São José dos Campos - SP
LV SBPO
LV Simpósio Brasileiro de Pesquisa Operacional
São José dos Campos, SP - 6 a 9 de novembro de 2023

AN EVOLUTIONARY ALGORITHM APPLIED TO CONTINUOUS BI-OBJECTIVE OPTIMIZATION PROBLEMS*

Luis Henrique Pauleti Mendes
Instituto de Computação - UNICAMP
Av. Albert Einstein, 1251, Cidade Universitária, 13083-852, Campinas, SP, Brasil
luis.mendes@ic.unicamp.br

Fábio Luiz Usberti
Instituto de Computação - UNICAMP
Av. Albert Einstein, 1251, Cidade Universitária, 13083-852, Campinas, SP, Brasil
fusberti@ic.unicamp.br

Mário César San Felice
Departamento de Computação - UFSCar
Rod. Washington Luís, Km 235, 13565-905, São Carlos, SP, Brasil
felice@ufscar.br

ABSTRACT

This paper studies a novel Multi-Objective Evolutionary Algorithm (MOEA), the Non-dominated Sorting Biased Random-Key Genetic Algorithm (NS-BRKGA), designed for Multi-objective Optimization Problems (MOPs). The NS-BRKGA integrates elements from both the Multi-Parent Biased Random-Key Genetic Algorithm with Implicit Path-Relinking (BRKGA-MP-IPR) and the Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II). By managing the population diversity within the variable space, it enhances solution quality and diverges from traditional approaches. The paper details a comparative study conducted using the ZDT test suite for continuous bi-objective problems and includes several other algorithms, such as NSPSO, MOEA/D-DE, MHACO, and IHS. Despite the NSGA-II emerging as the most effective algorithm, the NS-BRKGA demonstrated competitive performance, underlining its potential applicability to continuous MOPs even though it was primarily designed for combinatorial problems.

KEYWORDS. Multi-Objective Optimization. BRKGA. NSGA-II.

ADM - Apoio à Decisão Multicritério. MH - Metaheurísticas

*This work was supported by FAPESP (2015/11937-9) and by Santander.

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

1. Introduction

Multi-objective Optimization Problems (MOPs) constitute an important field of Operations Research (OR). Their practical relevance is due to the fact that the majority of real-world problems require the consideration of several different points of view. These points of view can be modeled using multiple, often conflicting, objectives [Deb, 2014]. Formally, a MOP can be stated as

$$\min_{x \in \mathcal{X}} \mathbf{f}(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \ ,$$

wherein $\mathcal{X} \subset \mathbb{R}^n$ represents an n-dimensional decision space, and $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^m$ consists of m real-valued objective functions, where each one needs to be minimized. Moreover, $\mathcal{Y} = \{\mathbf{f}(x) : x \in \mathcal{X}\} \subset \mathbb{R}^m$ denotes an attainable objective space.

The m objective functions within $\mathbf{f}$ often conflict with each other, i.e., to improve one, others must be worsened. Thus, a single solution optimizing all objectives simultaneously is typically unattainable. Instead, the best trade-off solutions are of interest to a decision-maker. Such solutions are known as Pareto-optimal solutions, in homage to the Italian economist Vilfredo Pareto. The Pareto-optimality concept uses the dominance relation. Given two attainable objective points $y', y'' \in \mathcal{Y}$, y' is said to dominate y'', denoted as $y' \prec y''$, if and only if y' is no worse than y'' in every objective (i.e., $\forall i \in \{1, \ldots, m\}(y'_i \leq y''_i)$) and y' is better than y'' in at least one objective (i.e., $\exists j \in \{1, \ldots, m\}(y'_j > y''_j)$). An attainable objective point $y^\star \in \mathcal{Y}$ is characterized as Pareto-optimal if and only if there is no attainable objective point dominating it (i.e., $\nexists y \in \mathcal{Y}(y \prec y^\star)$). Likewise, a feasible solution $x^\star \in \mathcal{X}$ is deemed Pareto-optimal if and only if there is no feasible solution that dominates it (i.e., $\nexists x \in \mathcal{X}(\mathbf{f}(x) \prec \mathbf{f}(x^\star))$). The Pareto-optimal front $\mathcal{PF} \subseteq \mathcal{Y}$ encompasses all Pareto-optimal attainable objective points (i.e., $\mathcal{PF} = \{y^\star \in \mathcal{Y} | \nexists y \in \mathcal{Y}(y \prec y^\star)\}$) and the Pareto-optimal set $\mathcal{PS} \subseteq \mathcal{X}$ includes all Pareto-optimal feasible solutions (i.e., $\mathcal{PS} = \{x^\star \in \mathcal{X} | \nexists x \in \mathcal{X}(\mathbf{f}(x) \prec \mathbf{f}(x^\star))\}$).

The classical approaches to solving MOPs predominantly focused on the process of decomposition, entailing the formulation of several related Single-objective Optimization Problems (SOPs) derived from the original problem. These approaches offer the advantage of allowing the use of existing techniques tailored for SOPs. Nevertheless, they generally require several applications of an algorithm to find multiple optimal solutions and, on some occasions, such applications do not guarantee to find any optimal solutions [Deb, 2014]. In contrast, more recent methodologies tend to focus on population-based metaheuristics, such as Evolutionary Algorithms (EAs). These approaches use a population of candidate solutions to search for a set of representative Pareto-optimal solutions [Saini e Saha, 2021]. In addition to the generic ability to handle large complex real-world problems inherent in traditional EAs, Multi-Objective Evolutionary Algorithms (MOEAs) present a unique advantage. This advantage consists in their ability to find a set of representative Pareto-optimal solutions in a single run, rather than merely finding one solution. This is primarily due to their inherent parallel character and group strategy.

