DIE DYNAMISCHE OPTIMIERUNG BEIM GRADIENTENENTWURF

Wilhelm CASPARY
Hansbert HEISTER
Walter WELSCH

In: *CASPARY, Wilhelm / WELSCH, Walter (Hrsg.) [1982]:*
Beiträge zur großräumigen Neutrassierung
Schriftenreihe des Wissenschaftlichen Studiengangs Vermessungswesen der Hochschule der Bundeswehr München, Heft 6, S. 140-158
ISSN: 0173-1009

DIE DYNAMISCHE OPTIMIERUNG BEIM GRADIENTENENTWURF

von W. Caspary, H. Heister und W. Welsch

1. Einführung

Für das Problem der Gewinnmaximierung auf dem Gebiet der Unternehmensforschung wurden seit Beginn der fünfziger Jahre in Form der linearen und nichtlinearen Programmierung mathematische Lösungsmethoden entwickelt, die in den letzten Jahren mehr und mehr auch auf technische Problemstellungen übertragen wurden und sich ebenfalls dort als sehr wirkungsvoll erwiesen haben. Wesentlich dazu beigetragen hat die rapide Steigerung der Leistungsfähigkeit digitaler Rechenanlagen.

Aus derselben Zeit werden die ersten Versuche (MILLER (1958)) berichtet, die oft aufwendigen Trassierungsarbeiten zu automatisieren; dabei beschäftigte er sich noch hauptsächlich mit der Erstellung eines geeigneten digitalen Informationsmodells. Erst nachdem man die hierbei auftretenden Probleme gelöst hatte, war es möglich, eine automatische Trassenfindung anzugehen. Dies führte schließlich zur Anwendung verschiedener Optimierungsalgorithmen, worüber vor allem in den Arbeiten von HINTZEN (1965), BOESEFELD (1970), SCHECK (1973), KOCH (1975), MANOLOPOULOS (1979) berichtet wird. Hierzu mußten die bei der manuellen Bearbeitung meist intuitiv berücksichtigten Beurteilungskriterien mathematisch formuliert und den vorhandenen Optimierungsmodellen angepaßt werden.

Wie in dem Beitrag CASPARY, HEISTER und WELSCH (1982b) eingehend erläutert, entschied man sich bei der Bearbeitung der eigentlich dreidimensionalen Entwurfsaufgabe für eine Trennung in Lage und Höhe. Wegen der Vielschichtigkeit der Lageoptimierung wurde hier kein strenger mathematischer Weg beschritten, vielmehr wurde eine Lösung durch interaktives systematisches Probieren gesucht. Somit ist unter Beibehaltung der bereits optimierten Trasse die optimale Gradiente zu finden. Prinzipiell sind beide Probleme stark korreliert, jedoch erscheint zur Zeit eine ganzheitliche strenge mathematische Lösung nicht praktikabel.

Da bei der Gradientenbearbeitung weit weniger einengende Randbedingungen zu berücksichtigen sind sowie bereits geringfügige Änderungen der Entwurfspara-

meter großen Einfluß auf die Kosten haben, erscheint es hier sinnvoll, ein strenges mathematisches Modell zur Kostenoptimierung einzuführen.

Unter dem Begriff mathematische Optimierung faßt man eine Reihe von Methoden zusammen, die geeignet sind, Systeme hinsichtlich einer vorher definierten Zielgröße - der Zielfunktion - unter Einhaltung gewisser Nebenbedingungen - den Restriktionen - zu optimieren. Dabei werden diese Methoden nicht auf das System selbst, sondern auf eine modellhafte Nachbildung des Systems angewendet. Es wird daher eigentlich nicht das System, sondern das Modell optimiert. Nur bei hinreichend genauer Übereinstimmung kann deshalb der erzielte Optimierungseffekt auf das System übertragen werden. Um bei der vorliegenden Problemstellung eine möglichst geringe Klaffung zwischen System und Modell zu erreichen, sind unter Zugrundelegung des in CASPARY, HEISTER und WELSCH (1982a) beschriebenen digitalen Informationsmodells folgende Faktoren bei der Formulierung von Zielfunktion und Restriktionen zu berücksichtigen:

a) Der Gradientenverlauf soll möglichst kostengünstig sein. Dies ist hauptsächlich abhängig von den Größen Grunderwerb, Damm, Einschnitt, Höhe der Aufständerung, Geländequerneigung, Tunnel- und Brückenkosten.

b) Die Gradiente soll möglichst gut den fahrdynamischen Eigenschaften angepaßt sein; darunter ist eine flüssige Linienführung unter Einhaltung vorgegebener Mindestparameter zu verstehen.

c) Die Gradiente soll in vorgegebenen Abschnitten der systemtypischen Bauweise von Hochgeschwindigkeitsbahnen entsprechen.

d) Die Gradiente soll vorgegebene Zwangspunkte durchlaufen.

Nach einer Gegenüberstellung der verschiedenen Optimierungsverfahren, auf die hier nicht näher eingegangen werden soll, erscheint der Einsatz der dynamischen Optimierung als besonders zweckmäßig, da die Berücksichtigung der oben angegebenen Faktoren und somit eine gute Modellanpassung möglich ist; dabei braucht auf eine strenge mathematische Lösung nicht verzichtet zu werden. Das Modell und die erforderlichen Rechenregeln wurden von BELLMANN (1957) für die Optimierung sequentieller Entscheidungsprozesse entwickelt.

Um die Übertragung der dynamischen Optimierung auf die vorgegebene Aufgabenstellung besser nachvollziehen zu können, wird zunächst eine kurze Einführung in das Verfahren und die Terminologie der Unternehmensforschung gegeben.

2. Das mathematische Konzept

2.1 Dynamische Optimierung

Man geht davon aus, daß der Zustand eines Systems zu jedem Zeitpunkt durch einen Zustandsvektor

$$z_i^T = (z_i^1, z_i^2, \ldots, z_i^p) \tag{1}$$

festgelegt werden kann. Unterteilt man das System in i Stufen, auf denen jeweils eine Entscheidung

$$e_i^T = (e_i^1, e_i^2, \ldots, e_i^q) \tag{2}$$

gefällt werden kann, die das gesamte System beeinflußt, dann kann man für den Übergang von einer Stufe i zur Stufe i+1 folgende Transformation anschreiben, um den neuen Zustandsvektor zu bestimmen:

$$z_{i+1} = v_i(z_i, e_i). \tag{3}$$

Weiterhin wird der Gewinn auf die Stufe i, der abhängt von dem Zustand z_i und der vorgenommenen Entscheidung e_i, mit $u_i(z_i, e_i)$ bezeichnet. Der Gesamtgewinn über N Stufen ergibt sich dann zu:

$$U = \sum_{i=1}^{N} u_i(z_i, e_i) \tag{4}$$

Trifft man nun auf jeder Stufe nur solche Entscheidungen, die die Gewinnfunktion U maximiert, so ist die Lösung des dynamischen Programms die optimale Strategie

$$E^* = (e_1^*, e_2^*, \ldots, e_N^*) \tag{5}$$

Bei vorgegebenem Anfangszustand z_1 ist dann über die vorgegebene Transformation (3) mit dem optimalen Entscheidungsvektor e_1^* der optimale Zustand z_2^* zu berechnen. So fährt man fort bis die entsprechende Zustandsfolge erreicht ist.

