Ernane Rosa Martins
(Organizador)

Pesquisa Operacional e sua Atuação Multidisciplinar

Atena Editora
Ano 2019

Ernane Rosa Martins

(Organizador)

Pesquisa Operacional e sua Atuação Multidisciplinar

Atena Editora

2019

Editora Executiva: Profª Drª Antonella Carvalho de Oliveira
Diagramação: Karine de Lima
Edição de Arte: Lorena Prestes
Revisão: Os Autores

Dados Internacionais de Catalogação na Publicação (CIP)
(eDOC BRASIL, Belo Horizonte/MG)

P474 Pesquisa operacional e sua atuação multidisciplinar [recurso eletrônico] / Organizador Ernane Rosa Martins. – Ponta Grossa, PR: Atena Editora, 2019.

Formato: PDF
Requisitos de sistema: Adobe Acrobat Reader
Modo de acesso: World Wide Web
Inclui bibliografia
ISBN 978-85-7247-478-8
DOI 10.22533/at.ed.788191107

1. Pesquisa operacional. I. Martins, Ernane Rosa.
CDD 658.51

Elaborado por Maurício Amormino Júnior – CRB6/2422

Atena Editora
Ponta Grossa – Paraná - Brasil
www.atenaeditora.com.br
contato@atenaeditora.com.br

Atena Editora
Ano 2019

APRESENTAÇÃO

A Pesquisa Operacional (PO) utiliza a matemática, a estatística e a computação para auxiliar na solução de problemas reais, com foco na tomada das melhores decisões nas mais diversas áreas científicas e de atuação humana, buscando otimizar e melhorar suas performances. Através do uso de técnicas de modelagem matemática e eficientes algoritmos computacionais, a PO vem cada vez mais atuando na análise dos mais variados aspectos e situações de problemas complexos em demandas de inúmeras áreas, principalmente por conta de sua flexibilidade de aplicação e interação multidisciplinar, permitindo a tomada de decisões efetivas e a construção de sistemas mais produtivos.

Esta obra reúne importantes trabalhos que envolvem o uso de PO, realizados em diversas instituições de ensino do Brasil, abordando assuntos atuais e relevantes, tais como: modelos matemáticos; otimização multiobjectivo; heurísticas; algoritmos; otimização geométrica; metodologia SODA; soft systems methodology; strategic choice approach; procedimentos metodológicos de análise estatística; jogos cooperativos; algoritmos genéticos; método VIKOR; regressão linear múltipla; algoritmos de aprendizado de máquina; análise de decisão multicritério e composição probabilística de preferências.

A importância desta coletânea está na excelência dos trabalhos apresentados e na contribuição dos seus autores em temos de experiências e vivências. A socialização destes estudos no meio acadêmico, permite ampla análise e inúmeras discussões sobre diversos assuntos pertinentes referentes a atuação multidisciplinar da PO. Por fim, agradeço a todos que contribuíram na construção desta belíssima obra e desejo a todos os leitores, boas reflexões sobre os assuntos abordados.

Ernane Rosa Martins

SUMÁRIO

CAPÍTULO 9

BIASED RANDOM-KEY GENETIC ALGORITHM ACCORDING TO LEVY DISTRIBUTION FOR GLOBAL OPTIMIZATION

Mariana Alves Moura
Universidade Federal de Pernambuco
Recife – Brasil

Ricardo Martins de Abreu Silva
Universidade Federal de Pernambuco
Recife – Brasil

ABSTRACT: Biased Random-Key Genetic Algorithm (BRKGA) is an evolutionary metaheuristic that consists of representing solutions of a problem as real-key vectors defined by randomly generated values in the continuous interval [0,1) and uses a deterministic decoder to map these vectors into feasible solutions to the problem. In this work, an improvement to the algorithm was proposed from the use of the Levy distribution to generate its random keys and the addition of a local search procedure. The new approach was validated using a set of global continuous optimization benchmark functions widely used in the literature, whose global minimum values are known. The results obtained by the new approach was quite competitive in relation to the literature, both in terms of quality and performance.

KEYWORDS: Global Optimization. Genetic Algorithms. Levy Distribution.

RESUMO: O Algoritmo Genético de Chaves Aleatórias Viciadas, do inglês Biased Random-Key Genetic Algorithm (BRKGA) é uma metaheurística evolucionária que consiste em representar soluções de um problema como vetores reais definidos por valores gerados aleatoriamente no intervalo contínuo [0,1) e usa um decodificador determinístico para mapear esses vetores em soluções viáveis para o problema. Neste trabalho, uma melhoria para o algoritmo foi proposta a partir do uso da distribuição de Levy para gerar suas chaves aleatórias e da adição de um procedimento de busca local. A nova abordagem foi validada utilizando um conjunto de funções de benchmark de otimização global contínua amplamente utilizadas na literatura, cujos valores mínimos globais são conhecidos. Os resultados obtidos pela nova abordagem foram bastante competitivos em relação à literatura, tanto em termos de qualidade quanto em desempenho.

