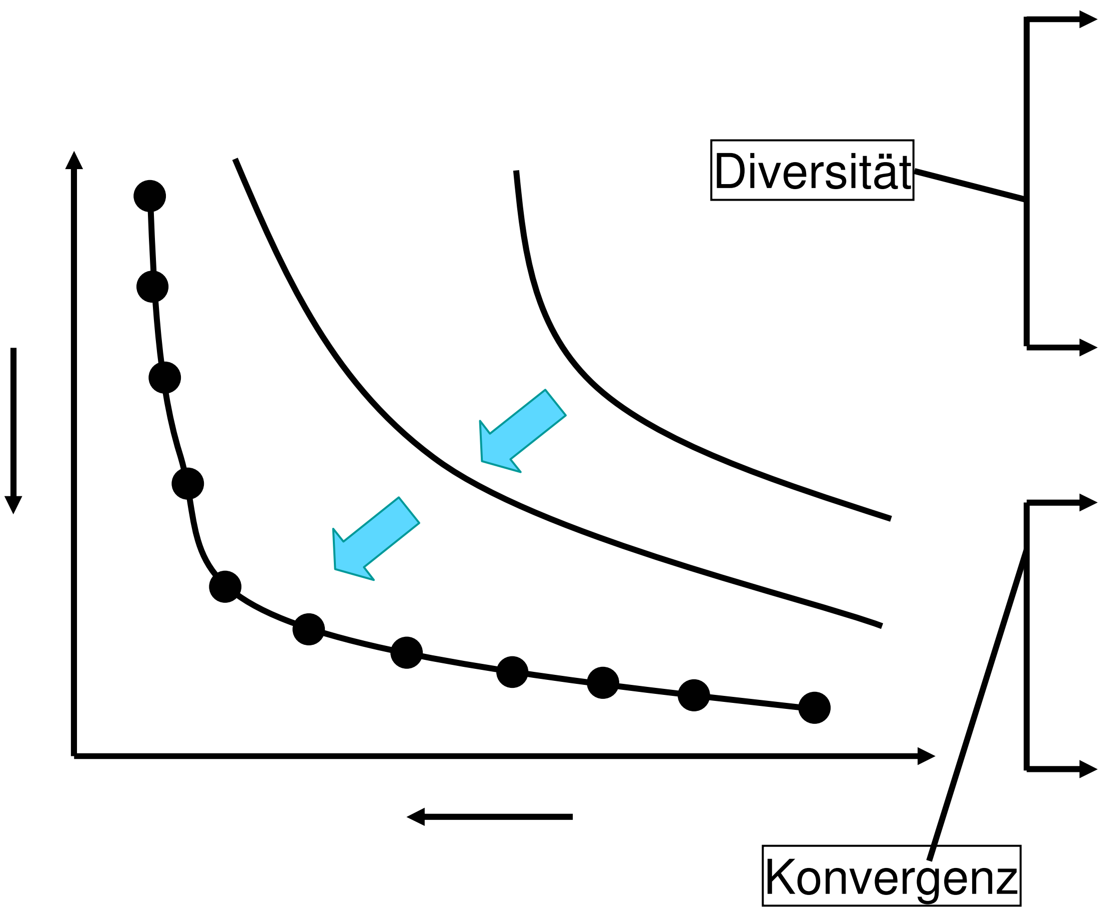
$$f_j(x^{(1)}) \prec= f_j(x^{(2)}) \text{ für alle } j = 1,2\ldots M$$

2) Die Lösung $x^{(1)}$ ist strikt besser als $x^{(2)}$ bezüglich mindestens einer Zielfunktion, oder

$$f_j(x^{(1)}) \prec f_j(x^{(2)}) \text{ für mindestens eins } j \in [1, M]$$

Das Ermitteln der nicht-dominierten Menge mit M-Funktionen und N-Variablen benötigt: $O(N(\log N)^{M-2})$

Ziele der mehrkriteriellen Optimierung



- Wie kann die Diversität erhalten werden?

  Dichte-Schätzung

- Wie kann sichergestellt werden, dass nicht-dominierte Lösungen nicht verschwinden?

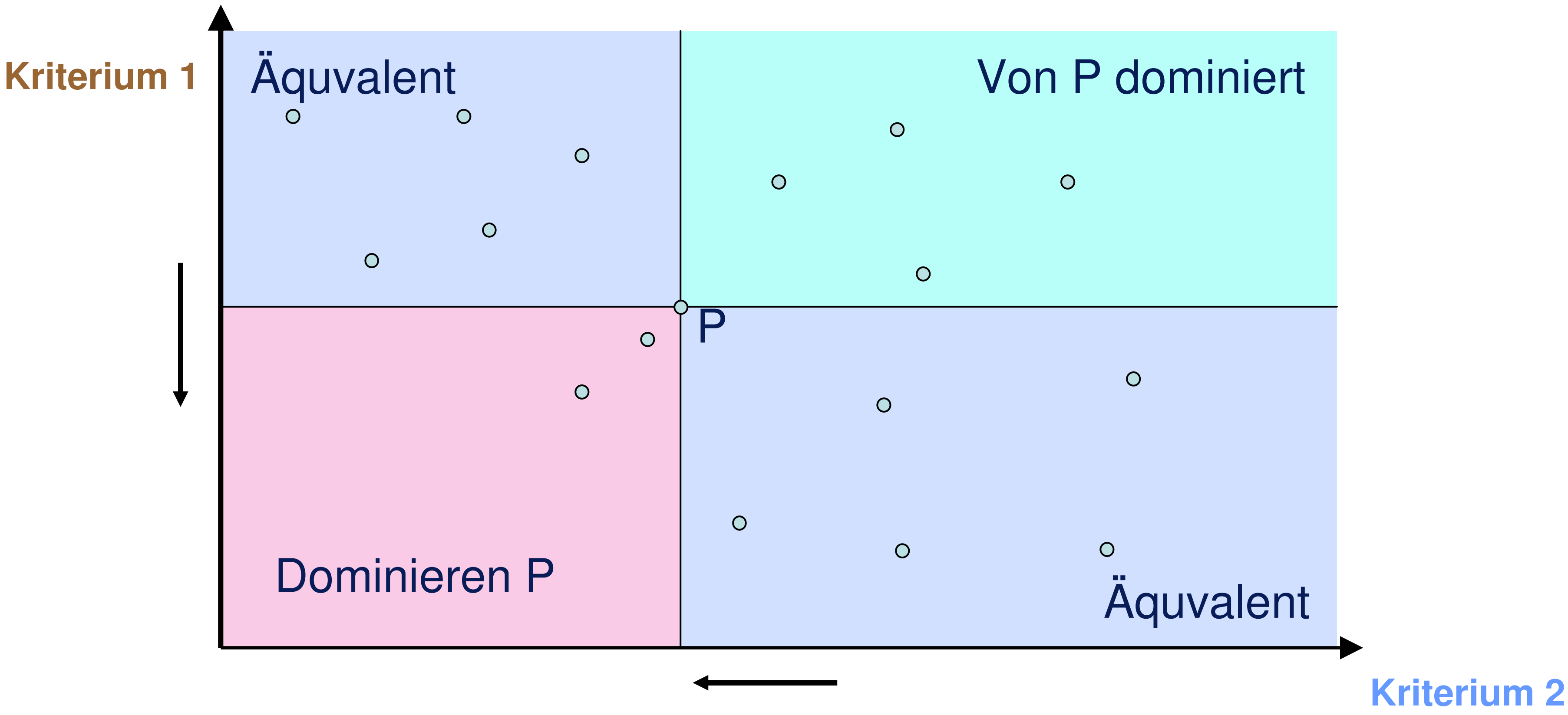
  Selektion der Umgebung

- Wie soll in Richtung Pareto-Menge optimiert werden?

