FernUniversität in Hagen

Lehrstuhl für
Betriebswirtschaftslehre,
insb. Operations Research
Prof. Dr. Wilhelm Rödder

Allgemeine Betriebswirtschaftslehre
Planungs- und Entscheidungstechniken

Sandra Rudolph, Dr. Friedhelm Kulmann*

Ergänzungen vom 10.02.2006 in Kapitel 7

Überblick

Motivation

5. Varianten der Linearen Programmierung als unternehmerisches Planungsinstrument

5.4. Die Behandlung von Nichtlinearität bei Separablen Programmen
- 5.4.1. Optimale leitungsgebundene Wärmeversorgung in Städten
- 5.4.2. Stückweise Linearisierung bei Separablen Programmen
- 5.4.3. Algorithmische Lösung von Separablen Programmen

Klassifizierung von Optimierungsproblemen

Klasse	Zielfunktion	Restriktionen
Lineare Optimierung	linear	linear
Quadratische Optimierung	quadratisch	linear
Nichtlineare Opt. mit linearen Restrikt.	nichtlinear	linear
Nichtlineare Opt. (nichtlineare Restr.)	nichtlinear	nichtlinear

Ziel ist es unter anderem,
die Lösbarkeit von Optimierungsproblemen einschätzen und gegebenenfalls Transformationen zwecks Vereinfachung vornehmen zu können.

5.4.2. Stückweise Linearisierung bei Separablen Programmen

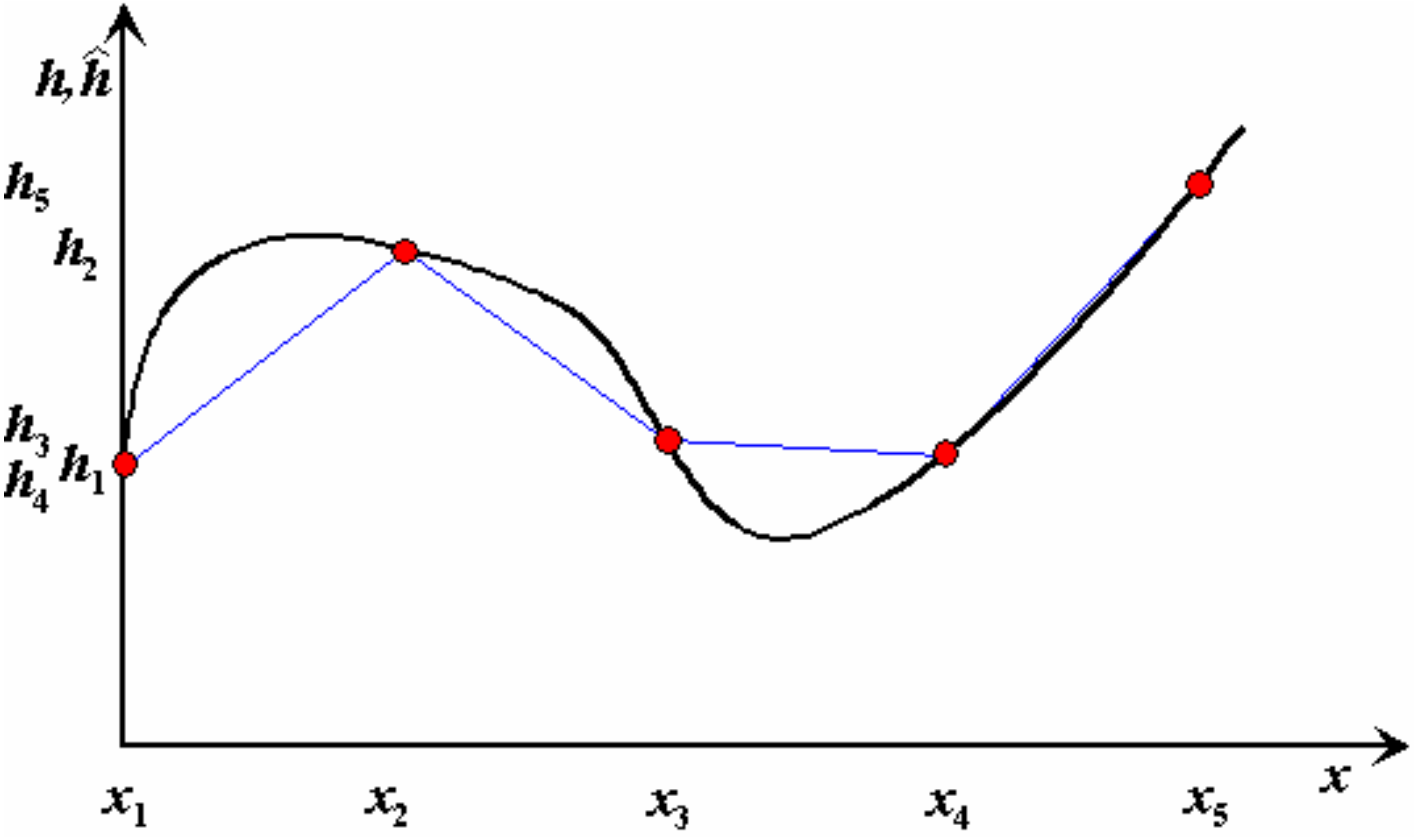
Zielfunktion $f(x)$ und alle Restriktionen $g_i(x)$ lassen sich als Summe einer einzigen Variablen und einer Konstanten schreiben.

