

# Über die Adaptation von allgemeinen, Koordinatensystem-unabhängigen, normalverteilten Mutationen in der Evolutionsstrategie: Die Erzeugendensystemadaptation

Nikolaus Hansen, Andreas Ostermeier & Andreas Gawelczyk\*

Januar 1995, letzte Änderung August 1995

## Zusammenfassung

Es wird ein neues Verfahren zur Adaptation von *beliebig* normalverteilten Mutationen in der  $(\mu, \lambda)$ -Evolutionsstrategie (ES) vorgestellt. Die Adaptation ist unabhängig vom gewählten Koordinatensystem und somit insbesondere unabhängig von beliebigen Drehungen der Zielfunktion. Es werden zuverlässig Mutationsverteilungen adaptiert, die Hyperellipsen mit hohen Achsverhältnissen in eine Hypersphäre transformieren. Das Verfahren wird in Simulationen mit zwei anderen Adaptationsmechanismen verglichen, die ebenfalls nicht-achsparallele Mutationsellipsoide erzeugen können.

## 1 Einführung

In der ES (siehe z.B. Rechenberg 1994) ist es bei kontinuierlichen Zielfunktionen üblich eine Mutation durch Addition eines  $(\vec{0}, \mathbf{A})$ -normalverteilten Zufallsvektors auf den Objektvariablenvektor zu erzeugen. Dabei repräsentiert die positiv semidefinite, symmetrische  $n \times n$ -Matrix  $\mathbf{A}$  die Verteilungsparameter (Varianzen und Kovarianzen). Um bei unbekannter Topologie der Zielfunktion einen zuverlässigen Fortschritt zu erzielen, muß  $\mathbf{A}$  im allgemeinen adaptiert werden. Die einfachste Form der Adaptation ist die Regelung nur einer globalen Schrittweite  $\delta$ . In diesem Fall ist  $\mathbf{A} = \delta^2 \mathbf{I}$ , wobei  $\mathbf{I}$  die Einheitsmatrix darstellt. Diese Verteilung ist isotrop. Werden anisotrope Verteilungen adaptiert, so kann zwischen speziellen Verteilungen, deren Isodichteflächen ausschließlich achsparallele Hyperellipsoide sind, und allgemeineren, deren Isodichteflächen beliebig orientierte Hyperellipsoide sein können, unterschieden werden. Bei ersteren sind die Mutationen in den Koordinatenachsen unkorreliert, und es werden nur die Achsverhältnisse verändert –  $\mathbf{A}$  ist eine Diagonalmatrix. Letztere können in den Koordinatenachsen korrelierte Mutationen erzeugen,  $\mathbf{A}$  ist nicht mehr zwangsläufig diagonal. Im folgenden werden drei Verfahren diskutiert, die Verteilungen mit in dem gegebenen Koordinatensystem korrelierten Mutationen adaptieren können. Die neu vorgestellte Erzeugendensystemadaptation erzeugt allgemeine Normalverteilungen und zeichnet sich dadurch aus, daß der Adaptationsprozeß völlig unabhängig vom gewählten Koordinatensystem ist.

Alle Verfahren arbeiten mit einem  $(\mu, \lambda)$ -Reproduktionsschema, d.h. mit *truncation selection* der  $\mu$  besten aus einer Population von  $\lambda$  Individuen.

## 2 Adaptationsschemata

Zunächst wird die neue Erzeugendensystemadaptation zur Adaptation einer beliebigen Normalverteilung vorgestellt. Danach wird ein weiteres Verfahren erläutert, das nicht *beliebige* Normalverteilungen erzeugen kann, aber auf einer Reihe von Zielfunktionen, die Mutationsverteilungen mit Korrelationen erfordern, gute Ergebnisse liefert. Die beiden ersten Algorithmen werden im Abschnitt 2.4 formal dargestellt. Diese Verfahren nennen wir *entstochastisiert*, weil nicht die Strategieparameter selbst einem Mutation-Selektions-Prozeß unterliegen, sondern die *gleiche Zufallsrealisation*, die auf die Objektparameter wirkt, auch die Strategieparameter verändert. Ein Mutations-Selektions-Prozeß auf Strategieparametererebene ist naturgemäß stark gestört, da das Selektionskriterium durch die Einstellung der Objektparameter bestimmt ist und die Strategieparameter nur in Wahrscheinlichkeit mit den Objektparameteränderungen korrespondieren. Ein Rekombinationsmechanismus wird für diese Verfahren nicht diskutiert.

Das dritte Verfahren wurde von Schwefel (1981) vorgeschlagen und kann beliebige Normalverteilungen erzeugen. Schwefel führte damit die Idee der Selbstadaptation *aller* Verteilungsparameter der zentrierten Normalverteilung ein.

Im Nachfolgenden werden wir die Adaptationsstrategien mit AI, AII bzw. AIII referenzieren.

---

\*Fachgebiet Bionik und Evolutionstechnik der TU Berlin.  
e-mail: {hansen, ostermeier, gawelczyk}@fb10.tu-berlin.de

## 2.1 Entstochastisierte Adaptation des Erzeugendensystems (AI)

In der folgenden Darstellung der Erzeugendensystemadaptation (AI) legen wir besonderen Wert auf das Verständnis des Adaptationsprozesses, weniger auf mathematische Strenge. Der Einfachheit halber, betrachten wir zunächst die Situation für  $\mu = 1$ .

