EVO File Documentation

GLEAM

Version 1.0

Wilfried Jakob
KIT, Campus North, Institute for Automation and Applied Informatics (IAI)
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
email: wilfried.jakob@partner.kit.edu

Contents

Release Notes

Changes to V1.0:
1.

1 Introduction

Reference is made to the description of the use of genetic operators to produce offspring from a mating in the user documentation [1, section 3.4]. Knowledge of these is prerequisite for understanding this documentation.

There is a maximum of 10 operator groups (constant `ANZ_PAAR_OPS` in `evo.h`). Each operator group can consist of a maximum of 16 genetic operators (constant `ANZ_GEN_OPS` in `go_gsw.c`). Since each group of operators can produce up to two offspring, up a maximum of 20 offspring per mating is possible. However, lower values in the range between 5 and 9 should make sense.

2 Mutation Rates per Mutation Operator

Mutation operators can be applied several times to the genes or segments of a chromosome. The frequency (mutation rate) is determined randomly depending on the chromosome length. For this purpose, one of four algorithms is assigned to each of the 22 predefined mutation operators and parameterized by the EVO file. According to the selected algorithm, the number of applications of a mutation operator `amount` is diced between 1 and `range`, whereby `range` is limited to the chromosome length.

```
Id = 0 (lin1) : range = round (x/a + b);    Min (range) = c
                if (x < a2)
                   amount = 0;
                else
                   amount = irand (range) + 1;
Id = 1 (lin2) : range = round (x/a + b);    Max (range) = c
                if (x < a2)
                   amount = 0;
                else
                   amount = irand (range) + 1;
Id = 2 (lin3) : if (x < c)
                   range = round (x/a  + b);
                   amount = irand (range) + 1;
                else
                   range = round (x/a2 + b2);
                   if (range < 2)
                      amount = 1;
                   else
                      amount = irand (range) + 1;
Id = 3 (quad) : range = round (a*SQR(x) + b*x + c);
                if (range < 2)
                   amount = 1;
                else
                   amount = irand (range) + 1;
```

The corresponding software is included in the module `go_gsw.c` of the `evo` package. Please refer to the documentation there.

3 Structure of an EVO File

Unlike an EXP or TSK file, an EVO file is format-bound. This means that the comment lines must be adhered to and, in principle, they could also be blank lines. Furthermore, each line can contain any text after the required data entries.

In order to be able to adapt the execution probabilities of the operator groups and the operators contained therein to the evolutionary progress, three parameter sets are provided, which are used depending on the fitness of the parent. For this purpose, three fitness areas[1] are defined in the header of an EVO file using the two fitness limits (F_UL1 and F_UL2). Accordingly, there are three data sets for the parameterization of the mutation rates and three data sets for the configuration of the operator groups and the specification of the corresponding probabilities.

3.1 EVO File Header

The header of an EVO file has the following content:

```
;================================ GLEAM/LESAK ================================
; Parameters for all genetic operators.                    File:lsk_stnd_gb.evo
; Standard settings for LESAK                           last update: 30.06.1998
;============================================================================
GLEAM/AE
40000    60000   Fitness upper limits F_UL1 and F_UL2 for parameter sets 1 and 2
22    9          # of mutation operators (LAST_GO + 1)    # of operator groups
```

After four comment lines follow the lines with the program identification GLEAM/AE and the specification of the two note limits `F_UL1` and `F_UL2`. In the example, these have the values 40000 and 60000, followed by the number of mutation operators and the number of configured operator groups, here 22 and 9.

3.2 Parameterization of the Mutation Rates

This is followed by three parameter sets for the parameterization of mutation rates as a function of chromosome length. Each parameter set begins with four comment lines, e.g.:

```
;============================================================================
; Parameter set 1 of "mut_data" for chromosomes with poor fitness (fit < F_UL1):
;id   a         b         c         a2        b2
;============================================================================
```

The first column is the identifier `Id` of the algorithm for calculating `range` and the other columns contain the parameters for the selected algorithm, see section 2. All parameters must be specified, even if the chosen algorithm only requires some of them.

3.3 Configuration of Operator Groups and Specification of Probabilities

After the three parameter sets for the parameterization of mutation rates, there are three parameter sets for the assignment of genetic operators to operator groups together with the respective execution probabilities.

Each of these parameter sets starts with five comment lines, e.g:

```
;============================================================================
; Parameter set 1 of op. groups "go_list" for chr. with poor fitn (fit < F_UL1):
; p_akt  p_min  eins  op_anz             operator or group designation
;   op_list                   p_list                              8.76 offspring
;============================================================================
```

1: It should be remembered that in GLEAM the fitness has a value range of 0 to 100000, with 0 being the lowest fitness.