$$\min x_0 = \sum_{j=1}^{n} f_j(x_j)$$

unter den Nebenbedingungen

$$\sum_{j=1}^{n} g_{ij}(x_j) \le b_i, \quad i = 1,...,m, \quad x_j \ge 0$$

Approximation einer nichtlineare Funktion h durch $\hat{h}$.



5.4.3. Algorithmische Lösung von Separablen Programmen

Bei Angabe von K Stützstellen benötigt man K neue Variable λ_k, für die gelten soll, dass $\sum_{k=1}^{K} \lambda_k = 1$ und $\lambda_k \geq 0$ für alle k.
Es berechnet sich:

$$\hat{h}(x) = \lambda h(x_k) + (1-\lambda) h(x_{k+1}),\ x = \lambda x_k + (1-\lambda) x_{k+1}$$

geschlossene Darstellung der Funktion $\hat{h}(x)$:

$$\hat{h}\left(\sum \lambda_k x_k\right) = \sum \lambda_k h(x_k),\ \ \lambda_k \geq 0,\ \ \sum \lambda_k = 1$$

Approximation des ursprünglichen Optimierungsproblems:

$$\min \lambda_0 = \sum_{j \in L} f_j(x_j) + \sum_{j \notin L} \sum_{k=1}^{K_j} f_{jk} \lambda_{jk}$$

unter den Nebenbedingungen

$$\sum_{j=1}^{n} g_{ij}(x_j) + \sum_{j \notin L} \sum_{k=1}^{K_j} g_{ijk} \lambda_{jk} \leq b_i, \quad i = 1,...,m$$

$$\sum_{k=1}^{K_j} \lambda_{jk} = 1 \text{ für alle } j \notin L, \lambda_{jk} \geq 0 \text{ für alle } k \text{ und } j \notin L$$

$$x_j \geq 0 \text{ für } j \in L \text{ und } \mathrm{NN}(\lambda_{jk}) \text{ für alle } j \notin L$$

Methodische Vorgehensweise

Ausgangsproblem:

$$\min x_0 = x_1^2 + (x_2 - 4)^2 + 2x_2$$

unter den Nebenbedingungen

$$\begin{aligned} x_1 - x_2 &\leq 3 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Stützstellen:

$$\begin{aligned} x_{11} &= 0, \quad x_{12} = 1, \quad x_{13} = 2, \\ x_{21} &= 2, \quad x_{22} = 3, \quad x_{23} = 4. \end{aligned}$$

Methodische Vorgehensweise

Durch Berechnung der Werte der Funktionen $f(x)$ und $g(x)$ an den Stützstellen erhält man folgendes Approximationsproblem:

$$\min \lambda_0 = \lambda_{12} + 4\lambda_{13} + 8\lambda_{21} + 7\lambda_{22} + 8\lambda_{23}$$

unter den Nebenbedingungen

$$\begin{aligned}
\lambda_{12} + 2\lambda_{13} - 2\lambda_{21} - 3\lambda_{22} - 4\lambda_{23} &\leq 3 \\
\lambda_{11} + \lambda_{12} + \lambda_{13} &= 1 \\
\lambda_{21} + \lambda_{22} + \lambda_{23} &= 1 \\
\lambda_{jk} \geq 0,\ k = 1,2,3,\ j = 1,2 &
\end{aligned}$$

Höchstens zwei benachbarte λ_{jk} sind positiv für $j = 1,2$.

Hinweis zur praktischen Lösung

Eine Menge von Variablen mit fester Reihenfolge, von denen genau eine (mindestens eine, aber höchstens zwei benachbarte) einen Wert ungleich Null annehmen muss (müssen), heißt Special Ordered Set vom Typ 1 (Typ 2).

