# Biased Random-Key Genetic Algorithms

## Algoritmos Genéticos de Chaves Aleatórias Viciadas

Jonatas B. C. Chagas [1]

[1] Departamento de Computação
Universidade Federal de Ouro Preto

8 de dezembro de 2017

Este material foi desenvolvido baseando-se em
`http://mauricio.resende.info/talks/2012-09-CLAIO2012-brkga-tutorial-both-days.pdf`

# Summary

1. Genetic algorithms (GAs)
2. Random-key genetic algorithms (RKGAs)
3. Biased random-key genetic algorithm (BRKGAs)

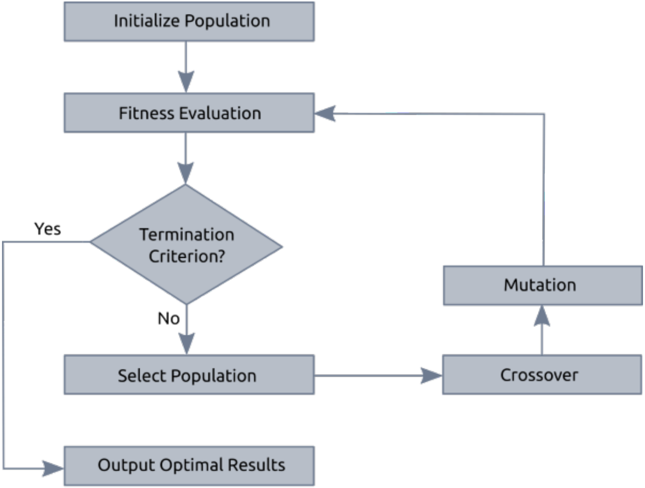
# Genetic algorithms

- Genetic algorithms (GAs) are metaheuristics inspired by the process of natural selection
- GAs evolve population of individuals (solutions) applying Darwin's principle of survival of the fittest
- A GA maintains a population of candidate individuals (solutions) for the problem at hand, and makes it evolve by iteratively applying a set of **stochastic operators** (selection, recombination and mutation)
  - selection: replicates the most successful individuals found in a population at a rate proportional to their relative quality
  - recombination (crossover): decomposes two or more distinct individuals and then randomly mixes their parts to form novel solutions
  - mutation: randomly perturbs a candidate individual

- Encoding principles (gene, chromosome)
- Initialization procedure (creation)
- Selection of parents (reproduction)
- Genetic operators (mutation, recombination)
- Evaluation function (environment)
- Termination condition

# Genetic algorithms

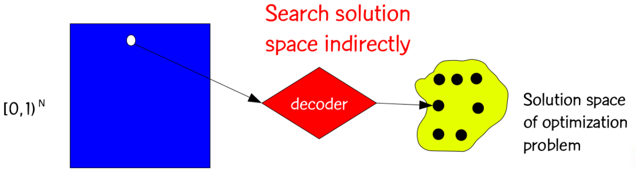
## Evolutionary cycle



# Random-key genetic algorithms

## Introduction

- Introduced by Bean (1994)[1] for sequencing problems
- A random-key is a real random number in the continuous interval [0,1)
- Individuals (solutions) of optimization problems can be encoded by random-keys
- Individuals are strings of real-valued numbers (random-keys)
- A decoder is a deterministic algorithm that takes a vector of random-keys as input and outputs a solution of the optimization problem




[1]Bean, J. C. (1994). Random-key genetic algorithms for sequencing and optimization. ORSA journal on computing, 6(2), 154-160.

- Bean (1994)[1] proposed decoders based on sorting the random-key vector to produce a sequence
- Encoding:

$$s = \langle \underset{1}{0.25}, \underset{2}{0.19}, \underset{3}{0.67}, \underset{4}{0.05}, \underset{5}{0.89} \rangle$$

- Decode by sorting vector of random-keys:

$$s' = \langle \underset{4}{0.05}, \underset{2}{0.19}, \underset{1}{0.25}, \underset{3}{0.67}, \underset{5}{0.89} \rangle$$

- Therefore, the vector of random-keys:

$$s = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$

encodes the sequence: $\langle 4, 2, 1, 3, 5 \rangle$

[1] Bean, J. C. (1994). Random-key genetic algorithms for sequencing and optimization. ORSA journal on computing, 6(2), 154-160.