Ein Mutationsschritt wird üblicherweise durch Addition der Realisierung eines normalverteilten Zufallsvektors  $\mathbf{z}'$  auf den Objektvariablenvektor des Elter ausgeführt. Für eine Strategie mit nur einer (globalen) Schrittweite wird der Zufallsvektor  $\mathbf{z}'$  nach dem folgenden Verfahren erzeugt:

$$\mathbf{z}' := \delta (z_1 \mathbf{b}_1 + \dots + z_n \mathbf{b}_n) \quad (1)$$

mit

- $n$  Zahl der Objektparameter (Dimension des Objektvariablenvektors),
- $\delta$  globale Schrittweite,
- $z_i$   $i = 1, \dots, n$ , unabhängige,  $(0, 1)$ -normalverteilte Zufallszahlen,
- $\mathbf{b}_i := \mathbf{e}_i$   $i$ -ter Einheitsvektor im  $\mathbb{R}^n$ ; die  $\mathbf{b}_i$  bilden eine Orthonormalbasis im Objektparameterraum.

$\mathbf{z}'$  ist  $(\vec{0}, \delta^2 \mathbf{I})$ -verteilt, wobei  $\mathbf{I}$  die Einheitsmatrix im  $\mathbb{R}^n$  ist. Ein achsparalleles Verteilungs-ellipsoid bekommt man durch Multiplikation jedes Summanden  $z_i \mathbf{b}_i$  in Gleichung (1) mit einer individuellen Schrittweite  $\sigma_i$ . Eine allgemeinere Modifikation der Verteilung läßt sich durch Austausch der  $\mathbf{b}_i$  erreichen. Dadurch löst man sich vom gegebenen Koordinatensystem und kann – durch geeignete Wahl der  $\mathbf{b}_i$  – jede zentrierte Normalverteilung erzeugen, denn zu jeder Normalverteilung existieren z.B. orthogonale  $\mathbf{b}_i$ , die diese Verteilung erzeugen. Andererseits erzeugen beliebig gewählte  $\mathbf{b}_i$  immer eine Normalverteilung, da einzelne (entartete) Normalverteilungen addiert werden.

### Der Adaptationsmechanismus

Die Koordinatensystem-unabhängige Adaptation wird durch sukzessiven Austausch der Vektoren  $\mathbf{b}_i$  erreicht. Gesucht ist jeweils ein neuer Vektor, der die Mutationsverteilung in geeigneter Weise modifiziert. Setzt man voraus, daß sich die „beste“ Mutationsverteilung in der Generationenfolge nur langsam ändert, liefert die Differenz der Objektvariablenvektoren von bestem Nachkommen und Elter, die wir als selektierten Mutationsschritt  $\mathbf{y}_{\text{sel}}$  bezeichnen wollen, die beste verfügbare Information über die optimale Mutationsverteilung. Wählt man den (geeignet normierten) selektierten Mutationsschritt  $\mathbf{y}_{\text{sel}}$  zum Austausch eines  $\mathbf{b}_i$ , werden im Folgenden Vektoren, die  $\mathbf{y}_{\text{sel}}$  ähnlich sind, mit größerer Wahrscheinlichkeit auftreten als bei der Wahl jedes anderen Vektors. Das auf diese Weise sukzessive erzeugte Vektorentupel hängt nur von der Selektion vorausgegangener Mutationen, d.h. von der Topologie der Zielfunktion, nicht jedoch vom Koordinatensystem ab.

Zur Realisierung dieses Adaptationsmechanismus werden die folgenden Punkte berücksichtigt:

- Es werden nicht  $n$ , sondern zwischen  $n^2$  und  $2n^2$  Vektoren  $\mathbf{b}_i$  verwendet. Die Vektoren bilden ein Gedächtnis der in der Generationenfolge selektierten Mutationsschritte, das sich für die Anpassung aller  $(n^2 + n)/2$  Verteilungsparameter als notwendig erweist. Daß sich ein Gedächtnis für die Adaptation der Strategieparameter als nützlich erweisen könnte, hatte schon Rudolph (1992) vermutet. Die benötigte Selektionsinformation kann prinzipiell auch innerhalb eines Generationsschrittes durch Vergrößerung der Population und insbesondere  $\mu$  gewonnen werden. Die Populationsgröße skaliert dann allerdings mit  $n^2$ .

Alle oben genannten Eigenschaften des Verfahrens bleiben von einer Vergrößerung der Vektorenzahl unbeeinflusst, insbesondere wird unabhängig von Vektorenzahl und –lage immer eine Normalverteilung erzeugt.