Überblick

7. Optimierung mit intelligenten Strategien

- 7.1. Übersicht der Verfahren
- 7.2. Genetischer Algorithmus
  - 7.2.1. Einführung
  - 7.2.2. Der Algorithmus
- 7.3. Bandabgleichproblem und Genetischer Algorithmus
  - 7.3.1. Beschreibung des Bandabgleichproblems
  - 7.3.2. Ausgestaltung des Algorithmus

(mit Ergänzungen vom 10.02.2006)

Ausblick

Wirtschafts- und Sozialwissenschaften
➢ endliche Mengen von interagierenden Individuen und deren Verhalten

Zweite Hälfte des 20. Jahrhunderts

“Welchen Weg sollte man wählen, um eine bestimmte Anzahl von Städten auf einem möglichst kurzen Weg zu besuchen?“

“In welcher Reihenfolge sollten Aufträge auf einer Maschine bearbeitet werden, um die Durchlaufzeit zu minimieren?“

“Wie sieht der Stundenplan für ein vierzügiges Gymnasium aus, wenn alle Fächeranforderungen für die Klassen zu erfüllen sind und die Anzahl der Freistunden in den Klassen möglichst gering sein soll?“

Menge möglicher Antworten auf diese Fragen ist zwar endlich,

das Ermitteln einer sehr guten Lösung aber nicht trivial!

7.1 Übersicht der Verfahren

Man unterscheidet:

Bestensuchverfahren

Bestimmung einer optimalen Lösung durch zielgerichtete Suche

Nachbarschaftssuchverfahren

Iterative lokale Suche mit dem Ziel der Verbesserung einer aktuellen Lösung

Genetische Algorithmen

Gleichzeitiges Nutzen von Verbesserungspotential in einer sehr großen Anzahl von aktuellen Lösungen mit dem Ziel einer Lösungsverbesserung auch durch Kombination guter Teillösungen

7.2 Genetischer Algorithmus

Erzeugung einer Ausgangspopulation von Individuen

Bewertung der Individuen

Reproduktion und Rekombination unter Anwendung der genetischen Operatoren Selektion, Mutation und Crossover

Erzeugung einer Folgegeneration durch Auswahl von Individuen aus dem Pool der potentiellen Nachfolger

Erzeugung einer Ausgangspopulation von Individuen

Kodierung in Form von Zeichenketten

Phänotyp: tatsächliche Erscheinung
Genotyp: Satz von Genen (z.B. als Bitstring)

Folge von 6 Farben aus {rot, gelb, blau, violett}

Binärkodierung: {rot = 00, gelb = 01, blau = 10, violett = 11}
Bsp.: 001001111101

weitere Möglichkeit: {rot = r, gelb = g, blau = b, violett = v}
Bsp.: rbgvvg

Population mit 12 Individuen: {rbgvvg, gvbbrv, vrrvbg, ggbvrg, bbrgbv, vgrvrr, rrvbbg, bbvvrv, rgrgrg, vbvrbb, rrbgvb, bvvvrb}

Bewertung der Individuen

Tauglichkeit / Fitness der Individuen

Bewertungsfunktion

im einfachen Fall:	**gegebene Zielfunktion**
oder:	**verbale Beschreibung**
umsetzen in:	**formale Bewertungsvorschrift**

Beispiel

Sei z ein Individuum mit z_j aus der Menge $F = \{r, g, b, v\}$
($j = 1,...,6$ gibt Position im aktuellen Individuum an).
y entspreche der Zielfarbkombination,

$$f(z) = \sum_{j=1}^{6} x_j \text{ mit } x_j = 1, \text{ falls } z_j = y_j;\ x_j = 0, \text{ sonst.}$$

Bewertung der Individuen

Zielfarbkombination *y* sei: rgbbgr

Population mit 12 Individuen und Fitnesswert in Klammern:
{rbgvvg (1), gvbbrv (2), vrrvbg (0), ggbvrg (2), bbrgbv (0),
vgrvrr (2), rrvbbg (2), bbvvrv (0), rgrgrg (2), vbvrbb (0),
rrbgvb (2), bvvvrb (0)}

Reproduktion und Rekombination unter Anwendung der genetischen Operatoren Selektion, Mutation und Crossover

- **Reproduktion**
  - **Selektion eines Individuums;**
  - **Übernahme in den Pool der potentiellen Nachfolgegeneration**
- **Rekombination (1)**
  - **Selektion eines Individuums**
  - **Mutation, d.h. punktuelle Veränderung dieses Individuums;**
  - **Übernahme in den Pool der potentiellen Nachfolgegeneration**
- **Rekombination (2)**
  - **Selektion zweier Individuen**
  - **Crossover, d.h. abschnittweiser Austausch von Genen;**
  - **Übernahme beider neuen Individuen in den Pool der potentiellen Nachfolgegeneration**

Reproduktion und Rekombination unter Anwendung der genetischen Operatoren Selektion und Mutation

Ausgangspopulation: {rbgvvg, gvbbrv, vrrvbg, ggbvrg, bbrgbv, vgrvrr, rrvbbg, bbvvrv, rgrgrg, vbvrbb, rrbgvb, bvvvrb}

Reproduktion

Selektion eines Individuums etwa über Zufallszahl $1 \leq i \leq 12$;
$i := 7$ => rrvbbg
wird in den Pool der pot. Nachfolgegeneration übernommen.