$$Z^* = (z_1, z_2^*, z_3^*, \ldots, z_{N+1}^*) \; . \tag{6}$$

Jetzt ist nur noch die Frage zu klären: Wie bestimmt man die optimale Strategie? Dazu bedient man sich der Rückwärtsrechnung mit folgenden Rechenregeln:
Ausgehend von der Stufe N berechnet man zunächst alle möglichen Gewinne u_N durch Einsetzen aller Entscheidungsmöglichkeiten:

$$u_N = u_N(z_N, e_N) + u_{N+1}^* \tag{7}$$

mit $u_{N+1}^* = u_{N+1}(z_{N+1}) = 0$.

Von diesen Werten ist aber nur der optimale für die weitere Rechnung von Bedeutung:

$$u_N^* = \max \{u_N(z_N, e_N)\} \tag{8}$$

Geht man zur nächsten Stufe N-1 zurück, dann lassen sich wieder Gewinne u_{N-1} berechnen, die mit dem optimalen Gewinn u_N^* folgende Gewinnsummen ermöglichen:

$$w_{N-1} = u_{N-1}(z_{N-1}, e_{N-1}) + u_N^* \tag{9}$$

Sucht man jetzt wieder den maximalen Wert von u_{N-1}, dann erhält man auf der Stufe N-1:

$$\max w_{N-1} = u_{N-1}^* = \max \{u_{N-1}(z_{N-1}, e_{N-1}) + u_N^*\} \ . \tag{10}$$

Allgemein gilt für Stufe i:

$$u_i^* = \max w_i = \max \{u_i(z_i, e_i) + u_{i+1}^*\}, \ i = N, N-1, \ldots , 1. \tag{11}$$

Setzt man die Rückwärtsrechnung bis zur 1. Stufe fort, ergibt sich der optimale Wert der Zielfunktion (4) zu:

$$U^* = u_1^* = \max w_1 = \max \{u_1(z_1, e_1) + u_2^*\} \ . \tag{12}$$

Hat man so den optimalen Gewinn berechnet, können die optimale Strategie E^* und Z^* durch Vorwärtsrechnung bestimmt werden; bei der Rückwärtsrechnung wurden nämlich neben den Funktionswerten u_i^* auch die optimalen Entscheidungen für jeden Zustand abgespeichert. Dann läßt sich aus (12) bei vorgegebenem Anfangszustand z_1 und bekannten u_1^*, u_2^* die optimale Entscheidung e_i^* festlegen. Anschließend kann über die Transformation (3) der optimale Zustand z_2^* ermittelt werden. Für die i-te Stufe gilt

$$z_i^* = v_{i-1}(z_{i-1}^*, e_{i-1}^*) \tag{13}$$

mit

$$e_{i-1}^* = e_{i-1}(z_{i-1}^*) \ . \tag{14}$$

Damit ist das Ziel der dynamischen Optimierung erreicht, nämlich nach Bestimmung des optimalen Wertes U^* die optimale Strategie E^* und Zustandsfolge Z^* zu kennen.

Dieser zunächst abstrakte Lösungsweg soll durch ein kleines Beispiel auch graphisch näher erläutert werden:

2.2 Beispiel

Um direkt den Bezug zur Trassierung herzustellen, wird eine Gradiente vereinfacht durch ein Tangentenpolynom dargestellt (siehe Abbildung 1a). Der optimale Verlauf dieses Polynoms ist nur durch Veränderung der Tangentenanstiege zu erreichen. Durch die Anzahl der Tangenten ist das System in N=4 Stufen unterteilt. Auf jeder Stufe sind die Entscheidungsmöglichkeiten zur Änderung der Tangente durch den Entscheidungsvektor e_i (i = 1, ... , 4) festgelegt. Zum Beispiel kann man auf der Stufe 1 den Anstieg der Tangente auf 0 %, 3 % und 6 % festlegen. Ist der Zustand

$$z_1 = 0 \text{ [m]} , \tag{15}$$

so wird gemäß (3) mit dem Abstand s_1 der Tangentenschnittpunkte der Zustand zu Beginn der 2. Stufe durch die Transformation berechnet:

$$z_2 = v_1(z_1, e_1) , \tag{16}$$

$$z_2 = z_1 + s_1 e_1 = \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix} + 100 \cdot \begin{vmatrix} 0{,}00 \\ 0{,}03 \\ 0{,}06 \end{vmatrix} = \begin{vmatrix} 0{,}00 \\ 3{,}00 \\ 6{,}00 \end{vmatrix} . \tag{17}$$

Dieser neue "Zustand" ist in der Abbildung 1a durch die Gitterpunkte (100,0; 100,3; 100,6) graphisch dargestellt. Weiterhin kann jetzt auch für diese Stufe je nach Entscheidung die Gewinnfunktion u_i berechnet werden. Hier wurde ein Kostenmodell simuliert, das von der Fläche zwischen Tangente und Gelände abhängt. Je nach Anstieg betragen die Kosten, die auch als negative Gewinne berechnet werden können, -20, -10 oder -6.

Auf der 2. Stufe können jetzt in jedem Gitterpunkt wieder die Entscheidungen e_2 gefällt werden. Dies führt wieder zu neuen Gitterpunkten und entsprechenden Kosten. So fährt man fort bis zur 4. Stufe. Alle Möglichkeiten sind in Abbildung 1a aufgezeigt.

Ziel der Optimierung ist die Bestimmung desjenigen Tangentenpolynoms, das minimale Kosten verursacht oder maximalen Gewinn bringt.

$$\sum_{i=1}^{4} u_i(z_i, e_i) = U \rightarrow \text{Max.} \tag{18}$$

Nach (7) bestimmt man alle möglichen Gewinne u_4 auf der 4. Stufe, wobei die Endkosten mit $u_5^* = 0$ angesetzt werden. Die Einzelwerte sind der Abbildung 1b zu entnehmen. Hiervon sind jedoch nur die minimalen Werte u_4^* für den weiteren Rechengang interessant. Geht man jetzt auf die 3. Stufe zurück, bildet man die Kostensummen w_3 nach (9). Für jeden Knotenpunkt zu Beginn der 3. Stu-