In this example the comment lines contain a kind of minimal documentation and it was also noted how many offspring are generated on average per mating based on the selected probabilities. Since the specifications `go_list`, `p_akt`, `p_min`, `eins` `op_anz`, `op_list` and `p_list` correspond to the variable names in the corresponding code, they were not translated.

This is followed by two lines for the configuration of each operator group. The number of these groups is given in the file header. The first line contains the following information:

- the group probability: $0 \leq$ `p_akt` ≤ 1.0
- a currently unused item `p_min`
- an indication (`eins`) whether one or two offspring should result from the application of the operators of this group: 0: two offspring, 1: one offspring
- The number of operators in this group, at least 1.

The second line contains the operator list followed by a list of execution probabilities for each operator in the first list.

Three examples may illustrate this:

```
0.8     0.3      0       1                    recomb
  24                              1.0
```

The group contains one operator and two descendants are created (column `eins` contains a 0). The group probability is 80%. The operator 24 is in the operator list of the group, which corresponds to the recombination. In GLEAM, recombination means an n-point crossover. The probability of this operator is 100%, which is obvious for a single-element list.

The second example combines the 1-point crossover with two mutations and the group is executed with a 60% probability:

```
0.6     0.15     0       3                    cross_over
  23 1 9                          1.0 0.8 0.5
```

After the crossover, there is an 80% probability of a parameter change of the first child followed by a gene shift with 50%. The second child remains always unchanged.

The third example mutates a clone of the parent with a group probability of 90% and carries out three mutations with different probabilities. If none of these mutations occur or if nothing changes by chance, the descendant is deleted.

```
0.9     0.2      1       3                    Neu-Parametrierung
   0  2  4                          0.6  0.7  0.8
```

The mutations carried out here are:

1. New parameter value (0) with 60 % probability
2. New parameterization of a gene (2) with 70 % probability
3. Reparameterization of all genes of a segment (4) with 80 % probability

4 Standard Operators of GLEAM

4.1 Standard Mutations of GLEAM

The table below shows the operator code of the standard mutations (see also `evo.h` in the `evo` package). The mutations are described in [2, paragraph 4.4]. They can be applied several times to a chromosome, as described above.

Operator-code	Short Description	Designator in an EVO file
0	Random redetermination of a randomly selected gene parameter	par_change_new
1	Change of one or more randomly selected parameter values of a gene	par_change_rel
2	Randomly select all parameters of a randomly selected gene	act_new_param
3	Change of one or more parameters of all genes of a randomly selected segment	segm_change_rel
4	All parameters of all genes of a randomly selected segment are randomly chosen	segm_new_param
5	Replacement of a randomly selected gene by a randomly newly determined	act_exchange
6	Insertion of a new gene at a random position	add_new_act
7	Insertion of a copy of a randomly selected gene behind this gene	double_act
8	Deletion of a randomly selected gene	delete_act
9	Shifting a randomly determined gene to a randomly determined position	act_translocation
10	A randomly created new segment replaces a randomly selected segment	segm_exchange
11	Insertion of a copy of a randomly determined segment behind this segment	double_segm
12	Deletion of a randomly selected segment	delete_segm
13	Shifting a randomly determined segment behind another randomly determined segment	segm_transl
14	Arranging the genes of a randomly selected segment in reverse order	segm_inversion
15	A randomly selected segment is merged with the following one	integr_nachb_segm
16	Shifting a randomly determined segment before another randomly determined segment and merging the two	integr_segm
17	A randomly determined segment border is shifted to the right or left by a randomly determined number of genes up to the next segment border at maximum	schieb_segm_grenz
18	A randomly determined segment is divided at a randomly determined location, provided it has at least two genes	teile_segm
19	A new randomly generated segment is inserted after a randomly selected one	add_new_segm
20	Small change in one or more randomly selected parameter values of a gene	par_change_small
21	Small modification of one or more parameters of all genes of a randomly selected segment	segm_change_small

Table of Standard Mutations of GLEAM

Note that the 3 mutations `integr_nachb_segm`, `schieb_segm_grenz` and `teile_segm` only change the segment structure and thus leave the phenotype of the chromosome unchanged. Since unaltered offspring are deleted immediately, these mutations should always be used in conjunction with other mutations, unless they are used as mutations of the first offspring of a recombination or crossover.

The two mutations `par_change_small` and `segm_change_small` only operate over a partial range of the respective current change interval, whereby the default value is one thousandth of the respective current possible change range, see also [2, Section 4.4.3]. This value can be adjusted in the MOD file by an optional entry behind the specification of the "`akt_roh_erg_wertes (Ext)Simu`" in line 6, e.g. to one hundredth by specifying 0.01.

