

# MOD-File-Dokumentation

## GLEAM

Version 1.2

Wilfried Jakob

KIT, Campus Nord, Institut für Automation und angewandte Informatik (IAI)  
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany  
email: wilfried.jakob@partner.kit.edu

### Inhalt

|  |          |
|--|----------|
| <b>1 Einführung.....</b>   | <b>3</b> |
| <b>2 MOD-File-Aufbau.....</b>  | <b>3</b> |
| 2.1 MOD-File-Kopf.....   | 3        |
| 2.2 LSV-Zeilen.....  | 6        |
| 2.3 Gentyp-Definitionen.....   | 6        |
| <b>3 Beispiele.....</b>  | <b>7</b> |
| 3.1 Einfaches Genmodell für die Benchmarkfunktion Shekel's Foxholes..... | 7        |
| 3.2 Aktionsmodell der LESAK-Anwendung.....                               | 8        |
| <b>4 Literatur.....</b>  | <b>9</b> |

## Änderungen:

Änderungen gegenüber V1.1:

1. Redaktionelle Änderungen und Präzisierungen
2. Anmerkungen zur maximalen Anzahl von ganzzahligen und reellen Parametern per Gentypdefinition.

# 1 Einführung

GLEAM unterscheidet zwischen einer statischen und einer dynamischen Interpretation von Chromosomen. Letztere spielt bisher nur bei der Codierung der Anwendung „kollisionsfreie Roboterbahnplanung“ (LESAK) eine Rolle. Da GLEAM mit dieser Anwendung entstanden ist, nimmt ein Großteil der Dokumentation auf die damit verbundene Nomenklatur Bezug. Daher folgende Erläuterungen:

1. Die Unterscheidung ist nur bei der Chromosomeninterpretation relevant.
2. Es wird sprachlich zwischen Gen- und Aktionsmodell wie folgt unterschieden:

|                                   |   |
|-----------------------------------|---|
| Statische Interpretation:         | Dynamische oder zeit-bezogene Interpretation: |
| Genmodell                         | Aktionsmodell (Handlungsmodell)               |
| Chromosom                         | Aktionskette (AK)                             |
| Gen                               | Aktion  |
| Parameter (Entscheidungsvariable) | Parameter                                     |

Eine MOD-Datei enthält das Gen- oder Aktionsmodellmodell und besteht aus einem beschreibenden Kopf, Regeln zur Konstruktion von Chromosomen (Aktionsketten) und der Beschreibung der Gene (Aktionen) und ihrer Parameter sowie deren Grenzen (explizite Restriktionen). Damit wird ein Aufgabengebiet insofern umrissen, als mit Hilfe des Genmodells eine Beschreibung der Lösung möglich sein muss. Das Genmodell steht in engem Zusammenhang mit dem Simulator, der die Gene bzw. Aktionen interpretieren und „ausführen“ muss, um die im Chromosom steckende Lösung bewerten zu können. Eine erweiterte Version für HyGLEAM enthält darüber hinaus Angaben zu verwendeten lokalen Suchverfahren (LSVs) und Heuristiken. Zunächst wird die GLEAM-Version behandelt.

Während die Anzahl der Gentyp-Definitionen nicht begrenzt ist (abgesehen vom verfügbaren Speicher), ist es die Anzahl der Parameter pro Gentyp sehr wohl. Die Grenzen für ganzzahlige und reelle Parameter sind in der Datei `chaindef.h` zu finden: `I_PAR_ANZ_MAX` und `R_PAR_ANZ_MAX`. Die aktuellen Werte sind 12 für die ganzzahligen und 8 für die reellen Parameter pro Gentyp. Ab Version 2.2.2 können diese Einschränkungen durch Setzen des Compiler-Schalters `WITH_DYN_GENES` in der `schalter.h`-Datei außer Kraft gesetzt werden. Da es nur wenig Erfahrung mit dieser Erweiterung gibt, ist Vorsicht geboten und es wird empfohlen, bei Problemen lieber die genannten Konstanten höher zu setzen.

Die MOD-Datei ist ein ASCII-File und wird mit einem normalen Text-Editor erstellt.

