Anwendung evolutionärer Algorithmen

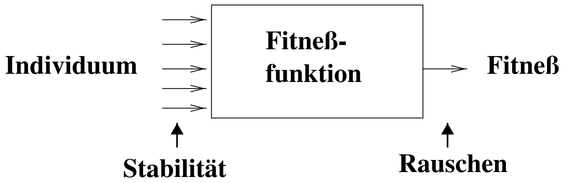
Vorlesung 13: Rauschen und stabile Lösungen

Karsten Weicker
Universität Stuttgart
Institut für Informatik, Abteilung Formale Konzepte
Breitwiesenstr. 20–22, 70565 Stuttgart
E-Mail: Karsten.Weicker@informatik.uni-stuttgart.de

Ursachen für Rauschen

- Experimente zur Fitneßauswertung
  - Meßfehler bei der Erfassung der Zielgrößen
  - ungenaue Einstellbarkeit
- stochastische Simulationen von Vorgängen
- Heranziehen von Stichproben zur Fitneßbestimmung
- Interaktion mit einem Anwender oder Beurteiler

Arten von Noise



Stabilität: die Lösung soll stabil sein gegenüber leichtem Rauschen in den Eingangsgrößen

Rauschen: der Fitneßwert kann nicht exakt ermittelt werden

Auswirkungen von Noise

- Es kann zu extremen Ausreißern (z.B. bei Meßfehlern) kommen – diese sollten erkannt und nicht zur Bewertung herangezogen werden.
- mangelnde Zuverlässigkeit
  - schlechte Lösungen können völlig überschätzt werden
  - gut Lösungen unterbewertet
- Veränderung der Fitneßlandschaft:
  - negativ: gute Individuen können nur schwer identifiziert werden, da ein Vergleich nicht repräsentativ ist
  - positiv: lokale Optima können durch Rauschen miteinander verschmolzen werden, so daß auch aus einem lokalen Optima noch Wege herausführen können.

Mögliche Ansätze zum Umgang mit Rauschen

- Individuen sollten immer wieder neu bewertet werden
- Individuen werden mehrfach bewertet und dann über die Ergebnisse gemittelt
  - Wieviele Bewertungen sind heranzuziehen?
  - Möglichkeiten die Kosten zu reduzieren?
  - Mathematische Berechnung der optimalen Anzahl an Samples
- veränderte Mutationen
  - Anpassung bei der Evolutionsstrategie
- Lernen während der Lebenszeit eines Individuums
- andere Vorgehensweise für stabile Lösungen?

Einfluß der Sampling Size

- Populationsgröße M
- Anzahl der Evaluation pro Kandidat: N
- Dauer der Auswertung einer Generation: T – für gewöhnlich gilt: $T = MN$
- Falls T vorgegeben ist: optimales M (und damit N) bestimmen
  $\Rightarrow$ Trade-off zwischen der Sample size (Anzahl der Evaluierungen) und der Populationsgröße
- Die Gesamtkosten C einer Generation ergeben sich aus
  - Kosten β pro Evaluation
  - evaluationsunabhängigen Kosten α pro Individuum

  als $C = (\alpha + \beta * N) * M$.
- Standardabweichung des Fitneßwertes
  - bei einer Auswertung: σ
  - bei N Auswertungen: $\frac{\sigma}{\sqrt{N}}$

Ergebnisse bei GA

- theoretische und empirische Untersuchung für feste vorgegebene Anzahl an Evaluationen pro Generation
- Es werden die Kosten α und β berücksichtigt; abhängig von $\frac{\alpha}{\beta}$ und der Populationsgröße ergeben sich unterschiedliche Anzahlen an Generationen
- für $\alpha = 0$: eine größere Population ist einer größeren Sample Size vorzuziehen ($M = 1000$, $N = 2$)
- für $\frac{\alpha}{\beta} > 1$: eine größere Sample size gewinnt an Bedeutung (für $\frac{\alpha}{\beta} = 3$: $M = 100$, $N = 20$; für $\frac{\alpha}{\beta} = 32$: $M = 50$, $N = 40$)

Ergebnisse bei ES

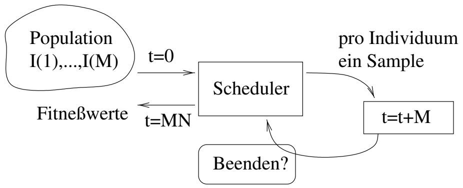
- theoretische Analysen für $(1, \lambda)$-Strategien, empirische Untersuchungen für (μ, λ)-Strategien
- mehrfache Bewertung reduziert den beobachtbaren Fehler – dies verbessert die Konvergenzpräzision und die Verlässlichkeit
- dies läßt sich nicht durch Vergrößerung der Kinderpopulation erreichen
- Vergrößerung der Elternpopulation verbessert die Konvergenzpräzision (Ausreißer fallen nicht so sehr ins Gewicht)
- Rekombination hat sich als nützlich erwiesen
- Ergebnisse stehen im Gegensatz zu den Ergebnissen bei GA