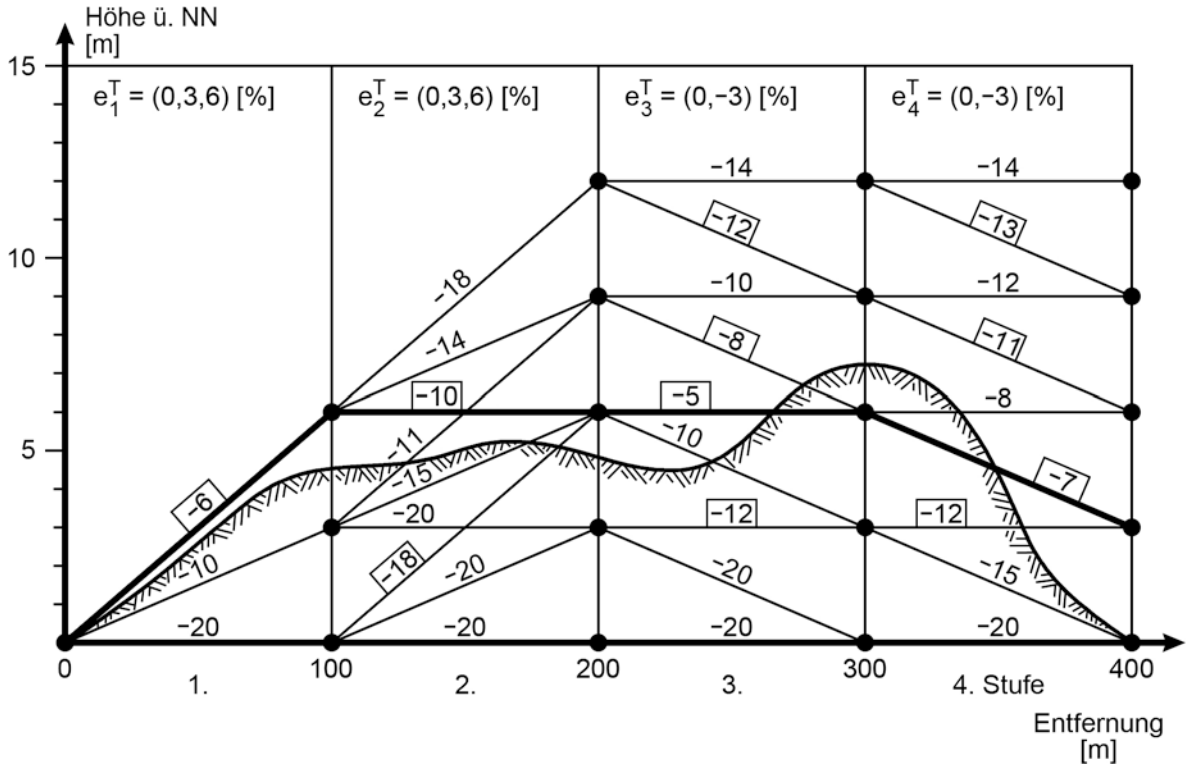

Abb. 1a: Darstellung aller möglichen Tangentenpolynome

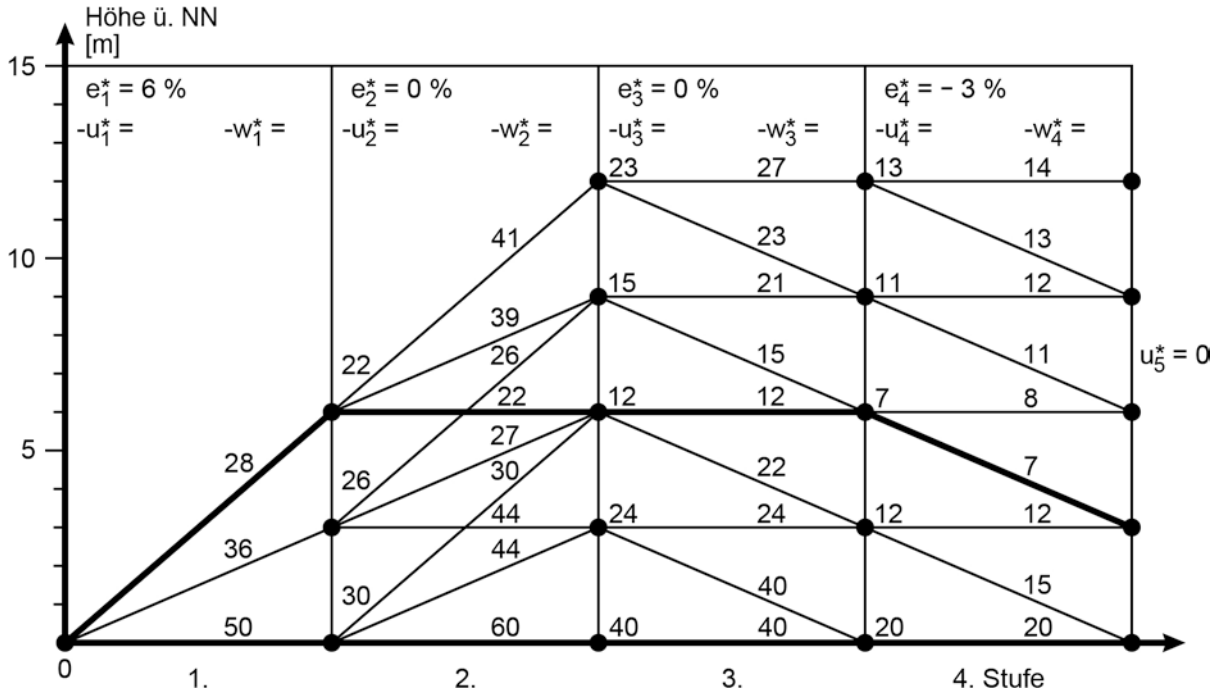

Abb. 1b: Darstellung der Rückwärtsrechnung

fe wird jetzt wieder das Optimum u_3^* gesucht. Gleichzeitig wird die hierfür relevante Entscheidungsvariable notiert; sie wurde in der Abbildung 1a jeweils durch das Symbol ☐ gekennzeichnet.

Fährt man in diesem Schema bis zur ersten Stufe fort, so ergibt sich der optimale Wert der Zielfunktion (18) zu:

$$U^* = u_1^* = -28 \ .$$

Mit Hilfe der Vorwärtsrechnung läßt sich dann gemäß (13) und (14) die optimale Strategie $\mathcal{E}^*$ und die Zustandsfolge $\mathcal{Z}^*$ eindeutig bestimmen:

$$\begin{aligned}
&z_1 = 0 \\
&z_2^* = z_1 + s_1\, e_1^* = 0 + 100 \cdot 0{,}06 = 6 \\
&\qquad e_1^* = e_1\,(z_1) = 0{,}06 \\
&z_3^* = z_2^* + s_2\, e_2^* = 6 + 100 \cdot 0{,}00 = 6 \\
&\qquad e_2^* = e_2\,(z_2^*) = 0{,}00 \\
&z_4^* = z_3^* + s_3\, e_3^* = 6 + 100 \cdot 0{,}00 = 6 \\
&\qquad e_3^* = e_3\,(z_3^*) = 0{,}00 \\
&z_5^* = z_4^* + s_4\, e_4^* = 6 + 100 \cdot (-0{,}03) = 3 \\
&\qquad e_4^* = e_4\,(z_4^*) = -0{,}03
\end{aligned} \qquad (19)$$

Damit liegt der Verlauf des kostengünstigsten Tangentenpolynoms fest.