## 2 MOD-File-Aufbau

### 2.1 MOD-File-Kopf

Der Kopf besteht aus mindestens 7 Zeilen und hat folgenden Aufbau:

#### 1. Zeile:

Kommentarzeile:

```
===== ... ===== Genmodell fuer GLEAM/AE ===== ... =====
```

**2. Zeile:**

MOD-File-Identifikationszeile:

GLEAM/AE <Anwendungsklasse> <Anwendung> [withLS] [`<TimeStamp>`]

mit: <Anwendungsklasse> Klasse der Anwendung und der Simulatoranbindung.  
 <Anwendung> Bezeichnung der Anwendung bzw. des Simulators. Muss je nach Anwendungsklasse vorhanden sein.  
 withLS optionale LSV-Kennung. Sie gibt an, ob bei HyGLEAM LSVs genutzt werden sollen und die entsprechende Parametrierung in der MOD-Datei enthalten ist.

Alles nach der optionalen withLS-Kennung wird überlesen. Der Zeitstempel dient nur der Dokumentation. Derzeit sind folgende Anwendungsklassen implementiert, siehe auch [1]:

|             |  |
|-------------|--|
| MathFkt     | interner Simulator. Es gibt folgende Anwendungen (Benchmarkfunktionen):  |
|             | MBF-Ackley Funktion nach Ackley.   |
|             | MBF-Beale Funktion nach Beale.   |
|             | MBF-Bigg Funktion nach Bigg.   |
|             | MBF-Bracken Funktion nach Bracken.   |
|             | MBF-Fletcher Funktion nach Fletcher und Powell.  |
|             | MBF-Foxholes Shekel's Foxholes   |
|             | MBF-Fractal Fraktale Funktion  |
|             | MBF-Griewank Funktion nach Griewank  |
|             | MBF-HValley Helical Valley   |
|             | MBF-Rast verallgemeinerte Rastrigin-Funktion   |
|             | MBF-RBerg Funktion nach Rechenberg   |
|             | MBF-Sphere Kugelfunktion   |
|             | MBF-WSphere gewichtete Kugelfunktion   |
| LESAK/Plus  | interner Roboter-Simulator. Keine Angabe einer Anwendung.  |
| Matlab      | Matlab-Simulator. Die Anwendung ist eine Kennung für MatPower, die angibt, ob mit OPF (OPF) oder ohne (PF) gerechnet werden soll.  |
| ExtSimuServ | Externe Simulationsdienste (ESS). Keine Angabe einer Anwendung, da dies durch die Auswahl eines Simulatormodells, welches die externen Simulationsdienste kennen und anbieten müssen, abgedeckt wird. Die Angabe stellt sicher, dass alle für die ESS erforderlichen Programmparameter gesetzt wurden. |
| OPAL/V      | OPAL/V Scheduling-Anwendung für Testzwecke, ohne Angabe einer Anwendung.   |

Alte Anwendungsklassen, die zwar noch bekannt, wahrscheinlich nicht mehr nutzbar sind:

GADO-V2 mit den Simulatoren (Anwendung) Mathematica, Eldo und GenExtSim für die allgemeine auf Pipes basierende Schnittstelle zu externen Simulatoren.

**3. Zeile:**

Die Zeile besteht aus drei Zahlenwerten und nachfolgenden Kommentaren, die in der Regel die Variablenamen im „hmod“-Package darstellen und deren zulässige Wertebereiche angeben. Es handelt sich um Generierungsvorgaben für die drei Chromosomentypen (die beiden ersten Zahlen) und für die Segmentierung (3. Zahl):

<gen\_len\_mode> <gen\_akt\_mode> <gen\_segm\_vert>

Die ersten beiden Zahlenwerte bestimmen den Chromosomentyp wie folgt:

Chromosom mit fester Länge und irrelevanter Genreihenfolge (Typ 1):

gen\_len\_mode: 1 fixe Länge, die Mindestkettenlänge (Zeile 4) bestimmt die AK-Länge

gen\_act\_mode 1 jede Aktion mit einer Priorität >0 wird in aufsteigender Reihenfolge genau einmal erzeugt

Chromosom mit fester Länge und relevanter Genreihenfolge (Typ 2):

gen\_len\_mode: 1 fixe Länge, die Mindestkettenlänge (Zeile 4) bestimmt die AK-Länge  
 gen\_act\_mode 2 jede Aktion mit einer Priorität >0 wird in beliebiger Reihenfolge genau einmal erzeugt