Vergleich mit lokaler Suche

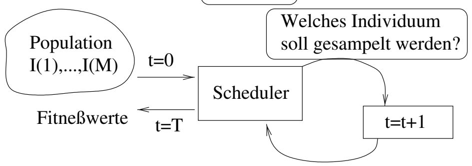
- Vergleich zwischen ES, GA und TA (und anderen punktbasierten Suchmethoden)
- die Sample size N wird variiert
- alle Verfahren profitieren von einer größeren Sample size – die punktbasierten allerdings mehr (da sich dort die Reduktion des beobachtbaren Fehlers direkter auswirkt)
- lokale Verfahren haben selbst mit moderatem Rauschen große Probleme
- GA und ES sind robust bei allen Rauschstärken
- GA und ES produzieren bei mittleren und starkem Rauschen bessere Ergebnisse mit $N = 1$ als lokale mit $N = 10$

Dynamisches Scheduling der Samples

- feste Populationsgröße
- es wird bestimmt, wieviele Samples pro Individuum erfaßt werden sollen



- feste Populationsgröße
- feste Anzahl der Auswertungen T
- die Aufteilung der Auswertungen auf die einzelnen Individuen wird bestimmt



- der Suchprozeß wird als statistischer Auswahlprozeß modelliert, woraus sich Formeln für die jeweiligen Entscheidungen ableiten
- bessere Ergebnisse als statische Verfahren

Methoden die Kosten zu verringern

- mehrfache Auswertungen (große Sample sizes) bringen große Berechnungskosten mit sich
- bei einer (μ, λ)-Strategie sind allerdings eigentlich nur die μ besten Individuen aus der Elternpopulation von Interesse $\Rightarrow$ nicht alle Individuen müssen gleich häufig ausgewertet werden
- Idee:
  - wenigen Auswertungen für alle Individuen
  - die Ordnung in der Population aufgrund von statistischen t-Tests bestimmen
  - dort, wo noch keine klare Aussage gemacht werden kann, sind mehr Tests notwendig
- Anzahl der Gesamtauswertungen wird signifikant reduziert

Mathematisch hergeleitete Sample Size

- es wird mathematisch die optimale Sample Size für GA hergeleitet
- Fitneßfunktion: OneMax
- Annahmen:
  - normalverteilte Fitneßwerte in der Population
  - Unabhängigkeit der Building Blocks
- Optimale Sample Size hängt von zwei Größen ab:
  - Verhältnis von α zu β
  - Verhältnis der Varianz des Rauschens zur Varianz der Fitneßwerte
- Untere Grenze für die optimale Sample Size: $coef\sqrt{\frac{\alpha}{\beta}}\sqrt{\frac{\sigma^2_{noise}}{\sigma^2_{fitness}}}$

Art der ES-Schrittweitenanpassung

- Verschiedene Regeln für die Anpassung der Schrittweiten
  Evolutionsstrategien: $\sigma_i' = \sigma_i e^{\tau' N(0,1) + \tau N_i(0,1)}$
  Evolutionäres Programmieren: $\sigma_i' = \sigma_i + \alpha \sigma_i N(0,1)$
- Regel der Evolutionsstrategien ist der zweiten Regel normalerweise überlegen (schnellere Anpassung)
- Mit zunehmendem Noise kann jedoch die zweite Regel der ersten überlegen sein.
- Erklärung: Die ES-Regel läßt sich leichter durch fehlerhafte Gradienteninformation in die Irre führen.

Kappa-Ka-Methode

- Modifikation der Schrittweiten-Anpassung
- Prinzip: groß variieren – klein vererben
- (Aufgrund von Rauschen) als gut befundene Veränderungen werden nur abgeschwächt weitergegeben
- Für die Bewertung wird das neue Individuum so berechnet:

$$\begin{aligned} \sigma' &= \sigma(e^{\tau N(0,1)})^{\kappa} \\ x_i' &= x_i + \sigma' k N(0,1) \end{aligned}$$

- Für die Übernahme in die nächste Generation so:

$$\begin{aligned} \sigma' &= \sigma e^{\tau N(0,1)} \\ x_i' &= x_i + \sigma' N(0,1) \end{aligned}$$

Lifetime Fitness Learning

- Bei Steady-State-Algorithmen werden Individuen nur einmal bewertet und nicht jede Generation
- Problem bei Noise: Bewertung sollte so exakt wie möglich sein
- Idee: Fortwährende zusätzliche Bewertung der Individuen
  - die Individuen werden bei Erzeugung mehrfach bewertet
  - spaeter werden immer wieder ein paar Individuen gezogen und zusätzlich bewertet
  - als Fitneß ergibt sich der Mittelwert über die letzten z.B. 20 Bewertungen
- Dies hat sich im Rahmen von koevolutionären Algorithmen bei Rauschen verursacht durch Auswahl von Stichproben bewährt.

Unterschiede bzgl. stabiler Lösungen

- Strategien zur Berechnung stabiler Lösungen
  - Mittelwert mehrerer Auswertungen, mit leicht variiertem Genotyp
    $\Rightarrow$ sehr zeitaufwendig
  - Mehrfachauswertung nur bei dem/den Besten Individuum/Individuen
  - Bewertung durch nur eine Auswertung an einem leicht variierten Genotyp
  - Heranziehen der bisher berechneten Individuen
    * Fitneß der Individuen aus der Nachbarschaft geht mit ein
    * alle Individuen der momentanen Population oder ein Speicher für die letzten z.B. 5000 Individuen
    * Neue Fitneß $f'(x) = \frac{\sum_y w(y) f(y)}{\sum_y w(y)}$ wobei w von der Entfernung bezüglich des Genotyps abhängt (z.B. $w(y) = \max\{0, 1 - \text{dist}(x,y)\}$)