Our contribution. This work focuses on a new MOEA, called Non-dominated Sorting Biased Random-Key Genetic Algorithm (NS-BRKGA), that combines the benefits of Biased Random-Key Genetic Algorithms (BRKGAs) [Gonçalves e Resende, 2011; Lucena et al., 2014; Andrade et al., 2021] with the Elitist Non-Dominated Sorting Algorithm (NSGA-II) [Deb et al., 2002]. The NS-BRKGA, which can be easily applied to any MOP, has already been shown to be efficient in tackling combinatorial MOPs [Mendes et al., 2023, 2022] . In this study, the performance of our methodology was evaluated, in comparison with other MOEAs from the literature, using the widespread ZDT test suite of continuous bi-objective optimization problems [Zitzler et al., 2000].

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

2. Base Algorithms

Here we describe the frameworks that served as foundation for the algorithm we propose.

2.1. Random-Key Genetic Algorithms

Random-Key Genetic Algorithms (RKGAs) were initially introduced by Bean [1994] for solving combinatorial SOPs involving sequencing. In an RKGA, chromosomes are represented as vectors of randomly generated real numbers within the interval $[0, 1)$, that is, as points in the n-dimensional unit hypercube $[0, 1)^n$. A deterministic algorithm, called a decoder, maps any individual to a feasible solution of the optimization problem, for which a fitness value can be computed.

The initial population P_0 comprises vectors of random-keys, where each allele is independently and randomly generated within the interval $[0, 1)$. Given any generation $t \in \mathbb{N}$, the fitness of each individual in the population P_t is computed via the decoder. Subsequently, P_t is partitioned into a smaller subset of elite individuals P_t^e (i.e., those with the best fitness value) and a larger subset of non-elite individuals $P_t^{\bar{e}} = P_t \setminus P_t^e$ (i.e., containing the remaining individuals). The elite set P_t^e is copied into the next generation $t+1$. Additionally, a small set P_{t+1}^m of random solutions, referred to as mutants, is introduced into the new population P_{t+1}. Considering that $|P_{t+1}^e| + |P_{t+1}^m| < |P_{t+1}|$, the population is completed by offspring, generated through the process of mating.

2.2. Biased Random-Key Genetic Algorithm

A Biased Random-Key Genetic Algorithm (BRKGA) [Gonçalves e Resende, 2011] distinguishes itself from an RKGA in its approach towards parent selection for mating. In an RKGA [Bean, 1994], both parents are randomly selected from the entire population. In contrast, in BRKGA, each offspring is generated by combining one individual from the elite set P_t^e and one individual from the non-elite set $P_t^{\bar{e}}$. Since $|P_t^e| < |P_t^{\bar{e}}|$, the probability that a given elite individual is selected for mating $(1/|P_t^e|)$ is greater than that of a given non-elite individual $(1/|P_t^{\bar{e}}|)$. Consequently, an elite individual has a higher likelihood of passing on his characteristics to future generations in comparison to a non-elite individual. Furthermore, mating is performed using parameterized uniform crossover [Spears e De Jong, 1995], where a gene is taken from the elite individual with a probability of $\rho > 0.5$, or otherwise, it is taken from the other parent.

2.3. Multi-Parent Biased Random-Key Genetic Algorithm

While natural mating processes are limited to at most two parents, the crossover operator within GAs does not have such a restriction. Taking advantage of this opportunity, the Multi-Parent Biased Random-Key Genetic Algorithm (BRKGA-MP) [Lucena et al., 2014] employs multiple parents to generate new offspring. Within a BRKGA-MP, π parents are selected and ranked according to their fitness. Among these, π^e are elite parents and $\pi^{\bar{e}} = \pi - \pi^e$ are non-elite parents, with $\pi^e \geq \pi^{\bar{e}}$. A parameterized uniform crossover is subsequently performed, such that each allele is taken from a chosen parent. The probability of selecting a parent is determined by a non-increasing weighting bias function $\Phi : \mathbb{N} \to \mathbb{R}_+^*$ over on its rank r. The following bias functions, proposed by Bresina [1996], are available: *logarithmic*: $\Phi(r) = \frac{1}{\log_2(r+1)}$; *linear*: $\Phi(r) = \frac{1}{r}$; *polynomial (in d)*: $\Phi(r) = \frac{1}{r^d}$; *exponential*: $\Phi(r) = \frac{1}{e^r}$; *constant*: $\Phi(r) = \frac{1}{\pi}$.

2.4. Multi-Parent Biased Random-Key Genetic Algorithm with Implicit Path-Relinking

Path-Relinking (PR) [Glover et al., 2000] is an intensification strategy that exploits the neighborhood constructed along the trajectories between elite solutions. The algorithm typically initiates with two solutions: a base solution, and a guide solution. The search for better solutions involves the exploration of one or more paths connecting these solutions, constructed through a series of local moves on the guide solution that integrate attributes from the guide solution. The

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

intermediate solutions found along these paths are expected to be of high quality, as they combine elements from both elite solutions.