Chromosom mit dynamischer Länge und relevanter Genreihenfolge (Typ 3):

gen\_len\_mode: 2 dynamische Länge, die Generierungslänge der AK liegt zwischen der Mindest- und der Maximalkettenlänge von Zeile 4.  
 gen\_act\_mode 3 jede Aktion mit einer Priorität >0 wird in beliebiger Reihenfolge entsprechend der durch die Prioritäten gebildeten Verteilung erzeugt. Die Wahrscheinlichkeit zur Generierung einer Aktion  $j$  beträgt also

$$prio_j / \sum prio$$

Damit kann eine Aktion keinmal, einmal oder mehrfach vorkommen.

Dritter Zahlenwert gen\_segm\_vert: Vorgabe zur Segmentierung bei der Chromosomengenerierung und der Nachsegmentierung bei Genetic Repair, wie es bei Crossover-Operatoren vorkommen kann:

- 1 Die Segmentlänge wird gleichverteilt innerhalb der in Zeile 4 festgelegten Grenzen für Segmente bestimmt. Dies ist die Standardeinstellung.
- 2 Die Segmentlänge wird mit fallender Verteilung innerhalb der in Zeile 4 festgelegten Grenzen für Segmente bestimmt. Für die LESAK-Anwendung oder andere Anwendungen mit dynamischen Chromosomen.

#### **4. Zeile:**

Die Zeile besteht aus vier Zahlenwerten und nachfolgenden Kommentaren, die in der Regel die Variablennamen im „hmod“-Package darstellen. Es handelt sich um Generierungsvorgaben für die Ketten- und Segmentlänge bestehend aus je zwei Werten:

```
<min_ketten_len> <max_ketten_len> <min_abschn_len> <abschn_delta>
```

Bei AK-Typ 1 und 2 wird die min\_ketten\_len benutzt. Die max\_ketten\_len sollte auf den gleichen Wert gesetzt werden und es müssen mindestens min\_ketten\_len Gentypen definiert werden.

Bei AK-Typ 3 muss gelten: min\_ketten\_len ≤ max\_ketten\_len.

Ein Segment hat bei Generierung mindestens die Länge min\_abschn\_len und darf daher nicht größer als min\_ketten\_len sein. abschn\_delta gibt an, wie viel weitere Aktionen (Gene) ein Abschnitt maximal haben darf.

#### **5. Zeile:**

Die Zeile gibt die Anzahl der Gentypen des Genmodells an. Im nachfolgenden Genbeschreibungsteil müssen neben der immer vorhandenen Definition einer Unterkette<sup>1</sup> genauso viel Gendefinitionen enthalten sein.

---

1: Unterketten spielen keine Rolle mehr

**6. Zeile:**

Anzahl der Ergebniswerte des Simulators `akt_roh_erg_werte` und ein optionaler Steuerungsparameter für die genetischen Operatoren für kleine Parameteränderungen, `par_change_small` und `segm_change_small`, siehe EVO-File. `<small_change_frac>` gibt den zu verwendenden Anteil am tatsächlich möglichen Änderungsbereich an, welcher sich aus dem aktuellen Wert der Entscheidungsvariablen und ihrer Wertebereichsgrenzen ergibt..Default ist 0.001.

```
<akt_roh_erg_werte>  [<small_change_frac>]
```

Beim Laden eines MOD-Files wird ein Feld der Länge `akt_roh_erg_werte` angelegt, welches zum Empfang der Ergebniswerte eines Aufrufs des (externen) Simulators dient und auf das die Bewertung zugreift. Daher darf die Bewertung (BEW-File) nicht mit mehr Bewertungskriterien arbeiten, als durch `akt_roh_erg_werte` vorgegeben wird. Dies wird überprüft und führt negativenfalls bei einer CLV-Version zum Abbruch. Bei TUI-Versionen gibt es Fehlermeldungen und eine Sperrung entsprechender Funktionen.

**7. Zeile:**

Anzahl der anwendungsspezifischen Zusatzfiles:

```
<anw_file_anz>
```

Diese Dateien werden dem System nachfolgend bekannt gemacht und sind für den Simulator gedacht. Ihre Angabe erfolgt in der Experiment-Datei (EXP-Datei).