- Es wird jeweils der älteste Vektor  $\mathbf{b}_i$  überschrieben. Dadurch wird immer die aktuellste Selektionsinformation genutzt.
- Das Verfahren wird mit einer globalen Schrittweitenregelung kombiniert, weil dadurch die wichtige Adaptation der Größe des Mutationsschrittes schneller erfolgen kann. Die Schrittweitenregelung erfolgt mutativ mit Dämpfung der vererbten Schrittweitenänderung durch den Exponenten  $\beta$ .
- Der neue Vektor  $\mathbf{b}_i$  wird durch (exponentiell abklingend) gewichtete arithmetische Mittelung aus allen bisherigen Mutationsschritten berechnet. Durch diese Kumulation erhält das

Vektorentupel nicht-lokale<sup>1</sup> Selektionsinformation, wodurch zusätzlich die Orientierung<sup>2</sup> der Mutationsvektoren die Adaptation beeinflussen kann, denn die *Vektorsumme* ist nicht unabhängig von der Orientierung der summierten Vektoren.

Der zuerst genannte Punkt ist essentiell, da die Verteilungen bei zu kleiner Vektorenzahl degenerieren. Dämpfung der *globalen* Schrittweite und Kumulation der Selektionsinformation sind dagegen verzichtbar. Die Speicherkapazität für die  $\mathbf{b}_i$  kann von  $\mathcal{O}(n^3)$  auf  $\mathcal{O}(n^2)$  reduziert werden. Dazu werden die Kovarianzmatrizen der Verteilungen der  $z_i \mathbf{b}_i$  aufsummiert. Um die gewünschte Normalverteilung zu erzeugen, können z.B. Eigenwerte und -vektoren der resultierenden Matrix bestimmt werden. Eine Diskussion der daraus resultierenden Modifikationen des Verfahrens geht über den Rahmen dieses Reports hinaus.

## 2.2 Entstochastisierte Adaptation individueller Schrittweiten und einer Richtung (AII)

Die entstochastisierte Adaptation individueller Schrittweiten und einer Richtung (AII) ist im Kern kein neues Verfahren. Sie ist eine Erweiterung der entstochastisierten individuellen Schrittweitensteuerung (Ostermeier et al. 1994). Ein ganz ähnliches Verfahren wurde schon in Ostermeier et al. (1993) beschrieben.

AII adaptiert  $n$  individuellen Schrittweiten und eine Richtung. Die Adaptation von Schrittweiten- und Richtungsvektor erfolgt mit Hilfe von Information aus den realisierten Zufallsschritten nach dem Schema der entstochastisierten individuellen Schrittweitenadaptation. Die erzeugte Mutationsverteilung ist eine spezielle, vom Koordinatensystem abhängige Normalverteilung, die durch Addition eines achsparallelen Hyperellipsoids (individuelle Schrittweiten) und einer beliebig gedrehten normalverteilten Linienmutation (Richtungsvektor mal Zufallszahl) erzeugt wird. Insbesondere läßt sich *nicht jede* Normalverteilung erzeugen.

## 2.3 Adaptation von $n$ Standardabweichungen und $n(n - 1)/2$ Rotationswinkeln (AIII)

Eine  $n$ -dimensionale Normalverteilung kann durch  $n$  Varianzen  $\sigma_i^2$  und  $n(n - 1)/2$  Drehwinkel  $\omega_j$  eindeutig beschrieben werden: Die Matrix  $\mathbf{A}$  wird durch Anwendung von  $n(n - 1)/2$  geeigneten Drehmatrizen auf die Diagonalmatrix  $((\sigma_{ik})) := ((\delta_{ik} \sigma_i))$  eindeutig festgelegt.<sup>3</sup> Die Matrix  $((\sigma_{ik}))$  steht für ein paralleles Mutationsellipsoid, das durch die Drehmatrizen in jeder kanonischen Drehebene um einen festgelegten Winkel  $\omega_j \in [-\pi, \pi]$  gedreht wird. In einem von Schwefel (1981) vorgeschlagenen Verfahren wird durch Mutation der Standardabweichungen  $\sigma_i$  und der Drehwinkel  $\omega_j$  eine Adaptation der Mutationsverteilung vorgenommen:

$$\begin{aligned}\sigma_i^{(g+1)} &= \sigma_i^{(g)} z_i \\ \omega_j^{(g+1)} &= \left( \omega_j^{(g)} + z_j^\omega + \pi \right) \bmod 2\pi - \pi\end{aligned}$$

mit

- $g$  Generationszähler,
- $z$   $\left(0, (1/\sqrt{2n})^2\right)$  logarithmisch normalverteilt, nur eine Realisierung für alle  $\sigma_i$ ,
- $z_i$   $i = 1, \dots, n$ ;  $\left(0, (1/\sqrt{2\sqrt{n}})^2\right)$  logarithmisch normalverteilt,
- $z_j^\omega$   $j = 1, \dots, n(n - 1)/2$ ;  $\left(0, (\frac{5}{180} \pi)^2\right)$ -normalverteilt.

Mit diesem Adaptationsschema kann jede beliebige zentrierte Normalverteilung erzeugt werden.

## 2.4 Algorithmen zu AI und AII

Es folgt eine formale Darstellung der Algorithmen AI und AII aus 2.1 und 2.2 für  $\lambda > \mu \geq 1$ . Alle hier verwendeten Zufallszahlen sind voneinander unabhängig. Vektoren sind in Spaltenschreibweise

<sup>1</sup>i.e. in Bezug auf die Generationenfolge nicht-lokal.