  Fitness-Zuordnung

Paretodominanz

- Paretoäquivalenz – Paretodominanz



Multi-Objective Optimization
Formal Problem Definition

- **Pareto Optimalität**: Ein Entscheidungsvektor $\vec{x} \in X_f$ ist *non-dominated* bezüglich der Menge $A \subseteq X_f$, wenn $\nexists \vec{a} \in A : \vec{a} \prec_p \vec{x}$. Des Weiteren ist $\vec{x}$ Pareto optimal wenn $\vec{x}$ nicht dominiert wird von X_f.

- **Nicht dominierte Menge und Front**: Sei $A \subseteq X_f$. Die Funktion *nd(A)* ergibt die nicht-dominierte Untermenge von A:

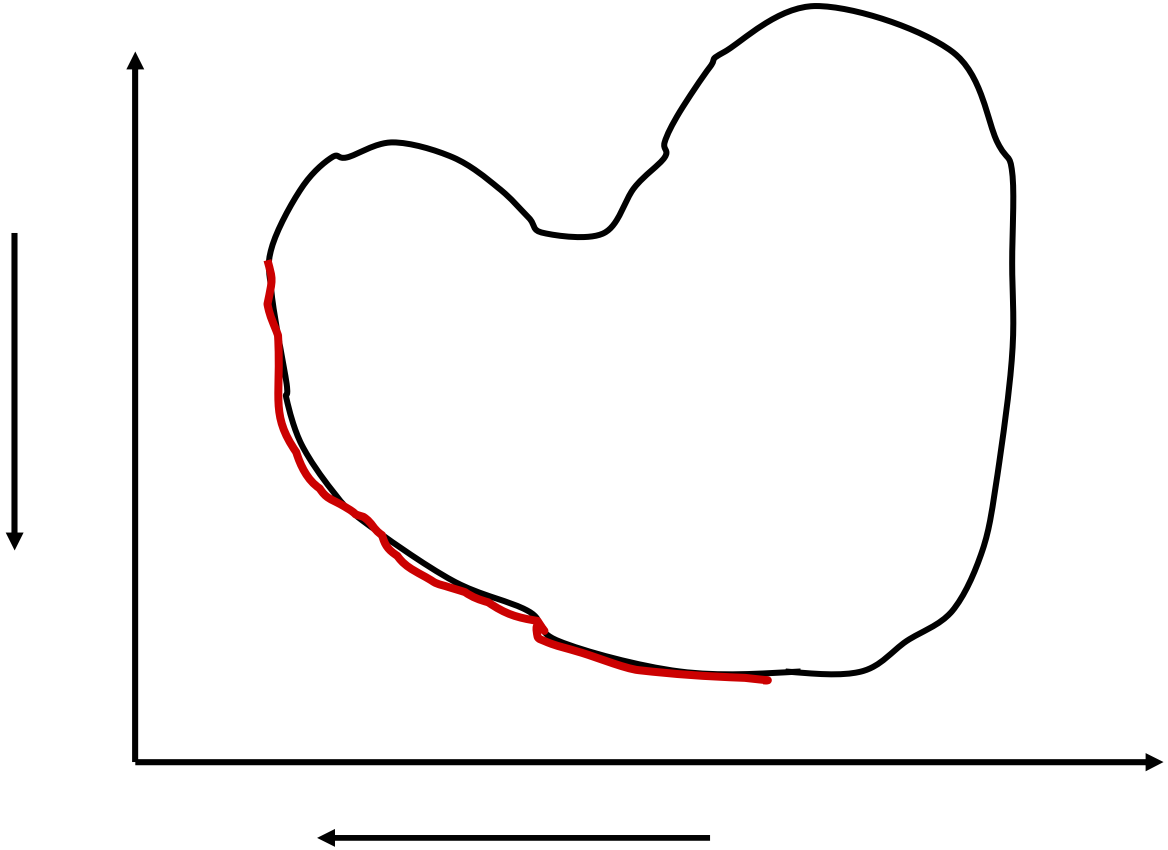
$$nd(A) = \{\vec{x} \in A \mid \nexists \vec{x}' \in A \, with \, \vec{x}' \prec_p \vec{x}\}$$

Die zugehörigen Objektvektoren $\vec{f}(X_P)$ bilden die nicht-dominierte Front bezüglich A. Des Weiteren wird die Menge $X_P = nd(X_f)$ die **Pareto optimale Menge** und die Menge $\vec{f}(nd(X_f))$ die **Pareto-Front** genannt.