- Other decodings:
    - subset selection (select 3 of 5 elements)

- Other decodings:
  - subset selection (select 3 of 5 elements)

    encoding:

    $$s = \langle \underset{1}{0.099}, \underset{2}{0.216}, \underset{3}{0.802}, \underset{4}{0.368}, \underset{5}{0.658} \rangle$$

    decode by sorting vector of random-keys:

    $$s' = \langle \underset{1}{0.099}, \underset{2}{0.216}, \underset{4}{0.368}, \underset{5}{0.658}, \underset{3}{0.802} \rangle$$

    encodes the subset: $\{1, 2, 4\}$

# Random-key genetic algorithms
## Encoding/decoding principles

- Other decodings:
  - subset selection (select 3 of 5 elements)

    encoding:

    $$s = \langle \underset{1}{0.099}, \underset{2}{0.216}, \underset{3}{0.802}, \underset{4}{0.368}, \underset{5}{0.658} \rangle$$

    decode by sorting vector of random-keys:

    $$s' = \langle \underset{1}{0.099}, \underset{2}{0.216}, \underset{4}{0.368}, \underset{5}{0.658}, \underset{3}{0.802} \rangle$$

    encodes the subset: $\{1, 2, 4\}$
  - subset selection (select $k$ of 5 elements, where $0 \leq k \leq 5$)

- Other decodings:
  - subset selection (select 3 of 5 elements)

    encoding:

    $$s = \langle \underset{1}{0.099}, \underset{2}{0.216}, \underset{3}{0.802}, \underset{4}{0.368}, \underset{5}{0.658} \rangle$$

    decode by sorting vector of random-keys:

    $$s' = \langle \underset{1}{0.099}, \underset{2}{0.216}, \underset{4}{0.368}, \underset{5}{0.658}, \underset{3}{0.802} \rangle$$

    encodes the subset: $\{1, 2, 4\}$
  - subset selection (select $k$ of 5 elements, where $0 \leq k \leq 5$)

    encoding:

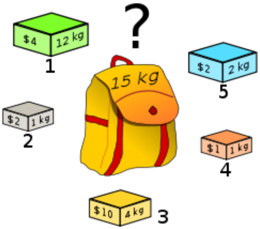
    $$s = \langle \underset{1}{0.82}, \underset{2}{0.12}, \underset{3}{0.54}, \underset{4}{0.89}, \underset{5}{0.26} \rangle$$

    decoding: if $s_i \geq 0.5$ then select $i$

    encodes the subset: $\{1, 3, 4\}$

- For some cases of complex decoders is necessary to adjust the chromosome in order to produce a feasible solution

- Knapsack problem:




encoding:

$$s = \langle \underset{1}{0.82}, \underset{2}{0.12}, \underset{3}{0.54}, \underset{4}{0.89}, \underset{5}{0.26} \rangle$$

decoding: if $s_i \geq 0.5$ then get item $i$

encodes the subset items: $\{1, 3, 4\}$

$\sum_{i=1,3,4} w_i \geq W \;\; (12 + 4 + 1 \geq 15)$

so, a **deterministic** strategy must be applied to make the solution viable

- Initial population is made up of $P$ random-key vectors, each with $N$ keys, each having a value generated uniformly at random in the interval [0,1).

$$s_1 = \langle \mathrm{KEY}_1^1, \mathrm{KEY}_2^1, \cdots, \mathrm{KEY}_N^1 \rangle$$
$$s_2 = \langle \mathrm{KEY}_1^2, \mathrm{KEY}_2^2, \cdots, \mathrm{KEY}_N^2 \rangle$$
$$\cdots$$
$$s_P = \langle \mathrm{KEY}_1^P, \mathrm{KEY}_2^P, \cdots, \mathrm{KEY}_N^P \rangle$$

$$\mathrm{KEY}_j^i \leftarrow \mathrm{rand}\,[0,1) \qquad \forall i = 1..P, \forall j = 1..N$$

