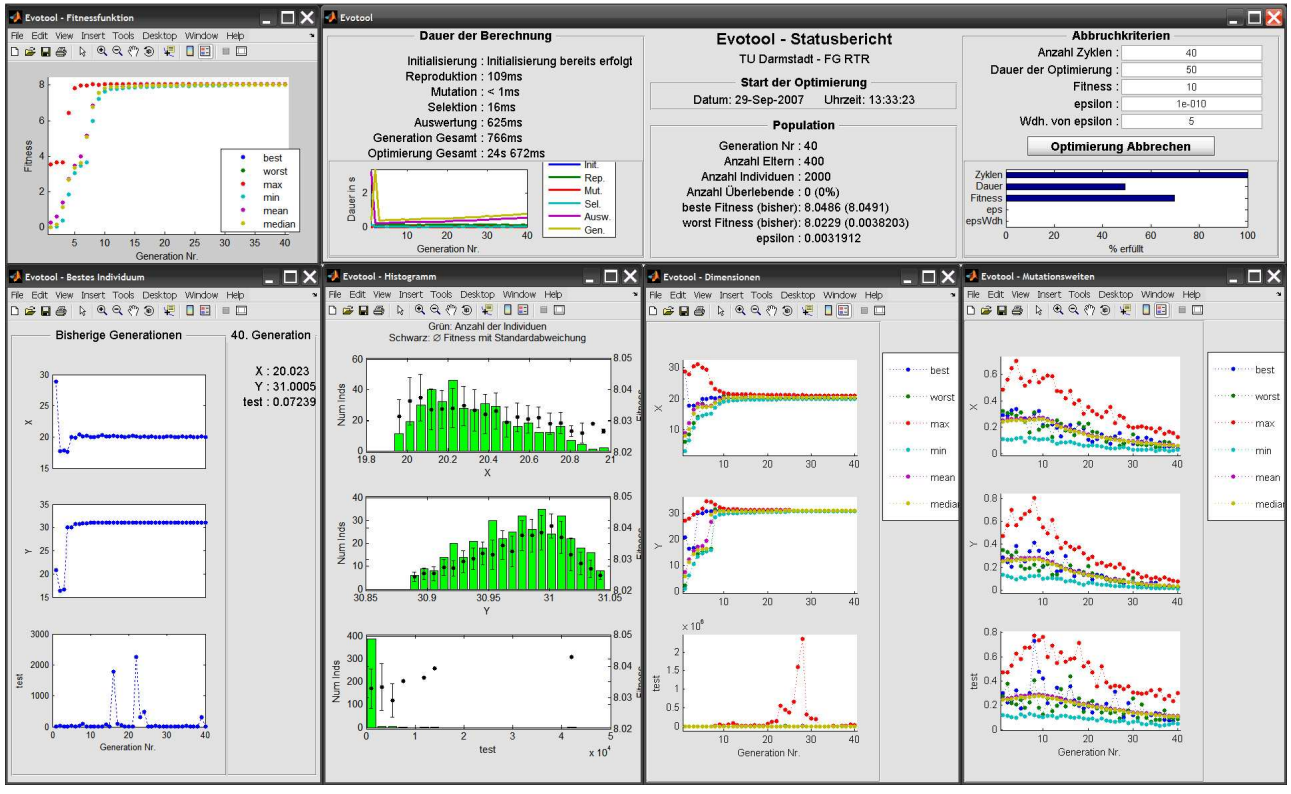
Evotool

Eine MATLAB-Toolbox für Evolutionsstrategien

Christian Voigt



TECHNISCHE UNIVERSITÄT DARMSTADT

September 2007

REGELUNGSTHEORIE UND ROBOTIK rtr

Agenda

Agenda

Agenda

Agenda

Agenda

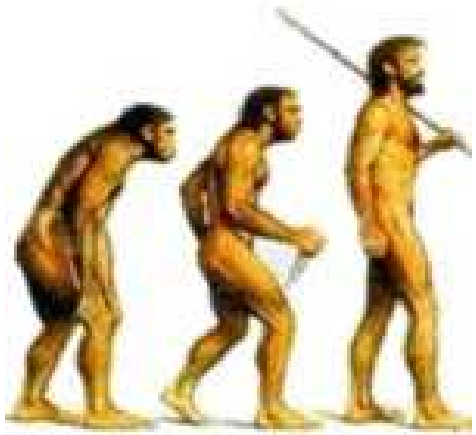
Was ist Evotool?

- Eine Toolbox für die Optimierung mittels evolutionärer Strategien.
- Umgesetzt als Klasse in MATLAB.
  - einfache Programmierung.
  - einfache Verwaltung der Optimierungsläufe als „World“'s.
  - Trennung von Optimierungsaufgabe und -methode.
- Optionale graphische Ausgabe:
  - Einfaches Statusfenster mit elementaren Eingriffsmöglichkeiten.
  - Statusfenster zur Entwicklung der Fitness.
  - Ausgabe des besten Individuums.
  - Ausgabe eines Histogramms über die einzelnen Dimensionen mit durchschnittlicher Fitness und Standardabweichung.
  - Statusfenster zur Entwicklung der Dimensionen.
  - Statusfenster zur Entwicklung der Mutationsweiten.
- Verwendung auf der Kommandozeile mit MATLAB ohne Java trotzdem möglich.

Evolutionäre Strategien

Was sind Evolutionäre Strategien?

- Algorithmen zur Optimierung
- Simulieren den Vorgang der Evolution
  - Reproduktion
  - Mutation
  - Selektion
- Keine genetische Algorithmen



Der zugrundeliegende Algorithmus

... kann dem Skript zur Vorlesung „Fuzzy-Logik“ von Prof. J.Adamy [1] entnommen werden.

Inbetriebnahme

Einmalig den Pfad setzen

1. Menü `Datei -> Pfad setzen` bzw. `File -> Set Path`
2. `Ordner hinzufügen` bzw. `Add Folder`. Den Ordner, indem sich der Ordner `@World` befindet, auswählen. Nicht `@World` selbst.
3. `Speichern` bzw. `Save` drücken, damit die Änderung nicht bei erneutem Starten von MATLAB erfolgen muss.

Für jeden Optimierungslauf

1. Ein Initialisierungsskript erstellen.
2. Eine passende Fitnessfunktion erstellen.
3. Den Optimierungslauf starten und beobachten.

1.Schritt: Eine `World` erstellen

`Worldname=World;`

Eine `World`-Variable mit dem Namen *`Worldname`* mit folgenden Eigenschaften erstellen:

- Die `World` ist unbevölkert.
- Es sind keine Dimensionen definiert.
- Die Reproduktionsmethode ist auf `'diskret+inter'` und ohne Aussterben der Eltern eingestellt.
- Die Mutationsweiten τ_0 und τ_1 sind gleich 0.15.
- Die Optimierungsaufgabe ist eine nichtstatische Minimierungsaufgabe und wird nach einer Generation abgebrochen.
- Es erfolgt lediglich eine einfache Statusausgabe auf der Kommandozeile und die Welt wird nicht nach jeder Generation gespeichert.

2.Schritt: Optimierung einstellen

`Worldname.optType='max'/'min';`

Falls nötig, auf eine Maximierungsaufgabe umstellen.

`Worldname.fitFun=@Fitnessfunktionsname;`

Den „Function-handle“ zur Fitnessfunktion mit folgenden Eigenschaften einstellen:

- Die Fitnessfunktion heißt `Fitnessfunktionsname.m`.
- Die Fitnessfunktion muss im gleichen Verzeichnis oder im Pfad liegen.
- Das einzige Eingangsargument ist die Welt `w` selbst.
- Das einzige Ausgabeargument ist ein Vektor mit den Fitnesswerten zu den Individuen.

`Worldname.staticFitFun=true/false;`

Ist die Fitnessfunktion statisch oder ändert sie sich nach jeder Generation?

3.Schritt: Algorithmus einstellen

Worldname.nachkommenProElter=(int>0);

Die Anzahl der Nachkommen pro Elter einstellen.

Worldname.elternAussterben=true/false;

Sollen die Eltern doch aussterben?

Worldname.mutTau0,_Worldname_.mutTau1=(double>0);

Die Mutationsweiten τ_0 und τ_1 auf andere Werte einstellen.