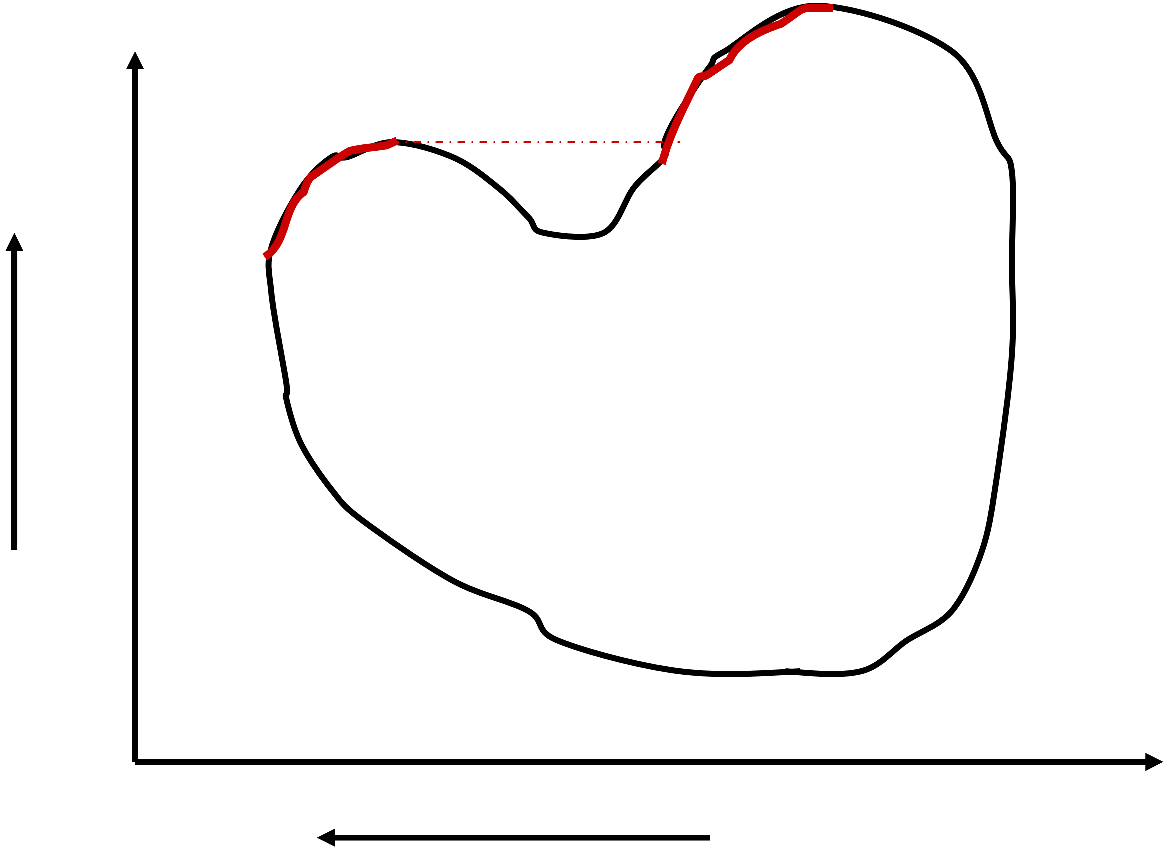
Übung

- Welche Aussage ist richtig:
  - Die Pareto-Front beschreibt alle Entscheidungsvektoren, die von anderen Lösungen nicht dominiert werden.
  - Die Pareto-Menge beschreibt alle Entscheidungsvektoren, die von anderen Lösungen nicht dominiert werden.

Pareto-optimale Lösungen (min-min)

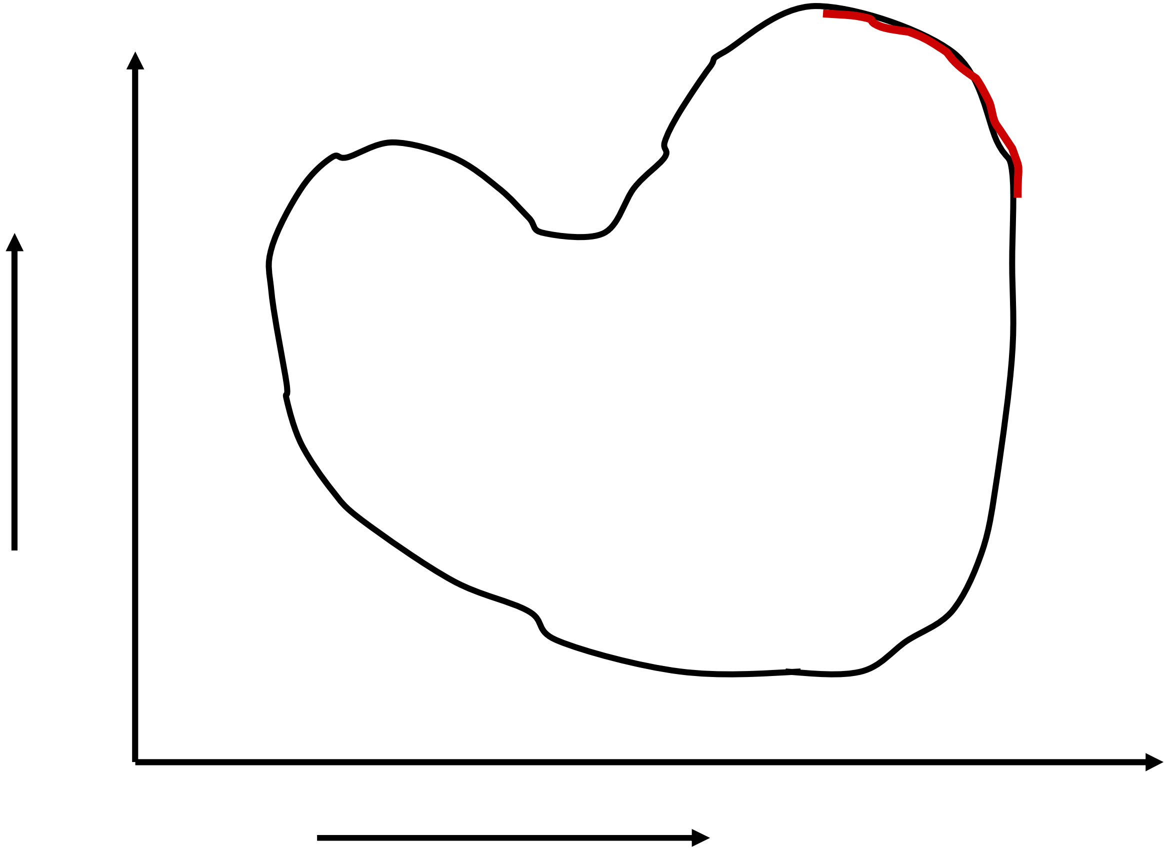


Pareto-optimale Lösungen (min-max)



Pareto-optimale Lösungen (max-min)

Pareto-optimale Lösungen (max-max)



Übung

1. Gegeben seien folgende Nutzenfunktionen

   $$\text{Minimiere } (-x_1^3)$$

   $$\text{Maximiere } \exp(x_1^2)$$

   Wenn beide Nutzenfunktionen optimiert werden, kann es dabei Pareto-optimale Lösungen geben? Begründen Sie.

Übung

2. Überprüfen Sie ob die erste Lösung die zweite dominiert.

a) $(\min, \min): f^{(1)} = (1{,}2;\ 3{,}5)^T,\ f^{(2)} = (1{,}5; 3.0)^T$

b) $(\min, \max, \min): f^{(1)} = (10{,}5; 1{,}5;\ -10{,}0)^T,\ f^{(2)} = (5{,}0; 0{,}5; -12)^T$

c) $(\min, \min, \min): f^{(1)} = (10{,}5; 1{,}5;\ -10{,}0)^T,\ f^{(2)} = (5{,}0; 0{,}5; -12)^T$

d) $(\max, \max, \min): f^{(1)} = (10{,}5; 1{,}5;\ -10{,}0)^T,\ f^{(2)} = (5{,}0; 0{,}5; -12)^T$

e) $(\max, \max, \max): f^{(1)} = (10{,}5; 1{,}5;\ -10{,}0)^T,\ f^{(2)} = (5{,}0; 0{,}5; -12)^T$