PALAVRAS-CHAVE: Otimização Global. Algoritmos Genéticos. Distribuição de Levy.

1 I INTRODUCTION

Many real problems can be mathematically modeled, always raising the interest of researchers in developing efficient software solutions to solve them.

A very common subclass of problems, which occurs for example in many practical engineering applications and generally generate rather difficult to solve modeling, are the problems in which the system to solve it must meet certain criteria, usually subject to limited resources.

The values given to these resources are called **optimization variables**. When the performance or cost of the system is described as a non-convex mathematical function that has more than one local optimum value, the problem of writing algorithms that can find the best among the local optimum values (or global optimum) is known as **global optimization problem**.

The objective of global optimization is to find the optimum (maximum or minimum) value of a mathematical function named **Objective Function**. An objective function can be defined as $f : S \subset \mathbb{R}^n \to \mathbb{R}$ subject to certain constraints, where represents the set of feasible points in the search space (JAMIL & YANG, 2013). When a problem is restricted only to the upper and lower bounds of its variables, we can call it **global optimization problem with box constraints**. These problems can be formally defined as:

$$\text{min/max } f(x) \quad (1)$$

subject to

$$l_i \leq x_i \leq u_i, \ \ i = 1, \ldots, n \quad (2)$$

where x is the vector of decision variables, l_i and u_i represent the lower and upper bounds of each decision variable .

Several techniques are proposed in the literature to try to solve this type of problem. Many of them seek to always find the optimal solution, such as the dynamic programming (BELLMAN, 2013) and branch and bound (LAWLER & WOOD, 1966), but they are limited to small and less complex problems. Other techniques are based on approximate methods, whose objective is to find good results that approach the optimum, but do not guarantee optimality. Another alternative that has been widely used in the area of optimization is heuristics and metaheuristics, which try to find quality solutions guided by some idea, in a good or acceptable computational time. Among the latter, we can cite a few examples, such as Simulated Annealing (KIRKPATRICK et al., 1983), GRASP (FEO & RESENDE, 1995) and Genetic Algorithms (GOLDBERG, 2006).

In this work we will use a variation of the Genetic Algorithm, called Biased Random-Key Genetic Algorithm (BRKGA), proposed by Gonçalves & Resende (2011) for which a modification will be introduced in the random key generator, from the use of the Levy distribution.

2 I BIASED RANDOM-KEY GENETIC ALGORITHM

BRKGA is a metaheuristic proposed by Gonçalves & Resende (2011) as a variant of Random Key Genetic Algorithm (RKGA), described by Bean, (1994), which consists of representing solutions as a vector of n random keys in which each key is a real number defined by a randomly generated value in the continuous interval [0,1) and uses a deterministic decoder that maps this vectors of random keys in solutions of the optimization problem.

The main difference between the RKGA and the BRKGA is in the selection of the parents for the crossover. While in RKGA the two parents are randomly selected from the entire population p, in BRKGA one parent is always selected from the elite set (pe) while the other parent is selected from the non-elite set (p-pe), or in some cases, of the entire population. The repetition of a parent in the selection for the crossover is allowed, so an individual can produce more than one offspring. Since pe<p/2 is required, the probability of an elite individual being selected for mating is (1/pe) greater than that of a non-elitist individual (1/p-pe) and therefore, elite individuals are more likely to pass on their characteristics to later generations.

BRKGA, like the RKGA, uses the uniformly parameterized crossing method of Spears & De Jong (1995). Thus, as in BRKGA one parent (α) is always selected from the elite set, and knowing that the probability of each parent gene a to be transmitted to the offspring is given by the parameter $p_{\alpha}>1/2$, the offspring will always be more likely to inherit the genes of the elitist parent, while in RKGA this is not a rule.

BRKGA is based on a metaheuristic framework for general purposes, in which there is a clear division between a dependent part and a non-dependent part of the problem. In this framework, the independent part of the algorithm has no knowledge about the problem to be solved, being limited to perform a search in the domain of the random keys. The BRKGA connection to the problem is made by the dependent part of the algorithm, in which a decoder produces the solutions to the problem from the random keys and calculates the suitability of these solutions. Thus, to specify a BRKGA heuristic for a given problem it is only necessary to define the representation of the chromosome and the decoder.

3 I USING LEVY DISTRIBITION ON BRKGA RANDOM-KEY GENERATOR

Levy distribution has properties that allow it several possibilities of symmetry and kurtosis, making it adaptable to different kinds of problems from the configuration of its four parameters described below:

- The parameter α, called law index, stability index or yet characteristic exponent and varies between 0 and 2. Different values of the parameter can represent different distributions, as can be seen in Figure 1 from the left. For example, when *α = 2*, a distribution takes the form of a Gaussian and when

$\alpha = 1$, its form corresponds to the Cauchy distribution.