Rekombination (1)

Selektion eines Individuums: $i := 5$ => bbrgbv
Mutation, d.h. punktuelle Veränderung dieses Individuums;
bbrgbv => bvrgbv
wird in den Pool der pot. Nachfolgegeneration übernommen.

Rekombination unter Anwendung des Crossover-Operators

Ausgangspopulation: {rbgvvg, gvbbrv, vrrvbg, ggbvrg, bbrgbv, vgrvrr, rrvbbg, bbvvrv, rgrgrg, vbvrbb, rrbgvb, bvvvrb}

Rekombination (2.1)

Selektion zweier Individuen: *i* := 7, 12 => rrvbbg & bvvvrb

1-Punkt-Crossover: *j* := 2

Crossover nach der 2ten Position

rr|vbbg & bv|vvrb => bvvbbg & rrvvrb

Übernahme beider neuen Individuen in den Pool der potentiellen Nachfolgegeneration

Rekombination unter Anwendung des Crossover-Operators

Ausgangspopulation: {rbgvvg, gvbbrv, vrrvbg, ggbvrg, bbrgbv, vgrvrr, rrvbbg, bbvvrv, rgrgrg, vbvrbb, rrbgvb, bvvvrb}

Rekombination (2.2)

Selektion zweier Individuen: $i := 1, 4$ => rbgvvg & ggbvrg

2-Punkt-Crossover: $j_1 := 1, j_2 := 5$

Crossover nach der 1ten und 5ten Position

r|bgvv|g & g|gbvr|g => rgbvrg & gbgvvg

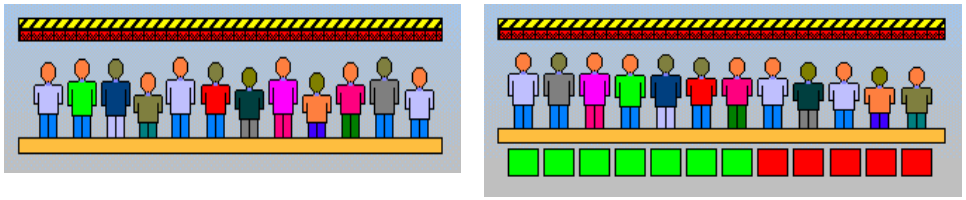
Übernahme beider neuen Individuen in den Pool der potentiellen Nachfolgegeneration

Erzeugung einer Folgegeneration durch Auswahl von Individuen aus dem Pool der potentiellen Nachfolger

Bestenauswahl

gemäß Fitnesswerten

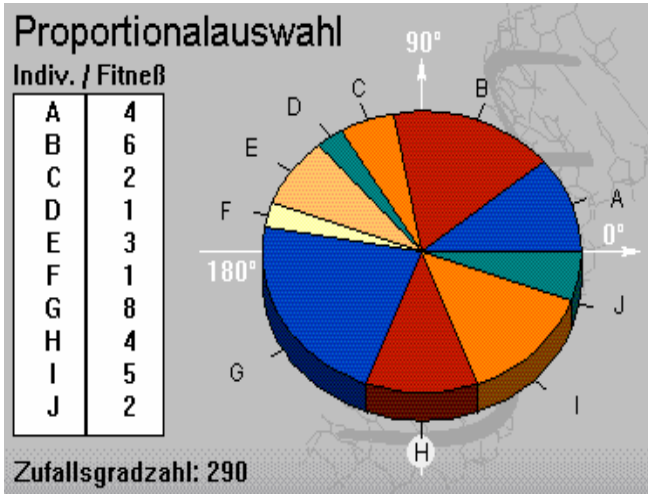
Auswahl von *n* Individuen



Proportionalauswahl

gemäß Anteil an Gesamtfitness

Auswahl von *n* Individuen stochastisch



Erzeugung einer Folgegeneration durch Auswahl von Individuen aus dem Pool der potentiellen Nachfolger

Pool für Nachfolgegeneration mit Fitnessbewertung:
{rrvbbg (2), bvrggv (1), bvvbbg (1), rrvvrb (1), rgbvrg (3), gbgvvg (0), rbgvvg (1), gvbbrv (2), vrrvbg (0), ggbvrg (1), bbrgbv (0), vgrvrr (2), rrvbbg (2), bbvvrv (0), rgrgrg (2), vbvrbb (0), rrbgvb (2), bvvvrb (0)}

Bestenauswahl

blau markierte Auswahl von $n = 12$ Individuen

Erzeugung einer Folgegeneration durch Auswahl von Individuen aus dem Pool der potentiellen Nachfolger

Pool für Nachfolgegeneration mit Fitnessbewertung:
{rrvbbg (0,1), bvrggv (0,05), bvvbbg (0,05), rrvvrb (0,05), rgbvrg (0,15), gbgvvg (0,0), rbgvvg (0,05), gvbbrv (0,1), vrrvbg (0,0), ggbvrg (0,05), bbrgbv (0,0), vgrvrr (0,1), rrvbbg (0,1), bbvvrv (0,0), rgrgrg (0,1), vbvrbb (0,0), rrbgvb (0,1), bvvvrb (0,0)}