- Two parents $a$ and $b$ are randomly selected from the entire $P$ population
- Mating is done using parameterized uniform crossover (Spears & DeJong , 1990)[2] for sequencing problems.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$

- For each gene, flip a biased coin to choose which parent passes the allele (key, or value of gene) to the child.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$
$$c = \langle$$

[2] Spears, W. M., & De Jong, K. D. (1995). On the virtues of parameterized uniform crossover. NAVAL RESEARCH LAB WASHINGTON DC.

- Two parents $a$ and $b$ are randomly selected from the entire $P$ population
- Mating is done using parameterized uniform crossover (Spears & DeJong , 1990)[2] for sequencing problems.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$

- For each gene, flip a biased coin to choose which parent passes the allele (key, or value of gene) to the child.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$
$$c = \langle 0.25,$$

[2] Spears, W. M., & De Jong, K. D. (1995). On the virtues of parameterized uniform crossover. NAVAL RESEARCH LAB WASHINGTON DC.

- Two parents $a$ and $b$ are randomly selected from the entire $P$ population
- Mating is done using parameterized uniform crossover (Spears & DeJong , 1990)[2] for sequencing problems.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$

- For each gene, flip a biased coin to choose which parent passes the allele (key, or value of gene) to the child.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$
$$c = \langle 0.25, 0.90,$$

[2] Spears, W. M., & De Jong, K. D. (1995). On the virtues of parameterized uniform crossover. NAVAL RESEARCH LAB WASHINGTON DC.

- Two parents $a$ and $b$ are randomly selected from the entire $P$ population
- Mating is done using parameterized uniform crossover (Spears & DeJong , 1990)[2] for sequencing problems.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$

- For each gene, flip a biased coin to choose which parent passes the allele (key, or value of gene) to the child.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$
$$c = \langle 0.25, 0.90, 0.76,$$

[2] Spears, W. M., & De Jong, K. D. (1995). On the virtues of parameterized uniform crossover. NAVAL RESEARCH LAB WASHINGTON DC.

# Random-key genetic algorithms

## Selection of parents and recombination

- Two parents $a$ and $b$ are randomly selected from the entire $P$ population
- Mating is done using parameterized uniform crossover (Spears & DeJong , 1990)[2] for sequencing problems.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$

- For each gene, flip a biased coin to choose which parent passes the allele (key, or value of gene) to the child.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$
$$c = \langle 0.25, 0.90, 0.76, 0.05,$$

[2] Spears, W. M., & De Jong, K. D. (1995). On the virtues of parameterized uniform crossover. NAVAL RESEARCH LAB WASHINGTON DC.

# Random-key genetic algorithms

## Selection of parents and recombination

- Two parents $a$ and $b$ are randomly selected from the entire $P$ population
- Mating is done using parameterized uniform crossover (Spears & DeJong , 1990)[2] for sequencing problems.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$

- For each gene, flip a biased coin to choose which parent passes the allele (key, or value of gene) to the child.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$
$$c = \langle 0.25, 0.90, 0.76, 0.05, 0.89 \rangle$$

[2] Spears, W. M., & De Jong, K. D. (1995). On the virtues of parameterized uniform crossover. NAVAL RESEARCH LAB WASHINGTON DC.

# Random-key genetic algorithms

## Selection of parents and recombination

- Two parents $a$ and $b$ are randomly selected from the entire $P$ population

- Mating is done using parameterized uniform crossover (Spears & DeJong , 1990)[2] for sequencing problems.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$

- For each gene, flip a biased coin to choose which parent passes the allele (key, or value of gene) to the child.

$$a = \langle 0.25, 0.19, 0.67, 0.05, 0.89 \rangle$$
$$b = \langle 0.63, 0.90, 0.76, 0.93, 0.08 \rangle$$
$$c = \langle 0.25, 0.90, 0.76, 0.05, 0.89 \rangle$$

If every random-key array corresponds to a feasible solution: Mating always produces feasible offspring.