- The parameter β, called the law asymmetry and varies between -1 and 1. When $\beta = 0$, the distribution is symmetric, if $\beta > 0$ is asymmetric to the right, and if $\beta < 0$ is asymmetric to the left. This modification can be verified graphically in Figure 1 from the right.
- The parameter γ or *c*, known as the scale parameter, can be any positive number and determines the spread of the distribution.
- The parameter δ or μ called the location parameter, and translates the distribution to the right if $\delta > 0$, and to the left if $\delta < 0$.
-

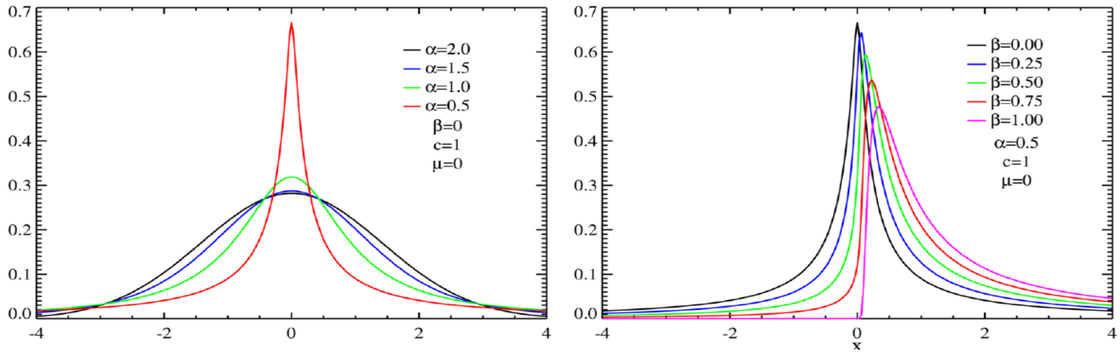


Figure 1: Variation of parameters α (to the left) and β (to the right) in Levy distribution.

The main motivation for choosing this distribution was its applicability in other works of the global optimization area, such as Lee & Yao (2004), Yang & Deb (2009), Yang & Deb (2010), Gandomi et al. (2013) and Yang & Deb (2014), among others, which propose search methods nature-inspired, simulating behaviors that follow properties of this distribution and presenting very promising results. Based on this observation and the study of the distribution, our hypothesis is that the main contribution to the good results of these methods is actually the use of Levy distribution.

Therefore, in this this work was developed and evaluated a variant of BRKGA that uses the Levy distribution to generate its random keys. It was verified if the replacing of uniform distribution by Levy distribution could improve the results of the algorithm applied to global optimization problems subject to box constraints.

4 I METHODOLOGY

This work was composed of 3 phases of development and 3 phases of experimental analysis that are described below.

4.1 Development

In the first phase of development a BRKGA Application Programming Interface (API) was developed in Python, based on the work previously proposed by Toso & Resende (2015).

In the second phase a version of the API was developed including a local search procedure for optimization problems in continuous spaces based on Hirsch et al. (2010) and Silva et al. (2013a), where the search space is discretized by transforming it into a grid using a discretization parameter h and the neighborhood of a specific solution x are the projections of the other grid points on a circumference with center in x and radius h.

In the third phase, a study was performed with the Levy distribution and the insertion of this distribution into BRKGA as a method used to generate the random keys of the algorithm. This modification was inserted in the traditional version and version with local search. For the development of this phase, we used Levy's variable-generator module *levy_stable.rvs*, which is available in the Python Scipy library, and it is only necessary to develop mechanisms for results to be generated within the upper and lower bounds of the decision variables. For this, the module operation was used in all the necessary situations.

Thus, we have four versions of the API, which will be named in this work as:

- **BRKGA1:** traditional BRKGA version, without local search and no Levy distribution;
- **BRKGA2:** version of BRKGA without local search but using Levy distribution;
- **BRKGA3:** version of BRKGA with local search and without Levy distribution;
- **BRKGA4:** version of BRKGA with local search and with Levy distribution.

An important modification that has also been made in the versions that use Levy distribution is in the space in which the random keys are generated. In traditional BRKGA we use the interval [0,1) to later transform into the solution by decoder. In Levy versions these keys are generated in a user-defined interval, which can be used, including the range of variables of the problem. Thus, the new version of the algorithm supports using Levy numbers directly as solution variables.

4.2 Experimental analysis

The first phase of the experimental analysis was characterized by the automatic tuning of the Levy distribution parameters for each function used in the experiments and also the tuning of the parameters of the four versions of the BRKGA using the *irace* library available on CRAN (Comprehensive R Archive Network).

An important observation perceived in this first phase was that the parameters of the distribution varied greatly for each function. From this information we can say that the choice of the parameters of the Levy distribution is very sensitive to the problem that will be solved, becoming a decisive factor for obtaining good results and performances by the algorithms.