Proportionalauswahl

$$p_i = \frac{f(z_i)}{\sum_{j=1}^{18} f(z_j)} = \frac{f(z_i)}{20}$$

Auswahl von $n = 12$ Individuen stochastisch

Beispiel „Rundreiseproblem“

Gesucht sei eine Rundreise minimaler Länge durch die Städte Aachen, Berlin, Cottbus, Dresden, Emden, Frankfurt und Greifswald. Jede Stadt muss in der Rundreise enthalten sein.

Treffen Sie Vorbereitungen, um das Problem mit einem Genetischen Algorithmus lösen zu können.

Im Folgenden werden vor allem die Aspekte „Kodierung“ und „Rekombination“ behandelt.

Beispiel „Rundreiseproblem“ – Kodierung

Kodierung in Form von Zeichenketten
Phänotyp: tatsächliche Erscheinung
Genotyp: Satz von Genen (z.B. als Bitstring)

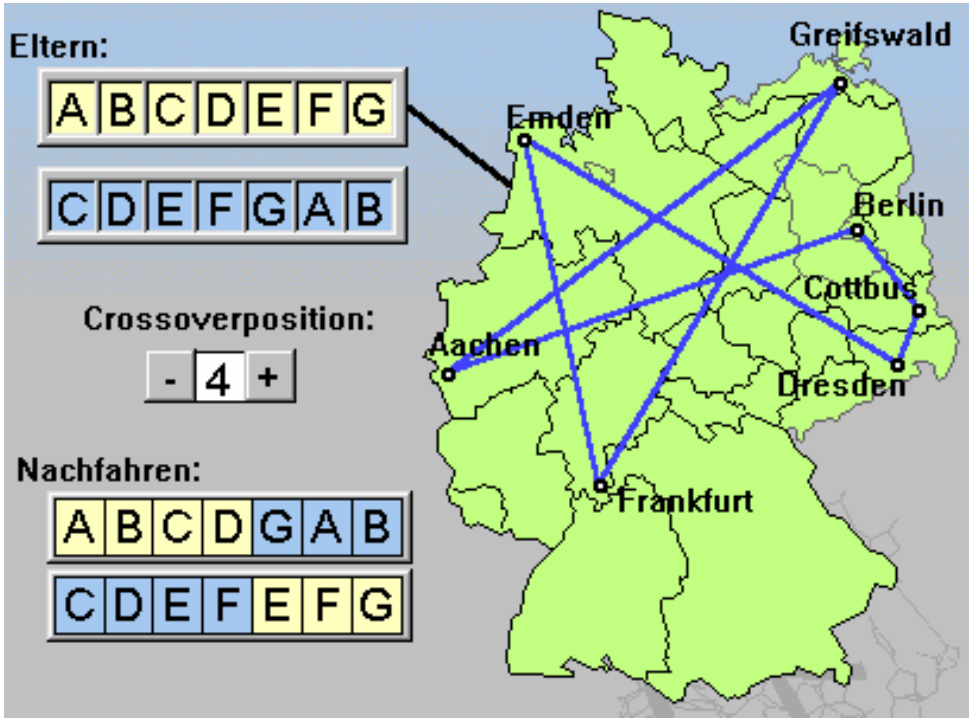
7 Städte. Aachen, Berlin, Cottbus, Dresden, Emden, Frankfurt und Greifswald

Binärkodierung: {Aachen (000), Berlin (001), Cottbus (010), Dresden (011), Emden (100), Frankfurt (101), Greifswald (110)}
Problem ? **Bsp.: 010110001101010100011**

weitere Möglichkeit: {Aachen (A), Berlin (B), Cottbus (C), Dresden (D), Emden (E), Frankfurt (F), Greifswald (G)}
Bsp.: AGBFCED