**Zusatzfile-Zeilen (ab Zeile 8):**

Die Spezifikation von Zusatzfiles entfällt, wenn `anw_file_anz = 0`. Pro Zusatzfile gibt es eine Zeile, die folgendes enthält:

```
<io_code>  <ext>  <datei_bezeichnung>  <menue_eintrag>
```

|                                 |  |
|---------------------------------|--|
| <code>io_code:</code>           | 0=lesen, 1=lesen/schreiben, 2=schreiben          |
| <code>ext:</code>               | Datei-Erweiterung                                |
| <code>datei_bezeichnung:</code> | Dateibezeichnung für Dialoge (max. 15 Zeichen)   |
| <code>menue_eintrag:</code>     | Eintrag im entsprechenden Menü (max. 10 Zeichen) |

**2.2 LSV-Zeilen**

Dieser Bereich ist nur vorhanden, wenn im Kopf (Zeile 2) die Kennung `witLS` enthalten ist. Ansonsten entfällt der Bereich vollständig. Wenn er vorhanden ist, wird er durch zwei Zeilen mit den Kennungen `BEGIN_LS` und `END_LS` geklammert.

Eine Dokumentation des Inhalts erfolgt später.

**2.3 Gentyp-Definitionen**

Der Bereich der Gentypdefinitionen besteht aus 7 einleitenden Kommentarzeilen, welche die Bedeutung der Einträge einer Gentypdefinition beschreiben. Häufiges Aussehen:

```
=====
1) 2) 3) 4) 5) 1: Prio. der Aktion 2: Anzahl Integer-Parameter
3: Anzahl Real-Parameter 4: Untere ParamGrenze
5: Obere ParamGrenze MaxBezLen=20 MaxParBezLen=20
-----
Bezeichner/ParBez
=====
```

Die Definition eines Gentypts hat folgenden Aufbau:

|        |           |           |               |             |
|--------|-----------|-----------|---------------|-------------|
| <Prio> | <IParAnz> | <RParAnz> | <AktionsName> |             |
|        |           | <ug>      | <og>          | <ParamName> |
|        |           |           |               | <DimString> |
|        |           | .         | .             | .           |
|        |           | <ug>      | <og>          | <ParamName> |
|        |           |           |               | <DimString> |
|        |           | <ug>      | <og>          | <ParamName> |
|        |           |           |               | <DimString> |
|        |           | .         | .             | .           |
|        |           | <ug>      | <og>          | <ParamName> |
|        |           |           |               | <DimString> |

1. IntPar

1. IntPar

n. IntPar

n. IntPar

1. RealPar

1. RealPar

m. RealPar

m. RealPar

mit: <Prio>

Ganze Zahl. Wahrscheinlichkeitsanteil des Gens während der Chromosomengenerierung (0=aus, für Chromosomentyp 1 und 2 ist jede Zahl größer als 0 für eine Aktivierung ausreichend. Nur für den Chromosomentyp 3 sind größere Werte im Sinne einer Wahrscheinlichkeitsverteilung sinnvoll. Die Wahrscheinlichkeit eines Gens  $j$  ist gleich

$$\frac{prio_j}{\sum_{i=1}^{genTypAnzahl} prio_i}$$

wobei *genTypAnzahl* die Anzahl der Gentypten ist

|               |  |
|---------------|--|
| <IparAnz>     | Anzahl der Integer-Parameter des Gens (entspricht obigem $n$ )   |
| <RparAnz>     | Anzahl der Real-Parameter des Gens (entspricht obigem $m$ )  |
| <ug>          | Ganze oder reelle Zahl. Parameter-Untergrenze  |
| <og>          | Ganze oder reelle Zahl. Parameter-Obergrenze   |
| <DimString>   | Die Dimensionsangabe dient nur zu Anzeigezwecken und besteht aus drei durch Blanks getrennten Einträgen: Dem Dimensionsbezeichner, einer kleiner/größer-Angabe (<,>) und des alternativen Dimensionsbezeichners, wenn der Wert kleiner oder größer als 1/1000 bzw. 1000 ist (! ! ! = leerer Eintrag, die maximale Länge eines Dimensionsbezeichners beträgt 12 Zeichen (EINHEIT_MAX)). Der alternative Dimensionsbezeichner kommt nur bei der Darstellung von Simulationsergebnissen zum Zuge, sofern die Anwendung so konfiguriert wurde, dass die Entscheidungsvariablen dargestellt werden. |
| <AktionsName> | Name der Aktion oder des Gens für Anzeigezwecke (maximal 20 Zeichen (ACT_NAME_MAX))  |
| <ParamName>   | Name des Parameters für Anzeigezwecke (maximal 20 Zeichen (PAR_NAME_MAX))  |