The second phase corresponded to the comparison between the four BRKGA approaches to determine whether significant improvements have been achieved in relation to the traditional approach.

In the third phase, the best of the BRKGA versions was compared with an implementation of the C-GRASP algorithm proposed in Hirsch et al. (2007) and Hirsch et al. (2010) which is also applied to continuous global optimization problems and uses a local search procedure similar to what was inserted in BRKGA.

For the accomplishment of the experiments were used 27 global optimization benchmark functions widely used in the literature, available in Jamil & Yang (2013) and Hirsch et al. (2010), whose best solutions and objective function values are known. Each function was run 30 times for each algorithm, using the settings obtained by *irace* execution.

The optimality gap between the results and the known optimal values of the functions was defined by the function $GAP = |f(x) - f(x^*)|$, where x is the best current solution found by heuristics and is the known global minimum vector-solution.

As a quality stop criterion, we consider that a heuristic solved a certain problem when

$$GAP \leq \begin{cases} \epsilon, & if \quad f(x^*) = 0 \\ \epsilon \cdot f(x^*), & if \quad f(x^*) \neq 0 \end{cases} \tag{3}$$

where ϵ = *0.001*, as well as in Hirsch et al. (2010).

At the end, the mean (**Med**) value of the results of the functions and computational times was calculated, as well as the minimum (**Min**), maximum (**Max**), standard deviation (**S.D.**) and percentage of iterations that reached the quality stop criterion (**%QUAL**).

All experiments were run on an Ubuntu 14.04.5 LTS server with an Intel Xeon E5-2603 v3 processor, with 1.60GHz and 12 cores, 2TB HD and 32GB RAM.

5 I RESULTS AND DISCUSSION

5.1 Comparison between BRKGA approaches

For the analysis of the results, the functions were divided into groups according to their size and modality characteristics.

In terms of size, the functions were divided into small (2 decision variables), medium (4 decision variables) and large (10 decision variables).

Regarding the modality, we divide the functions into unimodal and multimodal. A function whose landscape has a single peak, that is, determining a single global optimum value, there being no other local optimum, it is called **unimodal function**. When a function has multiple peaks, it means that it has more than one local optimal value, which does not correspond to the global optimum. This type of function is called **multimodal function**.

Therefore, 6 groups were created:

- **Group 1:** Functions with 2 decision variables and unimodal;
- **Group 2:** Functions with 2 decision variables and multimodal;
- **Group 3:** Functions with 4 decision variables and unimodal;
- **Group 4:** Functions with 4 decision variables and multimodal;
- **Group 5:** Functions with 10 decision variables and unimodal;
- **Group 6:** Functions with 10 decision variables and multimodal.

The results of the analysis are summarized in Table 1.

Function Groups	GAP	Approaches			
		BRKGA1	BRKGA2	BRKGA3	BRKGA4
Group 1 (n = 2)	Min	0.00000	0.00000	0.00000	0.00000
UNIMODALS	Max	4.00206	0.00194	8.94870	0.00027
	Mean	0.40105	0.00058	0.89487	0.00004
	S.D.	1.20034	0.00053	2.99144	0.01653
	%QUAL	82.67	94.67	90.00	100.00
Group 2 (n=2)	Min	0.00018	0.00000	0.00000	0.00000
MULTIMODALS	Max	1.05594	1.04077	1.00001	0.00195
	Mean	0.08587	0.08317	0.12606	0.00024
	S.D.	0.27043	0.26687	1.00001	0.04413
	%QUAL	71.67	84.05	81.19	100.00
Group 3 (n = 4)	Min	0.00071	0.00059	0.00004	0.00003
UNIMODALS	Max	20.08117	0.56966	100.75711	0.00106
	Mean	2.62142	0.08015	12.59490	0.00042
	S.D.	6.60228	0.18564	10.03778	0.03251
	%QUAL	30.42	58.33	87.50	98.33
Group 4 (n=4)	Min	0.00059	0.00000	0.00002	0.00000
MULTIMODALS	Max	0.39462	0.05113	351.01688	0.00182
	Mean	0.10504	0.01101	31.91171	0.00063
	S.D.	0.13552	0.01714	18.73544	0.04266
	%QUAL	52.73	84.24	90.61	100.00
Group 5 (n = 10)	Min	0.01498	0.00083	0.00012	0.00033
UNIMODALS	Max	3.95e+12	4.46e+11	1.61e+10	5.52e+09
	Mean	5.65e+11	6.38e+10	2.3e+09	7.89e+08
	S.D.	1.38e+12	1.56e+11	1.27e+05	7.43e+04
	%QUAL	1.90	21.43	57.14	71.43
Group 6 (n = 10)	Min	0.00386	0.00013	0.00014	0.00062
MULTIMODALS	Max	1302.18217	24.96266	1.49e+06	21.25683
	Mean	236.95022	4.16527	2.49e+05	3.55018
	S.D.	478.29324	9.30088	1222.40893	4.61051
	%QUAL	12.22	76.67	65.00	70.00

Table 1: Comparison between BRKGA approaches.

