

RFreak

An R Package for Evolutionary Computation

Robin Nunkesser

Department of Computer Science, TU Dortmund

Statistical Computing 2008

Outline

- 1 Introduction
 - Motivation
 - Evolutionary Computation and FrEAK
 - The RFreak Package
- 2 Schoolbook Example
 - $(1 + 1)$ EA on OneMax
- 3 Application Examples
 - Genetic Association Studies
 - Robust Regression

Last Year's Talk...

- Title: “Evolutionary Computation for Problems in Computational Statistics”
- One slide said: “Adaptions to easily access FrEAK from R with rJava”
- Now: R Package resulting from this

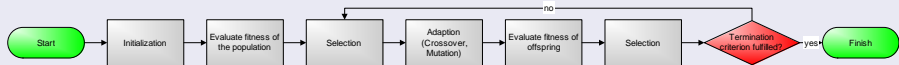
Evolutionary Computation (EC)

- General term for the usage of search heuristics inspired by natural evolution
 - possible solutions are represented by *individuals*
 - a set of individuals (a *population*) undergoes variation (*crossover* and *mutation*)
 - the *fitness* of the individuals is evaluated
 - a new generation is derived after a *selection* process
- Black-box optimization
- Often used for problems that are not easy to solve by conventional methods
 - Combinatorial optimization
 - Learning
 - Genetic association studies, robust regression, evolutionary clustering, time series modeling. . .

Layout of Algorithms for Evolutionary Computation

- ① Create an initial random population.
- ② Evaluate the fitness values of the population.
- ③ Perform the following steps on the current generation:
 - ① Select individuals in the population based on a selection scheme.
 - ② Adapt the selected individuals.
 - ③ Evaluate the fitness value of the adapted individuals.
 - ④ Select adapted individuals for the next generation according to a selection scheme.
- ④ If the termination criterion is fulfilled, then output the final population. Otherwise, set the next generation as current and go to step 3.

Graphical Representation



The modular view of FrEAK

LEGOLAND – LEGOLAND Deutschland

http://www.legoland.de/ | legoland günzburg

ALLE PARKS | WWW.LEGO.COM | LEGOLAND FERIENDORF | HILFE & SERVICES | VERZEICHNIS | SUCHEN

LEGOLAND DEUTSCHLAND

WELDEN GESUCHT

TICKETS KAUFEN | UNTERKUNFT

Eintrittskarten online kaufen

IM PARK | PARKBESUCH | JAHRESKARTE | GRUPPEN | UNTERNEHMEN

NEUIGKEITEN | EVENTS | ATTRAKTIONEN | PARKÜBERSICHT | SHOWS | SHOPS | ESSEN & TRINKEN

PAUSCHALANGEBOTE AB 75 €

1 Übernachtung + 1 Eintritt ab 75 Euro

Kinder (3 - 11 Jahre)
50% Ermäßigung

LEGOLAND® FERIENDORF

NEU ab Juni

FUSSBALL-ACTION

Fußballer gesucht!

Bitte beachten Sie, dass der Park vom 26. Mai bis 11. Juni sowie vom 16. bis 24. September 2008 jeweils montags, dienstags und mittwochs geschlossen ist. Alle weiteren Informationen dazu finden Sie in unserer [Übersicht](#).

NEU ab Juni 2008: [LEGOLAND Feriendorf](#).

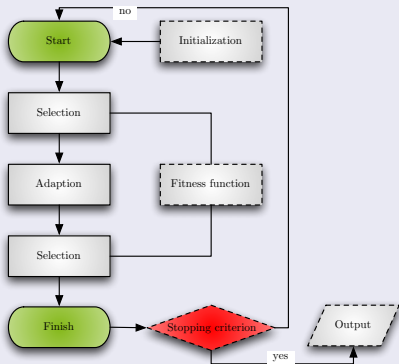
Eintrittskarten 2008

Eintrittskarten online bestellen

The modular view of FrEAK

- Base of R package: Free Evolutionary Algorithm Kit (FrEAK)
<http://sourceforge.net/projects/freak427/>
- Unified view of EC (De Jong, 2006): interchangeable modules