2.3 Diskretisierung

Bei der numerischen Behandlung größerer Probleme wird man sich darauf beschränken, die Funktionen u_i^* nur in einem gewissen Bereich zu bestimmen, um hier nicht sinnvolle Lösungswege, die den Rechenvorgang unnötig aufblähen, auszuschalten. Diese Begrenzung wird durch die Vorgabe von Maximal- und Minimalwerten der Zustandsvariablen erreicht:

$$\begin{aligned}
&z_{iMin}^k \le z_i^k \le z_{iMax}^k \ . \\
&i = 1, \ \dots \ , N+1 \\
&k = 1, \ \dots \ , p
\end{aligned} \qquad (20)$$

Zusätzlich kann dieser Zustandsbereich durch Vorgabe einer Schrittweite Δz und eines Diskretisierungsfaktors l diskretisiert werden, so daß nur auf Gitterpunkten eines vorgegebenen Rasters die Funktionsberechnung nach (11) durchgeführt wird. Legt man noch die Anzahl der Entscheidungsmöglichkeiten auf jeder Stufe j fest, dann sind die Bereiche von folgender Form:

$$z_j = \left\{ \begin{pmatrix} z_j^1 \\ \vdots \\ z_j^p \end{pmatrix} , \; z_j^i = z_{jMin} + \lambda^i \, \Delta z_j^i \; , \; \lambda^i = 0, \, 1, \, \ldots, \, l_j^i \right\}, \qquad (21)$$

$$i = 1, \, 2, \, \ldots \, , \, p$$

$$\mathcal{E}_j = \left\{ \begin{pmatrix} e_j^1 \\ \vdots \\ e_j^q \end{pmatrix} , \; e_j^\nu \in \mathcal{E}_j^\nu \; , \; \nu = 1, \, 2, \, \ldots, \, q \right\} . \qquad (22)$$

$$j = 1, \, \ldots \, , \, N$$

Ist dabei r_j^ν die Anzahl von Entscheidungsmöglichkeiten der Komponente e_j^ν, dann gibt es auf der Stufe j insgesamt

$$R_j = \prod_{\nu=1}^{q} r_s^\nu \qquad (23)$$

Entscheidungen. Ist die Anzahl der Gitterpunkte

$$G_j = \prod_{i=1}^{p} l_j^i \; , \qquad (24)$$

dann ergibt sich für die Gesamtzahl der Funktionsberechnungen auf einer Stufe

$$n_j = \sum_{k=1}^{G_j} R_j^k \; . \qquad (25)$$

Die Beschränkung und Unterteilung des Zustandsbereiches bedingt, daß der erhaltene Gewinn U^* nur optimal bezüglich der vorgegebenen Diskretisierung ist. Je feiner diese gewählt wird, um so dichter wird man das wahre Optimum erreichen. Dies bedeutet jedoch eine wesentliche Steigerung des Gesamtrechen-

aufwandes, der folgendermaßen abschätzbar ist.

Es gibt bedingt durch die Anzahl G_j der Gitterpunkte auf jeder Stufe j insgesamt

$$G = \prod_{j=1}^{N} G_j \tag{26}$$

Möglichkeiten, den Gewinn U zu berechnen. Geht man von der Vereinfachung aus, daß die Anzahl der Gitterpunkte auf jeder Stufe gleich ist, dann sind $G = G_j^N$ Berechnungen notwendig. Bei der dynamischen Optimierung ist jedoch diese Anzahl wesentlich geringer, da durch die lokale Maximierung auf jedem Gitterpunkt nach (11) alle nicht optimalen Entscheidungen entfallen. Ist die Anzahl der Gitterpunkte bzw. Entscheidungsmöglichkeiten für alle Stufen gleich, dann ergibt sich aus (24), (23) bzw. (25) für jede j-te Stufe

die Anzahl der Gitterpunkte zu $G_j = (l_j+1)^p$,
die Anzahl der Entscheidungsmöglichkeiten zu $R_j = (r_j)^q$,
die Anzahl der Funktionswertberechnungen zu $n_j = (l_j+1)^p \cdot (r_j)^q$.

Mit $l_j+1 = r_j = 1$ kann schließlich näherungsweise zur Abschätzung des Rechenaufwandes die Anzahl n der zu berechnenden w_i nach (11) über N Stufen mit

$$n = N \cdot l^{(p+q)} \tag{27}$$

angegeben werden. Dabei beträgt die Anzahl $\bar{n}$ der lokalen Maximalwerte u_i^*, die zur Vorwärtsrechnung abgespeichert werden müssen

$$\bar{n} = N \cdot l^2 \tag{28}$$

Sind zum Beispiel p=q=3, N=10 und l=10, dann wäre die Gesamtzahl aller möglichen Berechnungen $G = (10^3)^{10} = 10^{30}$ (!); die Anzahl der durch die dynamische Optimierung notwendigen Funktionsberechnungen beträgt nach (27) jedoch nur $n = 10 \cdot 10^6 = 10^7$. Die Zahl der abgespeicherten Funktionswerte ist letztlich $\bar{n} = 10 \cdot 10^2 = 1\ 000$.

Dieses Beispiel zeigt, daß bei mehrdimensionalen Problemen dem Bellmanschen Lösungsprinzip, trotz der starken Reduzierung der Funktionswertberechnungen, Grenzen gesetzt sind, da der Rechenaufwand mit der Dimension etwa exponentiell ansteigt. Deshalb ist bei der Formulierung des mathematischen Modells darauf zu achten, daß

a) die Dimensionen des Zustands- und Entscheidungsvektors möglichst klein sind,
b) der Diskretisierungsfaktor klein und
c) die Zielfunktion sowie die Transformation möglichst einfach sind.

Sind der erlaubte Zustandsbereich sowie sein Diskretisierungsfaktor groß und wird zusätzlich eine hohe Genauigkeit gefordert, empfiehlt sich folgendes iterative Vorgehen:

Man bestimmt zunächst mit grober Diskretisierung Δz_j^{i0} eine vorläufige optimale Zustandsfolge $Z^{*0} = (z_1^0, z_2^0, \ldots, z_N^0)$ aus den erlaubten Zustandsmengen $Z_1^0, Z_2^0, \ldots, Z_N^0$ und erhält den Gewinn U^{*0}. Man legt für den nächsten Iterationsschritt die Zustandsmengen so fest, daß um die vorläufig optimale Zustandsfolge mit halbem Rasterwert $\Delta z_j^{i1} = 1/2\ \Delta z_j^{i0}$ eine neue Zustandsmenge $Z_1^1, Z_2^1, \ldots, Z_N^1$ gebildet werden kann. In diesem "Suchschlauch" wird nun die nächste optimale Zustandsfolge Z^{*1} berechnet. Man fährt so fort, bis ein definiertes Abbruchkriterium erfüllt ist. Ob das Iterationsverfahren gegen das gesuchte Optimum konvergiert, ist allgemein nicht nachweisbar. Man kann jedoch mit großer Wahrscheinlichkeit sagen, daß das gesuchte Optimum innerhalb des Suchschlauchs liegt, wenn ein zweitbester Zustandsvektor noch innerhalb der Grenzen des neuen Zustandsbereiches liegt. Deshalb ist es sinnvoll, beim k-ten Iterationsschritt bereits die Zustandsgrenzen des (k+1)-ten Schrittes zu bestimmen und zu überprüfen, ob die "zweitbeste" Zustandsfolge innerhalb dieser Grenzen liegt.