As can be seen, the BRKGA4 version, which uses both the local search procedure and the Levy distribution in the generation of random keys, overcame the other versions in almost all executions, except in the execution of the larger multimodal functions (10 variables), in the which BRKGA2 version stood out. The traditional version, which does not present local search nor distribution of Levy presented the worst results in all groups of this analysis.

Another important observation in the results is the relevance of Levy's contribution to the algorithms. Of the two versions without local search, the version that uses the distribution has surpassed the traditional version. This has also occurred among versions that implement local search. It is also important to note that for some functions, the version that uses only the Levy distribution, without local search (BRKGA2) was able to overcome the version that only has local search (BRKGA3).

5.1.1 Performance and convergence

To analyze the performance and convergence of the BRKGA approaches, *tttPlots* graphs were used, some of which were attached in this section, as well as the computational times of the algorithm executions as support.

The *tttPlots* technique (Ribeiro & Rosseti, 2015) uses the computational times of the sample to generate and display the probability of finding the optimal or near optimal value at a given execution time.

To demonstrate this analysis, the graphs of a function of each of the groups defined in the section above were selected for each BRKGA version studied in this work.

The graphs can be analyzed in Figures 2, 3 and 4.

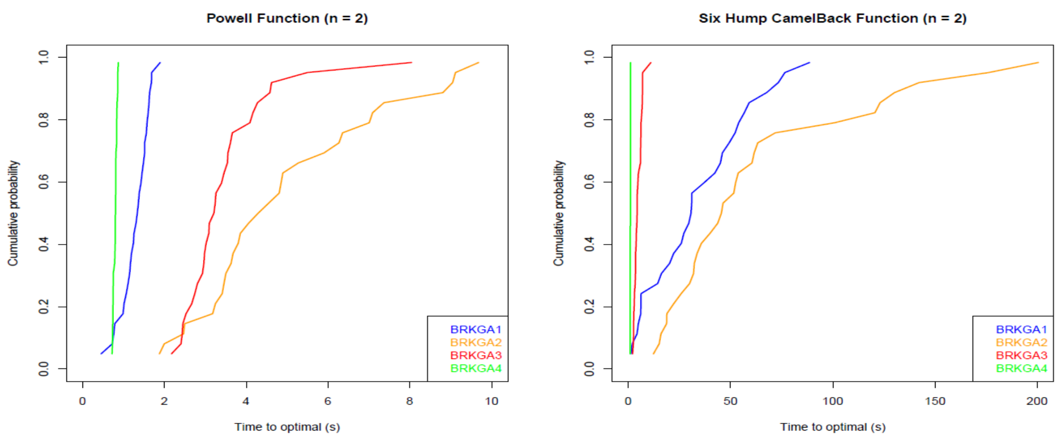


Figure 2: Examples of time to target plots to compare the BRKGA variants in small functions.

Figure 2: Examples of *tttPlots* to compare the BRKGA variants in small functions.

From the analysis of results and possible information that the BRKGA4 version is also the most important in terms of performance and convergence in most of the functions tested, obtaining a significant difference in relation to others.

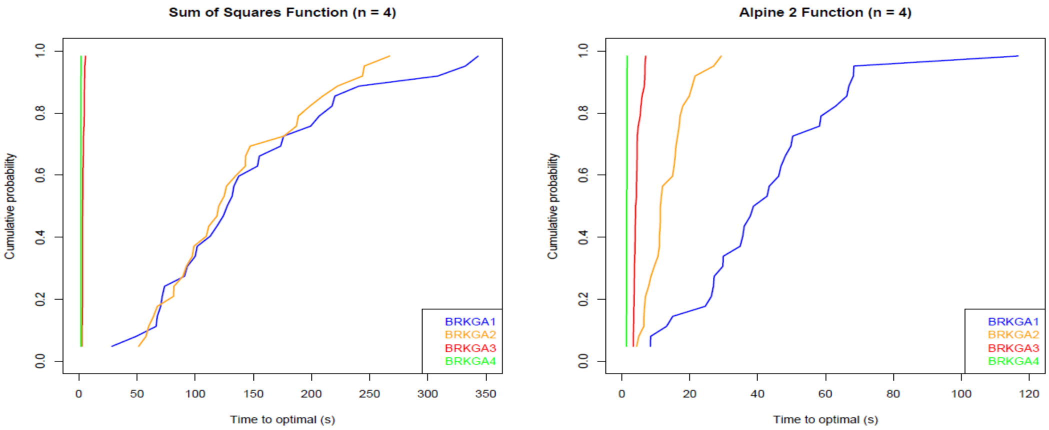


Figure 3: Examples of time to target plots to compare the BRKGA variants in median functions.