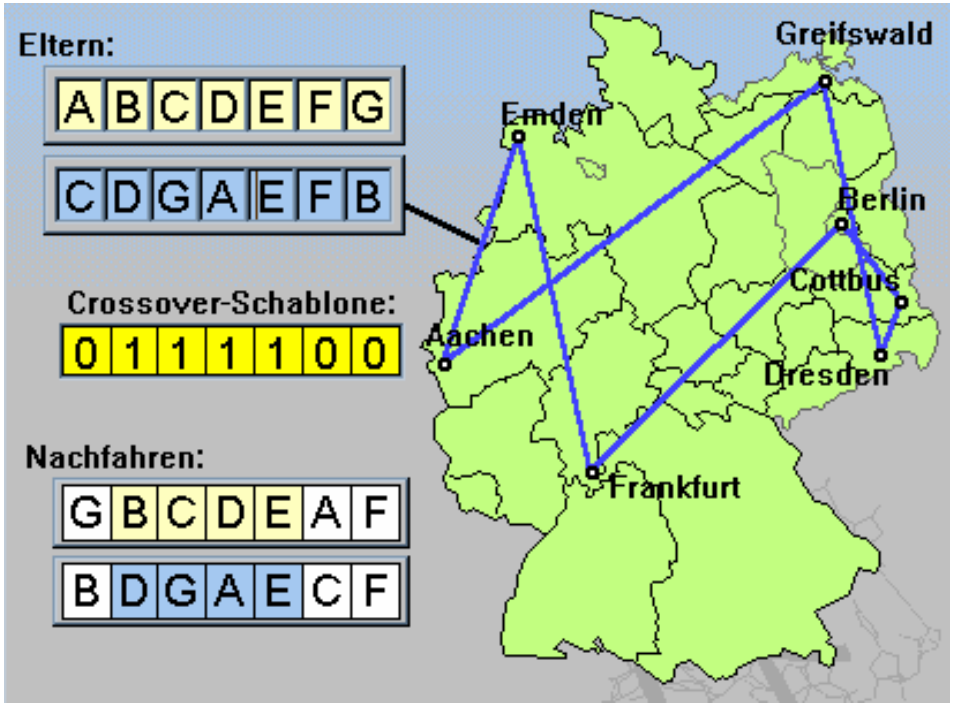
Beispiel „Rundreiseproblem“ – Crossover

1-Punkt-Crossover: $j := 4$
Crossover nach der 4ten Position



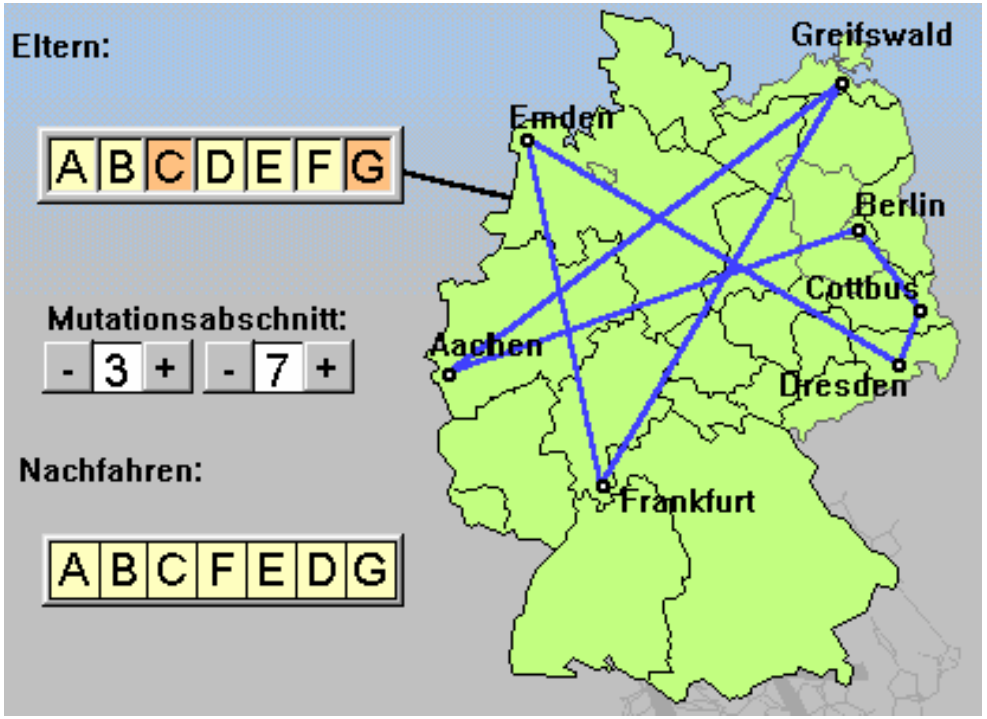
Beispiel „Rundreiseproblem“ – Crossover

Uniform Order−Based Crossover – 0/1-Muster der Länge 7
Vertausche 0-Positionen so, dass die Gene in die Reihenfolge des jeweils anderen Strings gebracht werden.