3. Die Gradientenoptimierung

3.1 Entscheidungs- und Zustandsparameter bei der Gradientenbearbeitung

Gewöhnlich wird die Gradiente nicht - wie im oben angegebenen Beispiel - durch einen Parameter beschrieben, sondern durch mehrere, wie Gerade und Kreis im vorliegenden Fall. Für die Anwendung der dynamischen Optimierung muß die Gradientenberechnung so umformuliert werden, daß sie sich in das Schema des Optimierungsalgorithmus fügt; dabei ist für die praktische Durchführung wichtig, daß Zielfunktion (4) und Transformation (3) einen möglichst einfachen mathematischen Ausdruck bilden. Denn wie bereits aus dem Beispiel ersichtlich, werden diese Funktionen sehr oft berechnet und bilden somit den Hauptanteil an Rechenzeitbedarf.

Die Gradiente des Vorentwurfs wird deshalb in Abschnitte unterteilt, in de-

nen jeweils die Parameterfolge Gerade - Kreis oder Kreis - Gerade erlaubt ist. Diese Parameterfolge entspricht jeweils einer Stufe. Wichtig hierbei ist, daß die Anzahl N der Stufen durch den Optimierungsalgorithmus nicht geändert werden kann. Der Entwurfsingenieur hat hier bereits die Möglichkeit, auf eine ausgewogene, fahrdynamisch günstige Linienführung Einfluß auszuüben.

Der Zustand zu jedem Abschnittsbeginn ist festgelegt durch den dreidimensionalen Zustandsvektor

$$\boldsymbol{z}_i^T = (r_i,\ h_i,\ m_i)\ \boldsymbol{z}_i \in Z_i\ , \tag{29}$$

wobei

r die Länge der rektifizierten Trasse [m],
h die Höhe der Gradiente ü. NN [m],
m der Anstieg [‰],
i der Stufenindex (i=1, ..., N)

bedeutet. Auf jeder Stufe können jetzt Entscheidungen gefällt werden, die mehr oder weniger kostengünstig sind

$$\boldsymbol{e}_i^T = (r_i,\ h_i,\ m_i)\ \boldsymbol{e}_i \in E_i\ . \tag{30}$$

Diese Entscheidungen implizieren die Festlegung einer Parameterfolge, wie leicht aus Abb. 2 ersichtlich ist:

Im Punkt A auf der Stufe i ist der Zustand festgelegt durch die Koordinaten r_i, h_i und den Anstieg m_i. Der Punkt E kann in einem vorgegebenen Raster durch die Entscheidungsmöglichkeiten für r_i, h_i und m_i verändert werden. Gesucht ist natürlich die optimale Entscheidung r_i^*, h_i^* und m_i^*. Dadurch ist indirekt der Tangentenschnittpunkt TS festgelegt, der wiederum eine eindeutige Bestimmung der Gradientenparameter Gerade - Kreis zuläßt.

Die Transformation nach (3) ist durch Wahl der Größen $\boldsymbol{e}$ und $\boldsymbol{z}$ trivial:

$$\boldsymbol{z}_{i+1} = \boldsymbol{e}_i\ . \tag{31}$$

Hierdurch ist ein stetiger Übergang von einer Parameterfolge zur nachfolgenden gewährleitstet, so daß der Gradientenverlauf glatt ist.

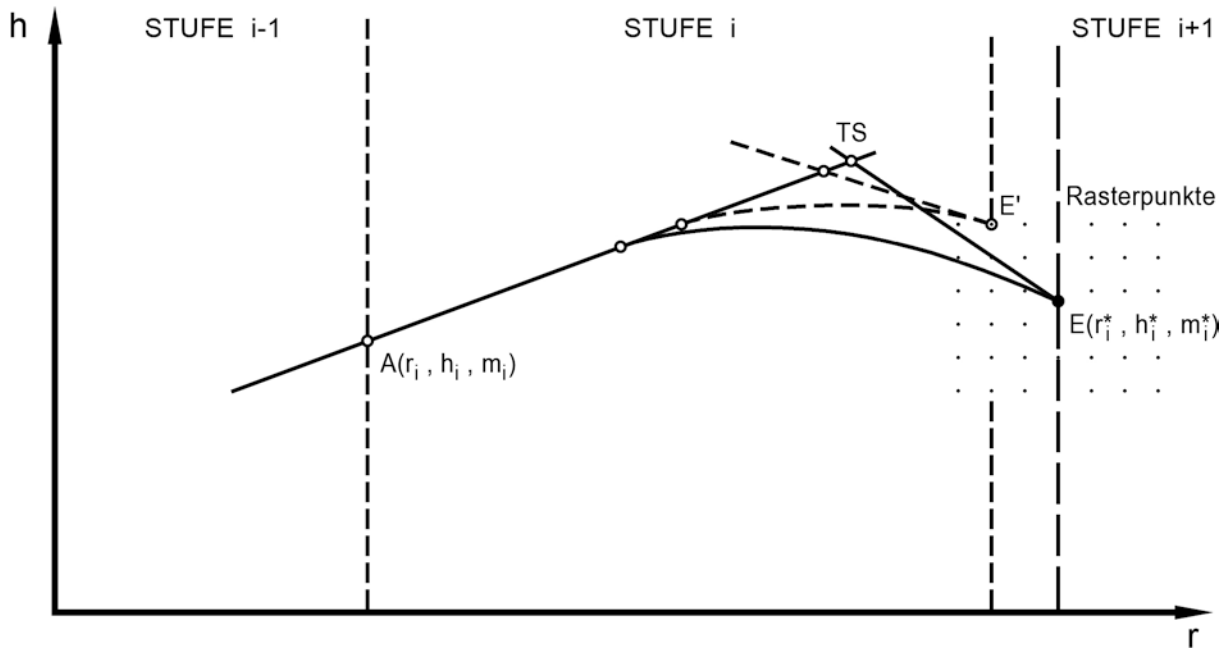

Abb. 2: Darstellung einer Parameterfolge

3.2 Zielfunktion und Restriktionen

Von der Ausbildung der Zielfunktion hängt weiterhin ganz wesentlich die Güte des Optimierungsergebnisses ab. Hier steht man im Zwiespalt zwischen möglichst exakter Modellanpassung und wirtschaftlich vertretbarem Rechenaufwand; schließlich kann sogar die Praktikabilität des Optimierungsverfahrens für große Systeme in Frage gestellt werden.