Worldname.rep='diskret+inter';

Die Reproduktionsmethode einstellen. Mögliche Einstellungen sind:

`'klonen'`	Reines Klonen der Nachkommen. Keine Rekombination.
`'inter'`	Sowohl die Eigenschaften als auch die Mutationsweiten der Individuen werden intermediär rekombiniert.
`'diskret'`	Sowohl die Eigenschaften als auch die Mutationsweiten der Individuen werden diskret rekombiniert.
`'diskret+inter'`	Die Eigenschaften werden diskret, die Mutationsweiten intermediär rekombiniert.

4.Schritt: Dimensionen einstellen

Die Dimensionen werden als MATLAB-`structure`-Vektor mit folgenden Feldern übergeben.

`name`	Der Name der jeweiligen Dimension
`startRange`	Ein Zeilenvektor mit 2 Elementen, die den Bereich der Anfangsverteilung angeben.
`startDistribution`	Anfangsverteilung und Mutation erfolgen entweder linear (`'lin'`) oder logarithmisch (`'log'`).
`numEltern`	Die Anzahl der Stützstellen bei der ersten Generation für diese Dimension.
`boundaryCheck`	Ein Zeilenvektor, der den erlaubten Bereich dieser Dimension angibt.
`start0`	Anfängliche Mutationsweite dieser Dimension.

Dimensionen einstellen - Ein Beispiel

Folgender Beispielcode initialisiert 2 Dimension `X` und `Y`, bei denen in der ersten Generation 20 Eltern logarithmisch im Intervall $[1, 30]$ verteilt sind (Es gibt daher $20 \cdot 20 = 400$ Eltern).
In den folgenden Generationen dürfen Individuen nur im Intervall $[0, 40]$ existieren. Die Anfänglichen Mutationsweiten sind jeweils 0.25:

Beispiel für die Initialisierung der Dimensionen

```
FieldNames={'name', 'startRange', 'startDistribution',...
            'numEltern', 'boundaryCheck', 'start0'};
c= {'X', [1 30], 'log', 20, [0 40], .25; ...
    'Y', [1 30], 'log', 20, [0 40], .25;};
Worldname.dim=cell2struct(c,FieldNames,2);
```

5. Schritt: Abbruchkriterien

bk.numCycles=(int>0);

Die Optimierung wird nach dieser Anzahl an Generationen abgebrochen.

bk.time=(double>0);

Die Optimierung wird nach soviel Sekunden abgebrochen.

bk.f=(double);

Die Optimierung wird abgebrochen, wenn diese oder eine bessere Fitness erreicht wurde.

bk.epsilon=(double>0);,bk.epsilonTimes=(int>0);

Die Optimierung wird abgebrochen, wenn dieses oder ein geringeres ϵ epsilonTimes-mal erreicht wurde.

***Worldname*.break=bk;**

Wichtig ist die Übergabe der Abbruchkriterien-Structure an die Welt. Trifft eines der Abbruchkriterien zu, wird die Optimierung abgebrochen.

Abbruchkriterien einstellen - Ein Beispiel

Folgendes Beispiel initialisiert die Abbruchkriterien so, dass die Optimierung abgebrochen wird, wenn entweder:

- 100 Zyklen durchlaufen wurden,
- eine Fitness von 10 oder besser erreicht wurde, oder
- 5 mal hintereinander ein $\epsilon < 0.01$ auftritt.
- Die Optimierung wird nie aufgrund der Dauer abgebrochen!

Beispiel für die Initialisierung der Abbruchkriterien

```
...
bk.numCycles     =  100;
bk.time          =  inf;
bk.f             =   10;
bk.epsilon       = 0.01;
bk.epsilonTimes  =    5;

Worldname.break=bk;
...
```

6. Schritt: Ausgabe konfigurieren

Worldname.saveOnCycle='*Dateiname*';

String, der angibt, wie die mat-Datei heißt, in der, nach der erfolgreichen Berechnung, die Generation gespeichert wird. Ist er leer, werden die Generationen nicht nach jedem Zyklus abgespeichert.

Worldname.reportOnCycle=true/false;

Dieser Schalter schaltet die Berichterstattung, ob auf der Kommandozeile oder graphisch, während der Berechnung an und aus.

Beispielausgabe auf der Kommandozeile

```
============================================================
                  Zyklus nr: 11
     Zeit (Zyklus / Gesamt): 0.14 / 2.031
      Fitness best / worst : 8.0432 / 7.8362
============================================================
Dauer der Reproduktion      : 0.094 s
Dauer der Mutation          : 0.015 s
Dauer der Selektion         : 0.016 s
```

7.Schritt: Graphische Ausgabe konfigurieren

Die graphische Ausgabe wird über das Feld *Worldname*.Fig gesteuert. Für die Initialisierung muss zunächst eine temporäre Structure Fig mit folgenden Feldern konfiguriert werden:

Fig.on = true/false;

Dieser Schalter schaltet die graphische Berichterstattung an oder aus. Ist sie ausgeschaltet, werden alle anderen Felder von Fig ignoriert und eventuelle Statusberichte erfolgen nur auf der Kommandozeile.

Fig.histogramm,Fig.BestInd = true/false;

Diese Schalter schalten die Fenster mit der Anzeige der Eigenschaften des besten Individuums und dem Histogramm der Dimensionen ein und aus.

7.Schritt: Graphische Ausgabe konfigurieren (2)

`Fig.fitness,Fig.Inds,Fig.Mut=(cellstr)`

`Fig.fitness` Steuert das Fenster mit der Anzeige der Fitness.

`Fig.Inds` Steuert das Fenster mit der Anzeige der Dimensionen.

`Fig.Mut` Steuert das Fenster mit der Anzeige der Mutationsweiten.

Sind die jeweiligen Felder leere Cell-Arrays, bleiben die zugehörigen Fenster geschlossen. Die sonst im Cell-Array enthaltenen Strings steuern die Ausgabe. Gültige Werte sind:

`best` Ausgabe der jeweiligen Eigenschaft des besten Individuums.

`worst` Ausgabe der jeweiligen Eigenschaft des schlechtesten überlebenden Individuums.

`func` Eine beliebige MATLAB-Funktion `func`, die auf einen Vektor als Eingangsargument einen Skalar als Ausgangsargument liefert.

`Worldname``.Fig = Fig`

Wichtig: Übergabe der Einstellungen an die `World` *`Worldname`*.

Graphische Ausgabe einstellen - Ein Beispiel

Das auf der folgenden Folie gezeigte Beispiel konfiguriert die graphische Ausgabe der Welt *Worldname* so, dass...

- die graphische Ausgabe eingeschaltet ist.
- das Histogramm und die Anzeige des besten Individuums eingeschaltet sind.
- die Fitness des besten und des schlechtesten Individuums angezeigt werden.
- die Anzeige der Dimensionen ausgeschaltet ist.
- die durchschnittliche Mutationsweite einschließlich deren Standardabweichung geplottet werden.

Man beachte...

Die Darstellung der Standardabweichung der durchschnittlichen Mutationsweite als Gebiet um den Mittelwert herum erfordert die Erstellung zweier einfacher Funktionen die rechts dargestellt sind. Die Funktionen müssen im gleichen Verzeichnis oder Pfad liegen.
Mit dieser Methode lassen sich speziell an die Bedürfnisse des Nutzers angepasste Plots erstellen.

Graphische Ausgabe einstellen - Ein Beispiel(2)

Beispiel für die Initialisierung der Abbruchkriterien

```
...
Fig.on          = true;
Fig.histogramm  = true;
Fig.BestInd     = true;
Fig.Inds        = {};
Fig.fitness     = {'best',...
                   'worst'};
Fig.Mut         = {'meanpstd',...
                   'mean',     ...
                   'meanmstd'};

Worldname.Fig = Fig;
...
```

meanpstd.m

```
function out = meanpstd(v)
out = mean(v) + std(v);
```

meanmstd.m

```
function out = meanmstd(v)
out = mean(v) - std(v);
```

8.Schritt: Optimierung starten

`[Worldname_Finish breakKrit] = start(Worldname);`

Mit dieser Codezeile wird die Optimierung gestartet. Es sollte der letzte Befehl des Initialisierungsskriptes sein.
Die Funktion `start` liefert 2 Rückgabewerte:

- die Welt der letzten Generation und
- eine Structure mit dem Zustand aller Abbruchkriterien.

Aufbau der Fitnessfunktion

Funktionsname und -ort

- Die Fitnessfunktion heißt *Fitnessfunktionsname*.m.
- Die Fitnessfunktion muss im gleichen Verzeichnis wie das Initialisierungsskript oder im Pfad liegen.