Beispiel „Rundreiseproblem“ – Mutation

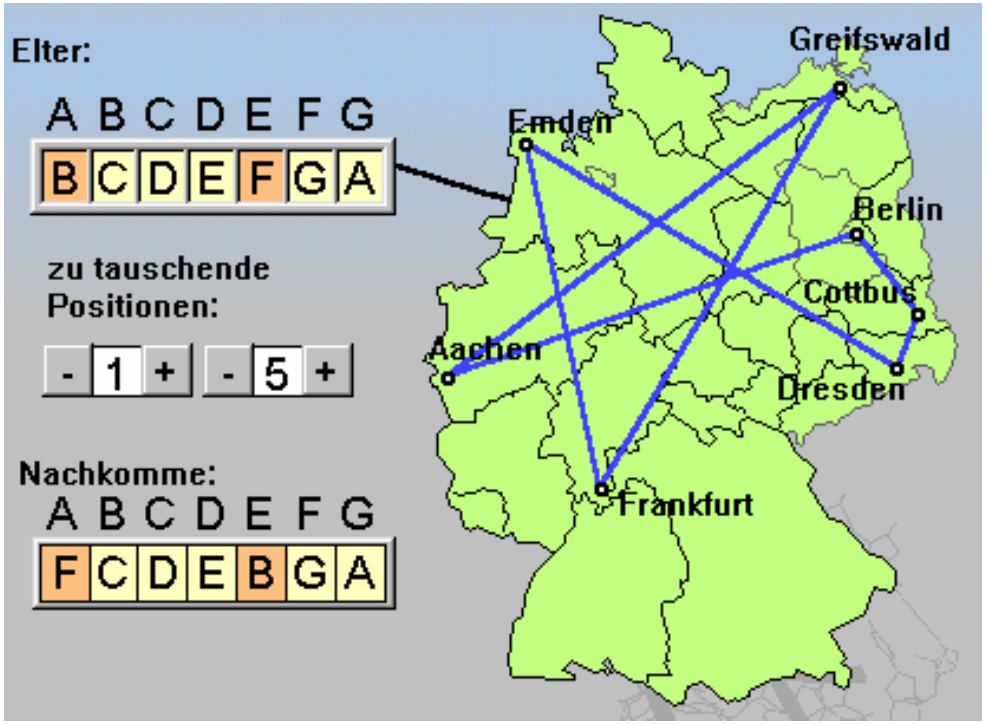
Bestimme zwei Positionen im String und durchlaufe die Städte dazwischen in umgekehrter Reihenfolge.



Beispiel „Rundreiseproblem“ – Mutation

Bestimme zwei Positionen im String und vertausche die jeweiligen Nachfolger.

Hinweis zur Kodierung:
Die Positionen im String (d.h. in einem Individuum) assoziiert man von vorneherein mit den Städten A, B, C, D, E, F, G und interpretiert die aktuell eingetragenen Buchstaben als Nachfolger dieser Städte in der Rundreise.
Oben ist also B Nachfolger von A, usw.; unten ist allerdings F Nachfolger von A, G Nachfolger von F usw.
Im Repetitorium wurden die daraus entstehenden Probleme diskutiert.



7.3 Bandabgleichproblem und Genetischer Algorithmus (10.02.2006)

Fließfertigung: Werkstücke von Station zu Station – gleiche Zeitspanne zwischen Verlassen aufeinander folgender Werkstücke – Taktzeit T

Planungsaufgabe: Wie viele Stationen sind einzurichten und welche Vorgänge sind auf welchen Stationen durchzuführen?

Restriktionen: Vorrangbeziehungen gegeben durch einen Graph

7.3.2 Ausgestaltung des Algorithmus

Der Bandwirkungsgrad $BW = (GZ/J)/T$ kann zur Bewertung der Fitness herangezogen werden. Dabei wird die minimale Taktzeit $T_{\min} := \lceil GZ/J \rceil$ durch die tatsächlich realisierte Taktzeit T dividiert. Das Aufrunden auf die nächst größere ganze Zahl, formalisiert durch die obere Gaußklammer $\lceil GZ/J \rceil$, stellt dabei eine zulässige Vereinfachung dar.

Algorithmus zur Fitnessbewertung

Schritt 1: Initialisierung

Berechnung der minimalen Taktzeit $T_{min} := \lceil GZ/J \rceil$; $T := T_{min}$

Für Individuum I führe Schritt 2 aus

Schritt 2: Iteration

Weise die Vorgänge in der Reihenfolge in I den Stationen 1 bis J zu.

Wird die Bearbeitungszeit in einer Station > T, eröffne eine neue.

Falls neue Statnr > J, setze $T := T + 1$ und beginne erneut mit Iteration.

Falls die Stationsbeschreibung eines Vorgangs eine frühere Station verlangt, setze Fitness (I) := 0.

Falls die Stationsbeschreibung eines Vorgangs eine spätere Station verlangt, eröffne diese.

Ist die Vorgangsliste von I leer, setze Fitness(I) := T_{min} /T.