Da die Kosten minimiert werden sollen, muß zunächst entschieden werden, ob eine einheitliche Funktion u_i zur Kostenberechnung in dem Planungsstadium bereits erstellt werden kann oder ob hierfür genäherte Größen wie zum Beispiel Fläche zwischen Gelände und Gradiente oder Höhendifferenz - Gradiente - Gelände genügen.

In zwei Projektstudien, für die dieses Optimierungskonzept entworfen wurde,

ist ein sehr detailliertes Kostenmodell (siehe SCHWINTZER, STOECKL und FISCHER (1982)) entwickelt worden, so daß zunächst eine rechenaufwendige Zielfunktion unumgänglich schien. Zusätzlich sollten auch noch die Restriktionen berücksichtigt werden, die sich aus den Forderungen b) bis d) des ersten Abschnitts ergeben.

Diese Bedingungen führten zu folgender Kostenzielfunktion:

$$KO = \sum_{i=1}^{N} \sum_{j=1}^{k^i} |\Delta h_j|\ K_j \cdot p_j \ . \qquad (32)$$

Hierin bedeuten (siehe auch Abb. 3)

k^i = Anzahl der Geländepunkte, die im Bereich der Stufe i liegen,

Δh_j = Höhendifferenz Gelände - Gradiente,

K_j = Kostenfaktor,

p_j = Gewicht.

$j = 1, \ldots, k^i$

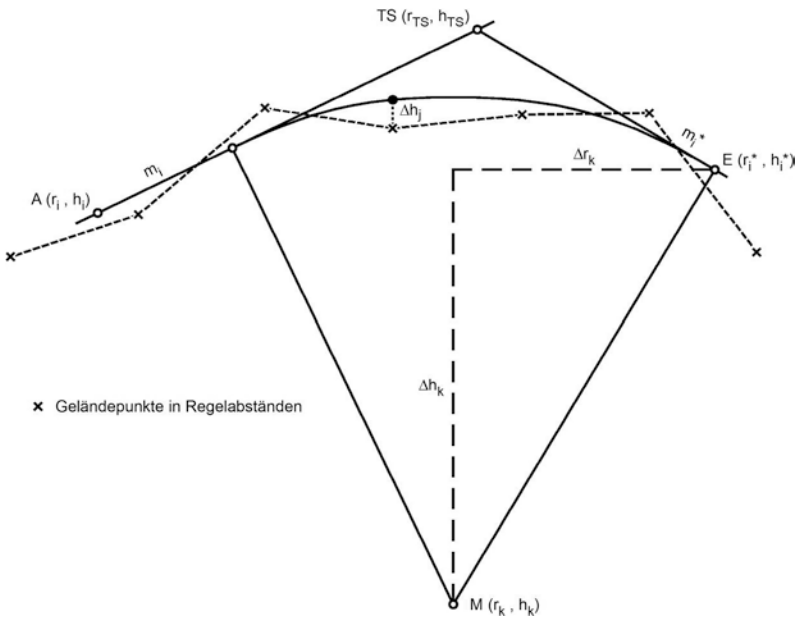

Abb. 3: Geometrische Darstellung zur Kostenfunktion

Diese einfache Gestalt der Zielfunktion ist dadurch erreicht worden, daß die Kostenfunktion durch die Faktoren K_j diskretisiert wurde. Um für jede Station

den richtigen Kostenfaktor zu finden, ist ein lokales Informationsmodell erstellt worden. Dieses digitale Modell enthält die Größen

- Geländehöhe in Regelabständen h_j,
- entsprechende Geländequerneigung q_j,
- Gewicht p_j, bestehend aus sieben Ziffern (xxxx.xx).

Hierbei dienen die Vorkommaziffern als Gewicht und ermöglichen die Berücksichtigung von Zwangspunkten. Die Nachkommaziffern regeln die Lage und Klassifizierung von Straßen zwischen zwei benachbarten Stützpunkten. Zu jedem Höhenunterschied Δh_j, der zum Beispiel in Regelabständen von 100 m bestimmt wird, kann dann unter zusätzlicher Berücksichtigung der Querneigung auf den entsprechenden Kostenfaktor K_j in einer Tabelle zugegriffen werden. Diese Tabelle hat zwei Eingänge und zwar Δh in der Schrittweite 1 m und die Querneigung in 2 %-Schritten. Sind die Nachkommastellen von p_j besetzt, so werden noch die Kosten für Kreuzungsbauwerke einer zweiten Tabelle entnommen und in Abhängigkeit der Entfernung Kreuzungsbauwerk - Stützpunkt proportional auf die Kostenfaktoren K_j und K_{j+1} verteilt.

Damit ist es gelungen, in einer einfachen Zielfunktion ein kompliziertes Kostenmodell zu berücksichtigen; gleichzeitig impliziert diese Funktion sogar noch gewisse Restriktionen wie die Berücksichtigung von Zwangspunkten durch individuelle Steuerung des Gewichtes p_j. Mindestparameter werden dadurch steuerbar, daß nur solche Entscheidungen zulässig sind, aus denen sich Parameter berechnen lassen, die nur im vorgegebenen Bereich liegen. Ein Sprungbefehl im Programm bewirkt, daß für unzulässige Entscheidungen keine Funktionswerte gemäß (11) berechnet werden können.

Schließlich sei noch erwähnt, daß zur Manipulation der Fahrdynamik entscheidend die Anzahl des Stufenindexes N ist. Je mehr Stufen bzw. Parameterfolgen für einen Abschnitt angesetzt werden, um so besser kann sich die Gradiente dem Gelände anpassen, um so unruhiger wird aber die Linienführung.

Weiterhin ist zu berücksichtigen, daß die Anzahl N der Stufen nach (27) mit verantwortlich für den Rechenaufwand ist. Bei numerischen Berechnungen hat sich $N \leq 15$ bewährt; dies bedeutete eine Länge zwischen 20 bis 30 km des zu bearbeitenden Abschnitts.

Weitere Details hierzu können dem Benutzerhandbuch TROP (1981) entnommen werden.

