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...
Introduction

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

Start Initialization Evaluate fitness of the population Selection Adaption (Crossover, Mutation) Evaluate fitness of offspring Selection Termination criterion fulfilled? yes Finish

Robin Nunkesser
RFreak
Statistical Computing 2008
The modular view of FrEAK
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/](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→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 to evaluate
- Directly possible via the method `cardinality` of class `BitSet`
Choosing a Genotype-Mapper

Remark
- Not necessary here
Introduction

Schoolbook Example

Application Examples

Summary and Outlook

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
Remark

- A special view is provided and preselected
Batches and Runs

Remark

- Batches are not supported
> executeSchedule()

Result obtained from FrEAK:

<table>
<thead>
<tr>
<th>Run</th>
<th>Generation</th>
<th>Obj. value</th>
<th>Individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>50</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10111101111111101111111111011111</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>50</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>111001111101111101001111111111</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>50</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100111100011101011111111111101</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>50</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11011010111111111100110111111111</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>49</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>011010111111111110111101111110</td>
</tr>
</tbody>
</table>
Genetic Association Studies

**GPAS**

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

### MCR of discrimination for GENICA and HapMap data set

<table>
<thead>
<tr>
<th></th>
<th>GP Algorithm</th>
<th>Logic Regression</th>
<th>CART</th>
<th>Bagging</th>
<th>Random Forests</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENICA</td>
<td>0.392</td>
<td>0.405</td>
<td>0.429</td>
<td>0.457</td>
<td>0.450</td>
</tr>
<tr>
<td>HapMap</td>
<td>0.011</td>
<td>0.144</td>
<td>0.356</td>
<td>0.022</td>
<td>0.011</td>
</tr>
</tbody>
</table>
Example

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

Result obtained from FrEAK:

<table>
<thead>
<tr>
<th>Run</th>
<th>Gen.</th>
<th>Obj. 1</th>
<th>Obj. 2</th>
<th>Obj. 3</th>
<th>Individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>1319</td>
<td>180</td>
<td>104</td>
<td>-2</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1185</td>
<td>197</td>
<td>32</td>
<td>-2</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>786</td>
<td>193</td>
<td>61</td>
<td>-2</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>537</td>
<td>175</td>
<td>114</td>
<td>-2</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>186</td>
<td>100</td>
<td>200</td>
<td>-2</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>167</td>
<td>119</td>
<td>183</td>
<td>-2</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>154</td>
<td>178</td>
<td>107</td>
<td>-2</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>134</td>
<td>138</td>
<td>161</td>
<td>-1</td>
</tr>
</tbody>
</table>

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

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

Obj value for an increasing number of regressors

R function
- LTSevol
Example

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

Result obtained from FrEAK:

<table>
<thead>
<tr>
<th>Run</th>
<th>Generation</th>
<th>Objective value</th>
<th>Individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>983</td>
<td>-2.932391</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>000000100010000010100</td>
</tr>
</tbody>
</table>

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