Ein- und Ausgangsargument

Für die Fitnessfunktion muss gelten:

- Das einzige Eingangsargument ist die Welt `w` selbst.
- Das einzige Ausgabeargument ist ein Vektor mit den Fitnesswerten zu den Individuen.

Beispiel:

```
function f=Fitnessfunktionsname(w)
```

Zugriff auf die zu testenden Individuen

Der Algorithmus übergibt automatisch nur die noch nicht getesteten Individuen:

structname = w.ind4FitFun;

structname enthält die zu testenden Individuen. Jedes Array-Element ist ein Individuum. Dessen Fitness wird im zugehörigen Wert im Ausgangsvektor f zurückgegeben.

Einfaches Beispiel

```
...
ind = w.ind4FitFun;
X   = ind.X;
Y   = ind.Y;
f = functionname(X,Y);
...
```

Berechnung der Fitness in Schleifen.

Falls möglich, ist die vektorielle Berechnung eleganter und schneller und sollte in MATLAB prinzipiell der Berechnung in Schleifen vorgezogen werden.

Die Berechnung in Schleifen ist jedoch trotzdem möglich:

Beispiel für die Berechnung der Fitness in Schleifen

```
...
ind = w.ind4FitFun;
for i = 1:numel(ind)
  x    = ind(i).X;
  y    = ind(i).Y;
  f(i) = functionname(x,y);
end
...
```

Abbruch während der Berechnung

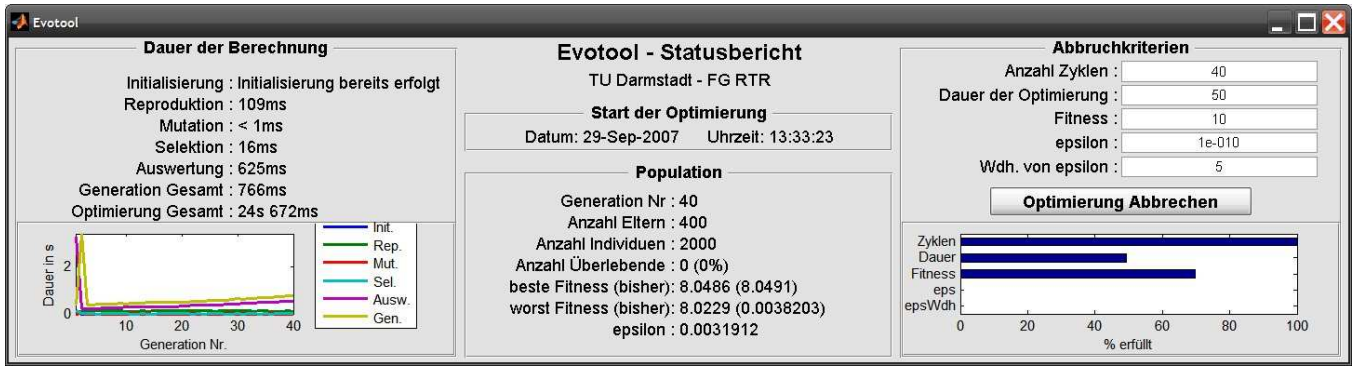
Falls die Fitnessfunktion über eine Eingriffsmöglichkeit verfügt, und hiermit ein manueller Abbruch erfolgen soll, so kann dies dem Algorithmus mitgeteilt werden, indem ihm als Fitness eine leere Menge zurückgegeben wird.
Dann wird die aktuelle Generation verworfen und die letzte vollständig berechnete `World` zurückgegeben.

Beispiel unter Verwendung von `GenWaitbar.m`

```
...
a = w.generation;
GenWaitbar(0,(a.nr+1));
ind = w.ind4FitFun;
for i = 1:numel(ind)
      ...
      if GenWaitbar(i/numel(ind));
          f=[];
          break
      end
 end
...
```

Das Statusfenster

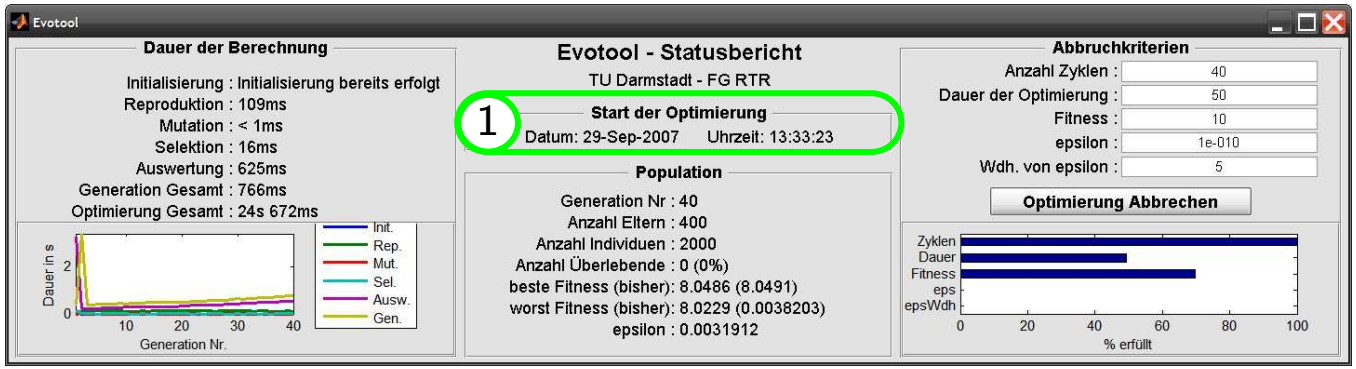
Das Statusfenster wird bei eingeschalteter graphischer Ausgabe immer angezeigt. Es liefert einen grundlegenden Feedback zum Ablauf der laufenden Optimierung und Interaktionsmöglichkeiten.



1. Beginn der Simulation
2. Grundlegenden Angaben zur Population.
3. Angaben zur Dauer der einzelnen Phasen der Optimierung der letzten Generation.
4. Graph mit Angabe der Dauer der Phasen der vergangenen Generationen.
5. Aktuelle Abbruchkriterien mit Interaktionsmöglichkeiten.
6. Statusbalken die den Erfüllungsgrad der einzelnen Abbruchkriterien angeben.

Das Statusfenster

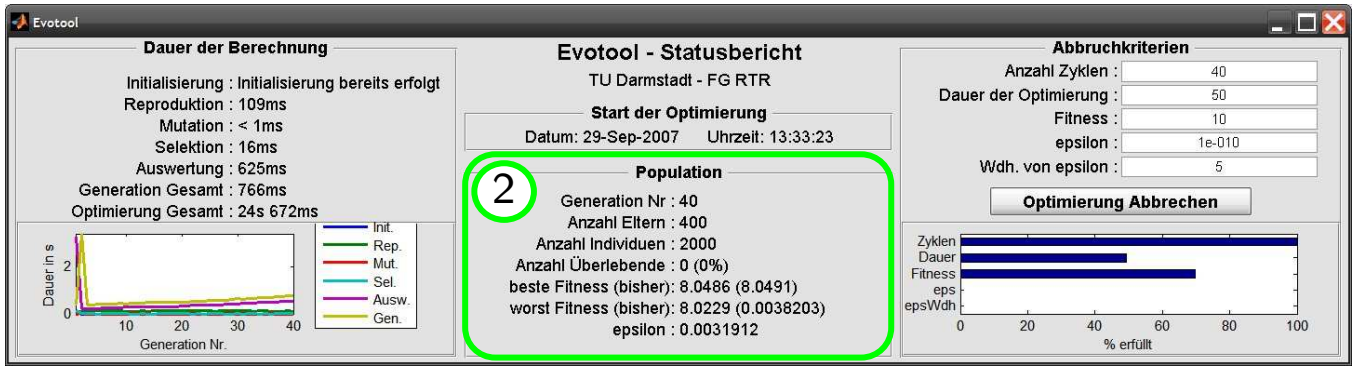
Das Statusfenster wird bei eingeschalteter graphischer Ausgabe immer angezeigt. Es liefert einen grundlegenden Feedback zum Ablauf der laufenden Optimierung und Interaktionsmöglichkeiten.



1. Beginn der Simulation
2. Grundlegenden Angaben zur Population.
3. Angaben zur Dauer der einzelnen Phasen der Optimierung der letzten Generation.
4. Graph mit Angabe der Dauer der Phasen der vergangenen Generationen.
5. Aktuelle Abbruchkriterien mit Interaktionsmöglichkeiten.
6. Statusbalken die den Erfüllungsgrad der einzelnen Abbruchkriterien angeben.