4.2 Standard Crossover Operators of GLEAM

In GLEAM, seven standard crossover or recombination operators are implemented, all of which operate at the segment boundaries. This means that crossover points cannot lie within a segment. Recombination in GLEAM refers to crossover operators that have more than 2 crossover points.

The table below shows the operator code of the standard crossover operators (see also evo.h in the `evo` package). The first three operators are suitable for all chromosome types, while the following four only make sense if there are sequence restrictions for combinatorial tasks and chromosome types 2 or 3 (see also [3] and [2, sections 4.2.2 and 4.3]) are used.

Operator-code	Short Description	Designator in an EVO file
22	Exchange of two randomly determined segmens of the two parent chromosomes	`segm_cross_over`
23	1-point crossover	`cross_over`
24	n-point crossover (recombination)	`recomb`
25	Order based crossover as 2-point crossover [5, 7]	`OX_XO`
26	Order based crossover as n-point crossover [5, 7]	`OX_RECO`
27	Position based crossover as 2-point crossover [6, 7]	`PPX_XO`
28	Position based crossover as n-point crossover [6, 7]	`PPX_RECO`

Table of the Standard Crossover Operators of GLEAM

For all operators, the parents must have at least two segments, and for the two OX operators there must be at least three. Otherwise, no descendants are generated by the affected operator group.

The OX and PPX operators pass on the relative gene order of the parents to the offspring and are therefore only useful in the context of combinatorial tasks with order constraints such as job shop scheduling problems. Of course, they can also be used for tasks without such restrictions, but in this case they do little more than the first three crossover operators in the table. In contrast to PPX operators, the OX operators have proven to be useful for scheduling tasks with order restrictions [7].

Both operator groups take the segmentation of GLEAM into account

- in that (one or more) segments are combined to form the gene sequences that form the basis of both operators and
- in that the segment structures of the parents are inherited.

For chromosome types 1 and 2, the use of a crossover operator may result in the offspring having too few or too many genes, with the missing genes being found as surplus genes on the other chromosome. The supernumerary genes are removed and attached to the chromosome end of the other offspring, whereby they are integrated into the last segment.

5 Application-related Genetic Operators in GLEAM

GLEAM contains an interface for the integration of application-related genetic operators. Their operator code is negative and can be specified in the operator list of a group. Mutation rates cannot be specified in the EVO file for application-related mutations. The corresponding SW is included in the `appl` package in the module `appl_go.c` in the form of two functions: `do_appl_mut()` for mutations and `do_appl_xo()` for crossover operators. The former is called if the operator code is negative and the entry for one for this operator group has the value 1. The operator code is passed to the function, so that in both functions the value range of valid codes starts at 0.

6 Literature

[1] W. Jakob: *HyGLEAM - User Manual*. Technical Paper, KIT, IAI, 2020.
See `HyGLEAM-Manual_V1.0.pdf`

[2] C. Blume, W. Jakob: *GLEAM - General Learning Evolutionary Algorithm and Method: Ein Evolutionärer Algorithmus und seine Anwendungen*. In German. Karlsruhe: KIT Scientific Publishing, 2009. doi: 10.5445/KSP/1000013553

[3] W. Jakob: *MOD-File-Documentation*. Technical Paper, V1.2, IAI, 2020.
See `MOD-File-Docu_V1.2.pdf`

[4] W. Jakob: *Eine neue Methodik zur Erhöhung der Leistungsfähigkeit Evolutionärer Algorithmen durch die Integration lokaler Suchverfahren*. Dissertation, Fak. f. Maschinenbau, Universität Karlsruhe, FZKA 6965, Forschungszentrum Karlsruhe, März 2004. Siehe auch: https://www.iai.kit.edu/~wilfried.jakob/HyGLEAM/

[5] L. Davis (ed): *Handbook of Genetic Algorithms.* V. Nostrand Reinhold, New York (1991)

[6] C. Bierwirth, D.C. Mattfeld, H. Kopfer: *On Permutation Representations for Scheduling Problems*. In: Voigt, H.-M. et al. (eds.): PPSN IV, LNCS 1141, Springer (1996) 310-318

[7] W. Jakob, A. Quinte, K.-U. Stucky, W. Süß: *Fast Multi-objective Scheduling of Jobs to Constrained Resources Using a Hybrid Evolutionary Algorithm*. In: G. Rudolph et al. (eds.): Conf. Proc. of Parallel Problem Solving from Nature X (PPSN 2008), LNCS 5199, Springer-Verlag, Berlin, 2008, S.1031-1040 doi: 10.1007/978-3-540-87700-4_102