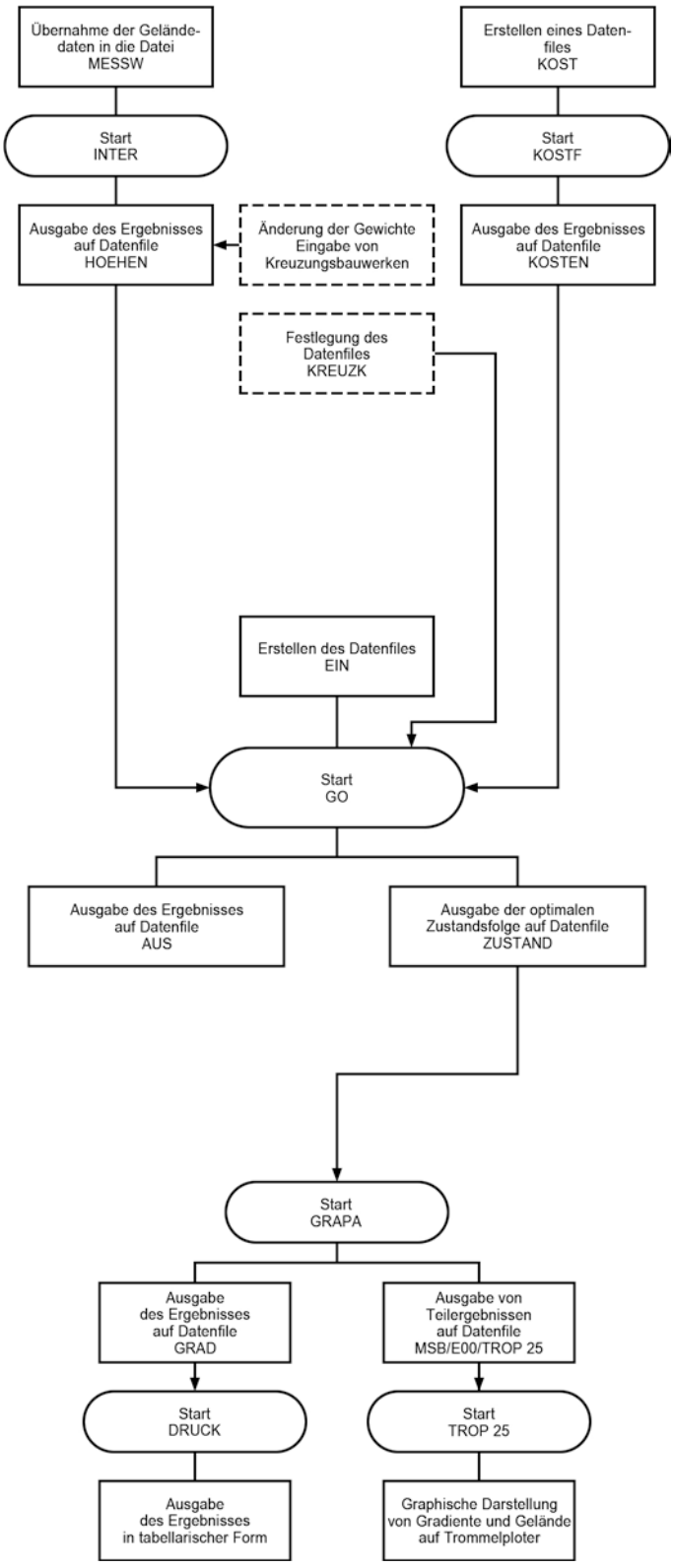

Abb. 4: Ablaufplan zur Gradientenoptimierung

3.3 Ein dialogfähiges Optimierungsprogramm

Nach dem dargelegten mathematischen Konzept wurde ein dynamisches Optimierungsprogramm mit dreidimensionalen Zustands- und Entscheidungsvektoren erstellt. Trotz der für den mathematisch weniger vorgebildeten Nutzer komplizierten Materie ist ein einfach handzuhabendes, dialogfähiges Programm EGO entstanden, dessen Arbeitsablauf in Abbildung 4 entnommen werden kann.

Um das Hauptprogramm zur Gradienten-Optimierung (GO) zu starten, müssen vorab vier Datenfiles erstellt werden: HOEHEN, KOSTEN, KREUZK und EIN. Der erste File enthält die geländebeschreibenden Werte Höhe und Querneigung in Trassenachse und Regelabstände von 100 m sowie das Gewicht eines jeden Stützpunktes, vorbesetzt mit Eins. Die Datei KOSTEN enthält die Erstellungskosten pro m Höhendifferenz und je laufendem m Trasse in Abhängigkeit von Höhendifferenz und Querneigung. In KREUZK ist die Kastentabelle für Kreuzungsbauwerke abgespeichert; als Eingangsgrößen zur Ermittlung der Kosten sind die Höhendifferenz und Klassifizierung der Straße notwendig. Die Datei EIN enthält schließlich die Steuerparameter für das Hauptprogramm; dies sind insbesondere

- der Stufenindex N,
- die Untergrenze des Zustandsbereiches auf jeder Stufe für jede Komponente,
- der Diskretisierungsfaktor für jede Stufe und jede Komponente,
- die Rasterweite,
- die Entwurfsgeschwindigkeit.

Die Ergebnisse der Optimierung werden an das Programm GRAPA zur Berechnung der Gradientenparameter übergeben, wo dann zwei Ergebnisdateien erstellt werden, die entweder einen sachgerechten Ausdruck über das Programm DRUCK ermöglichen oder aber über TROP 25 den in der Abbildung 5 dargestellten optimalen Gradientenverlauf erzeugen.

4. Bewertung der Ergebnisse und Ausblick

Das beschriebene Programmsystem EGO wurde in zwei Projektstudien zur Trassierung von Hochgeschwindigkeitsbahnen angewendet. Dies führte je nach Streckenabschnitt zu einer Reduzierung der Kosten zwischen 10 % bis 20 %. Da die reine Kostenminimierung eine Gradientenführung festlegt, die nicht in allen Bereichen unter den Gesichtspunkten des Fahrkomforts und der Umweltfreundlich-