<sup>2</sup>Gemeint ist hier die Orientierung des Vektors auf der Achse, in der der Vektor liegt („+“ oder „-“). Ohne den Mechanismus der Kumulation spielt die Orientierung des Vektors keine Rolle für die Adaptation. Denn bei der Erzeugung von  $\mathbf{z}'$  ist die Orientierung der  $\mathbf{b}_i$  irrelevant: Die Zufallszahl  $z_i$ , mit der der Vektor bei der Erzeugung eines Mutationssschrittes multipliziert wird, ist symmetrisch um null verteilt.

<sup>3</sup> $\delta_{ik}$  ist das Kronecker-Symbol. Ist  $\mathbf{B}$  das Resultat der Drehoperationen so gilt:  $\mathbf{A} = \mathbf{B}\mathbf{B}^t$ .

zu verstehen und immer fett gedruckt. Folgende Symbole werden in diesem Abschnitt mehrfach verwendet:

- $n$  die Zahl der Objektparameter (Dimension des Objektvariablenvektors),
- $\mathbf{I}$  Einheitsmatrix,
- $\mathbf{x} = (x_1, \dots, x_n)^t \in \mathbb{R}^n$  Objektvariablenvektor,
- $E_j$  Index für  $j$ -ten Elter,  $j = 1, \dots, \mu$ ,
- $N_k$  Index für den  $k$ -ten Nachkommen,  $k = 1, \dots, \lambda$ ,
- $\zeta_k = 1, \dots, \mu$  gleichwahrscheinlich, eine Realisation für jedes  $k = 1, \dots, \lambda$ ,
- $c_u = \sqrt{(2-c)/c}$  normiert die Varianz der linken Seite der jeweiligen Gleichung.

### Nachkommenerzeugung in AI

1. Erzeugen einer Realisation eines normalverteilten Zufallsvektors  $\mathbf{y}$ :

$$\mathbf{y}_k = \xi_k c_m \underbrace{\left( \mathbf{b}_1^{E_{\zeta_k}}, \dots, \mathbf{b}_m^{E_{\zeta_k}} \right)}_{n \times m\text{-Matrix}} \mathbf{z}_k \quad (2)$$

$$\parallel$$

$$\sum_{j=1}^m z_k^j \mathbf{b}_j^{E_{\zeta_k}}$$

2. Mutation der Strategie- und Objektparameter:

$$\begin{aligned} \mathbf{x}^{N_k} &= \mathbf{x}^{E_{\zeta_k}} + \delta^{E_{\zeta_k}} \mathbf{y}_k \\ \delta^{N_k} &= \delta^{E_{\zeta_k}} (\xi_k)^\beta \\ \mathbf{b}_{(g \bmod m)+1}^{N_k} &= (1-c) \mathbf{b}_{(g-1 \bmod m)+1}^{E_{\zeta_k}} + c (c_u \mathbf{y}_k) \\ \mathbf{b}_i^{N_k} &= \mathbf{b}_i^{E_{\zeta_k}} \quad \text{für } i = 1, \dots, m, \quad i \neq (g \bmod m) + 1 \end{aligned} \quad (3)$$

mit

- $\xi_k = 1,5; 1/1,5$  gleichwahrscheinlich, eine Realisation für jedes  $k = 1, \dots, \lambda$ ,
- $m \in \{n^2, \dots, 2n^2\}$  Anzahl der Vektoren des Erzeugendensystems. Je größer  $m$  ist, desto sicherer, je kleiner  $m$  ist, desto schneller ist die Adaptation.
- $c_m = (1/\sqrt{m})(1+1/m)$  bewirkt in Gleichung (2), daß die Länge von  $\mathbf{y}$  im Mittel etwa der Länge der  $\mathbf{b}_i$  entspricht. Dadurch bleiben die Vektorlängen der  $\mathbf{b}_i$  über die Generationenfolge ohne Selektion im Mittel annähernd konstant.
- $\mathbf{b}_1, \dots, \mathbf{b}_m \in \mathbb{R}^n$  Erzeugendensystem für die Zufallsvektoren. Die  $\mathbf{b}_i$  transformieren die im  $\mathbb{R}^m$  ( $\vec{0}, \mathbf{I}$ )-normalverteilte Zufallszahl  $\mathbf{z}$  in den  $\mathbb{R}^n$ . Initialisierung für  $g = 1$  mit  $\mathbf{b}_1^{E_j} = \vec{0}$  sowie  $\mathbf{b}_2^{E_j}, \dots, \mathbf{b}_m^{E_j}$  komponentenweise  $(0, 1/\sqrt{n})$ -normalverteilt.
- $\mathbf{z} \in \mathbb{R}^m$  in den Komponenten  $z^j$ ,  $j = 1, \dots, m$ ,  $(0, 1)$ -normalverteilte Zufallszahlen,
- $\delta$  globale Schrittweite,
- $g$  Generation, Start mit  $g = 1$ ,
- $c = 1/\sqrt{n}$  empirisch ermittelter Parameter für den Kumulationszeitraum. Für  $c = 1$  findet keine Kumulation statt.
- $\beta = 1/\sqrt{n}$  empirisch ermittelter Parameter zur Dämpfung der vererbten Schrittweitenänderung.

