

A Random Parameter Value Strategy for the Biased Random-Key Genetic Algorithms

Luisa Zambelli A. R. Vilela

Departamento de Engenharia Industrial
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de São Vicente, 225, Gávea, Rio de Janeiro, RJ, 22452-900
luzambelli@gmail.com

Luciana S. Pessoa

Departamento de Engenharia Industrial
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de São Vicente, 225, Gávea, Rio de Janeiro, RJ, 22452-900
lucianapessoa@puc-rio.br

Carlos E. Andrade

AT&T Labs Research
200 South Laurel Avenue, Middletown, NJ 07748 USA
cea@research.att.com

ABSTRACT

The Biased Random-Key Genetic Algorithm (BRKGA) is a population-based meta-heuristic designed to find optimal or near-optimal solutions for combinatorial problems. The performance of such algorithms heavily depends on parameter settings, which influence solution quality and search efficiency. One approach to this challenge is parameter control (or online tuning), where the algorithm adapts parameter values during the search process, utilizing accumulated information about the fitness landscape. This method also alleviates the user from manual tuning. This work proposes a novel approach by using random parameter values for each generation in BRKGA. Evaluations on three classical optimization problems—the Flowshop Permutation Problem, Set Covering Problem, and Traveling Salesman Problem—showed that this strategy produced competitive results compared to the tuned algorithm.

KEYWORDS. Biased Random-Key Genetic Algorithm. Parameter Control. Combinatorial Optimization.

Paper topics: MH (Metaheuristics)

1. Introduction

The Biased Random-Key Genetic Algorithm (BRKGA) [Gonçalves and Resende, 2011] is a population-based metaheuristic that relies on parameter setting to achieve optimal performance, as parameters significantly influence the algorithm's ability to find near-optimal solutions and efficiently explore the search space [Eiben et al., 2007]. This process, known as the algorithm configuration problem, involves identifying parameter settings that optimize performance on specific problem instances [Hoos, 2011]. While traditional approaches like parameter tuning involve offline analysis, parameter control dynamically adjusts values during execution, enhancing adaptability. This work proposes a parameter control method for BRKGA, comparing it with the state-of-the-art Iterated Race (irace) tuning method [López-Ibáñez et al., 2016]. Although irace is effective, it is time-consuming, and this new approach seeks to avoid tuning by adopting random parameter values for each generation, aiming for results comparable to irace.

2. Related Literature

Genetic algorithms (GAs) are population-based metaheuristics inspired by natural selection, where multiple solutions, represented as chromosomes, evolve simultaneously to explore the search space [Golberg, 1989]. In the Biased Random-Key Genetic Algorithm (BRKGA) [Gonçalves and Resende, 2011], solutions are encoded as vectors of random numbers, and a decoder converts these into problem-specific solutions, with fitness values guiding the evolution process. The population evolves through generations, where elite individuals form a group for future generations, and new chromosomes are introduced as mutants. Mating occurs between elite and non-elite individuals, with a bias controlled by the parameter ρ_e , giving elite genes a higher likelihood of inheritance [Gonçalves and Resende, 2011]. BRKGA's performance is highly dependent on tuning key parameters such as population size (p), elite size (p_e), and mutation rate (p_m) [Londe et al., 2024].

Parameter configuration aims to optimize algorithm performance by setting the appropriate values [Hoos, 2011]. Offline parameter tuning, where values are fixed before execution, can be expensive, particularly for algorithms like BRKGA that are sensitive to parameter interaction [Eiben et al., 2007]. On the other hand, online tuning dynamically adjusts parameters during the algorithm's run, enhancing adaptability and reducing the need for pre-tuning [Eiben et al., 1999]. This method is especially beneficial for evolutionary algorithms, offering better control over diversification and intensification [Karafotias et al., 2014]. Studies by Chaves et al. [2018] and Chaves and Lorena [2021] applied deterministic and reinforcement learning approaches to parameter control in BRKGA, achieving robust results. The recommended parameter ranges for BRKGA by Gonçalves and Resende [2011] include setting population size as $p = an$ (where $a \geq 1$ and n is chromosome length), with elite and mutant populations both set between 10-25% of p , and the elite allele inheritance probability (ρ_e) ranging from 50-80%.

3. Random Parameter Values - The Proposed Method

In this section, we propose using randomly selected values for the main BRKGA parameters at each generation to eliminate the need for tuning. Instead of fixing parameters throughout the algorithm's execution, which risks poor outcomes from bad initial values, random selection mitigates this risk and increases the likelihood of selecting good parameter configurations at different stages of evolution. The randomization is applied to the four key BRKGA parameters—population size (p), proportion of elite individuals (p_e), mutants (p_m), and elite-parent inheritance probability (ρ_e)—with values sampled from a uniform distribution within predefined bounds at each iteration (following the recommended parameter ranges by Gonçalves and Resende [2011]), as outlined in Algorithm 1 (lines 1-3). This dynamic approach follows the standard BRKGA procedure [Gonçalves and Resende, 2011], with the addition of resampling parameter values at the start of each generation (line 8).