The hybridization of BRKGA with PR was first proposed by Ribeiro et al. [2021], where it was utilized as a progressive crossover operation. The Implicit Path-Relinking (IPR) is an algorithm that leverages the separation between the problem space and solution space inherent in the BRKGA [Andrade et al., 2021]. This approach constructs the path entirely within the unit hypercube of the BRKGA, employing the decoder function to evaluate the intermediate vectors of random-keys. The IPR strategy was inserted into the BRKGA-MP framework, thereby giving rise to a new variant known as Multi-Parent Biased Random-Key Genetic Algorithm with Implicit Path-Relinking (BRKGA-MP-IPR) [Andrade et al., 2021].

Typically, random-keys are employed in two ways: 1) *Thresholds*: keys are associated with decision variables and denote their inclusion or exclusion; 2) *Permutations*: the random-key vector may induce a permutation of decision variables by first assigning each variable with an index and subsequently sorting the vector to obtain a permutation. Consequently, there are two distinct IPR routines [Andrade et al., 2021]: the *direct implicit path-relinking* and the *permutation-based implicit path-relinking*. In the direct path-relinking strategy, the algorithm does not assume a particular structure; therefore, the keys are directly exchanged between the incumbent and the guide solutions. Conversely, in the permutation-based implicit path-relinking, the keys are not directly exchanged; instead, the permutation from the guide solution is used to rearrange the keys of the base solution in such a way that they induce the same sub-permutation on the base solution.

2.5. Elitist Non-dominated Sorting Genetic Algorithm

The Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [Deb et al., 2002] is a renowned MOEA primarily known for its capability to find a diverse set of Pareto-optimal solutions for a given MOP. It is characterized by the following principles: elitism, diversity preservation, and emphasis on non-dominated solutions.

The beginning of the NSGA-II involves the creation of an initial population, denoted as P_0, where each individual is randomly generated. As the algorithm progresses to each new generation t, it creates an offspring population, denoted as Q_t. This is achieved through the application of a series of genetic operators to the current population P_t. These operators include binary tournament selection, simulated binary crossover, and polynomial mutation. To create the population of the subsequent generation, a new population R_t is formed by combining the current population P_t with the offspring population Q_t, having a size of $2|P_t|$. The algorithm then applies the fast non-dominated sort to the combined population R_t. Solutions in the highest level of non-dominance are assigned the best fitness, while solutions in subsequent levels are assigned worse fitness values. Since all members from previous and current populations are included in R_t, the algorithm provides a global non-domination check among both parent and offspring solutions, ensuring elitism. The best non-dominated set, $S_1^\star$, contains the best solutions in R_t and must be prioritized over all other solutions in R_t. If the size of $S_1^\star$ is less than the size of P_t, all members of $S_1^\star$ are selected for the next population, P_{t+1}. The remaining slots in P_{t+1} are filled with solutions from subsequent non-dominated sets, in order of their ranking. This process continues until no more non-dominated sets can be accommodated into P_{t+1}. In the case that the last acceptable non-dominated set, $S_l^\star$, contains more solutions than the available slots in P_{t+1}, a niching strategy is employed. This strategy involves selecting solutions from the least crowded region in the objective space, thereby increasing the diversity among the selected solutions.

During the early stages of the NSGA-II, the previously described selection process does not greatly impact the algorithm's proceedings. The reason behind this is that there are many non-

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

dominated sets in the combined population, R_t, and it is likely that many of these sets are already included in the new population, P_{t+1}, before they add up to the total of $|P_t|$. Consequently, the specific solutions selected to fill the remaining slots in P_{t+1} hold relatively low importance. However, as the algorithm evolves to its later stages, it becomes increasingly likely that the majority of solutions in the combined population, R_t, belong to the best non-dominated set, $S_1^\star$, and exceed the size of $|P_t|$. In this case, the algorithm's niching strategy, used to select a diverse set of solutions from R_t to populate P_{t+1}, is of crucial importance to guarantee a better spread among the solutions. Even if the entire population, P_t, converges towards the Pareto-optimal set, the algorithm will continue to employ this strategy to maintain diversity among the solutions, ensuring that the algorithm does not stagnate and continues to explore various parts of the solution space.

3. Non-dominated Sorting Biased Random-Key Genetic Algorithm (NS-BRKGA)

In this work, we extend the single-objective BRKGA-MP-IPR [Andrade et al., 2021] framework for solving MOPs. One of the main characteristics of the BRKGA and its variants lies in their clear division between the problem-independent component, responsible for the generation and modification of individuals, and the problem-dependent component, which decodes and evaluates individuals. Therefore, in order to effectively address MOPs, certain modifications to the problem-independent part of the algorithm are necessary. One critical issue that emerges is how to determine an order for the solutions within the populations to select the elite set and to define the bias of the parents, given that the fitness of each individual must take into account all objectives. To address this, we have employed the non-dominated sorting and the crowding distance sorting strategies from the NSGA-II [Deb et al., 2002].

Though the standard BRKGA has been previously hybridized with NSGA-II to tackle some specific MOPs [Chagas et al., 2021; Marchi e Parpinelli, 2022; Damm e Ronconi, 2021; Tangpattanakul et al., 2015], to the best of our knowledge, our approach, called Non-dominated Sorting Biased Random-Key Genetic Algorithm (NS-BRKGA), is the first general framework extended from the BRKGA-MP-IPR which can be easily applied to solve any MOP. Indeed, previous versions have already been successfully applied to the Bi-Objective Traveling Salesman Problem (BOTSP) [Mendes et al., 2023] and to the Multi-Objective Multi-Dimensional Knapsack Problem (MOMDKP) [Mendes et al., 2022]. Furthermore, new features were developed in our methodology to improve the population diversity and the resulting non-dominated frontier. These enhancements include polynomial mutation and selection bias towards solutions with high diversity in the genotype space. The NS-BRKGA framework, implemented in C++, is available at: `https://github.com/luishpmendes/ns-brkga`.