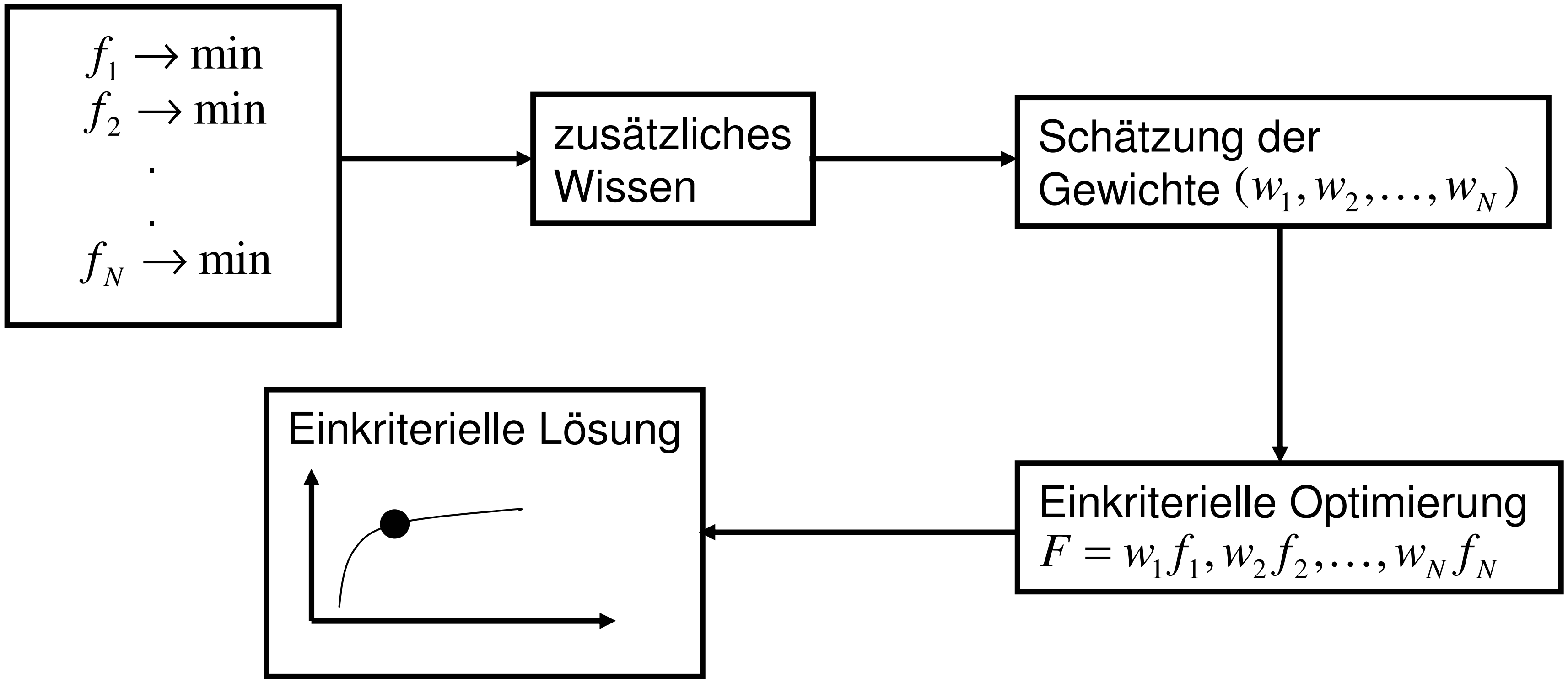
f) $(\min, \max, \max): f^{(1)} = (10{,}5; 1{,}5;\ -10{,}0)^T,\ f^{(2)} = (5{,}0; 0{,}5; -12)^T$

Klassische Methoden

Klassische Verfahren der mehrkriteriellen Optimierung:

- Gewichtete Summenmethode
- epsilon-bedingte Methode
- Gewichtete Metrik-Methode
- Werte-Funktion Methode
- Benson‘s Methode
- Ziel-Programmierung Methode

Methode 1: Präferenz-basierte Mehrkriterielle Optimierung



Gewichtete Summenmethode

- Gebräuchlichste klassische Methode
- Gewichte : definiert vom Anwender
- Maximiere: $F_m(x) = \sum_{m=1}^{M} \mathrm{w}_{\mathrm{m}} \cdot f_m(x)$
- mit:

$$g_j(x) \geq 0, \quad \mathrm{j} = 1,2\ldots J; \qquad \sum_{m=1}^{M} w_i = 1, w_i \geq 0$$

$$h_k(x) = 0, \quad k = 1,2\ldots K;$$

$$x_i^{(L)} \leq \mathrm{x}_{\mathrm{i}} \leq \mathrm{x}_{\mathrm{i}}^{(\mathrm{U})}, \quad \mathrm{i} = 1,2\ldots n;$$

- Vorteile:
  – Einfach zu nutzen
  – Für konvexe Fronten nachweislich
- Nachteile:
  – Gewichte sind sehr sensitiv
  – Unterschiedliche Gewichte führen nicht notwendiger Weise zu unterschiedlichen Lösungen

Übung

- Maximiere: $F_m(x) = \sum_{m=1}^{M} \mathrm{w}_{\mathrm{m}} \cdot f_m(x)$
- mit:

$$\sum_{m=1}^{M} w_i = 1, w_i \geq 0$$

$$g_j(x) \geq 0, \quad \mathrm{j} = 1,2 \ldots J;$$

$$h_k(x) = 0, \quad k = 1,2 \ldots K;$$

$$x_i^{(L)} \leq \mathrm{x_i} \leq \mathrm{x_i^{(U)}}, \quad \mathrm{i} = 1,2 \ldots n;$$

- Was muss variiert werden, um eine Pareto-Front zu erzeugen?

epsilon-bedingte Methode

- Minimiere: $f_\mu(x),$
- mit: $f_m(x) \le \varepsilon_m,\ m = 1,2\ldots M$ und $m \ne \mu$

$$g_j(x) \ge 0,\quad \mathrm{j} = 1,2\ldots J;$$
$$h_k(x) = 0,\quad k = 1,2\ldots K;$$
$$x_i^{(L)} \le \mathrm{x_i} \le \mathrm{x_i^{(U)}},\quad \mathrm{i} = 1,2\ldots n;$$

- Vorteile:
  – Methode funktioniert für konvexe und nicht-konvexe Suchräume
- Nachteile:
  – Für alle Nutzenfunktionen sind gute Informationen notwendig
  – Lösung hängt sehr vom gewählten epsilon ab

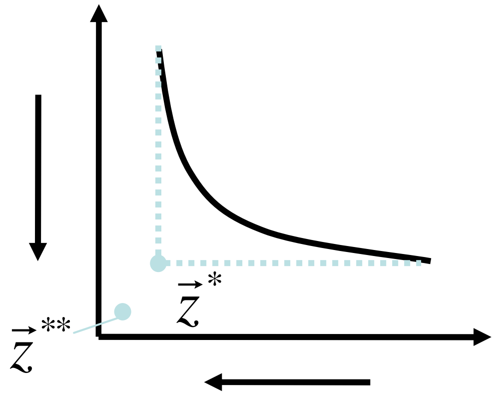
Gewichtete Metrik Methode

- Minimiere: $l_p(x) = \left(\sum_{m=1}^{M} w_m \,|\, f_m(x) - z_m^* \,|^p \right)^{1/p},$
- mit: $z^* - \text{ideale Lösung}$