Der Faktor  $c_u$  normiert  $\mathbf{y}$ , so daß die Varianz des exponentiell gewichteten Summenvektors  $\mathbf{b}_{(g \bmod m)+1}^{N_k}$  der Varianz von  $\mathbf{y}$  entspricht (vgl. Gleichung (3)).

### Nachkommenerzeugung in AII

Die Darstellung des Adaptationsverfahrens wurde gegenüber der Darstellung der individuellen Schrittweitenregelung nach Ostermeier et al. (1994) modifiziert, der Teil für individuelle Schrittweitenadaptation ist jedoch inhaltlich praktisch identisch.

1. Objektvariablen

$$x_i^{N_k} = x_i^{E_{\zeta_k}} + \delta_i^{E_{\zeta_k}} z_i^k + \delta_r^{E_{\zeta_k}} z_r^k r_i^{E_{\zeta_k}} \quad \text{für } i = 1, \dots, n$$

## 2. Individuelle Schrittweiten

$$\begin{aligned}
\mathbf{s}^{N_k} &= (1-c) \mathbf{s}^{E_{\zeta_k}} + c (c_u \mathbf{z}^k) \\
\delta_i^{N_k} &= \delta_i^{E_{\zeta_k}} \underbrace{\exp\{\beta (\|\mathbf{s}^{N_k}\| - \widehat{\chi}_n)\}}_{\substack{\text{„globale“} \\ \text{Schrittweitenadaptation,} \\ \text{für alle } i \text{ identisch}}} \underbrace{\exp\{\beta_{\text{ind}} (|s_i^{N_k}| - \widehat{\chi}_1)\}}_{\substack{\text{individuelle} \\ \text{Schrittweitenadaptation}}} \\
&\text{für } i = 1, \dots, n
\end{aligned}$$

## 3. Richtung

$$\begin{aligned}
s_r^{N_k} &= \max\{(1-c) s_r^{E_{\zeta_k}} + c (c_u z_r^k); 0\} \\
\mathbf{r}^{N_k} &= \underbrace{\left( (1-c_r) \delta_r^{E_{\zeta_k}} \mathbf{r}^{E_{\zeta_k}} + c_r (\mathbf{x}^{N_k} - \mathbf{x}^{E_{\zeta_k}}) \right)}_{=: \mathbf{r}'} / \|\mathbf{r}'\| \\
\delta_r^{N_k} &= \max\left\{ \delta_r^{E_{\zeta_k}} \exp\left\{ \beta_r (|s_r^{N_k}| - \sqrt{2/\pi}) \right\}; \frac{1}{3} \|\delta^{N_k}\| \right\}
\end{aligned}$$

mit

- $\mathbf{z} = (z_1, \dots, z_n)^t$  in allen Komponenten  $(0, 1)$ -normalverteilte Zufallszahlen,
- $z_r$   $(0, 1)$ -normalverteilte Zufallszahl, eine Realisation für jedes  $k = 1, \dots, \lambda$ ,
- $\mathbf{s} \in \mathbb{R}^n$  gewichtete Summe der realisierten Zufallszahlen in der Generationenfolge (Kumulation). Die Summe  $\mathbf{s}$  wird zur Anpassung der individuellen Schrittweiten benötigt. Der Faktor  $c_u := \sqrt{(2-c)/c}$  normiert die Varianzen von  $\mathbf{s}$ . Initialisierung mit  $\vec{0}$ .
- $s_r$  gewichtete Summe der realisierten Zufallszahlen der Richtungskomponente, die zur Anpassung der Schrittweite der Richtungskomponente benötigt wird. Der Faktor  $c_u$  normiert die Varianz. Da eine Richtung adaptiert wird, sind Zahlen kleiner 0 nicht sinnvoll.
- $\delta = (\delta_1, \dots, \delta_n)^t \in \mathbb{R}^n$  Vektor der individuellen Schrittweiten,
- $\delta_r$  Schrittweite der Richtungskomponente. Sie wird durch die „Gesamtschrittweite“  $\|\delta\|$  nach unten begrenzt, weil die Selektionsrelevanz der Richtungsadaptation für  $\delta_r \ll \|\delta\|$  klein ist und der Adaptationsprozess dann in einen random walk übergeht.
- $\mathbf{r} \in \mathbb{R}^n$  Richtungsvektor, mit dem die Linienmutation erzeugt wird,
- $\widehat{\chi}_n = \sqrt{n} \left(1 - \frac{1}{4n} + \frac{1}{21n^2}\right)$  Näherung für den Erwartungswert der  $\chi_n$ -Verteilung,
- $\widehat{\chi}_1 = \sqrt{2/\pi}$  Erwartungswert der  $\chi_1$ -Verteilung,

$$\left. \begin{aligned}
c &= \sqrt{1/n} \\
c_r &= 3/n \\
\beta &= 2/n \\
\beta_{\text{ind}} &= 1/(4n) \\
\beta_r &= \sqrt{1/(4n)}
\end{aligned} \right\} \text{ empirisch ermittelte Werte. } c \text{ bestimmt den Kumulationszeitraum, } \beta$$

ist der Dämpfungsparameter für die Schrittweitanpassung. Für  $c = 1$  findet keine Kumulation, für  $\beta_{\text{ind}} = 0$  keine individuelle Schrittweitanpassung statt. Driftet  $\delta_r$  oder ein  $\delta_i$  im Lauf der Optimierung weg, so muß der entsprechende Dämpfungsparameter verkleinert werden. Schnellere Umadaptation erreicht man durch Vergrößerung von  $\beta$ .