Algorithm 1: BRKGA with online random parameter values.

```

Data: Stop criteria.
Result: Best solution.
1 initialize best solution  $B^* \leftarrow \infty$  // minimization problems;
2 initialize generations counter  $k \leftarrow 1$ ;
3 uniformly sample random values for each parameter  $(p^k, p_e^k, p_m^k, \rho_e^k)$ ;
4 initialize population  $P$  with  $p^k$  random individuals;
5 evaluate fitness values of individuals and sort by fitness;
6 while a stopping criterion is not met do
7   if  $k > 1$  then
8     uniformly sample new values for each parameter  $(p^k, p_e^k, p_m^k, \rho_e^k)$ ;
9     if  $p^k > p^{k-1}$  then
10       inject  $p^k - p^{k-1}$  new individuals into  $P$ ;
11       decode and sort  $p^k - p^{k-1}$  new individuals;
12     else
13       remove  $p^{k-1} - p^k$  individuals with the least fitness;
14   split population  $P$  in elite and non-elite sets, copy elite individuals to next generation  $k + 1$ ;
15   insert mutants, perform mating for remaining population;
16    $B \leftarrow$  best solution of population  $P$ ;
17   if  $B > B^*$  then
18     update current best solution  $B^* \leftarrow B$ ;
19   update generation counter  $k \leftarrow k + 1$ ;
20 return  $B^*$ ;
  
```

4. Experimental Results and Discussion

In our experiments, we compared the proposed parameter control method for BRKGA with Iterated Racing (Irace) [López-Ibáñez et al., 2016] across three combinatorial problems: flowshop scheduling, traveling salesman, and set covering, representing scheduling, routing, and location challenges. We tested BRKGA with both pure and local search decoders [Londe et al., 2024] using the BRKGA-MP-IPR framework [Andrade et al., 2021] on a high-performance cluster. Irace was used to tune parameters based on Gonçalves and Resende [2011], with a budget of 2,000 experiments and 50 distributed machines. The tuning process was time-intensive, taking over 26 days for flowshop scheduling, and the traveling salesman experiment with local search faced early termination due to server issues. Despite these challenges, the benchmark allowed a comparison of the random parameter control method's effectiveness.

The results show that for flowshop scheduling (FSP), BRKGA-Tuned LS outperforms BRKGA-Random LS, with lower average and median deviations (2.87 vs. 4.60 and 2.60 vs. 4.60), suggesting that parameter tuning is more effective when local search is applied. However, without local search, BRKGA-Random NLS significantly outperforms BRKGA-Tuned NLS, with lower average and median deviations (3.28 vs. 7.18 and 3.45 vs. 6.09), indicating that random parameter selection may work better in such cases. For the traveling salesman problem (TSP), BRKGA-Random LS performs better on average than BRKGA-Tuned LS (485.96 vs. 566.44), though the median deviation is slightly higher (3.41 vs. 3.14), highlighting more variability with random selection. Without local search, BRKGA-Random NLS again outperforms BRKGA-Tuned NLS (1607.68 vs. 1934.32). In the set covering problem (SCP), BRKGA-Tuned LS provides slightly better results than BRKGA-Random LS (29.30 vs. 28.70 for average deviation), with more consistent performance when local search is applied. Without local search, BRKGA-Tuned NLS outperforms BRKGA-Random NLS (28430.00 vs. 123503.00), but both methods show significant variability.

5. Conclusions

In conclusion, this study demonstrates that online parameter control in the Biased Random-Key Genetic Algorithm (BRKGA) offers a strong alternative to traditional tuning methods like It-

Table 1: Comparison of BRKGA-Tuned and BRKGA-Random. The table shows the average deviation (Avg. Dev) and median deviation (Med. Dev) from the best-known solution for each method. BRKGA was evaluated with and without local search (LS and NLS). Lower values indicate better performance.

Method	FSP		TSP		SCP	
	Avg. Dev from Best	Med. Dev from Best	Avg. Dev from Best	Med. Dev from Best	Avg. Dev from Best	Med. Dev from Best
BRKGA-Tuned LS	2.87	2.60	566.44	3.14	29.30	20.90
BRKGA-Random LS	4.60	4.60	485.96	3.41	28.70	22.20
BRKGA-Tuned NLS	7.18	6.09	1934.32	908.77	28430.00	1269.00
BRKGA-Random NLS	3.28	3.45	1607.68	1247.36	123503.00	17136.00

erated Racing (irace). Our experiments show that random parameter selection, particularly without local search, performs competitively with tuned approaches for simpler problems such as Flowshop Scheduling and Traveling Salesman, while completely eliminating the need for offline tuning. Although tuning may still be more effective for complex problems like Set Cover, random selection proves to be a practical approach. The reduced time and simplicity of randomization make it a compelling choice, and future research should focus on improving random parameter control and exploring adaptive methods to enhance its performance.

References

- Andrade, C., Toso, R., Gonçalves, J., and Resende, M. (2021). The multi-parent biased random-key genetic algorithm with implicit path-relinking and its real-world applications. *European Journal of Operational Research*, 289:17–30.
- Chaves, A., Gonçalves, J., and Lorena, L. (2018). Adaptive biased random-key genetic algorithm with local search for the capacitated centered clustering problem. *Computers and Industrial Engineering*, 124:331–346.
- Chaves, A. A. and Lorena, L. H. N. (2021). An adaptive and near parameter-free BRKGA using Q-learning method. *2021 IEEE Congress on Evolutionary Computation (CEC)*, p. 2331–2338.
- Eiben, A., Hinterding, R., and Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141.
- Eiben, A., Michalewicz, Z., Schoenauer, M., and Smith, J. E. (2007). Parameter control in evolutionary algorithms. *Studies in Computational Intelligence*, 54:19 – 46.
- Golberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. *Addison-Wesley*, 1989(102):36.
- Gonçalves, J. F. and Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.
- Hoos, H. H. (2011). Automated algorithm configuration and parameter tuning. p. 37–71. Springer.
- Karafotias, G., Hoogendoorn, M., and Eiben, Á. E. (2014). Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 19(2): 167–187.
- Londe, M. A., Pessoa, L. S., Andrade, C. E., and Resende, M. G. (2024). Biased random-key genetic algorithms: A review. *European Journal of Operational Research*.
- López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., and Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.