  $g_j(x) \geq 0, \quad \mathrm{j} = 1,2 \ldots J;$

  $h_k(x) = 0, \quad k = 1,2 \ldots K;$

  $x_i^{(L)} \leq \mathrm{x_i} \leq \mathrm{x_i^{(U)}}, \quad \mathrm{i} = 1,2 \ldots n;$

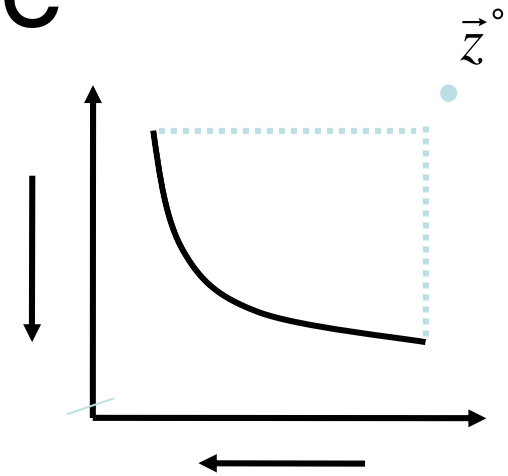


- Sonderform gewichtet Tchebycheff: $l_\infty = \max_{m=1}^{M} \left(w_m \,|\, f_m(x) - z_m^* \,| \right)$
- Vorteile:
  - gewichtetes Tchebycheff Problem findet alle Pareto-optimalen Lösungen, wenn z* ein utopian Vector ist (Miettinen 1999)
- Nachteile:
  - Werte können sich in Größenordnungen unterscheiden. Daher ist eine Normalisierung sinnvoll, die zusätzliches Wissen braucht.
  - Die ideale Lösung z* muss bekannt sein.

Benson‘s Method

$$\max \sum_{m=1}^{M} \max(0, (\overset{\circ}{z}_m - f_m(x)))$$

$$mit \;\; f_m(x) \le \overset{\circ}{z}_m$$



- „Vergrößere den Abstand zu einer schlechten Lösung“ $\overset{\circ}{z}_m$
- Problem: Optimierungsmethoden sind schwer nutzbar, da die Optimierungsfunktion nicht differenzierbar ist

Werte-Funktion Methode

$$\max U(f(x)); U : R^M \to R$$

$$f(x) = (f_1(x), f_2(x), \ldots, f_M(x))^T$$

- Probleme:
  - Werte-Funktion ist schwer zu bestimmen
  - Werte-Funktion muss überall im gültigen Lösungsraum definiert sein.

Ziel-Programmierungs-Methode

- Für jede Zielfunktion wird ein Ziel definiert

$$Ziel\,(f(x) = t)$$

- Oft Übersetzung in: $f(x) - \underbrace{p}_{\text{positive Abweichung}} + \underbrace{n}_{\text{negative Abweichung}} = t$

- Resultierendes Ziel: $\min\,(p + n)$
- z.B. gewichtete Zielprogrammierung:

$$\min \sum_{\mathrm{m=1}}^{\mathrm{M}} (\alpha_m p_m + \beta_m n_m)$$

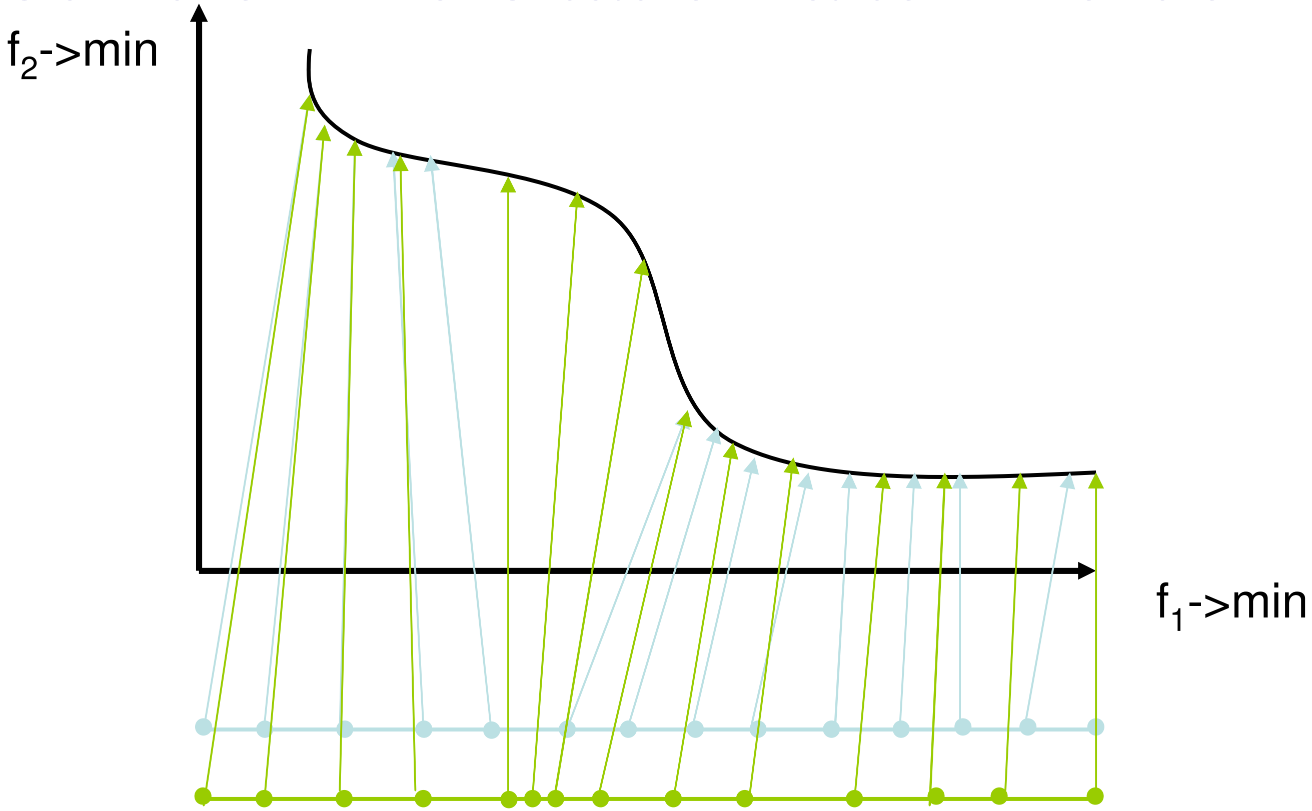
$$mit\ \mathrm{f_m}(x) - p_m + n_m = t,\ m = 1,2,\ldots,M$$

Übung

- Was kann mittels alpha und beta realisiert werden?