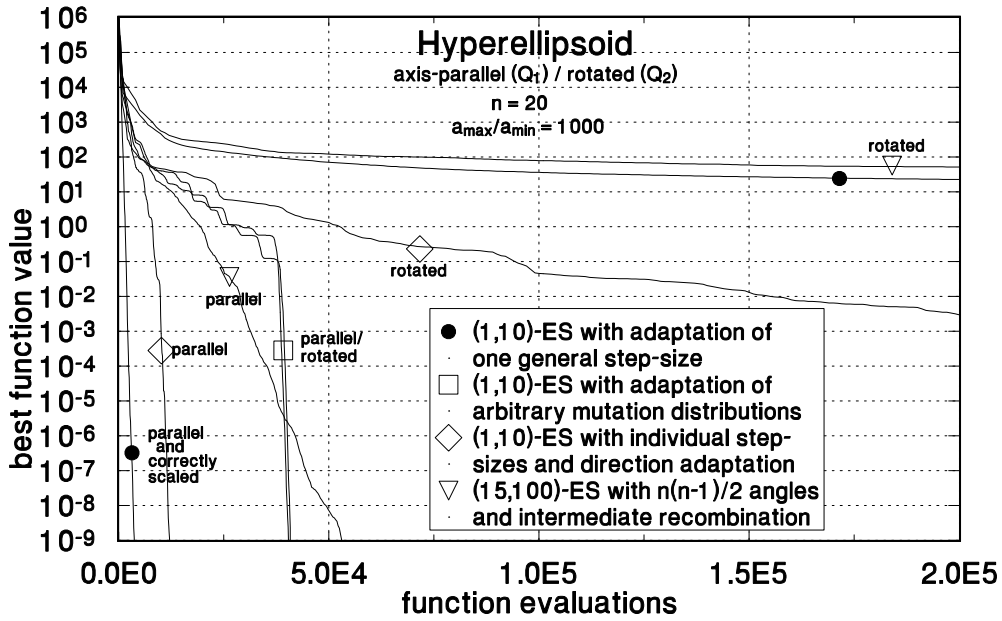
## 3 Zielfunktionen

Die Verfahren wurden an den folgenden Zielfunktionen mit  $n = 20$  getestet.

1. Hyperellipse mit konstantem Verhältnis (1,44 für  $n = 20$ ) zwischen „benachbarten“ Achsen:

$$Q_1(\mathbf{x}) = \sum_{i=1}^n \left( 1000 \frac{i-1}{n-1} \underbrace{\langle \mathbf{x}, \mathbf{e}_i \rangle}_{x_i} \right)^2, \quad Q_2(\mathbf{x}) = \sum_{i=1}^n \left( 1000 \frac{i-1}{n-1} \langle \mathbf{x}, \mathbf{o}_i \rangle \right)^2,$$

wobei  $\langle \cdot, \cdot \rangle$  das kanonische Skalarprodukt und die Vektoren  $\mathbf{o}_1, \dots, \mathbf{o}_n \in \mathbb{R}^n$  eine Orthonormalbasis mit zufälliger Orientierung im Raum sind.  $Q_1$  ist ein achsparalleles,  $Q_2$  ein



**Figure 1:** Simulation an der Hyperellipse.  $\square$  : AI,  $\diamond$  : AII,  $\nabla$  : AIII. Außer für die isotrope (1, 10)-ES ( $\bullet$ ) ist für jede Strategievariante je eine Simulation mit kanonischer ( $Q_1$ ) resp. zufälliger ( $Q_2$ ) Orthogonalbasis dargestellt. Zum Vergleich ist eine in allen Achsen identisch skalierte und eine korrekt skalierte (1, 10)-ES ( $\bullet$ ) am achsparallelen Hyperellipsoid dargestellt. Letztere entspricht einer Optimierung an der Hypersphäre und veranschaulicht – abgesehen von einer Regelung der globalen Schrittweite – den Fortschritt mit optimal adaptierter Mutationsverteilung. Startpunkt der Simulation war  $(1, \dots, 1)^t$ .

zufällig orientiertes Hyperellipsoid. Der  $i$ -te Basisvektor  $\mathbf{o}_i$  wird erzeugt, indem von einem Zufallsvektor mit normalverteilten Komponenten die Projektionen auf alle vorher erzeugten Vektoren abgezogen werden und das Resultat normiert wird. Mit diesem einfachen Verfahren kann jede Zielfunktion mit Definitionsbereich im  $\mathbb{R}^n$  in eine Koordinatensystem-unabhängige Lage gebracht werden.