The execution process of the NS-BRKGA is detailed next. The algorithm starts with the generation of an initial population of random key vectors. This population may consist solely of random solutions, or it can incorporate high-quality solutions known beforehand. The inclusion of existing solutions can enhance the algorithm's convergence to Pareto-optimal solutions. The algorithm then proceeds iteratively until the stopping criteria are satisfied. During each iteration or generation, the current population of random key vectors is evolved through a combination of genetic operators from NSGA-II and BRKGA-MP. Subsequent to the evolution process, the algorithm evaluates whether to apply IPR local search from BRKGA-MP-IPR among elite solutions. This procedure is executed periodically, with the frequency determined as determined by the parameter τ_{IPR}. Subsequently, the algorithm evaluates the necessity of applying the shaking procedure to the solutions within the current population. Shaking is executed when the number of iterations without improvement in the archive of the best solutions exceeds a threshold established by the parameter τ_{shake}. If shaking is required, a random subset of solutions is selected and subjected to shaking.

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

Furthermore, the algorithm evaluates the need to apply the reset procedure within the population. A reset occurs only when there has been no improvement in the archive of best solutions for a specified number of iterations, as denoted by the parameter τ_{reset}. During a reset, a random subset of solutions is replaced by random ones, similar to the random solutions comprising the initial population. Upon the conclusion of each iteration, the algorithm re-evaluates whether the stopping criteria have been fulfilled. If these criteria are yet not satisfied, the iterative process continues until the stopping criteria are eventually satisfied, and the algorithm reaches completion.

Figure 1 depicts the evolution procedure of the NS-BRKGA, illustrating how the subsequent population is generated from the current one using genetic operators from NSGA-II and BRKGA-MP. The algorithm takes as input the current population P_t, the number of objectives m, a

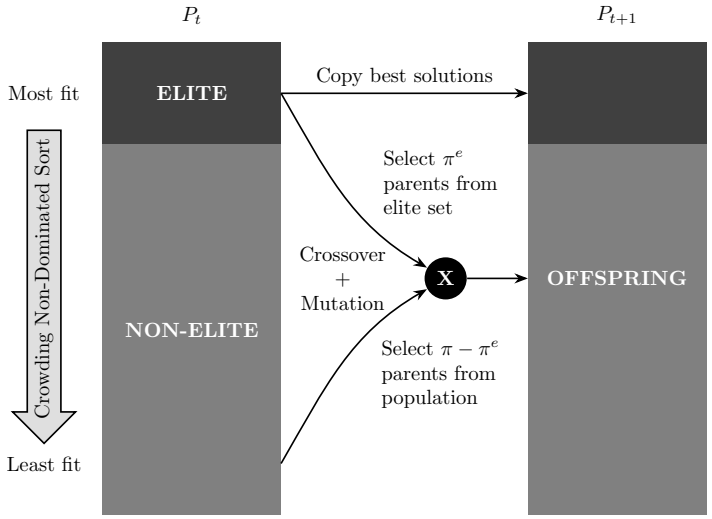


Figure 1: Evolution procedure of the NS-BRKGA.

diversity function within the genotype space $D : [0, 1)^n \to \mathbb{R}^+$, the minimum and maximum allowable sizes for the elite set, respectively, $p^e_{\min}$ and $p^e_{\max}$, a non-increasing bias function $\Phi : \mathbb{N} \to \mathbb{R}^+$ for the crossover operator, the total number of parents π used in the crossover, the number of elite parents π^e used in the crossover, the probability p_m of applying polynomial mutation, and the distribution index η_m utilized in the polynomial mutation. The output is a new population P_{t+1}. The algorithm starts by ordering the individuals in the current population by their fitness so that the fittest individuals are those that are non-dominated and reside in the less crowded region of the objective space, and by identifying the non-dominated set of solutions $S^\star$ within the current population P_t. Next, the elite set P^e_t is selected from P_t, prioritizing the non-dominated solutions $S^\star$, maximizing diversity D in the genotype space, and complying with the minimum and maximum allowable sizes, $p^e_{\min}$ and $p^e_{\max}$, for the elite set. The next population P_{t+1} is then initialized, containing all solutions from P^e_t without modifications, and completed with new solutions. Each new individual is generated through the Biased Multi-Parent Crossover from BRKGA-MP and then modified by the polynomial mutation from NSGA-II. The algorithm ensures that the m best individuals from P_t, i.e., those with the best value for each objective function, are selected for mating to generate the first m new individuals in the next population. Finally, the algorithm returns the next population.

The majority of current multi-objective optimizers do not actively control the population diversity within the variable space. This is a notable difference when compared to single-objective optimizers, wherein the preservation of diverse solutions is considered advantageous, given that it

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

enables a more thorough exploration of the search space. The quality of the solutions obtained can be enhanced by incorporating mechanisms to manage the diversity within the variable space explicitly [Segura et al., 2023]. Therefore, the NS-BRKGA adopts a strategy for selecting the elite set that maximizes the diversity within the variable space, namely, within the unit hypercube $[0, 1)^n$.