Die Beschreibung der ersten Aktion ist fix und besteht aus folgendem Eintrag:

0 0 0

Unterkette

### 3 Beispiele

Die Beschreibung des Aufbaus von MOD-Files soll an Hand von zwei Beispielen illustriert werden. Das erste Beispiel ist ein einfaches Genmodell für die mathematische Testfunktion „Shekel's Foxholes“ [1] bestehend aus 2 Gentypten, die auch noch identisch sind. Das zweite Beispiel zeigt ein Aktionsmodell für die LESAK-Anwendung [1, 2, 3].

### 3.1 Einfaches Genmodell für die Benchmarkfunktion Shekel's Foxholes

Die tatsächlich gelesenen Teile einer Zeile sind grün markiert.

```
***** Genmodell fuer GLEAM/AE *****
GLEAM/AE  MathFkt  MBF-Foxholes 15.05.1998
1 1 1      gen_len_mode (1,2)  gen_akt_mode (1,2,3)  gen_segm_vert (1,2)
2 2 1 0    min_ketten_len  max_ketten_len  min_abschn_len  abschn_delta
2          Anzahl der Gentypdefinitionen
1 0.0001   akt_roh_erg_werte des (Ext)Simu  small_change_frac (optional)
0          Anzahl der Zusatz-Files
*****
1) 2) 3) 4) 5) 1: Prioritaet der Aktion 2: Anzahl Integer-Parameter
3: Anzahl Real-Parameter 4: Untere Parametergrenze
5: Obere Parametergrenze MaxBezLen=20 MaxParBezLen=20
-----
Bezeichner/ParBez  Konstantennamen  Konstantenwert
*****
0 0 0      Unterkette
1 0 1      Param1
          -500.0 +500.0  p1
          ! ! !
1 0 1      Param2
          -500.0 +500.0  p2
          ! ! !
```

### 3.2 Aktionsmodell der LESAK-Anwendung

Die tatsächlich gelesenen Teile einer Zeile sind grün markiert.

```
***** Handlungsmodell für GLEAM/AE *****
GLEAM/AE  LESAK/Plus 01.07.1998 revised: 23.03.2016
2 3 2      gen_len_mode (1,2)  gen_akt_mode (1,2,3)  gen_segm_vert (1,2)
5 50 4 6    min_ketten_len  max_ketten_len  min_abschn_len  abschn_delta
13          Anzahl der Aktionstypdefinitionen
7 0.002     akt_roh_erg_werte des (Ext)Simu  small_change_frac (optional)
2          Anzahl der Zusatz-Files
0 kin Kinematik-Datei Kinematik
0 obs Hindernis-Datei Hindernis
*****
1) 2) 3) 4) 5) 1: Prioritaet der Aktion 2: Anzahl Integer-Parameter
3: Anzahl Real-Parameter 4: Untere Parametergrenze
5: Obere Parametergrenze MaxBezLen=20 MaxParBezLen=10
-----
Bezeichner/ParBez  Konstantennamen  Konstantenwert
*****
0 0 0      Unterkette  ACTIVITY_ACT  (00)
5 0 2      Setze_Motor1_U  MOTOR_1  (01)
          0.1 15.0  Rampe
          ! ! !
          -120.0 120.0  U_Wert_1
          V < mV
5 0 2      Setze_Motor2_U  MOTOR_2  (02)
          0.1 15.0  Rampe
          ! ! !
          -120.0 120.0  U_Wert_2
          V < mV
5 0 2      Setze_Motor3_U  MOTOR_3  (03)
          0.1 15.0  Rampe
          ! ! !
          -120.0 120.0  U_Wert_3
          V < mV
5 0 2      Setze_Motor4_U  MOTOR_4  (04)
```