Graphical Representation



Remarks

- Modules with solid lines are part of FrEAK's *Algorithm Graph*
- Modules with dashed lines implicitly influence the process

The RFreak Package

- <http://cran.r-project.org/web/packages/RFreak/>
- Interface to use FrEAK from R via rJava
- Basic idea is to incorporate parts of FrEAK's GUI
 - `launchScheduleEditor(saveTo='schedule.freak', load=NULL)`
 - `executeSchedule(freakFile='schedule.freak')`
- Redirect output to R
- Possible extension: S4 wrapper classes \rightsquigarrow stronger decoupling from Java

(1 + 1) EA on OneMax

(1 + 1) EA

- 1 Choose $x \in \{0, 1\}^n$ uniformly at random.
- 2 Define y in the following way. Each bit of x is flipped independently of the other bits with probability $1/n$.
- 3 If $\text{fitness}(y) \geq \text{fitness}(x)$, replace x by y .
- 4 If the stopping criterion is fulfilled, then output the final population. Otherwise go to step 2.

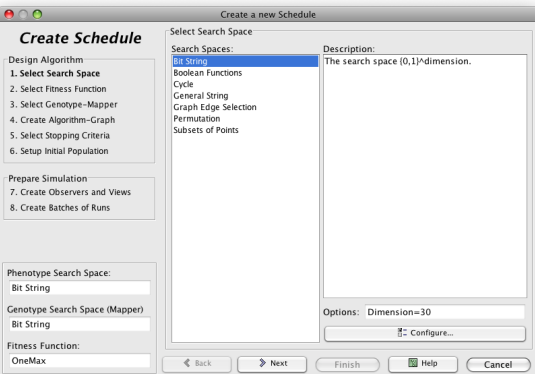
Definition

$$\text{OneMax}(x) := \sum_{i=1}^n x_i$$

- Main merit: Amenability to theoretical analysis

(1 + 1) EA on OneMax in RFreak

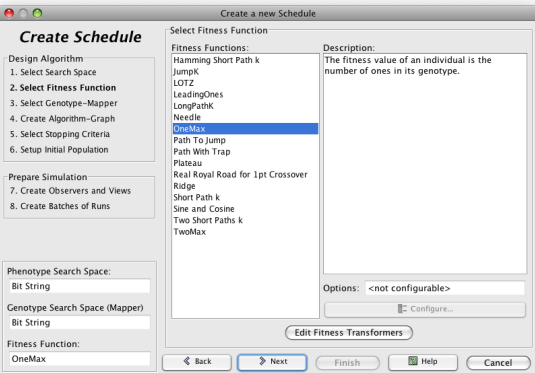
> launchScheduleEditor()



Behind the Scenes

- Essentially a Java BitSet
- Additional helper functions

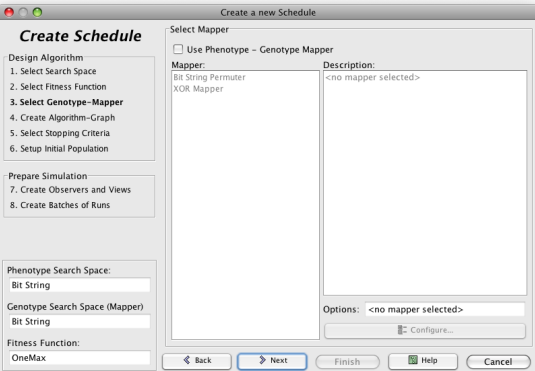
Choosing the Fitness Function



Behind the Scenes

- Provide a method evaluate
- Directly possible via method cardinality of class BitSet