Probleme der klassischen Ansätze

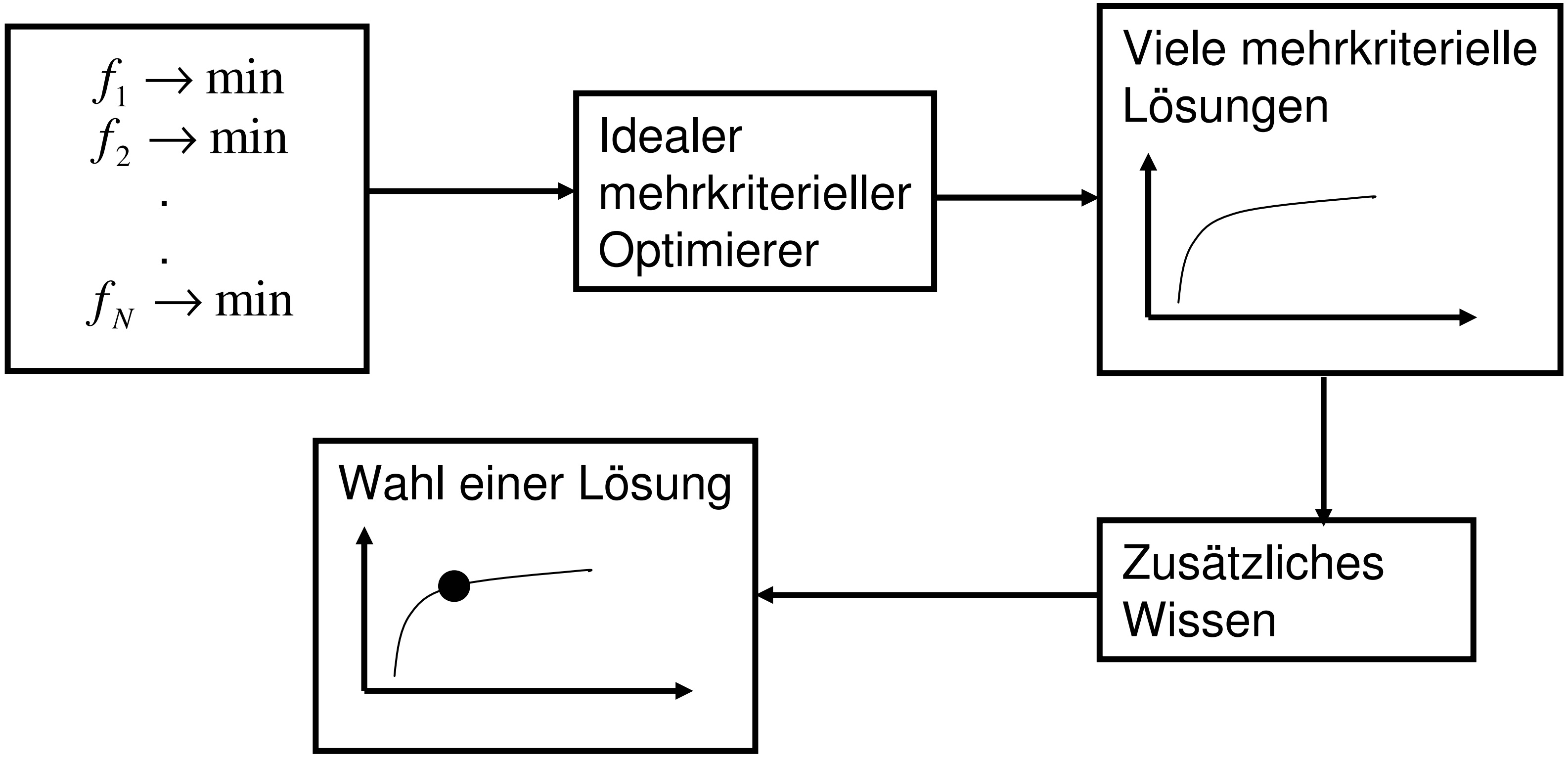
Wie soll die Diversität erhalten werden?



Klassische Methoden

- Vorteile:
  - Konvergenz (hier nicht gezeigt)
  - leicht zu implementieren
- Nachteile:
  - stets Umwandlung eines mehrkriteriellen Problems in ein einkriterielles Problem
  - Parametervariation notwendig um eine Pareto-Front zu erhalten (unterschiedliche Parameter müssten dann zu unterschiedlichen Lösungen führen)

Methode 2: Ideale Mehrkriterielle Optimierung



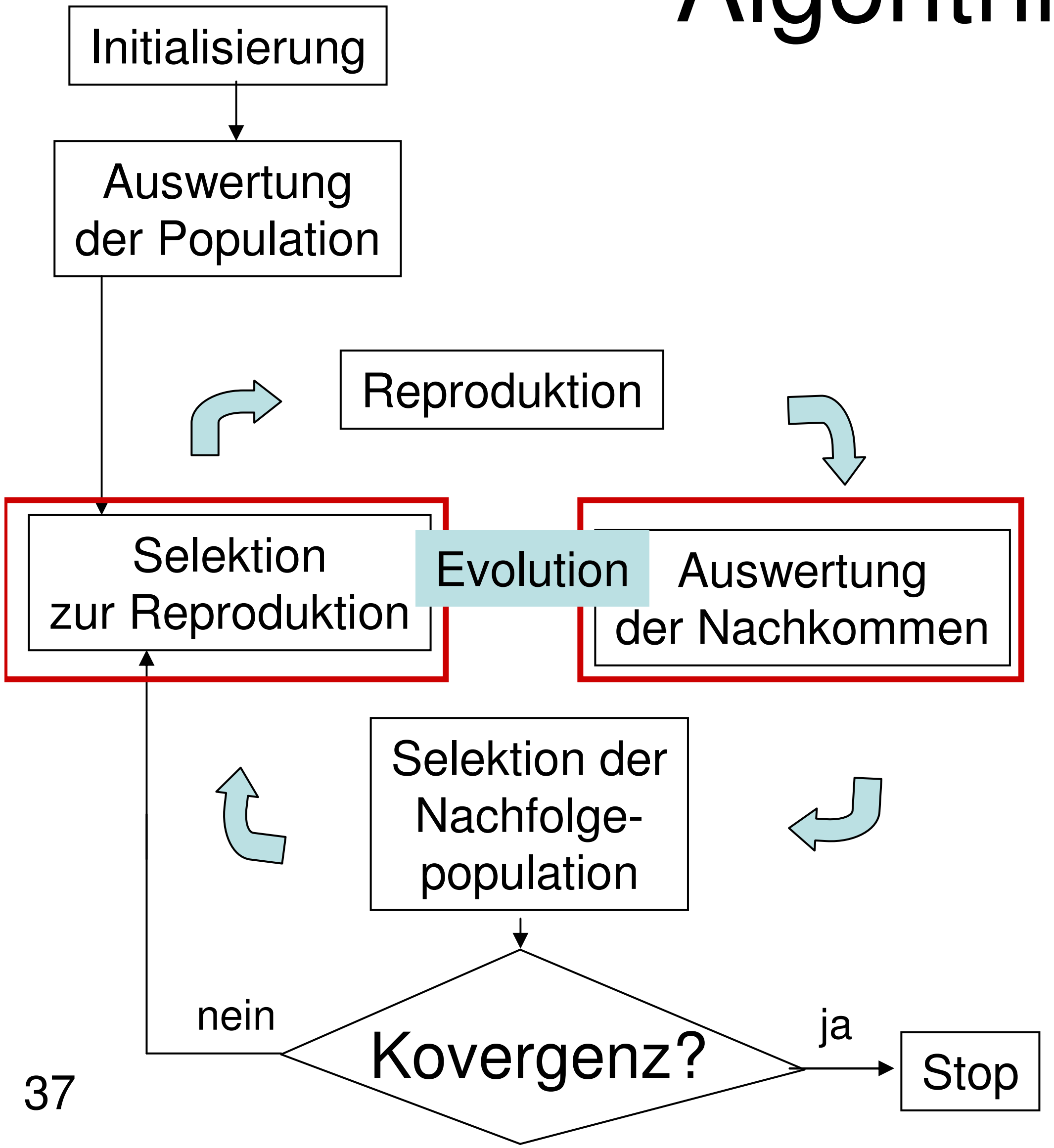
Wesentliche Vorteile mehr-kriterieller Evolutionärer Algorithmen