Algorithm 1 details the procedure for selecting the elite set from a given population.

Algorithm 1 Procedure for selecting the elite set

Input: Population P sorted by non-dominated sorting and crowding distance sorting; Non-dominated solutions $S^\star \subseteq P$; Diversity function $D : [0, 1)^n \to \mathbb{R}^+$; Sizes $p^e_{\min}$ and $p^e_{\max}$.

Output: The elite set $P^e \subseteq P$.

```
1:  SELECT-ELITE-SET(P, S*, D, p^e_min, p^e_max)
2:      P^e ← P[1 : p^e_min]
3:      if |P^e| < |S*| then
4:          P^e ← S*
5:      if |P^e| > p^e_max then
6:          P^e ← P_t[1 : p^e_max]
7:      for p ← |P^e| + 1 to p^e_max do
8:          if D(P^e) < D(P_t[1 : p]) then
9:              P^e ← P[1 : p]
10:     return P^e
```

In line 2, the elite set P^e is initialized with the first $p^e_{\min}$ solutions from the population P. Lines 3 and 4 guarantee the incorporation of all solutions from the non-dominated set $S^\star$ into the elite set P^e. Thereafter, lines 5 and 6 ensure that the maximum elite set size $p^e_{\max}$ is respected. If there are still slots in the elite set P^e to accommodate more solutions, the loop from lines 7 to 9 selects a prefix of the population P that maximizes the diversity function D. Finally, the algorithm returns the elite set P^e in line 10. The following diversity measure was used [Lacevic e Amaldi, 2011]: *average distance to centroid*: $D_v(P) = \frac{1}{|P|}\sum_{i=1}^{|P|} \|P[i] - \bar{P}\|$, where $\bar{P} = \frac{1}{|P|}\sum_{i=1}^{|P|} P[i]$ is the centroid of the population.

4. ZDT Test Suite

In this paper, we consider five continuous bi-objective test problems from the ZDT test suite, named after its authors Zitzler, Deb, and Thiele [Zitzler et al., 2000]. These functions, which have been widely used in the literature to assess the performance of MOEAs, possess specific features that may cause difficulties in converging to the Pareto-optimal front and maintaining diversity within the population. The ZDT1 test problem presents a convex Pareto-optimal front. Its counterpart, the ZDT2 test problem, exhibits a non-convex Pareto-optimal front. Representing discontinuities in the image space, the ZDT3 test problem introduces a more complex Pareto-optimal front that consists of several disconnected convex parts. The ZDT4 test problem presents a multimodal search space, containing 21^9 local Pareto-optimal fronts. Lastly, the ZDT6 test problem features a non-uniform search space in which Pareto-optimal solutions are not only non-uniformly distributed along the global Pareto-optimal front, but also exhibit a density that is lower near the Pareto-optimal front and higher away from it. The ZDT5 test problem was intentionally excluded from consideration due to its nature as a non-continuous problem. In this study, the number of decision variables, denoted as n, was chosen as $n = 30$ for the ZDT1, ZDT2, and ZDT3 test problems, and $n = 10$ for the ZDT4 and ZDT6 test problems, following the guidelines provided by Zitzler et al. [2000].

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

5. Computational experiments

The proposed algorithm was compared with the following algorithms, all of which have implementations available in the PAGMO2 library [Biscani e Izzo, 2020], which is maintained by the European Space Agency[1]. The algorithms used for comparison are the Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [Deb et al., 2002], the Non-dominated Sorting Particle Swarm Optimization (NSPSO) [Li, 2003], the Multi-Objective Evolutionary Algorithm by Decomposition with Differential Evolution (MOEA/D-DE) [Zhang e Li, 2007], the Multi-objective Hypervolume-based Ant Colony Optimizer (MHACO) [Acciarini et al., 2020], and the Improved Harmony Search (IHS) [Mahdavi et al., 2007]. The source code of the experiments and the obtained results can be accessed at: `https://github.com/luishpmendes/sbpo2023-zdt`.

5.1. Performance Metrics

The hypervolume ratio and the modified inverted generation distance were the performance metrics chosen to compare the algorithms, as they assess both the convergence and the diversity of the obtained solutions [Audet et al., 2021]. These metrics are discussed in this Section.

5.1.1. Hypervolume Ratio

The hypervolume indicator, also referred to as the Lebesgue measure or the S-metric, is associated with a front of attainable objective points $Y \subset \mathcal{Y}$, and a reference objective point $y^{ref} \in \mathbb{R}^m$, which respects the condition $\forall y \in Y (y \prec y^{ref})$. The hypervolume indicator can be described as the size of the portion of the objective space R^m that is weakly dominated by Y and bounded by y^{ref}. More formally, the hypervolume indicator is represented as

$$HV(Y, y^{ref}) = \Lambda_m(\{y' \in \mathbb{R}^m | \exists y \in Y (y \preceq y' \preceq y^{ref})\}) \ ,$$

where Λ_m denotes the m-dimensional Lebesgue measure. Given a reference front $Y^{ref} \subset \mathcal{Y}$, ideally consisting of the Pareto-optimal front $\mathcal{PF}$, the hypervolume ratio can be calculated as

$$HVR(Y, Y^{ref}, y^{ref}) = \frac{HV(Y, y^{ref})}{HV(Y^{ref}, y^{ref})} \ .$$

The greater the ratio, the better is the front Y.