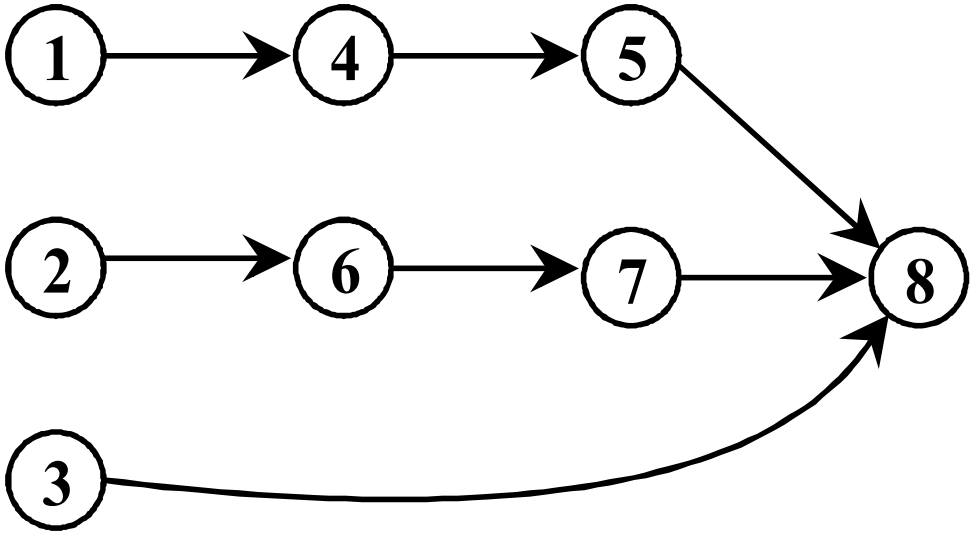
Abbruch: Eine Fitness-Zuweisung ist erfolgt.

Beispiel 7.1

Vorgänge: ①,②,③,④,⑤,⑥,⑦,⑧

Vorgangsdauern: 10, 11, 12, 11, 10, 13, 20, 11 [ZE]

Vorranggraph:



Für das Individuum ③①②⑥④⑤⑦⑧ führen wir folgende Schritte durch (Die Anzahl der Stationen sei auf 3 beschränkt):

$T = T_{\min} = \lceil 98/3 \rceil = 33$

STAT1	STAT2	STAT3	STAT4
③ 12	⑥ 13	⑤ 10	⑧ 11
① 10	④ 11	⑦ 20	
② 11			

Stationsnr. 4 > 3, setze T := 33 + 1 = 34 und beginne erneut mit Schritt 2.

STAT1	STAT2	STAT3	STAT4
③ 12	⑥ 13	⑦ 20	
① 10	④ 11	⑧ 11	
② 11	⑤ 10		

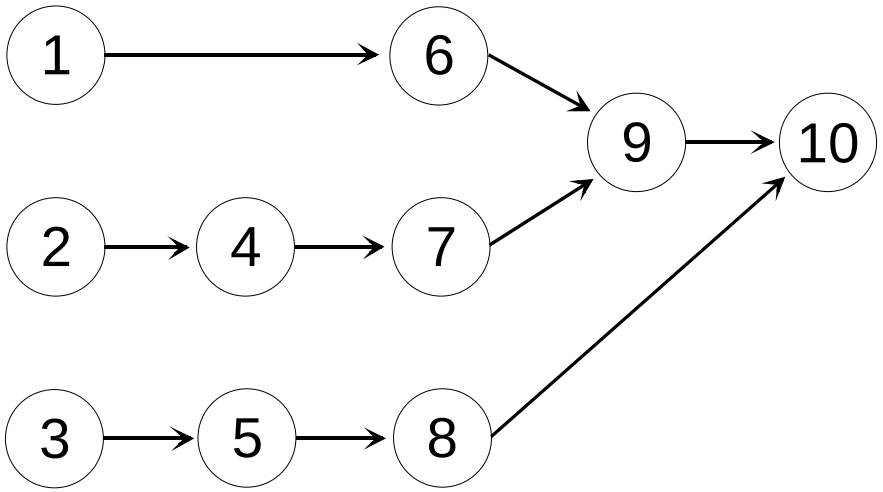
Vorgangsliste leer, setze Fitness (③①②⑥④⑤⑦⑧) := 33/34. Abbruch.

Aufgabe 6 des Repetitoriums

Zehn Vorgänge sollen auf drei Stationen angeordnet werden.

Vorgang:	1	2	3	4	5	6	7	8	9	10
Dauern: (ZE)	12	5	7	9	4	10	9	15	4	3

Vorranggraph:



Lösungshinweise zu Aufgabe 6 des Repetitoriums

a) $T_{\min} = \lceil 78/3 \rceil = 26$

b) **Für das Individuum ①②③④⑤⑥⑦⑧⑨⑩ kommt der letzte Schritt der Zuordnung nach Erhöhung von T auf 31 zu folgendem Ergebnis:**

STAT1	STAT2	STAT3	STAT4
① 12	④ 9	⑦ 9	
② 5	⑤ 4	⑧ 15	
③ 7	⑥ 10	⑨ 4	
		⑩ 3	

Der Fitnesswert berechnet sich somit zu
Fitness(①②③④⑤⑥⑦⑧⑨⑩)=26/31