[2] Spears, W. M., & De Jong, K. D. (1995). On the virtues of parameterized uniform crossover. NAVAL RESEARCH LAB WASHINGTON DC.

- **No mutation**: mutants are used instead (they play same role as mutation in GAs ... help escape local optima)
- A mutant is a new individual generated uniformly at random in the interval [0,1)

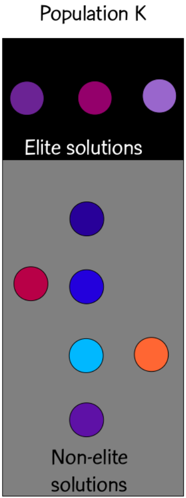
$$m = \langle \mathrm{KEY}_1^m, \mathrm{KEY}_2^m, \cdots, \mathrm{KEY}_N^m \rangle$$

$$\mathrm{KEY}_j^m \leftarrow \mathrm{rand}\,[0,1) \qquad\qquad \forall j = 1..N$$

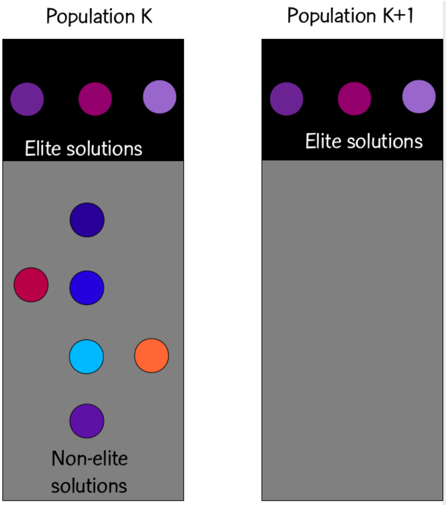
# Random-key genetic algorithms

## Next generation

- At the $k$-th generation, compute the cost of each solution and partition the individuals into two sets:
  - elite individuals ($P_e$)
  - non-elite individuals ($\overline{P}_e = P \setminus P_e$)
- Elite set should be smaller of the two sets and contain best individuals, i.e., $|P_e| < |P|/2$



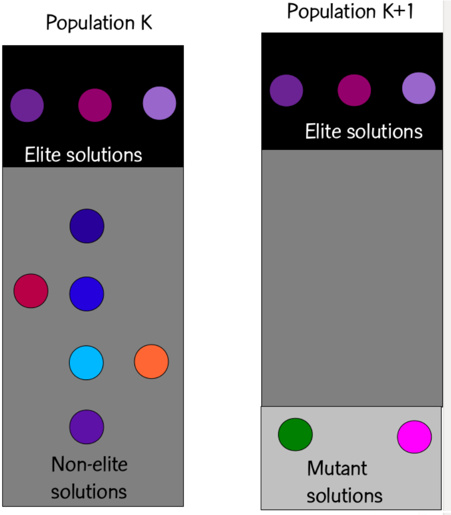
- Copy elite individuals $P_e$ from population $k$ to population $k+1$



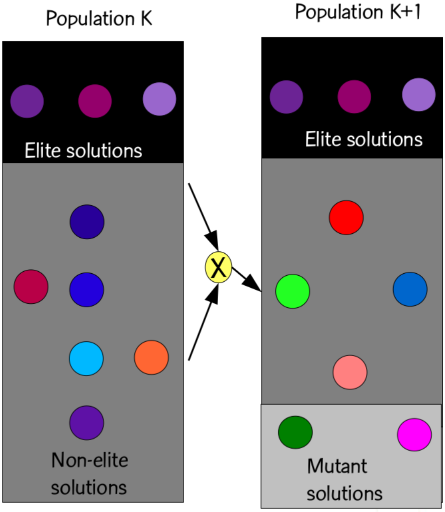
# Random-key genetic algorithms

## Next generation

- Copy elite individuals $P_e$ from population $k$ to population $k+1$
- Add $|P_m|$ random individuals (mutants) to population $k+1$



- Copy elite individuals $P_e$ from population $k$ to population $k+1$
- Add $|P_m|$ random individuals (mutants) to population $k+1$
- While $k$+1-th population $< P$ **do**
  - Use any two individuals in population $k$ to produce child in population $k$+1. Mates are chosen at random.