Das Statusfenster

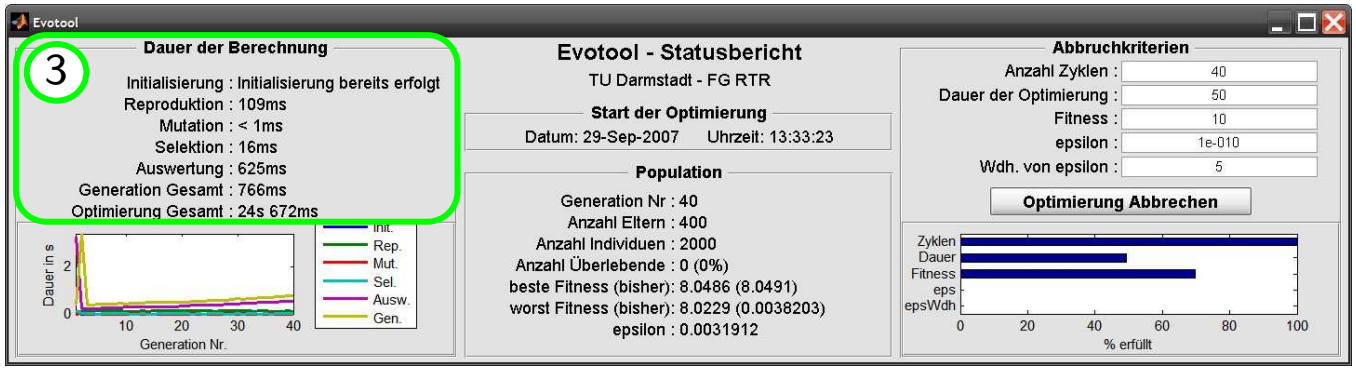
Das Statusfenster wird bei eingeschalteter graphischer Ausgabe immer angezeigt. Es liefert einen grundlegenden Feedback zum Ablauf der laufenden Optimierung und Interaktionsmöglichkeiten.



1. Beginn der Simulation
2. Grundlegenden Angaben zur Population.
3. Angaben zur Dauer der einzelnen Phasen der Optimierung der letzten Generation.
4. Graph mit Angabe der Dauer der Phasen der vergangenen Generationen.
5. Aktuelle Abbruchkriterien mit Interaktionsmöglichkeiten.
6. Statusbalken die den Erfüllungsgrad der einzelnen Abbruchkriterien angeben.

Das Statusfenster

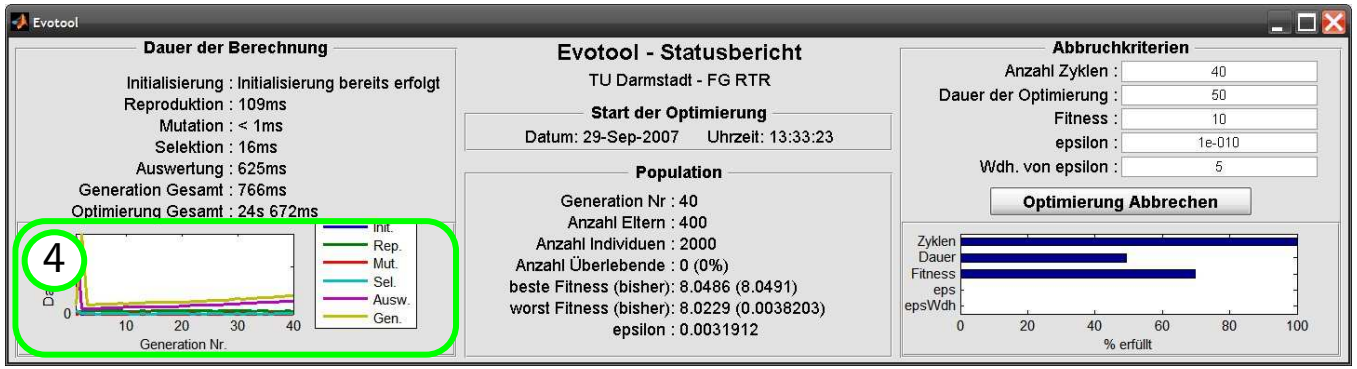
Das Statusfenster wird bei eingeschalteter graphischer Ausgabe immer angezeigt. Es liefert einen grundlegenden Feedback zum Ablauf der laufenden Optimierung und Interaktionsmöglichkeiten.



1. Beginn der Simulation
2. Grundlegenden Angaben zur Population.
3. Angaben zur Dauer der einzelnen Phasen der Optimierung der letzten Generation.
4. Graph mit Angabe der Dauer der Phasen der vergangenen Generationen.
5. Aktuelle Abbruchkriterien mit Interaktionsmöglichkeiten.
6. Statusbalken die den Erfüllungsgrad der einzelnen Abbruchkriterien angeben.

Das Statusfenster

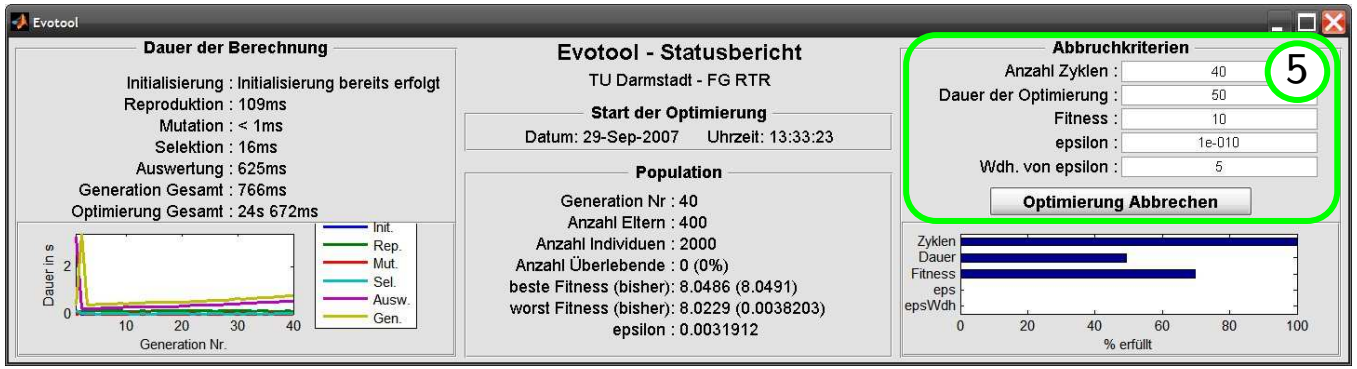
Das Statusfenster wird bei eingeschalteter graphischer Ausgabe immer angezeigt. Es liefert einen grundlegenden Feedback zum Ablauf der laufenden Optimierung und Interaktionsmöglichkeiten.



1. Beginn der Simulation
2. Grundlegenden Angaben zur Population.
3. Angaben zur Dauer der einzelnen Phasen der Optimierung der letzten Generation.
4. Graph mit Angabe der Dauer der Phasen der vergangenen Generationen.
5. Aktuelle Abbruchkriterien mit Interaktionsmöglichkeiten.
6. Statusbalken die den Erfüllungsgrad der einzelnen Abbruchkriterien angeben.

Das Statusfenster

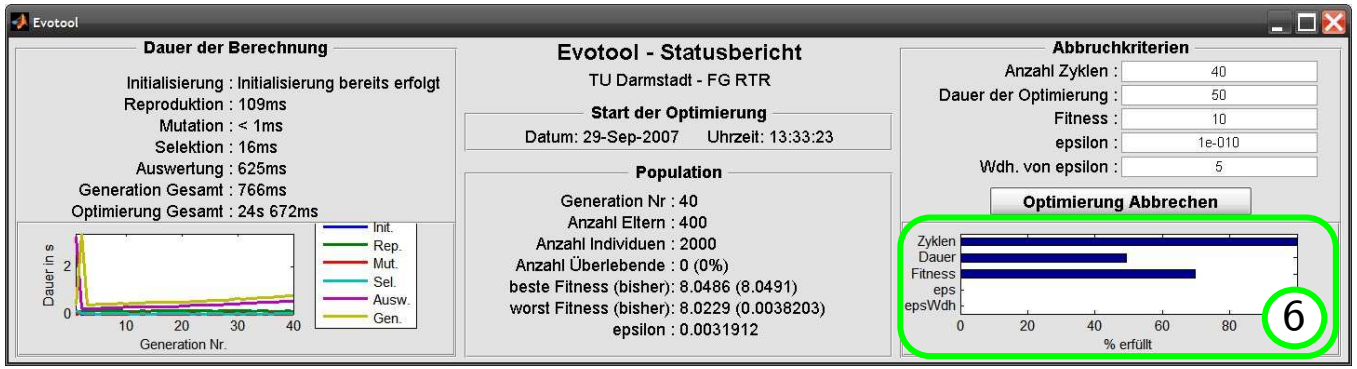
Das Statusfenster wird bei eingeschalteter graphischer Ausgabe immer angezeigt. Es liefert einen grundlegenden Feedback zum Ablauf der laufenden Optimierung und Interaktionsmöglichkeiten.