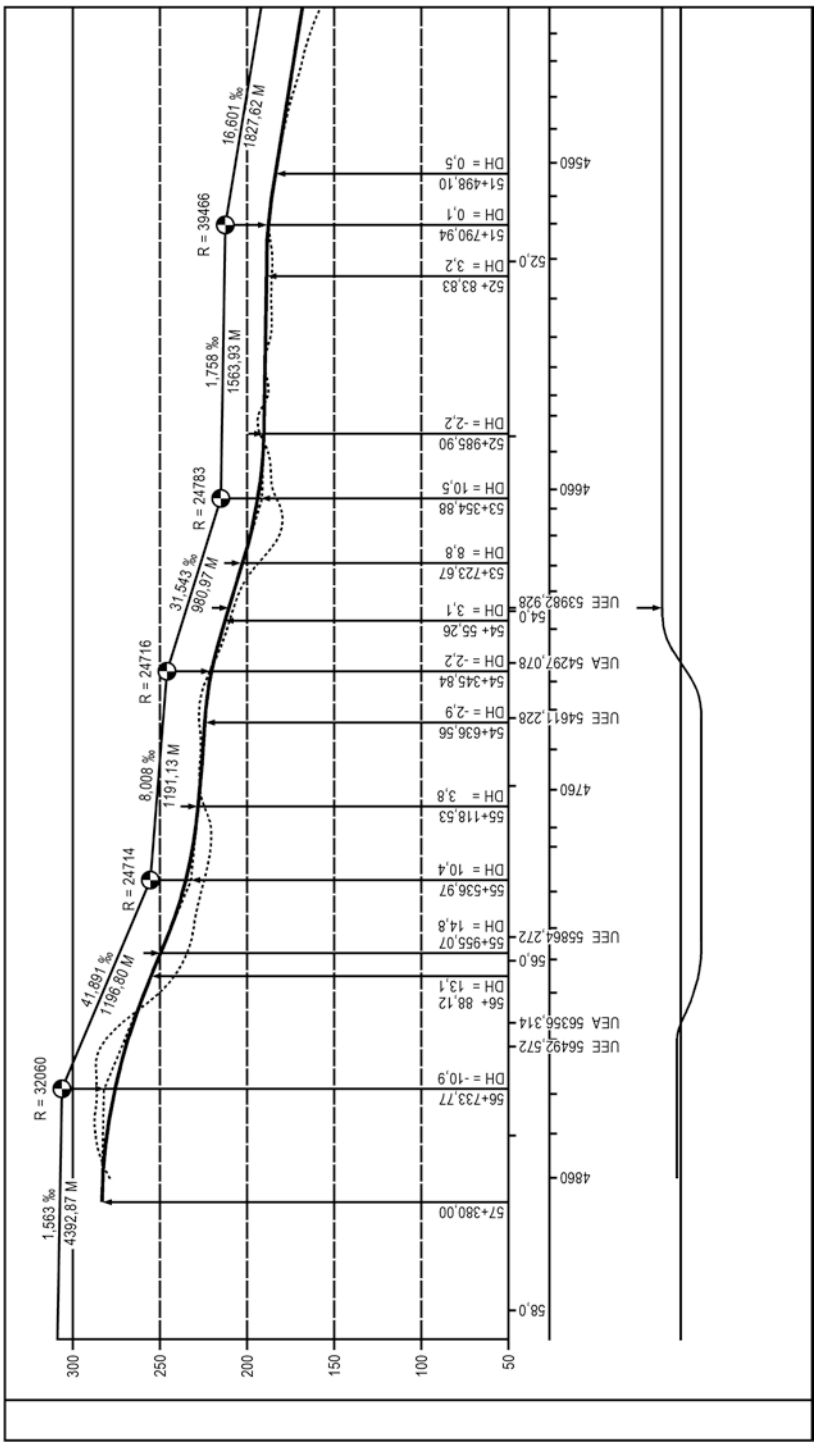

Abb. 5: Optimaler Gradientenverlauf

keit sowie der systemspezifischen Bauweise die Vorstellungen des Trassierungsingenieurs erfüllte, wurde in einzelnen Abschnitten die Gradiente nochmals überarbeitet. Es zeigt sich hierbei, daß dadurch der Optimierungsgewinn um circa 2 % reduziert wurde. Der verbleibende Gewinn dürfte jedoch die heute noch relativ hohen Rechnerkosten gerechtfertigt erscheinen lassen. Die Abwicklung der Studien machte deutlich, daß bei der Komplexität und der Vielfalt der beteiligten Personen eine möglichst "objektive" und automatisierte Bearbeitung durch den Computer sinnvoll ist; gleichzeitig wird dadurch der Entwurfsingenieur von Routinearbeiten entlastet und für die wichtigen Aufgaben der Beurteilung und Verbesserung freigestellt. Deshalb ist es wichtig, daß die Programmsysteme so erstellt sind, daß der Bearbeiter Änderungen schnell und an jeder Stelle einbringen und damit die Auswirkung unmittelbar in anschaulicher Form am Bildschirm bewerten kann. Diese Idealvorstellung eines dialogfähigen Systems konnte bei der Gradientenoptimierung in nicht allen Arbeitsschritten verwirklicht werden. Ein wesentliches Hindernis hierfür ist der bei der heutigen Hardware-Konfiguration noch hohe Rechenzeitbedarf, der befriedigende Antwortzeiten selbst bei leistungsfähigen Großrechenanlagen kaum erwarten läßt. Man steht hierbei im Zielkonflikt zwischen Modellvereinfachung und schnellem Ergebnis einerseits sowie möglichst exaktem Modell mit sicherem Optimierungsergebnis aber hohem Rechenzeitaufwand andererseits. Es wird aber bei der rasanten Entwicklung auf dem Hardware Sektor in naher Zukunft bestimmt eine Verbesserung erzielbar sein, die ein interaktives Optimieren am Bildschirm ohne Verzicht auf strenge mathematische Modelle ermöglicht.

Literatur

Bellmann, R.E. (1957): Dynamic Programming, Princeton

Benutzerhandbuch TROP (1981): Trassenoptimierung von Landverkehrswegen, GeoMeDa, München

Boesefeld, I. (1970): Optimierung einer Gradiente im Straßenbau nach der Monte Carlo Methode, Graz

Caspary, W., Heister, H., Welsch, W. (1980): Ein interaktives Programmsystem zur Entwicklung einer optimalen Trasse auf der Grundlage topographischer Karten, Allgemeine Vermessungsnachrichten, 87, S. 178-191

Caspary, W., Heister, H., Welsch, W. (1982a): Bearbeitung des Vorentwurfs für ausgewählte Varianten mit dem Programmsystem TROP, Schriftenreihe des Wissenschaftlichen Studiengangs Vermessungswesen der Hochschule der Bundeswehr, Heft 6, S. 159 - 174

Caspary, W., Heister, H., Welsch, W. (1982b): Gedanken zum interaktiven rechnergestützten Entwerfen von Trassen für Verkehrssysteme, Schriftenreihe des Wissenschaftlichen Studiengangs Vermessungswesen der Hochschule der Bundeswehr, Heft 6, S. 80 - 101

Hintzen, H. (1965): Die Bestimmung der optimalen Gradiente von Strassen mit Hilfe elektronischer Rechenanlagen, Straßen- und Tiefbau, 6, 1965

Koch, R. (1975): Optimierung der Linienführung von Straßen im Grundriß, Straße und Autobahn, 26, S. 223 - 229

Manolopoulos, N. (1979): Trassenfindung und Optimierung mit dem Programmsystem OPT-1, CAD-Berichte, Nr. 126, S. 109 - 162

Miller, C.L., Laflamme, R.A. (1958): The Digital Terrain Model-Theory and Application, Photogrammetric Engineering, Vol. XXIV, S. 433 - 442

Scheck, H.J. (1975): Optimierungsberechnungen und Sensitivitätsanalysen als Hilfsmittel bei der Entwurfsbearbeitung von Straßen, Straßenbau und Straßenverkehrstechnik, Heft 153, S. 31 - 147

Schwintzer, P., Stöckl, R. (1982): Investitionskostenrechnungen für den Vorentwurf, Schriftenreihe des Wissenschaftlichen Studiengangs Vermessungswesen der Hochschule der Bundeswehr, Heft 6, S. 114 - 139