In this paper, the adopted reference objective point y^{ref} consists of the worst values for the objective functions, which pertain to the solutions associated with the fronts yielded by the algorithms during the optimization process. This approach enables the computation of the hypervolume ratio for all such fronts. The reference front Y^{ref} consists of the best non-dominated attainable objective points obtained by all algorithms. However, due to the continuous nature of the problems under consideration, there may be a very large number of such objective points. Therefore, the crowding distance sort was used to select a well-spread subset of objective points.

5.1.2. Modified Inverted Generational Distance

The generational distance is defined as the average distance between each objective point y in a given front $Y \subset \mathcal{Y}$ and the closest objective point $y^\star$ in a given reference front $Y^{ref} \subset \mathcal{Y}$. Formally, the generational distance is calculated as follows:

$$GD(Y, Y^{ref}) = \frac{1}{|Y|} \left(\sum_{y \in Y} \min_{y^\star \in Y^{ref}} d(y, y^\star)^p \right)^{\frac{1}{p}} , d(y, y^\star) = \sqrt{\sum_{i=1}^{m} (y_i - y_i^\star)^2}.$$

[1] `https://esa.github.io/pagmo2/`

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

The inverted generational distance improves upon GD by reversing the order of the fronts considered as input by GD, i.e., $IGD(Y, Y^{ref}) = GD(Y^{ref}, Y)$, which is better because it is not sensitive to the size of the considered front Y.

Lastly, the modified inverted generational distance proposes a modification to the distance function. For objective points that are dominated by the reference front, the traditional Euclidean distance is used. However, for objective points that are non-dominated with respect to the reference point, only the dominated components of the objective points are used to compute the distance. Furthermore, it places the denominator $|Y^{ref}|$ under the exponent $\frac{1}{p}$. Formally, this is expressed as:

$$IGD^+(Y, Y^{ref}) = \left(\frac{1}{|Y^{ref}|} \sum_{y \in Y} \min_{y^\star \in Y^{ref}} d^+(y^\star, y)^p \right)^{\frac{1}{p}}, d^+(y^\star, y) = \sqrt{\sum_{i=1}^{m} (\max(y_i - y_i^\star, 0))^2}.$$

The smaller the IGD^+, the better the front Y is. The reference front Y^{ref} used here was the same used for the hypervolume ratio. Additionally, in this paper, the value $p = 1$ was used.

5.2. Experimental Settings

A series of twenty-five independent optimization runs was performed for each pair of algorithm and test problem. Each run was limited to ten minutes, resulting in a total of 125 hours of computational time. These optimization runs were carried out on a computer equipped with an Intel Xeon CPU E3-1230 V2 with 8 cores running at 3.30GHz, and 32GB of RAM. The experiments were conducted using 6 parallel threads, reducing the total time for the experiments to 20 hours and 50 minutes. For all algorithms, the population size was set to 100, and the size of the archive with the best solutions was set to 250. The size of the reference attainable objective front, which contains the best solutions found by all executions of all algorithms, was set to 500. For the NS-BRKGA, we used $\pi = 3$ parents, with $\pi^e = 2$ elite parents, logarithmic bias function, mutation probability equal to 0.01, mutation distribution index equal to 50, elite set size varying from 10% to 50% of the population size, and the direct implicit path-relinking. These parameters' values were selected empirically. For the other algorithms, we used the default parameter values from the PAGMO2 library [Biscani e Izzo, 2020].

5.3. Results Analysis

Table 1 presents the mean and standard deviation of the hypervolume ratio obtained by each algorithm for each ZDT test problem. The algorithm that obtained the best results was NSGA-II, obtaining the best mean hypervolume ratio for all problems. In fact, the average obtained by NSGA-II for the ZDT3 problem was greater than 1. This phenomenon can be explained by the fact that the reference front was computed using crowding distance sorting. However, this method does not assure that the hypervolume of the reference front is greater or equal to the hypervolume of the fronts obtained by the algorithms. The NS-BRKGA obtained the second-best mean overall, and the second-best average hypervolume ratio for 3 of the 5 evaluated problems.

Table 2 shows the mean and standard deviation of the modified inverted generational distance obtained by each algorithm for each ZDT test problem. The algorithm that achieved the best results was NSGA-II, which secured the best mean for 4 out of 5 problems, and the second-best mean for the remaining problem. The NS-BRKGA algorithm achieved the second-best mean for the ZDT4 problem, but for the other problems, its mean was worse than the third-best. However, taking all the problems into account, it can be seen that NS-BRKGA achieved the third-best mean overall. Thus, despite not ranking among the top three for 4 out of 5 problems, its results were not significantly worse, which allowed NS-BRKGA to stay in third place.

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

Table 1: The mean (μ) and the standard deviation (σ) of the hypervolume ratio obtained by each pair of algorithm and ZDT test problem.