Figure 3: Examples of *tttPlots* to compare the BRKGA variants in median functions.

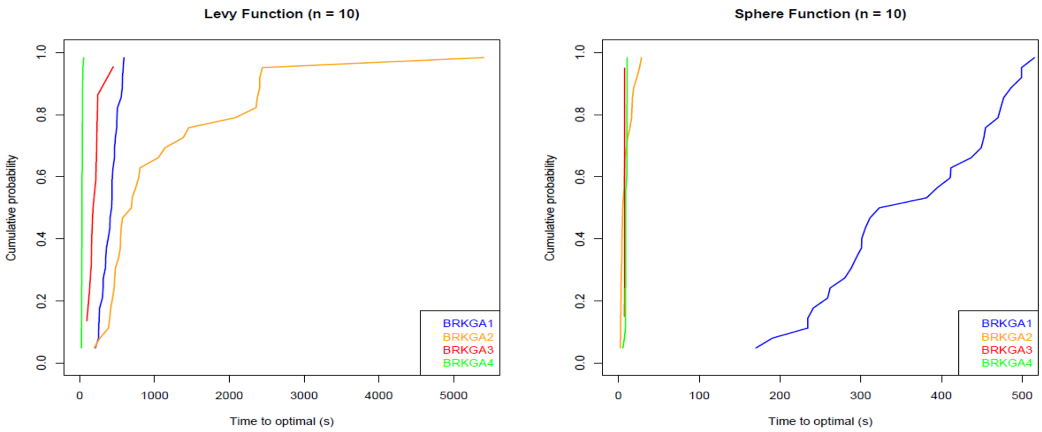


Figure 4: Examples of time to target plots to compare the BRKGA variants in big functions.

Figure 4: Examples of *tttPlots* to compare the BRKGA variants in large functions.

5.2 Comparison with the Continuous-GRASP algorithm

The second phase of the experiments consisted of comparing the BRKGA version that presented the best results in the previous section, with another technique from the literature, the Continuous GRASP or just C-GRASP.

This algorithm was initially proposed by Hirsch et al. (2007) and later improved in Hirsch et al. (2010). In this work, the implementation of this metaheuristic was used by Silva et al. (2017), which used Python embedded in C++ to implement the C-GRASP library proposed in Silva et al. (2013b), that proposes the use of C-embedded Python.

The parameters used to run the C-GRASP were adjusted according to the settings suggested in Hirsch et al. (2010).

We selected 10 functions for this analysis, for which the BRKGA4 achieved good results, achieving the quality stop criterion in 100% of executions. On the other hand, the C-GRASP, although it also presented good results, only in 50% of the analyzed functions the quality criterion was reached in the 30 executions. The results can be analyzed Table 2.

5.2.1 Performance and convergence

As in the analysis of the BRKGA approaches, we used *tttPlots* graphs together with the results of the computational times as support to compare the performance and convergence between the BRKGA4 and C-GRASP techniques in the functions experienced.

To exemplify the analysis performed, 3 functions of different sizes were chosen from the 10 tested. The *tttPlots* for these functions can be seen in Figure 5 (a), (b) and (c).

After analyzing the graphs and tables, we could verify that BRKGA4 was able to overcome C-GRASP in almost all functions in terms of performance and convergence, falling behind in only 3 of the 10 functions tested. Another interesting observation was that in the 4 largest functions tested, BRKGA4 stood out.

6 I CONCLUSION AND FUTURE WORK

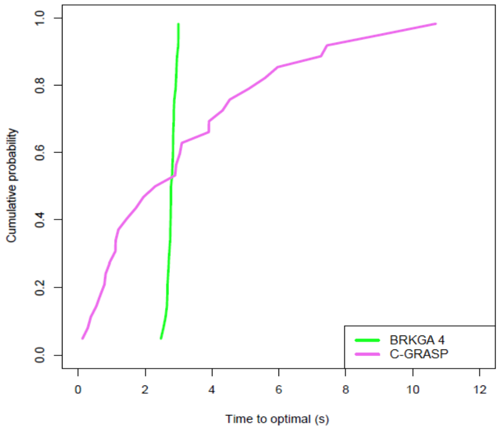
The purpose of this work was to propose a variation of the BRKGA framework, which consisted of changing its random number generator so that the random keys of the algorithm were generated using the Levy distribution. With the variations developed, four versions of BRKGA were generated so that the relevance of each change could be evaluated.

The algorithms were validated using 27 global optimization benchmark functions, widely used in the research area literature. The developed BRKGA approaches were compared with each other and also with the C-GRASP metaheuristic.