- A biased random-key genetic algorithm (BRKGA) is a random-key genetic algorithm (RKGA).
- BRKGA and RKGA differ in how mates are chosen for crossover and how parameterized uniform crossover is applied.

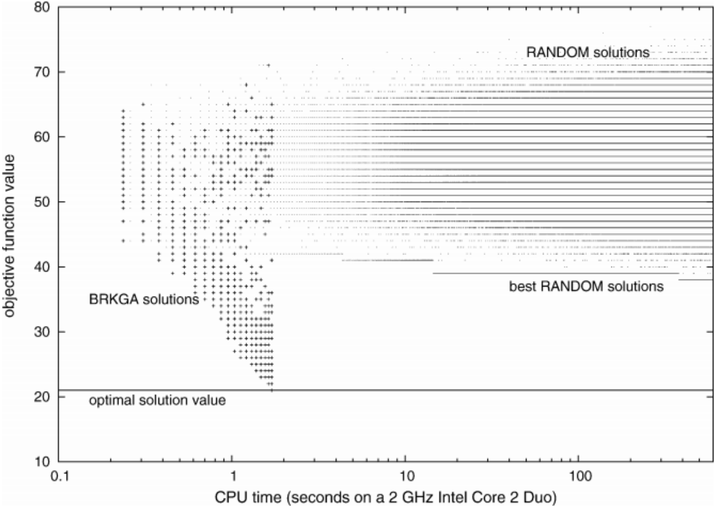
- A parent $a$ is randomly selected from the elite population $P_e$
- A parent $b$ is randomly selected from the non-elite population $\overline{P}_e$ **or** is randomly selected from entire population $P$
- For $i = 1, ..., n$, the $i$-th allele $c_i$ of the offspring $c$ takes on the value of the $i$-th allele $a_i$ of the elite parent $a$ with probability $\rho_e$ and the value of the $i$-th allele $b_i$ of the non-elite parent $b$ with probability $1 - \rho_e$

| | | | | |
|---|---|---|---|---|
| Parent $a$ | 0.32 | 0.77 | 0.53 | 0.85 |
| Parent $b$ | 0.26 | 0.15 | 0.91 | 0.44 |
| Random number | 0.58 | 0.89 | 0.68 | 0.25 |
| $\rho_e = 0.7$ | < | > | < | < |
| Offspring $c$ | 0.32 | 0.15 | 0.53 | 0.85 |

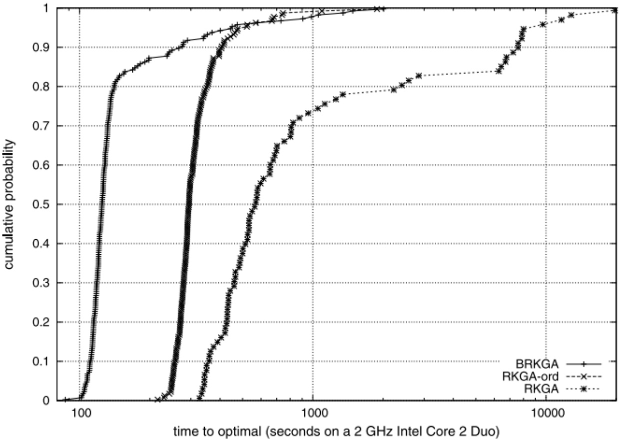
- In this way, the offspring is more likely to inherit characteristics of the elite parent than those of the non-elite parent.