Test Problem		NSGA-II	NSPSO	MOEA/D-DE	MHACO	IHS	NS-BRKGA
ZDT1	μ	***0.999904***	*0.999697*	**0.999867**	0.998605	0.992307	0.998261
	σ	0.000005	0.000017	0.000013	0.000104	0.001392	0.001802
ZDT2	μ	***0.999866***	*0.999674*	**0.999830**	0.950784	0.991832	0.996754
	σ	0.000005	0.000016	0.000013	0.000397	0.002181	0.003723
ZDT3	μ	***1.000025***	0.969324	*0.969332*	0.968824	0.957010	**0.995768**
	σ	0.000005	0.000007	0.000011	0.000293	0.001377	0.004164
ZDT4	μ	***0.999999***	0.981906	*0.999963*	0.800023	0.999132	**0.999986**
	σ	0.000000	0.023609	0.000004	0.016107	0.000409	0.000026
ZDT6	μ	***0.999943***	*0.940268*	0.940222	0.915730	0.742360	**0.998976**
	σ	0.000006	0.000005	0.000095	0.020137	0.017430	0.000859
All	μ	***0.999947***	0.978174	*0.981843*	0.926793	0.936528	**0.997949**
	σ	0.000059	0.024551	0.024036	0.070039	0.098895	0.003021

Table 2: The mean (μ) and the standard deviation (σ) of the modified inverted generational distance obtained by each pair of algorithm and ZDT test problem.

Test Problem		NSGA-II	NSPSO	MOEA/D-DE	MHACO	IHS	NS-BRKGA
ZDT1	μ	***0.000000206***	*0.000000506*	**0.000000232**	0.000005090	0.000155936	0.000038303
	σ	0.000000010	0.000000039	0.000000019	0.000000730	0.000056502	0.000082303
ZDT2	μ	***0.000000090***	*0.000000223*	**0.000000105**	0.016577252	0.000090137	0.000107046
	σ	0.000000005	0.000000017	0.000000009	0.000141936	0.000055435	0.000288683
ZDT3	μ	***0.000000032***	*0.000000057*	**0.000000039**	0.000000333	0.000158725	0.000160635
	σ	0.000000004	0.000000007	0.000000003	0.000000363	0.000047707	0.000263887
ZDT4	μ	***0.000000000***	0.002074863	*0.000000104*	0.149185200	0.000001646	**0.000000003**
	σ	0.000000000	0.003161862	0.000000511	0.024218105	0.000001533	0.000000007
ZDT6	μ	**0.000000015**	***0.000000011***	*0.000000509*	0.002534352	0.050227096	0.000000797
	σ	0.000000001	0.000000002	0.000002446	0.002112024	0.009943052	0.000001057
All	μ	***0.000000069***	0.000415132	**0.000000198**	0.033660445	0.010126708	*0.000061357*
	σ	0.000000076	0.001621483	0.000001112	0.059296415	0.020600783	0.000186903

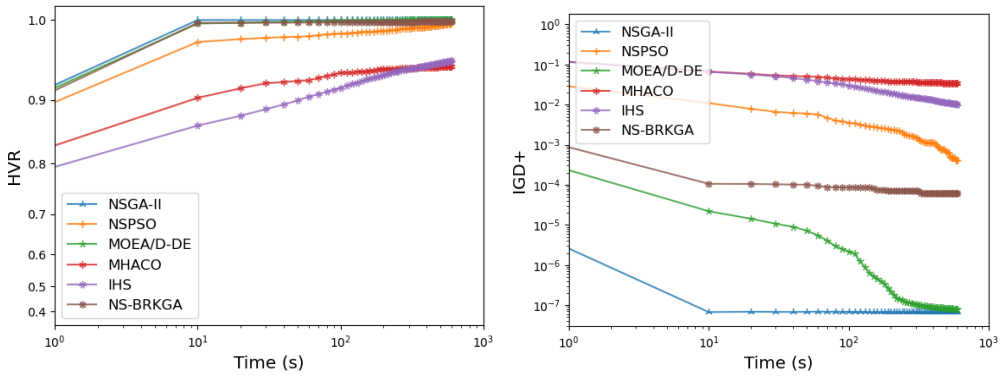
Figure 2 shows the graphs of the average hypervolume ratio and modified inverted directional distance obtained by each algorithm over time for all ZDT test problems. To plot these graphs, 60 snapshots of the archive with the best solutions obtained by the algorithms were saved, that is, a snapshot was saved every 10 seconds of the optimization process. Figure 2(a) reveals that within less than 30 seconds, the NS-BRKGA converged to the best values of the hypervolume ratio, along with the NSGA-II and MOEA/D-DE. In Figure 2(b), it can be observed that the NS-BRKGA, also in under 30 seconds, stagnated at values of the inverted directional distance on the order of 10^{-4}, whereas the NSGA-II and the MOEA/D-DE managed to reach values on the order of 10^{-7}. This stagnation of the NS-BRKGA may be attributed to a lack of exploitation of its search procedure, as its genetic operators were originally not designed for continuous MOPs.

Despite not achieving the best results, it can be observed that the NS-BRKGA's performance is competitive when compared with other algorithms in the literature. It is, however, important to emphasize that the multi-parent biased crossover operator used in the NS-BRKGA was initially designed for combinatorial problems. This could explain the NS-BRKGA's lower performance when solving continuous problems.

6. Final Remarks

This work studied the performance of the NS-BRKGA using the ZDT test suite [Zitzler et al., 2000]. Future work may assess the performance of the NS-BRKGA in solving other con-

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073



(a) (b)

Figure 2: The average of the hypervolume ratio and of the modified inverted generational distance, obtained by each algorithm over time for all ZDT test problems.

tinuous MOPs, as the DTLZ [Deb et al., 2005] and the WFG problem [Huband et al., 2006] test suits. As the genetic operators of NS-BRKGA were designed to deal with combinatorial MOPs, another line of research is to improve its performance by designing a crossover operator to better handle continuous MOPs. One of the drawbacks of BRKGAs is the large number of parameters that need to be tuned to achieve good performance. Works in the literature [Chaves e Lorena, 2021] attempt to solve this issue by using an adaptive strategy. Hence, another interesting research line is the incorporation of such strategies into the NS-BRKGA framework.