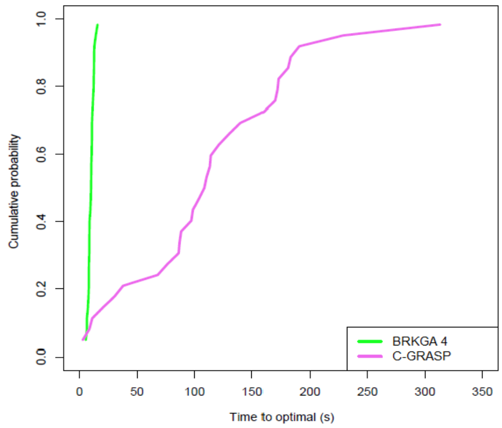
Table 2: Comparison between BRKGA4 and C-GRASP.

Function	n	Results	Approachs	
			C-GRASP	BRKGA4
Bohachevsky	2	Min	0.00003	0.00000
		Max	0.00093	0.00029
		Med	0.00047	0.00010
		S.D.	0.00029	0.00010
		%Qual	**100.00**	**100.00**
Booth	2	Min	0.00000	0.00000
		Max	0.00071	0.00001
		Med	0.00025	0.00000
		S.D.	0.00019	0.00000
		%QUAL	**100.00**	**100.00**
Branin	2	Min	0.39789	0.39789
		Max	0.39826	0.39789
		Med	0.39801	0.39789
		S.D.	0.00011	0.0000
		%QUAL	**100.00**	**100.00**
Colville	4	Min	0.00045	0.00015
		Max	0.00245	0.00100
		Med	0.00105	0.00059
		S.D.	0.00044	0.00020
		%QUAL	43.33	**100.00**
Perm0	4	Min	0.00007	0.00013
		Max	0.00100	0.00096
		Med	0.00049	0.00043
		S.D.	0.00028	0.00024
		%QUAL	**100.00**	**100.00**
Powersum	4	Min	0.00006	0.00011
		Max	0.00095	0.00097
		Med	0.00057	0.00062
		S.D.	0.00022	0.00022
		%QUAL	**100.00**	**100.00**
Rastrigin	10	Min	0.15991	0.00061
		Max	0.45480	0.00099
		Med	0.34767	0.00087
		S.D.	0.00010	0.00010
		%QUAL	0.00	**100.00**
Sum of Squares	10	Min	0.00351	0.00052
		Max	0.00924	0.00096
		Med	0.00605	0.00077
		S.D.	0.00136	0.00013
		%QUAL	0.00	**100.00**
Trid	10	Min	-124.66656	-209.97797
		Max	-124.66580	-209.79125
		Med	-124.66630	-209.81793
		S.D.	0.00015	0.04203
		%QUAL	0.00	**100.00**
Zakharov	10	Min	0.00085	0.00041
		Max	0.00370	0.00099
		Med	0.00263	0.00075
		S.D.	0.00059	0.00015
		%QUAL	3.33	**100.00**

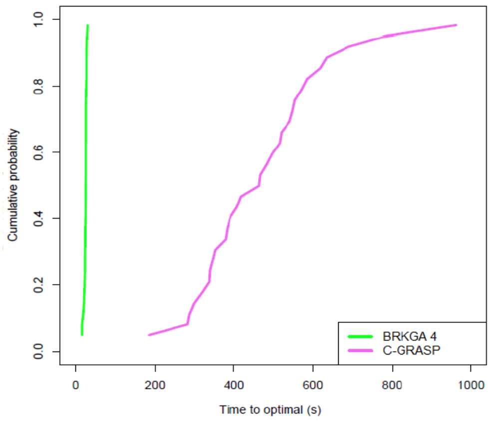
Table 2: Comparison between BRKGA4 and C-GRASP.



(a) Example of time to target plot to compare the BRKGA4 and C-GRASP algorithms in a small function (Bohachevsky Function, with n=2).



(b) Example of time to target plot to compare the BRKGA4 and C-GRASP algorithms in a median function (Colville Function, with n=4)



(c) Example of time to target plot to compare the BRKGA4 and C-GRASP algorithms in a small function (Zakharov Function, with n=10)

Figure 5: Example of tttPlots to compare the BRKGA4 and C-GRASP.

After the analysis of the experiments, we could conclude that the Levy distribution brought significant improvements for BRKGA in relation to the original version and even when applied to the version with local search, which already surpassed the original version. The modifications introduced significant improvements for the convergence of the algorithm to the global minimums, as well as for its performance.

Finally, a comparison of the best version of BRKGA with a metaheuristic selected from the literature, C-GRASP, was carried out in order to evaluate the relevance of this work to the area of global optimization research. As we observed in the experiments, BRKGA4 achieved results equal to or better than those of C-GRASP, both in quality and performance, validating the technique proposed in this work.