1. Beginn der Simulation
2. Grundlegenden Angaben zur Population.
3. Angaben zur Dauer der einzelnen Phasen der Optimierung der letzten Generation.
4. Graph mit Angabe der Dauer der Phasen der vergangenen Generationen.
5. Aktuelle Abbruchkriterien mit Interaktionsmöglichkeiten.
6. Statusbalken die den Erfüllungsgrad der einzelnen Abbruchkriterien angeben.

Das Statusfenster

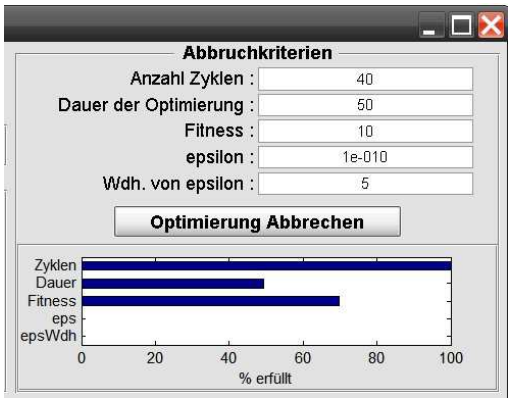
Das Statusfenster wird bei eingeschalteter graphischer Ausgabe immer angezeigt. Es liefert einen grundlegenden Feedback zum Ablauf der laufenden Optimierung und Interaktionsmöglichkeiten.



1. Beginn der Simulation
2. Grundlegenden Angaben zur Population.
3. Angaben zur Dauer der einzelnen Phasen der Optimierung der letzten Generation.
4. Graph mit Angabe der Dauer der Phasen der vergangenen Generationen.
5. Aktuelle Abbruchkriterien mit Interaktionsmöglichkeiten.
6. Statusbalken die den Erfüllungsgrad der einzelnen Abbruchkriterien angeben.

Interaktion über die GUI

Das Statusfenster erlaubt über die GUI-Elemente im unten abgebildeten rechten Panel eine einfache Interaktion mit der Optimierung während der Berechnung:



1. Alle Abbruchkriterien lassen sich während der Simulation beeinflussen. Dafür muss nur der Wert im Statusbericht gegen einen anderen Wert ausgetauscht werden. Wird ein ungültiger Wert eingegeben, wird die Eingabe verworfen und die ursprünglichen Werte werden wieder hergestellt.
2. Wird der Button „Optimierung abbrechen“ gedrückt, wird, nach Rückfrage, die Optimierung im Anschluss an die Berechnung der aktuellen Generation abgebrochen. Sie kann dann über denselben Button fortgesetzt werden.

Fenster schließen

Das Schließen des Fensters schließt alle anderen Statusfenster der aktuellen Optimierung ebenfalls. Es erfolgt keine weitere graphische Ausgabe, nur der Kommandozeilenbericht. Die Optimierung wird jedoch fortgesetzt.

Interaktion über die GUI

Das Statusfenster erlaubt über die GUI-Elemente im unten abgebildeten rechten Panel eine einfache Interaktion mit der Optimierung während der Berechnung:

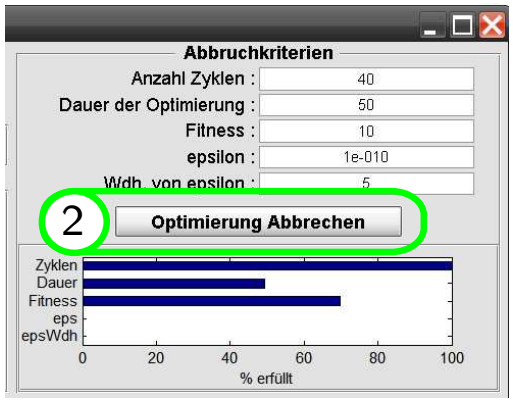
1. Alle Abbruchkriterien lassen sich während der Simulation beeinflussen. Dafür muss nur der Wert im Statusbericht gegen einen anderen Wert ausgetauscht werden. Wird ein ungültiger Wert eingegeben, wird die Eingabe verworfen und die ursprünglichen Werte werden wieder hergestellt.
2. Wird der Button „Optimierung abbrechen“ gedrückt, wird, nach Rückfrage, die Optimierung im Anschluss an die Berechnung der aktuellen Generation abgebrochen. Sie kann dann über denselben Button fortgesetzt werden.

Fenster schließen

Das Schließen des Fensters schließt alle anderen Statusfenster der aktuellen Optimierung ebenfalls. Es erfolgt keine weitere graphische Ausgabe, nur der Kommandozeilenbericht. Die Optimierung wird jedoch fortgesetzt.

Interaktion über die GUI

Das Statusfenster erlaubt über die GUI-Elemente im unten abgebildeten rechten Panel eine einfache Interaktion mit der Optimierung während der Berechnung:



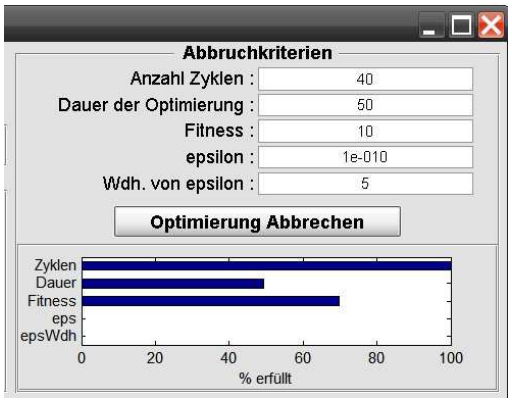
1. Alle Abbruchkriterien lassen sich während der Simulation beeinflussen. Dafür muss nur der Wert im Statusbericht gegen einen anderen Wert ausgetauscht werden. Wird ein ungültiger Wert eingegeben, wird die Eingabe verworfen und die ursprünglichen Werte werden wieder hergestellt.
2. Wird der Button „Optimierung abbrechen“ gedrückt, wird, nach Rückfrage, die Optimierung im Anschluss an die Berechnung der aktuellen Generation abgebrochen. Sie kann dann über denselben Button fortgesetzt werden.

Fenster schließen

Das Schließen des Fensters schließt alle anderen Statusfenster der aktuellen Optimierung ebenfalls. Es erfolgt keine weitere graphische Ausgabe, nur der Kommandozeilenbericht. Die Optimierung wird jedoch fortgesetzt.

Interaktion über die GUI

Das Statusfenster erlaubt über die GUI-Elemente im unten abgebildeten rechten Panel eine einfache Interaktion mit der Optimierung während der Berechnung:



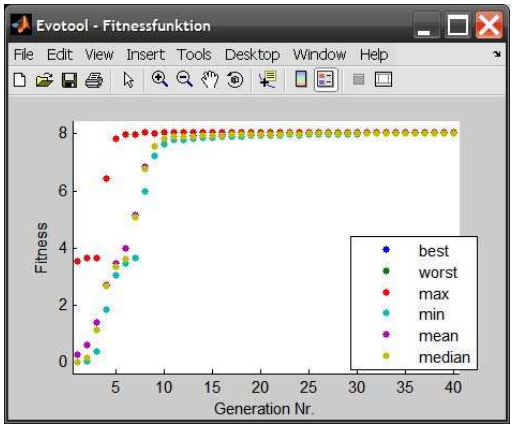
1. Alle Abbruchkriterien lassen sich während der Simulation beeinflussen. Dafür muss nur der Wert im Statusbericht gegen einen anderen Wert ausgetauscht werden. Wird ein ungültiger Wert eingegeben, wird die Eingabe verworfen und die ursprünglichen Werte werden wieder hergestellt.
2. Wird der Button „Optimierung abbrechen“ gedrückt, wird, nach Rückfrage, die Optimierung im Anschluss an die Berechnung der aktuellen Generation abgebrochen. Sie kann dann über denselben Button fortgesetzt werden.