- Pro Generation werden viele Lösungen gefunden
- Kein Bedarf nach idealen Lösungen/Gewichten etc.
- Alle nicht-dominierten Lösungen werden gleich betont
- Methoden zur Erhaltung der Diversität lassen sich besser integrieren
- Algorithmen können an viele verschiedene Probleme angepasst werden

Mehr-kriterielle Evolutionäre Algorihtmen (Beispiele)

- Nicht-elitäre Algorithmen
  - Vector Evaluated Genetic Algorithm (VEGA)
  - Non-dominated Sorting Genetic Algorithm (NSGA)
  - Predator-Prey Evolution Strategy
  - …. sehr viele andere
- Elitäre Algorithmen
  - Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II)
  - Strength Pareto Evolutionary Algorithm (SPEA 2)
  - S metric selection Multi-objective Evolutionary Algorithm (SMS-EMOA)
  - Indicator Based Evolutionary Algorithm (IBEA)
  - …. viele andere

Allgemeiner evolutionärer Algorithmus



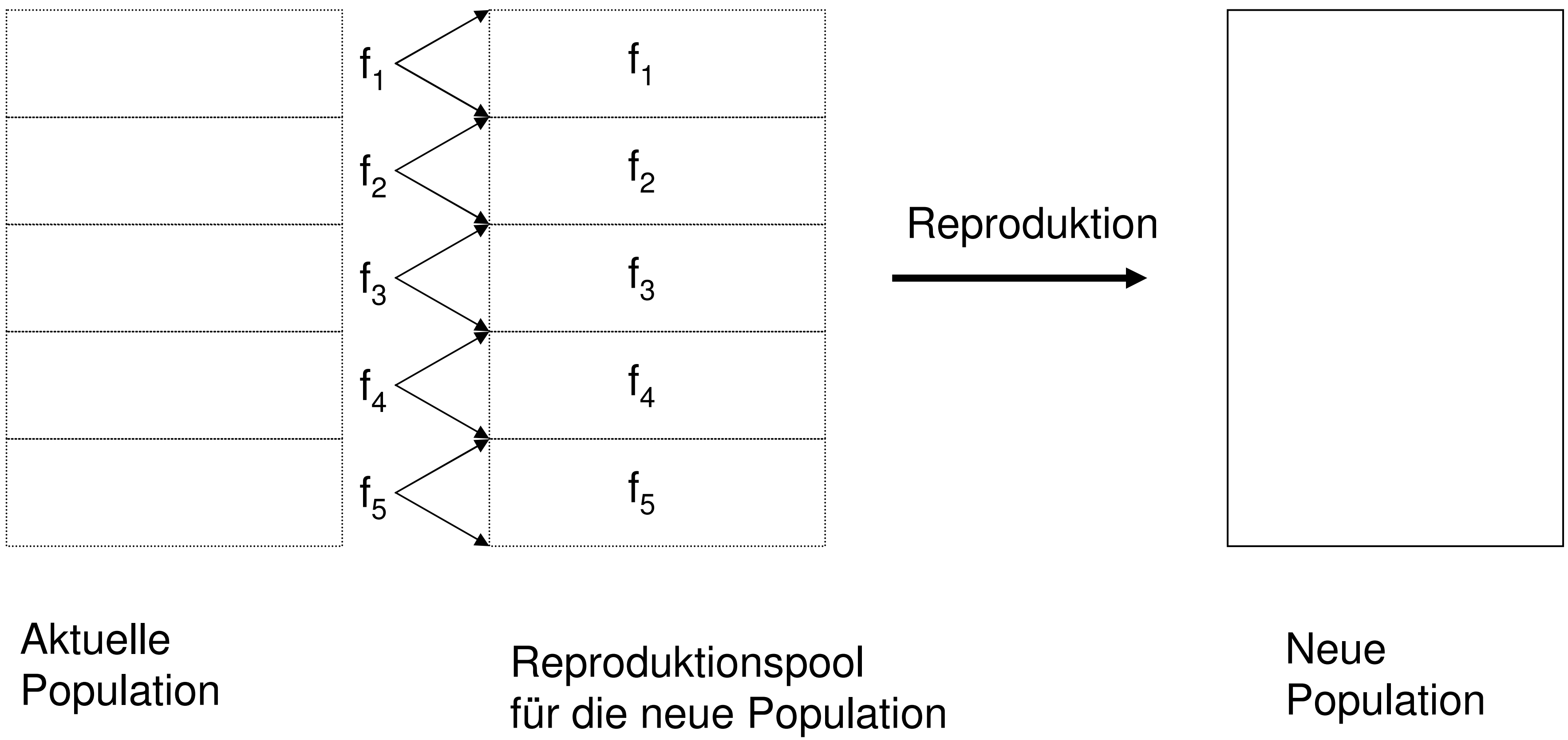
Hier liegen die wesentlichen Änderung. Dies ist im Wesentlichen auch für die meisten anderen mehrkriteriellen EAs so.