Future work includes:

- The use of the method on other kinds of problems for which BRKGA can also be applied;
- The implementation of improvements in local search, in order to further im-

prove the convergence and performance of the tool;

- Extend the use of the Levy distribution to the BRKGA API in C++ in order to get better performance.

REFERENCES

BEAN, James C. Genetic algorithms and random keys for sequencing and optimization. **ORSA journal on computing**, v. 6, n. 2, p. 154-160, 1994.

BELLMAN, Richard. **Dynamic programming**. Courier Corporation, 2013.

FEO, Thomas A.; RESENDE, Mauricio GC. Greedy randomized adaptive search procedures. **Journal of global optimization**, v. 6, n. 2, p. 109-133, 1995.

GANDOMI, Amir Hossein; YANG, Xin-She; ALAVI, Amir Hossein. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. **Engineering with computers**, v. 29, n. 1, p. 17-35, 2013.

GOLDBERG, David E. **Genetic algorithms**. Pearson Education India, 2006.

GONÇALVES, José Fernando; RESENDE, Mauricio GC. Biased random-key genetic algorithms for combinatorial optimization. **Journal of Heuristics**, v. 17, n. 5, p. 487-525, 2011.

HIRSCH, Michael J. et al. Global optimization by continuous grasp. **Optimization Letters**, v. 1, n. 2, p. 201-212, 2007.

HIRSCH, Michael J.; PARDALOS, Panos M.; RESENDE, Mauricio GC. Speeding up continuous GRASP. **European Journal of Operational Research**, v. 205, n. 3, p. 507-521, 2010.

JAMIL, Momin; YANG, Xin-She. A literature survey of benchmark functions for global optimization problems. **arXiv preprint arXiv:1308.4008**, 2013.

KIRKPATRICK, Scott; GELATT, C. Daniel; VECCHI, Mario P. Optimization by simulated annealing. **science**, v. 220, n. 4598, p. 671-680, 1983.

LAWLER, Eugene L.; WOOD, David E. Branch-and-bound methods: A survey. **Operations research**, v. 14, n. 4, p. 699-719, 1966.

LEE, Chang-Yong; YAO, Xin. Evolutionary programming using mutations based on the Lévy probability distribution. **IEEE Transactions on Evolutionary Computation**, v. 8, n. 1, p. 1-13, 2004.

RIBEIRO, Celso C.; ROSSETI, Isabel. tttplots-compare: A perl program to compare time-to-target plots or general runtime distributions of randomized algorithms. **Optimization Letters**, v. 9, n. 3, p. 601-614, 2015.

SILVA, Ricardo MA, et al. Generalized grasp for mixed-integer nonlinear optimization. In: **Proceedings of the 12th Metaheuristics International Conference**. 2017. p. 496–504.

SILVA, Ricardo MA et al. Biased random-key genetic algorithm for nonlinearly-constrained global optimization. In: **IEEE Congress on Evolutionary Computation**. 2013. p. 2201-2206.

SILVA, Ricardo MA et al. A Python/C library for bound-constrained global optimization with continuous GRASP. **Optimization Letters**, v. 7, n. 5, p. 967-984, 2013.

SPEARS, William M.; DE JONG, Kenneth D. **On the virtues of parameterized uniform crossover**. NAVAL RESEARCH LAB WASHINGTON DC, 1995.

TOSO, Rodrigo F.; RESENDE, Mauricio GC. A C++ application programming interface for biased random-key genetic algorithms. **Optimization Methods and Software**, v. 30, n. 1, p. 81-93, 2015.

YANG, Xin-She; DEB, Suash. Cuckoo search via Lévy flights. In: **Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on**. IEEE, 2009. p. 210-214.

YANG, Xin-She; DEB, Suash. Engineering optimisation by cuckoo search. **arXiv preprint arXiv:1005.2908**, 2010.

YANG, Xin-She; DEB, Suash. Cuckoo search: recent advances and applications. **Neural Computing and Applications**, v. 24, n. 1, p. 169-174, 2014.

SOBRE O ORGANIZADOR

Ernane Rosa Martins - Doutorado em andamento em Ciência da Informação com ênfase em Sistemas, Tecnologias e Gestão da Informação, na Universidade Fernando Pessoa, em Porto/Portugal. Mestre em Engenharia de Produção e Sistemas, possui Pós-Graduação em Tecnologia em Gestão da Informação, Graduação em Ciência da Computação e Graduação em Sistemas de Informação. Professor de Informática no Instituto Federal de Educação, Ciência e Tecnologia de Goiás - IFG (Câmpus Luziânia), ministrando disciplinas nas áreas de Engenharia de Software, Desenvolvimento de Sistemas, Linguagens de Programação, Banco de Dados e Gestão em Tecnologia da Informação. Pesquisador do Núcleo de Inovação, Tecnologia e Educação (NITE), certificado pelo IFG no CNPq.

Agência Brasileira do ISBN
ISBN 978-85-7247-478-8
9 788572 474788