Fenster schließen

Das Schließen des Fensters schließt alle anderen Statusfenster der aktuellen Optimierung ebenfalls. Es erfolgt keine weitere graphische Ausgabe, nur der Kommandozeilenbericht. Die Optimierung wird jedoch fortgesetzt.

Fenster: Fitnesswerte

Dieses Fenster stellt die Entwicklung der Fitnesswerte dar. Welche Werte der jeweiligen Genertation abgebildet werden ist von dem Inhalt des Cellstrings `w.Fig.fitness` abhängig. Zulässige Strings sind:



`'best'` Die Fitness des besten Individuums.

`'worst'` Die Fitness des schlechtesten überlebenden Individuums.

`'func'` Eine beliebige MATLAB-Funktion `func`, die einen skalaren Wert aus dem Vektor der Fitnesswerte berechnet. Zum Beispiel:

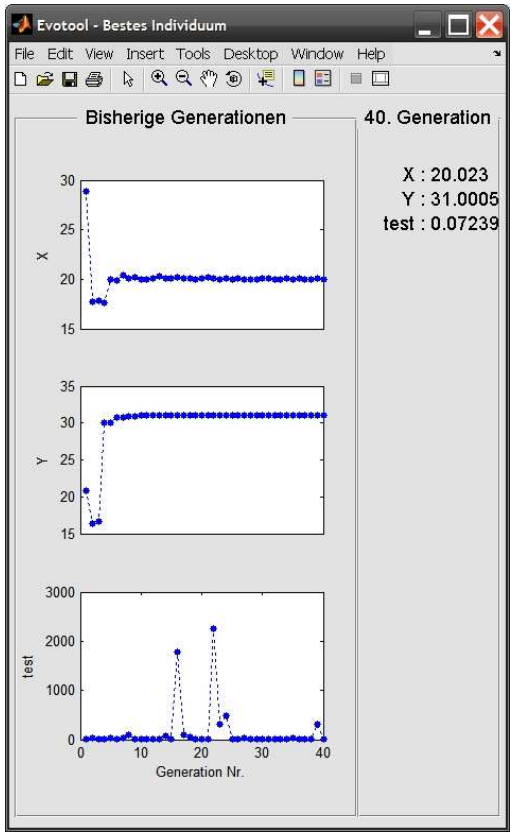
- `'mean'`
- `'median'`
- `'std'`

Fenster schließen

Wird das Fenster geschlossen, hat man die Wahl:

- Das Fenster kann dauerhaft geschlossen werden.
- Wahlweise kann es auch ab der nächsten Generation neu aufgebaut werden.

Fenster: Bestes Individuum



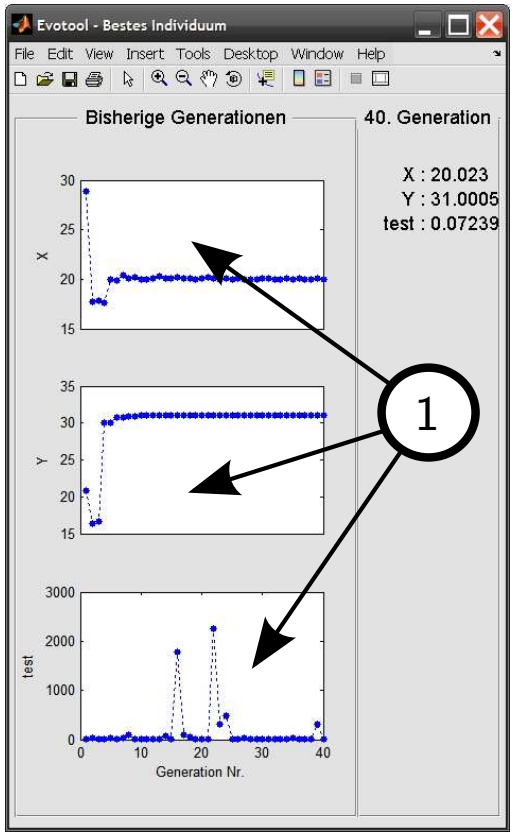
Dieses Fenster gibt Auskunft über Status und Entwicklung des jeweilig besten Individuums einer Generation und erscheint wenn `w.Fig.BestInd` auf `true` eingestellt wurde.

1. Die linke Seite zeigt einen Plot für jede Dimension (In diesem Fall drei). Jeder zeigt den Verlauf der Größe der jeweiligen Dimension über die vergangenen Generationen.
2. Die rechte Seite zeigt die aktuelle Größe der Dimensionen des besten Individuums. Hier lassen sich selbst kleinste Änderungen des besten Individuums beobachten.

Fenster schließen

Auch hier kann das Fenster entweder nur temporär oder dauerhaft geschlossen werden.

Fenster: Bestes Individuum



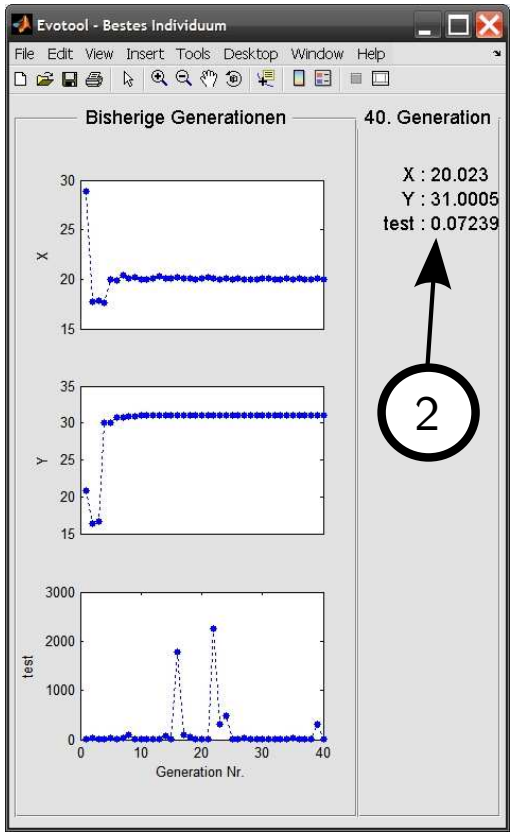
Dieses Fenster gibt Auskunft über Status und Entwicklung des jeweilig besten Individuums einer Generation und erscheint wenn `w.Fig.BestInd` auf `true` eingestellt wurde.

1. Die linke Seite zeigt einen Plot für jede Dimension (In diesem Fall drei). Jeder zeigt den Verlauf der Größe der jeweiligen Dimension über die vergangenen Generationen.
2. Die rechte Seite zeigt die aktuelle Größe der Dimensionen des besten Individuums. Hier lassen sich selbst kleinste Änderungen des besten Individuums beobachten.

Fenster schließen

Auch hier kann das Fenster entweder nur temporär oder dauerhaft geschlossen werden.

Fenster: Bestes Individuum



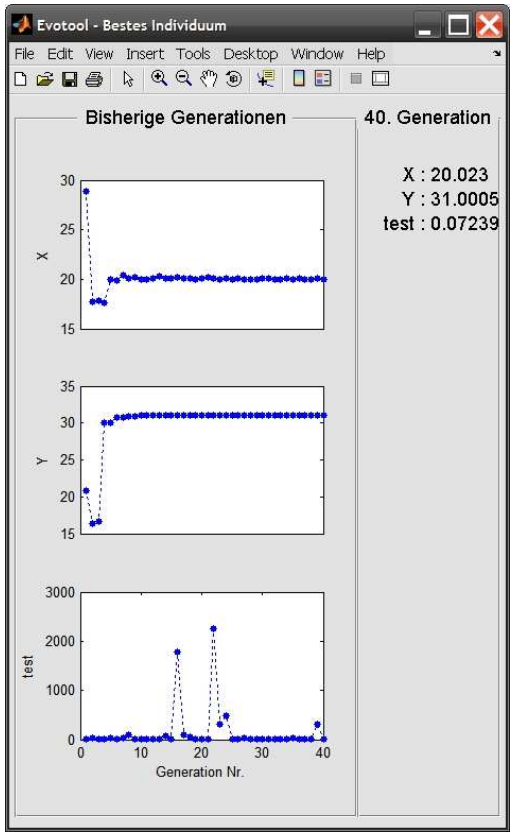
Dieses Fenster gibt Auskunft über Status und Entwicklung des jeweilig besten Individuums einer Generation und erscheint wenn `w.Fig.BestInd` auf `true` eingestellt wurde.