# Biased random-key genetic algorithm

## Random solutions vs BRKGA solutions



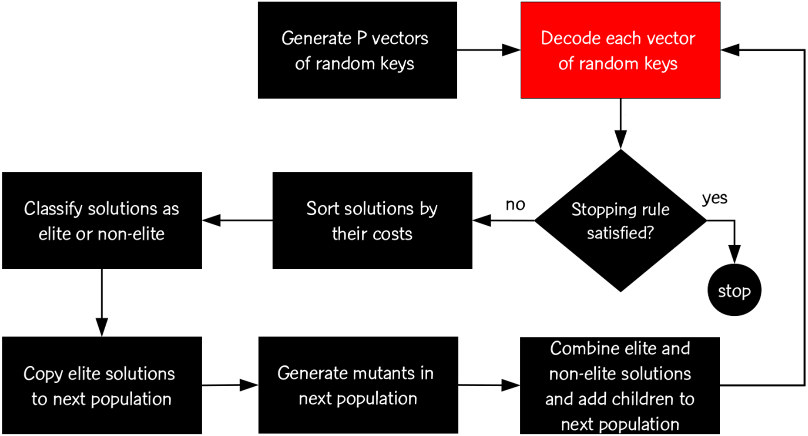

Comparing a BRKGA with a random multistart heuristic on an instance of a covering by pairs problem




Time to target plots compare running times needed to find the optimal solution of a 220 element covering by pairs problem with a BRKGA and two variants of a RKGA

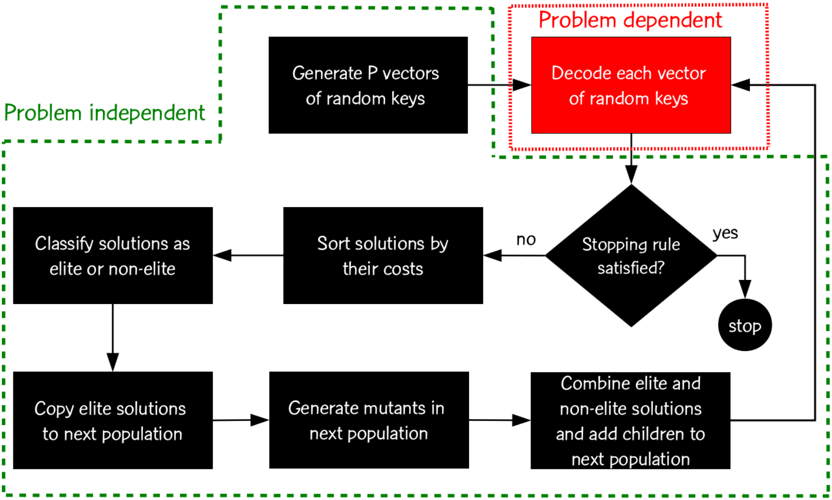
RKGA-ord is similar to a RKGA except that the offspring inherit the allele of the better fit of the two parents with probability $\rho_e$

# Biased random-key genetic algorithm
## Framework



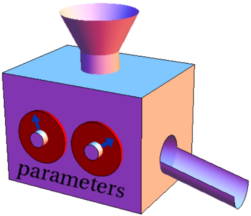
# Biased random-key genetic algorithm

## Framework



# Biased random-key genetic algorithm

## Parameters



| Parameter | Description | Recommended value |
|---|---|---|
| $P$ | size of population | $P = aN$, where $1 \leq a \in \mathbb{R}$ is a constant and $N$ is the length of the chromosome |
| $P_e$ | size of elite population | $0.10P \leq P_e \leq 0.25P$ |
| $P_m$ | size of mutant population | $0.10P \leq P_m \leq 0.30P$ |
| $\rho_e$ | elite allele inheritance probability | $0.5 < \rho_e \leq 0.8$ |
| $STOP$ | stopping criterion | e.g. time, # generations, solution quality, # generations without improvement |

- Efficient and easy-to-use object oriented application programming interface (API) for the algorithmic framework of BRKGA
- Download: `http://mauricio.resende.info/src/brkgaAPI/`

# References


- Resende, M. G. (2012). Biased random-key genetic algorithms: A tutorial
  http://mauricio.resende.info/talks/2012-09-CLAIO2012-brkga-tutorial-both-days.pdf
- Gonçalves, J. F., & Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. Journal of Heuristics, 17(5), pp. 487-525.