## 2. Rosenbrock-Funktion

$$Q_3(\mathbf{x}) = \sum_{i=1}^{n-1} \left( (x_i - 1)^2 + 100 (x_i^2 - x_{i+1})^2 \right)$$

und

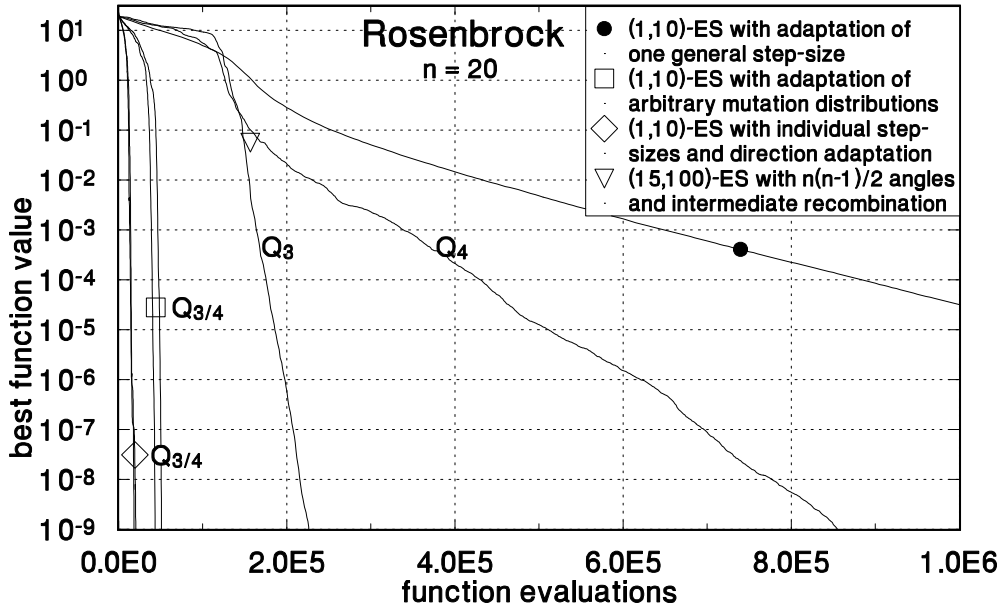
$$Q_4(\mathbf{x}) = \sum_{i=2}^n \left( (x_i - 1)^2 + 100 (x_i^2 - x_{i-1})^2 \right)$$

Die beiden Funktionen  $Q_3$  und  $Q_4$  unterscheiden sich nur durch die Umkehrung der Abzählrichtung der Variablen. Die Rosenbrock-Funktion erfordert in der Anfangsphase der Optimierung eine kontinuierliche Umadaptation der Verteilung.

## 4 Ergebnisse und Diskussion

Die beiden entstochastisierten Verfahren AI und AII wurden an einer (1, 10)-ES getestet. AIII benötigt aufgrund des Adaptationsmechanismus größere Populationen und wurde mit einer (15, 100)-ES und intermediärer Rekombination von jeweils zwei Eltern (i.e. arithmetische Mittelung der korrespondierenden Elternvariablen) auf Objekt- und Strategievariablen getestet. Die Ergebnisse wurden durch Änderung des Rekombinationstyps auf Strategie- und Objektvariablen (ohne/dominant) nicht verbessert.

Die Simulationsergebnisse an der Hyperellipse (Abb. 1) und der Rosenbrock-Funktion (Abb. 2) zeigen, daß AI ( $\square$ ) die Mutationsdichteverteilung zuverlässig an verschiedene Zielfunktionsstopologien anpaßt. Beliebige, auch gedrehte Hyperellipsen werden in die Hypersphäre transformiert.



**Figure 2:** Simulation an der Rosenbrock-Funktion.  $\square$  : AI,  $\diamond$  : AII,  $\nabla$  : AIII. Außer für die isotrope (1, 10)-ES ( $\bullet$ ) ist für jede Strategievariante je eine Simulation mit  $Q_3$  und  $Q_4$  dargestellt. Bei AII ( $\diamond$ ) sind die beiden Kurven praktisch identisch. Startpunkt der Simulation war  $\vec{0}$ .

Nach Abschluß der Adaptationsphase erreicht die Strategie etwa 80% der mit der optimalen, fest vorgegebenen Mutationsdichteverteilung möglichen Fortschrittsgeschwindigkeit. Insbesondere ist AI – in Übereinstimmung zu den theoretischen Überlegungen – unabhängig von Drehungen der Zielfunktion (vgl. Abb. 1) und Vertauschung von Koordinaten (vgl. Abb. 2).

Der Nachteil des Verfahrens AI liegt in der relativ langen Dauer des Adaptationsprozesses, der in Abb. 1 gut zu beobachten ist und hier etwa  $4 \cdot 10^4$  Funktionsaufrufe dauert. Sie skaliert wegen der Anzahl freier Parameter einer allgemeinen Normalverteilung mit  $n^2$ .

In der kürzeren Adaptationszeit liegt der Vorteil von AII ( $\diamond$ ). Die Mutationsdichteverteilung hat lediglich  $2n$  freie Parameter und die Adaptationsdauer skaliert mit  $n$ . Die Endgeschwindigkeit am achsparallelen Hyperellipsoid und an der Rosenbrock-Funktion entspricht der von AI. Allerdings können nur spezielle Mutationsverteilungen erzeugt werden. Insbesondere ist das Verfahren nicht unabhängig von Drehungen der Zielfunktion, und die Ergebnisse für eine zufällig orientierte Hyperellipse sind deutlich schlechter (vgl. Abb. 1).