1. Die linke Seite zeigt einen Plot für jede Dimension (In diesem Fall drei). Jeder zeigt den Verlauf der Größe der jeweiligen Dimension über die vergangenen Generationen.
2. Die rechte Seite zeigt die aktuelle Größe der Dimensionen des besten Individuums. Hier lassen sich selbst kleinste Änderungen des besten Individuums beobachten.

Fenster schließen

Auch hier kann das Fenster entweder nur temporär oder dauerhaft geschlossen werden.

Fenster: Bestes Individuum



Dieses Fenster gibt Auskunft über Status und Entwicklung des jeweilig besten Individuums einer Generation und erscheint wenn `w.Fig.BestInd` auf `true` eingestellt wurde.

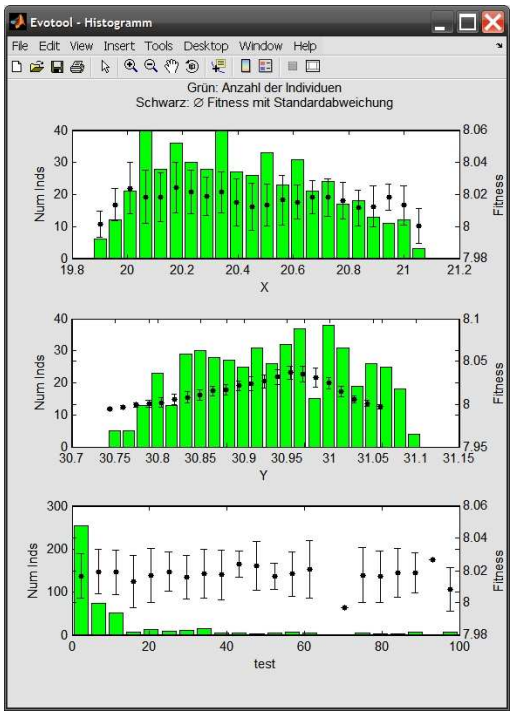
1. Die linke Seite zeigt einen Plot für jede Dimension (In diesem Fall drei). Jeder zeigt den Verlauf der Größe der jeweiligen Dimension über die vergangenen Generationen.
2. Die rechte Seite zeigt die aktuelle Größe der Dimensionen des besten Individuums. Hier lassen sich selbst kleinste Änderungen des besten Individuums beobachten.

Fenster schließen

Auch hier kann das Fenster entweder nur temporär oder dauerhaft geschlossen werden.

Fenster: Histogramm

Dieses Fenster gibt Auskunft über die Verteilung der Individuen der aktuellen Generation und deren durchschnittliche Fitness und erscheint, wenn `w.Fig.histogramm true` ist.



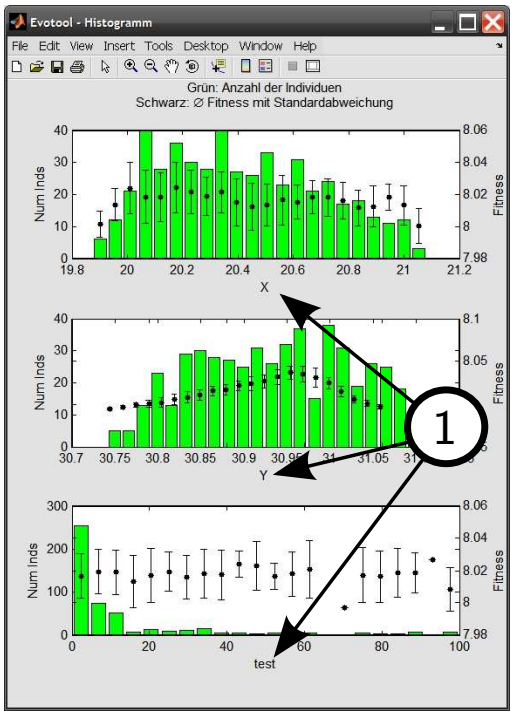
1. Jede Dimension erhält einen eigenen Plot. Die Dimension ist auf der x-Achse aufgetragen.
2. Die grünen Balken sind Ergebnis des Histogramms der Dimension, d.h. sie zeigen an, wie viele Individuen sich in diesem "Container"befinden. Die zugehörige Skalierung ist auf der linken y-Achse aufgetragen.
3. Die schwarzen Punkte geben die durchschnittliche Fitness der Individuen des jeweiligen Containers an. Die Balken nach oben und unten zeigen die Standardabweichung des Ergebnisses an. Die zugehörige Skalierung ist rechts aufgetragen.

Fenster schließen

Wie bei allen anderen Fenstern, kann dieses Fenster temporär oder dauerhaft geschlossen werden.

Fenster: Histogramm

Dieses Fenster gibt Auskunft über die Verteilung der Individuen der aktuellen Generation und deren durchschnittliche Fitness und erscheint, wenn `w.Fig.histogramm true` ist.



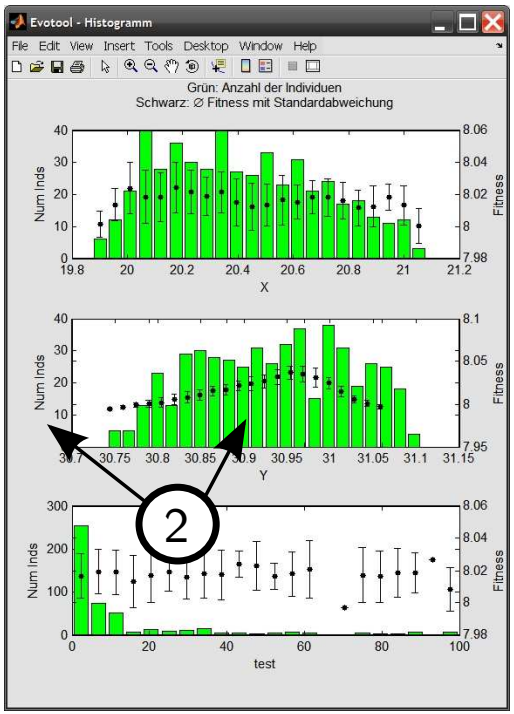
1. Jede Dimension erhält einen eigenen Plot. Die Dimension ist auf der x-Achse aufgetragen.
2. Die grünen Balken sind Ergebnis des Histogramms der Dimension, d.h. sie zeigen an, wie viele Individuen sich in diesem "Container"befinden. Die zugehörige Skalierung ist auf der linken y-Achse aufgetragen.
3. Die schwarzen Punkte geben die durchschnittliche Fitness der Individuen des jeweiligen Containers an. Die Balken nach oben und unten zeigen die Standardabweichung des Ergebnisses an. Die zugehörige Skalierung ist rechts aufgetragen.

Fenster schließen

Wie bei allen anderen Fenstern, kann dieses Fenster temporär oder dauerhaft geschlossen werden.

Fenster: Histogramm

Dieses Fenster gibt Auskunft über die Verteilung der Individuen der aktuellen Generation und deren durchschnittliche Fitness und erscheint, wenn `w.Fig.histogramm true` ist.



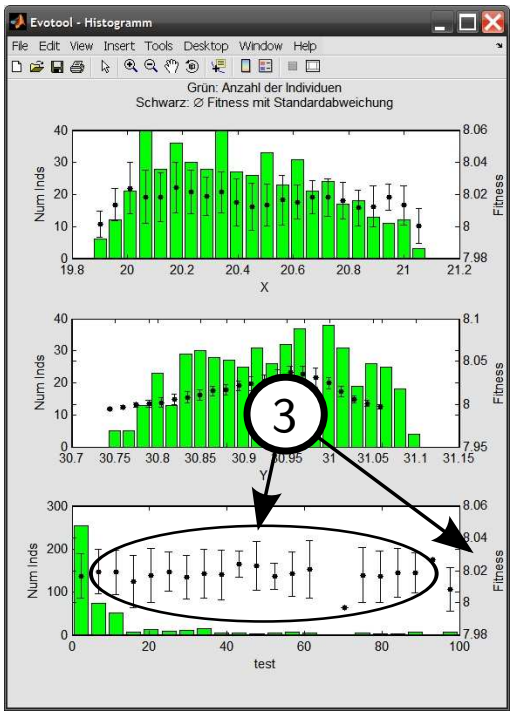
1. Jede Dimension erhält einen eigenen Plot. Die Dimension ist auf der x-Achse aufgetragen.
2. Die grünen Balken sind Ergebnis des Histogramms der Dimension, d.h. sie zeigen an, wie viele Individuen sich in diesem "Container"befinden. Die zugehörige Skalierung ist auf der linken y-Achse aufgetragen.
3. Die schwarzen Punkte geben die durchschnittliche Fitness der Individuen des jeweiligen Containers an. Die Balken nach oben und unten zeigen die Standardabweichung des Ergebnisses an. Die zugehörige Skalierung ist rechts aufgetragen.