Vector Evaluated Genetic Algorithm

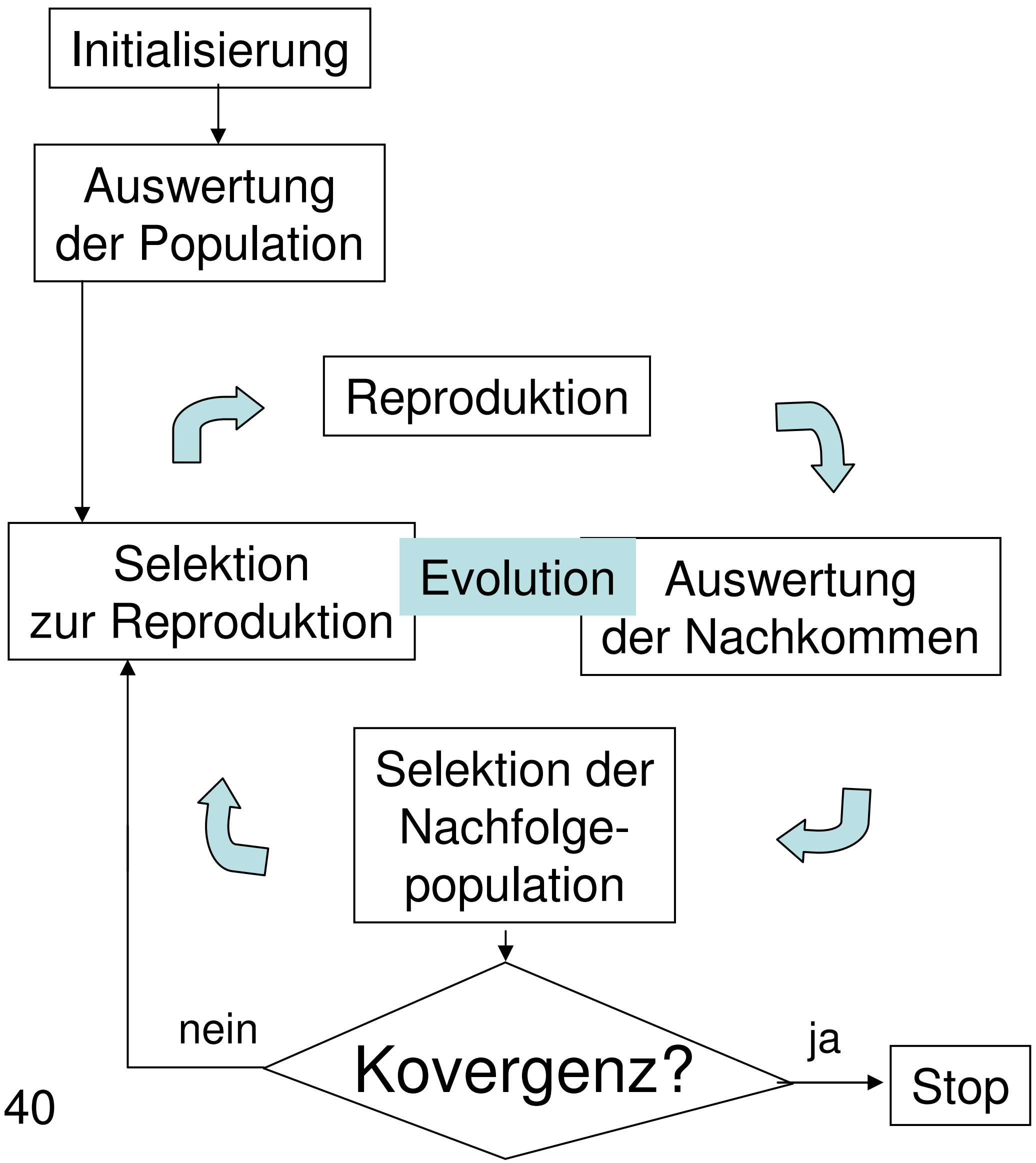
- Erster bekannter mehrkriterieller evolutionärer Algorithmus von Schaffer (1984)
- Nutzung eines Vektors für die verschiedenen Funktionswerte – ein Eintrag pro Funktion
- Extrem einfacher Algorithmus
- Eindeutige Zuweisung einer zufälligen, aber gleich großen Teilmenge der jeweiligen Population zu den jeweiligen Nutzenfunktionen
- Nutzung eines „normalen“ Genetischen Algorithmus“ zur eigentlichen Evolution

VEGA Auswertungsschema

In der Regel sollte die Zuordnung zufällig erfolgen.



Übung



Welcher Teil eines allgemeinen EAs wird bei VEGA verändert?

VEGA Selektion

- N- Anzahl der Individuen; M-Anzahl der Nutzenfunktionen

1) Setze i = 1 und q = N/M
2) Für alle j=1+(i-1)*q bis j=i*q:
   1) $F(x^j) = f_i(x^j)$ //wähle für Individum j die Nutzenfunktion i
3) Führe F-proportionale Selektion aller q Lösungen aus und erstelle den $Pool_i$
4) Wenn i=M, gehe zu 5. Sonst i++ und 2)
5) Erstelle den Reproduktionspool $P = \bigcup_{i=1}^{M} P_i$

VEGA Vor- und Nachteile

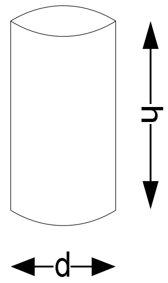
- Vorteile
  - Einfach und leicht zu implementieren
- Nachteile
  - Viele Lösungen liegen nahe an den optimalen Lösungen der einzelnen Nutzenfunktionen (schlechte Diversität)
  - Rekombinations-operator versagt häufig beim Erstellen von Zwischenlösungen zur Erzeugung der Pareto-Front

Übung

- Das Problem

Minimiere $f(d,h) = c(\frac{\pi d^2}{2} + \pi d h)$

Maximiere $g_1(d,h) = \frac{\pi d^2 h}{4}$



Variablen $0 < d \leq 32$

$0 < h \leq 32$

- Erzeugen Sie die Pareto-Front mit VEGA!
- Nutzen Sie pro Nutzenfunktion 15 zufällig ausgewählte Individuen.
- Nutzen Sie die Rang-basierte Selektion

Vorlesungsplanung

- 21.02.2012: Einkriterielle Evolutionäre Optimierung I (CF)
- 28.02.2012: Einkriterielle Evolutionäre Optimierung II (CF)
- 06.03.2012: Test (1+2), Mehrkriterielle Evolutionäre Optimierung I (CF)
- 13.03.2012: Statistische Lerntheorie I (JP)
- 20.03.2012: Statistische Lerntheorie II (JP)
- 27.03.2012: Test (4+5), Neuronale Netze (JP)
- 10.04.2012: Support Vector Maschinen I (JP)
- 02.05.2012: Mehrkriterielle Evolutionäre Optimierung II (CF)
- 08.05.2012: Genetische Fuzzy Systeme (CF)
- 15.05.2012: Test (3+8+9), Meta-Heuristiken (ACO, PSO) (CF)
- 22.05.2012: Simulated Annealing und andere Suchmethoden (CF)
- 29.05.2012: Support Vector Maschinen II (JP)
- 05.06.2012: Test (6+7+12), Clustering (JP)
- 12.06.2012: Lernen und Spieltheorie (JP)
- 26.06.2012: 1. Termin mündliche Prüfungen
- 03.07.2012: 2. Termin mündliche Prüfungen

Hausaufgabe

- Beenden Sie bitte alle in der Vorlesung nicht vollständig bearbeiteten Übungen+
- Lesen Sie das paper: „Indicator-Based Selection in Multiobjective Search” von Eckart Zitzler and Simon Künzli (beide ETH Zürich)