AIII ( $\nabla$ ) ist in der Lage, beliebige Mutationsdichteverteilungen zu erzeugen, verhält sich jedoch sehr sensibel gegenüber Drehungen der Zielfunktion (vgl. Abb. 1) und ist zudem abhängig von Permutationen der Koordinatenachsen (vgl. Abb. 2). Um dieses Verhalten zu verifizieren, betrachten wir für  $k = 1, \dots, 10$  die Zielfunktionen

$$q_k = x_k^2 + \sum_{\substack{i=1 \\ i \neq k}}^{10} (100 x_i)^2.$$

Die Zielfunktionen sind topologisch identisch. Sie haben eine lange und neun identische kurze Achsen mit einem Verhältnis von 100:1. Die lange Achse liegt jeweils in Richtung der  $k$ -ten Achse des Koordinatensystems. Für  $q_{10}$  hat AIII eine fast 10 mal höhere Fortschrittsgeschwindigkeit als für  $q_5$ . Liegt die lange Achse einer solchen Zielfunktion in einer nicht achsparallelen Richtung im Raum (vgl. Abschnitt 3), ist der Fortschritt noch geringer als für  $q_5$ . Diese Beobachtungen stehen nicht mit dem gewählten Rekombinationstyp in Verbindung.

Wir interpretieren das Strategieverhalten als Folge des geringen Selektionsdruckes auf die Winkelstellungen, die dadurch einer großen stochastischen Komponente unterliegen. Sind die Winkelstellungen einem stochastischen Prozeß unterworfen, so beeinflußt die (festgelegte) Reihenfolge der Drehungen das Verhalten der Strategie ganz wesentlich:

In der Monte-Carlo-Simulation zeigt sich, daß bei zufälliger, auf  $[-\pi, \pi]$  gleichverteilter Wahl der Drehwinkel  $\omega_j$  ein achsparalleles Mutationsellipsoid mit einer langen Achse mit hoher Wahrscheinlichkeit wieder in achsparallele Position rotiert wird. Hierbei werden abhängig von der Reihenfolge,

Richtung	Wkt.
Achse mit höchster Wkt.	$1,5 \cdot 10^{-2}$
Achse mit niedrigster Wkt.	$2,5 \cdot 10^{-4}$
Mittel aus 20 Zufallsvektoren	$8,3 \cdot 10^{-6}$
Mittel aus 20 Hauptdiagonalen	$3,2 \cdot 10^{-7}$

**Table 1:** Zufällig gewählte Einheitsvektoren wurden durch die in Abschnitt 2.3 beschriebenen Drehmatrizen mit auf  $[-\pi; \pi]$  gleichverteilten Winkeln gedreht. Der Öffnungswinkel  $\alpha$  zwischen dem so erzeugten Vektor und einem Vektor in der angegebenen Richtung wurde gemessen. Angegeben ist die relative Häufigkeit für  $\cos(\alpha) > 0,8$  in  $5 \cdot 10^6$  Messungen ( $n = 20$ ).

in der die Drehungen ausgeführt werden, jeweils unterschiedliche Achsen bevorzugt. Die Tabelle 1 zeigt das Ergebnis einer solchen Simulation. Die auftretenden Häufigkeiten von Richtungen entsprechen qualitativ den unterschiedlichen Fortschrittsgeschwindigkeiten von AIII in entsprechenden Zielfunktionen und können die Abhängigkeit sowohl von Drehungen des Koordinatensystems als auch von Vertauschungen der Koordinatenachsen erklären. Diese Ergebnisse legen nahe, daß für die Reproduzierbarkeit von Simulationsergebnissen von AIII an nicht-isotropen Zielfunktionalen immer die Reihenfolge der Drehmatrizen zu berücksichtigen ist.

## Literatur

- Ostermeier, A., Gawelczyk, A., & Hansen, N. (1993). *A Derandomized Approach to Self Adaptation of Evolution Strategies*. Technical Report TR-93-003, available via anonymous ftp: ftp-bionik.fb10.tu-berlin.de in /pub/papers/Bionik.
- Ostermeier, A., Gawelczyk, A., & Hansen, N. (1994). Step-size Adaptation Based on Non-local Use of Selection Information. In Davidor, Y., Schwefel, H.-P. & Männer, R. (Eds.), *Parallel Problem Solving from Nature – PPSN III, Proceedings*: pp. 189-198. Berlin: Springer.
- Rechenberg, I. (1994). *Evolutionsstrategie '94*. Stuttgart: fromman-holzboog.
- Rudolph, G. (1992). On Correlated Mutations in Evolution Strategies. In Männer, R. & Manderick, B. (Eds.), *Parallel Problem Solving from Nature, 2, Proceedings*: pp. 105-114. Amsterdam: North-Holland.
- Schwefel, H.-P. (1981). *Numerical optimization of computer models*. Chichester: Wiley.