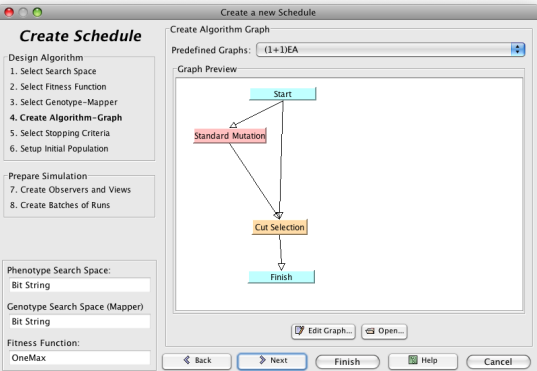
Choosing a Genotype-Mapper



Remark

- Not necessary here

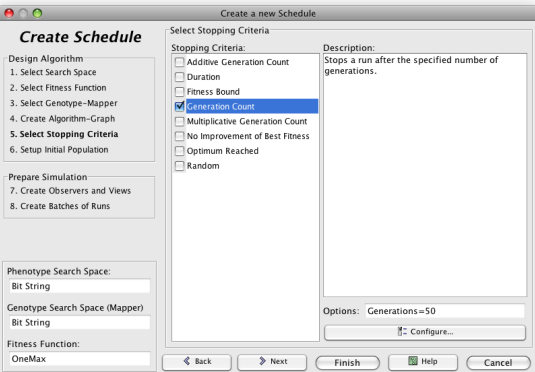
The Algorithm Graph



Remark

- Standard Mutation: Each bit flips with probability $1/n$
- Cut Selection: Select better individual or (in case of equal fitness) younger

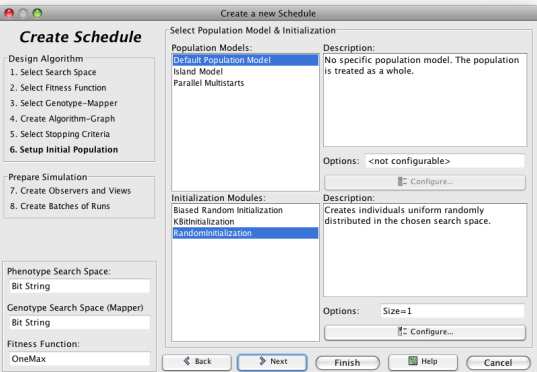
Stopping Criteria



Behind the Scenes

- Simple comparison

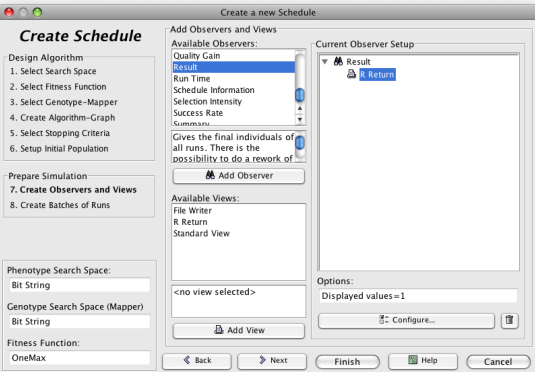
Population Model and Initialization



Behind the Scenes

- Choose n bits uniformly at random

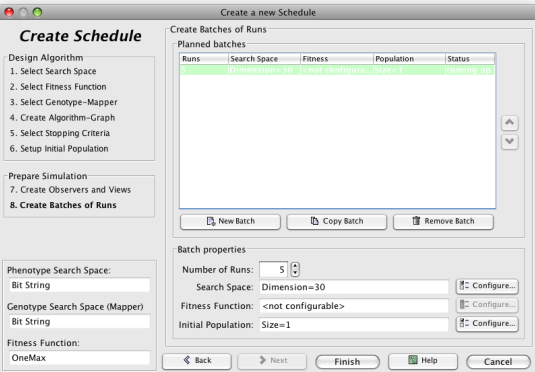
Observers and Views



Remark

- A special view is provided and preselected

Batches and Runs



Remark

- Batches are not supported

Result

```
> executeSchedule()
```

Result obtained from FrEAK:

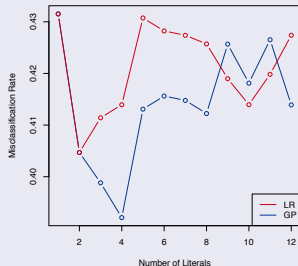
	Run	Generation	Obj. value	Individual
1	1	50	26	1011111011111111011111111011111
2	2	50	24	11100111110111110100111111111
3	3	50	22	10011110001110101111111111101
4	4	50	24	1101101011111111100110111111111
5	5	49	24	0110101111111111110111101111110

Genetic Association Studies

GPAS

- GP Algorithm
- Mainly for SNP data
 - GENICA
 - HapMap
- R functions:
 - GPASDiscrimination
 - GPASInteraction

MCR of LR and GPAS



MCR of discrimination for GENICA and HapMap data set

	GP Algorithm	Logic Regression	CART	Bagging	Random Forests
GENICA	0.392	0.405	0.429	0.457	0.450
HapMap	0.011	0.144	0.356	0.022	0.011

Example

```
> data(data.logicfs)
> GPASDiscrimination(cl.logicfs,data.logicfs)
```

Result obtained from FrEAK:

	Run	Gen.	Obj. 1	Obj. 2	Obj. 3	Individual
10	1	1319	180	104	-2	(SNP4==3) (SNP3==3)
12	1	1185	197	32	-2	(SNP4!=1) (SNP5!=1)
14	1	786	193	61	-2	(SNP4!=1) (SNP5==3)
16	1	537	175	114	-2	(SNP4==3) (SNP5==3)
21	1	186	100	200	-2	((SNP4==3) & (SNP2==1))
22	1	167	119	183	-2	((SNP4==3) & (SNP2!=3))
24	1	154	178	107	-2	(SNP4==3) (SNP6==1)
28	1	134	138	161	-1	(SNP4==3)

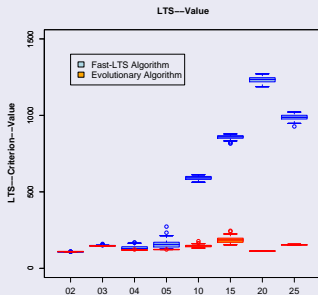
Another example: (SNP1==3) | ((SNP4==3) & (SNP2==1)) |
((SNP3==3) & (SNP5==3) & (SNP6==1))

Robust Regression

- Subset of observations is suitable for many robust regression methods

$$\hat{\beta}_{\text{LTS}} = \arg \min_{\hat{\beta} \in \mathbf{R}^p} \sum_{i=1}^h (r^2)_{i:n}$$

Obj value for an increasing number of regressors



R function

- LTSevol

Example

```
> data(stackloss)
> LTSevol(stackloss[,4],stackloss[,1:3],adjust=TRUE)
```

Result obtained from FrEAK:

	Run	Generation	Objective value	Individual
1	1	983	-2.932391	000000100010000010100

Chosen subset:

```
[1] 7 17 6 11 19 5 12 9 18 10 8 15 16
```

Coefficients:

```
[1] -37.32332647 0.74092106 0.39152672 0.01113454
```

Criterion:

```
[1] 2.932391
```






Summary

- Evolutionary Computation framework for R
- Modular layout for high reusability of code
- Growing number of application examples
- Easy to extend to further applications

Outlook

- More search spaces and fitness functions
- Wrap more functionality into S4 classes
- User wishes

Bibliography

-  De Jong, K. A., 2006. Evolutionary Computation: A Unified Approach. MIT Press.
-  Morell, O., Bernholt, T., Fried, R., Kunert, J., Nunkesser, R., 2008. An evolutionary algorithm for lts-regression: A comparative study. In: Proceedings of Compstat 2008. Accepted.
-  Nunkesser, R., 2008. Rfreak—an r package for evolutionary computation. Tech. rep., SFB 475, Technische Universität Dortmund.
-  Nunkesser, R., Bernholt, T., Schwender, H., Ickstadt, K., Wegener, I., 2007. Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. *Bioinformatics* 23 (24), 3280–3288.
-  Rousseeuw, P. J., 1984. Least median of squares regression. *Journal of the American Statistical Association* 79, 871–880.