7. Acknowledgments

This work was funded by FAPESP, process number 2015/11937-9, and by Santander.

References

Acciarini, G., Izzo, D., e Mooij, E. (2020). MHACO: a Multi-Objective Hypervolume-Based Ant Colony Optimizer for Space Trajectory Optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, p. 1–8. IEEE.

Andrade, C. E., Toso, R. F., Gonçalves, J. F., e Resende, M. G. C. (2021). The Multi-Parent Biased Random-Key Genetic Algorithm with Implicit Path-Relinking and its real-world applications. *European Journal of Operational Research*, 289(1):17–30.

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., e Salomon, L. (2021). Performance indicators in multiobjective optimization. *European Journal of Operational Research*, 292(2):397–422.

Bean, J. C. (1994). Genetic Algorithms and Random Keys for Sequencing and Optimization. *ORSA Journal on Computing*, 6(2):154–160.

Biscani, F. e Izzo, D. (2020). A parallel global multiobjective framework for optimization: pagmo. *Journal of Open Source Software*, 5(53):2338.

Bresina, J. L. (1996). Heuristic-biased stochastic sampling. In *AAAI/IAAI, Vol. 1*, p. 271–278.

Chagas, J. B., Blank, J., Wagner, M., Souza, M. J., e Deb, K. (2021). A non-dominated sorting based customized random-key genetic algorithm for the bi-objective traveling thief problem. *Journal of Heuristics*, 27(3):267–301.

Chaves, A. A. e Lorena, L. H. N. (2021). An Adaptive and Near Parameter-Free BRKGA Using Q-Learning Method. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, p. 2331–2338.

Damm, R. d. B. e Ronconi, D. P. (2021). A Multi-objective Biased Random-Key Genetic Algorithm for Service Technician Routing and Scheduling Problem. In *International Conference on Computational Logistics*, p. 471–486. Springer.

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073

Deb, K. (2014). *Multi-objective Optimization*, p. 403–449. Springer.

Deb, K., Pratap, A., Agarwal, S., e Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K., Thiele, L., Laumanns, M., e Zitzler, E. (2005). *Scalable Test Problems for Evolutionary Multiobjective Optimization*, p. 105–145. Springer.

Glover, F., Laguna, M., e Martí, R. (2000). Fundamentals of Scatter Search and Path Relinking. *Control and Cybernetics*, 29(3):653–684.

Gonçalves, J. F. e Resende, M. G. C. (2011). Biased Random-Key Genetic Algorithms for Combinatorial Optimization. *Journal of Heuristics*, 17(5):487–525.

Huband, S., Hingston, P., Barone, L., e While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5): 477–506.

Lacevic, B. e Amaldi, E. (2011). Ectropy of diversity measures for populations in Euclidean space. *Information Sciences*, 181(11):2316–2339.

Li, X. (2003). A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. In *Genetic and Evolutionary Computation Conference, GECCO 2003*, p. 37–48. Springer.

Lucena, M. L., Andrade, C. E., Resende, M. G., e Miyazawa, F. K. (2014). Some extensions of biased random-key genetic algorithms. In *Proceedings of the 46th Brazilian Symposium of Operational Research*, p. 1–12.

Mahdavi, M., Fesanghary, M., e Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 188(2):1567–1579.

Marchi, F. e Parpinelli, R. S. (2022). A Multi-objective Cluster-based Biased Random-Key Genetic Algorithm with Online Parameter Control Applied to Protein Structure Prediction. In *17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, p. 337–346. IEEE.

Mendes, L. H. P., Usberti, F. L., e San Felice, M. C. (2023). An Evolutionary Algorithm Applied to the Bi-Objective Travelling Salesman Problem. In *Metaheuristics: 14th International Conference, MIC 2022*, p. 519–524. Springer.

Mendes, L. H. P., Usberti, F. L., e San Felice, M. C. (2022). An Evolutionary Algorithm applied to the Multi-Objective Multi-Dimensional Knapsack Problem. In *LIV Simpósio Brasileiro de Pesquisa Operacional*.

Ribeiro, C. C., Riveaux, J. A., e Brandao, J. S. (2021). Biased Random-Key Genetic Algorithms Using Path-Relinking as a Progressive Crossover Strategy. In *5th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, p. 28–36. ACM.

Saini, N. e Saha, S. (2021). Multi-objective optimization techniques: a survey of the state-of-the-art and applications. *The European Physical Journal Special Topics*, p. 1–17.

Segura, C., Castillo, J. C., e Schütze, O. (2023). The importance of diversity in the variable space in the design of multi-objective evolutionary algorithms. *Applied Soft Computing*, p. 110069.

Spears, W. M. e De Jong, K. D. (1995). On the Virtues of Parameterized Uniform Crossover. Technical report, Naval Research Lab Washington DC.

Tangpattanakul, P., Jozefowiez, N., e Lopez, P. (2015). Biased Random Key Genetic Algorithm for Multi-user Earth Observation Scheduling. In *Recent Advances in Computational Optimization: Results of the Workshop on Computational Optimization, WCO 2013*, p. 143–160. Springer.

Zhang, Q. e Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

Zitzler, E., Deb, K., e Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195.

DOI: 10.59254/sbpo-2023-175073
http://live-proceedings.pantheonsite.io/p/175073