Fenster schließen

Wie bei allen anderen Fenstern, kann dieses Fenster temporär oder dauerhaft geschlossen werden.

Fenster: Histogramm

Dieses Fenster gibt Auskunft über die Verteilung der Individuen der aktuellen Generation und deren durchschnittliche Fitness und erscheint, wenn `w.Fig.histogramm true` ist.



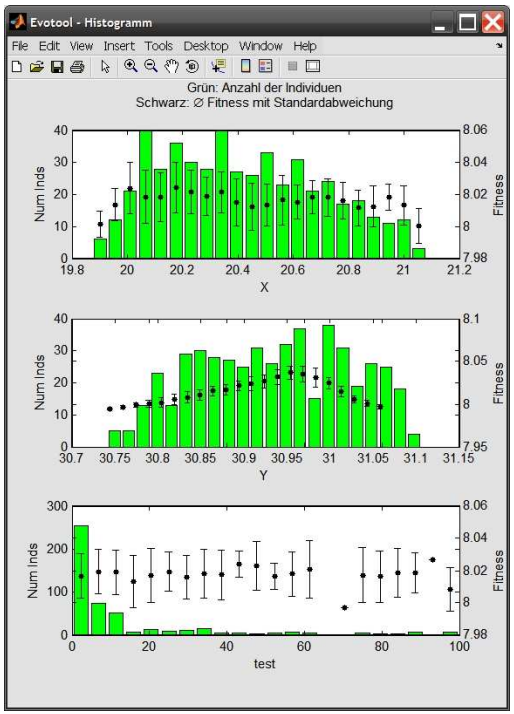
1. Jede Dimension erhält einen eigenen Plot. Die Dimension ist auf der x-Achse aufgetragen.
2. Die grünen Balken sind Ergebnis des Histogramms der Dimension, d.h. sie zeigen an, wie viele Individuen sich in diesem "Container"befinden. Die zugehörige Skalierung ist auf der linken y-Achse aufgetragen.
3. Die schwarzen Punkte geben die durchschnittliche Fitness der Individuen des jeweiligen Containers an. Die Balken nach oben und unten zeigen die Standardabweichung des Ergebnisses an. Die zugehörige Skalierung ist rechts aufgetragen.

Fenster schließen

Wie bei allen anderen Fenstern, kann dieses Fenster temporär oder dauerhaft geschlossen werden.

Fenster: Histogramm

Dieses Fenster gibt Auskunft über die Verteilung der Individuen der aktuellen Generation und deren durchschnittliche Fitness und erscheint, wenn `w.Fig.histogramm true` ist.

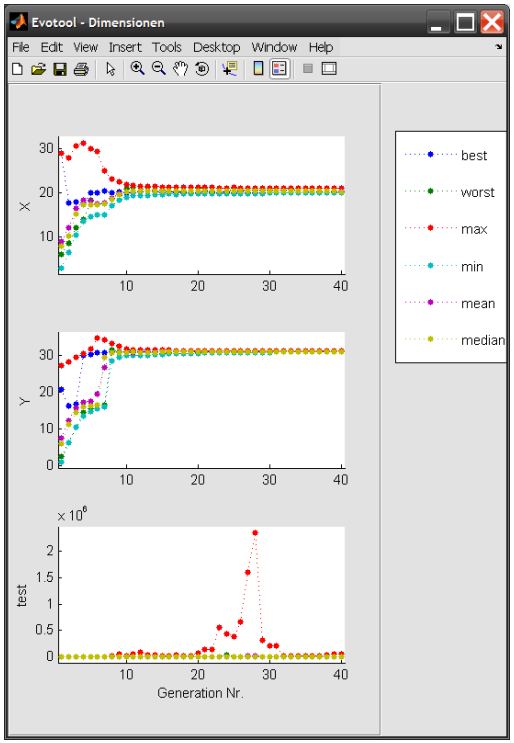


1. Jede Dimension erhält einen eigenen Plot. Die Dimension ist auf der x-Achse aufgetragen.
2. Die grünen Balken sind Ergebnis des Histogramms der Dimension, d.h. sie zeigen an, wie viele Individuen sich in diesem "Container"befinden. Die zugehörige Skalierung ist auf der linken y-Achse aufgetragen.
3. Die schwarzen Punkte geben die durchschnittliche Fitness der Individuen des jeweiligen Containers an. Die Balken nach oben und unten zeigen die Standardabweichung des Ergebnisses an. Die zugehörige Skalierung ist rechts aufgetragen.

Fenster schließen

Wie bei allen anderen Fenstern, kann dieses Fenster temporär oder dauerhaft geschlossen werden.

Fenster: Dimensionen und Mutationsweiten



Dimension

Dieses Fenster ermöglicht die Angabe von diversen Informationen über die Dimensionen bereits berechneter Generationen. Jede Dimension erhält einen eigenen Graph. Die angezeigten Information werden durch den CellString `w.Fig.Inds` festgelegt, völlig analog zu dem Fenster mit der Angabe der Fitness.

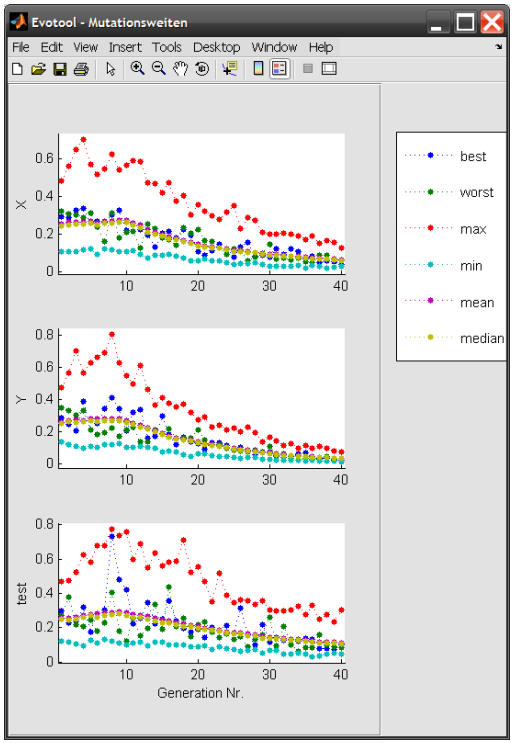
Mutationsweiten

Das Fenster mit den Mutationsweiten verhält sich genau, wie das mit den Dimensionen. Nur werden statt den Dimensionen selbst die Mutationsweiten der jeweiligen Dimension analysiert. Das Verhalten wird über `w.Fig.Mut` festgelegt.

Fenster schließen

Die beiden Fenster können temporär oder dauerhaft geschlossen werden.

Fenster: Dimensionen und Mutationsweiten



Dimension

Dieses Fenster ermöglicht die Angabe von diversen Informationen über die Dimensionen bereits berechneter Generationen. Jede Dimension erhält einen eigenen Graph. Die angezeigten Information werden durch den CellString `w.Fig.Inds` festgelegt, völlig analog zu dem Fenster mit der Angabe der Fitness.

Mutationsweiten

Das Fenster mit den Mutationsweiten verhält sich genau, wie das mit den Dimensionen. Nur werden statt den Dimensionen selbst die Mutationsweiten der jeweiligen Dimension analysiert. Das Verhalten wird über `w.Fig.Mut` festgelegt.

Fenster schließen

Die beiden Fenster können temporär oder dauerhaft geschlossen werden.

Zum vertiefen...

Fuzzy Logik, Neuronale Netze und Evolutionäre Algorithmen
Jürgen Adamy
Shaker Verlag, 2006

Parameteroptimierung eines bionischen Hörmodells zur Periodizitätsanalyse mittels evolutionärer Algorithmen.
Christian Langen
Studienarbeit, TU-Darmstadt 2004