|    |   |        |       |                |                   |      |
|----|---|--------|-------|----------------|-------------------|------|
|    |   | 0.1    | 15.0  | Rampe          |                   |      |
|    |   |        |       | !!!            |                   |      |
|    |   | -120.0 | 120.0 | U_Wert_4       |                   |      |
|    |   |        |       | V < mV         |                   |      |
| 5  | 0 | 2      |       | Setze_Motor5_U | MOTOR_5           | (05) |
|    |   | 0.1    | 15.0  | Rampe          |                   |      |
|    |   |        |       | !!!            |                   |      |
|    |   | -120.0 | 120.0 | U_Wert_5       |                   |      |
|    |   |        |       | V < mV         |                   |      |
| 5  | 0 | 1      |       | Motor1_aus     | MOTOR_1_AUS       | (06) |
|    |   | 0.1    | 12.0  | Rampe          |                   |      |
|    |   |        |       | !!!            |                   |      |
| 5  | 0 | 1      |       | Motor2_aus     | MOTOR_2_AUS       | (07) |
|    |   | 0.1    | 12.0  | Rampe          |                   |      |
|    |   |        |       | !!!            |                   |      |
| 5  | 0 | 1      |       | Motor3_aus     | MOTOR_3_AUS       | (08) |
|    |   | 0.1    | 12.0  | Rampe          |                   |      |
|    |   |        |       | !!!            |                   |      |
| 5  | 0 | 1      |       | Motor4_aus     | MOTOR_4_AUS       | (09) |
|    |   | 0.1    | 12.0  | Rampe          |                   |      |
|    |   |        |       | !!!            |                   |      |
| 5  | 0 | 1      |       | Motor5_aus     | MOTOR_5_AUS       | (10) |
|    |   | 0.1    | 12.0  | Rampe          |                   |      |
|    |   |        |       | !!!            |                   |      |
| 10 | 0 | 0      |       | Block_Beginn   | START_BLOCK_BEGIN | (11) |
| 10 | 0 | 0      |       | Block_End      | START_BLOCK_END   | (12) |
| 30 | 1 | 0      |       | Unveraendert   | UNVERAENDERT      | (13) |
|    |   | 1      | 20    | Takt_Anz       |                   |      |
|    |   |        |       | !!!            |                   |      |

Die beiden Spalten „Konstantennamen“ und „Konstantenwert“ dienen nur zur Dokumentation.

Es werden für jeden Motor des 5-achsigen Industrieroboters eigene Aktionen (Gene) zum Verfahren und zum Abschalten definiert. Danach folgen die drei Aktionen der Zeitsteuerung, siehe auch [2, Abschnitt 4.2] oder [3].

Anstelle der Definition eigener Aktionen für jeden Motor hätte man auch die Motornummer zu einem weiteren Parameter einer generellen Aktion jeweils zum Starten und Beenden einer Motorbewegung machen können. Dies hätte das Aktionsmodell vereinfacht. Davon wurde aber Abstand genommen, da sonst der neue Parameter *Motornummer* eine große phänotypische Bedeutung erhalten hätte und ein erhebliches Missverhältnis zwischen der phänotypischen Auswirkung einer kleinen Änderung dieses Parameters und einer ebenso kleinen Änderung eines der anderen Genparameter entstanden wäre. Damit wäre die Forderung nach möglichst starker Kausalität bei der Festlegung des Genmodells verletzt worden. Tatsächlich konnte auch gezeigt werden, dass das hier vorgestellte Handlungsmodell einem bestehend aus zwei universellen Bewegungsaktionen mit Motornummern als Parameter überlegen ist.

## 4 Literatur

- [1] W. Jakob: *Applications Included in GLEAM and Integration of New Ones*. Technical Paper, V1.0, IAI, 2020.  
See [GLEAM-Applications\\_Docu\\_V1.0.pdf](#)
- [2] C. Blume, W. Jakob: *GLEAM - General Learning Evolutionary Algorithm and Method: Ein Evolutionärer Algorithmus und seine Anwendungen*. Karlsruhe: KIT Scientific Publishing, 2009. doi: [10.5445/KSP/1000013553](https://doi.org/10.5445/KSP/1000013553)
- [3] C. Blume, W. Jakob: *GLEAM - An Evolutionary Algorithm for Planning and Control Based on Evolution Strategy*. In: E. Cantù-Paz (ed.): *Conf. Proc. of Genetic and Evolutionary Computation Conference (GECCO 2002)*, New York, Vol. Late Breaking Papers, 2002, pp.31-38