Hindawi
Scientific Programming
Volume 2021, Article ID 5513860, 14 pages
https://doi.org/10.1155/2021/5513860

Research Article

Ranking Methods for Multicriteria Decision-Making: Application to Benchmarking of Solvers and Problems

Joseph Gogodze

Georgian Technical University, 77, Kostava Str., Tbilisi 0171, Georgia

Correspondence should be addressed to Joseph Gogodze; jgogodze@gmail.com

Received 26 February 2021; Revised 21 May 2021; Accepted 28 May 2021; Published 13 July 2021

Academic Editor: Jianping Gou

Evaluating the performance assessments of solvers (e.g., for computation programs), known as the solver benchmarking problem, has become a topic of intense study, and various approaches have been discussed in the literature. Such a variety of approaches exist because a benchmark problem is essentially a multicriteria problem. In particular, the appropriate multicriteria decision-making problem can correspond naturally to each benchmark problem and vice versa. In this study, to solve the solver benchmarking problem, we apply the ranking-theory method recently proposed for solving multicriteria decision-making problems. The benchmarking problem of differential evolution algorithms was considered for a case study to illustrate the ability of the proposed method. This problem was solved using ranking methods from different areas of origin. The comparisons revealed that the proposed method is competitive and can be successfully used to solve benchmarking problems and obtain relevant engineering decisions. This study can help practitioners and researchers use multicriteria decision-making approaches for benchmarking problems in different areas, particularly software benchmarking.

1. Introduction

Recently, evaluating the performance of solvers (e.g., computer programs), that is, the problem of solver benchmarking, has attracted significant attention from scientists. Currently, most benchmarking tests produce tables that present the performance of each solver for each problem according to a specified evaluation metric (e.g., the central processing unit (CPU) time and number of function evaluations) and use various statistical tests for the conclusions. Thus, the selection of the benchmarking method currently depends on the subjective tastes and preferences of individual researchers. The following components of the benchmarking process, including the solver set, problem set, metric for performance assessment, and statistical tools for data processing, are chosen individually according to the researcher's preferences. For example, the performance profile method, which is currently the most popular and widely used method in practice (see [1]), is based on a comparative analysis of empirical probability distribution functions obtained in numerical experiments with different solvers.

In this study, we consider the benchmarking process based on the viewpoint that emphasizes natural relations between problems and solvers, as determined by their evaluation tables (see [2]). Specifically, we present data for benchmarking in the form of a so-called benchmarking context, that is, a triple $\langle S, P, J\rangle$, where S and P are sets of solvers and problems, respectively, and $J\colon S \times P \longrightarrow \mathbb{R}$ is an assessment function (a performance evaluation metric). Throughout the paper, the sets of solvers and problems are assumed to be finite. This concept is quite general and emphasizes that problems, solvers, and assessment functions must be considered closely related and not independent.

The benchmarking procedure presented in this study is described as follows. The data encapsulated by the given benchmarking context $\langle S, P, J\rangle$ are used to build the corresponding multicriteria decision-making (MCDM) problem $\langle A, C\rangle$, where $A = S$ is a set of alternatives, and $C = \{J(\cdot, p) | p \in P\}$ is a set of criteria. Hence, we define a decision matrix as a matrix whose elements exhibit the performance of different alternatives (i.e., solvers) concerning various criteria (i.e., problems) through the assessment function. Thus, the investigation of benchmarking problems was reduced to an MCDM problem. Moreover, for each MCDM problem, a corresponding benchmark context is presented. The rationale for such a consideration is that a vast array of different approaches for MCDM problems can be used for benchmarking problem analysis. In particular, such a multicriteria formulation allows the consideration of Pareto-optimal alternatives (i.e., solvers) as “good” solvers.

The next innovation presented in this study is that a recently proposed technique (see [3]) is used to solve the MCDM problem corresponding to a benchmarking problem. The multicriteria formulation is a typical starting point for theoretical and practical analyses of decision-making problems to clarify the essence of the new technique used in this study. Correspondingly, based on the fundamental concept of Pareto optimality, several methods and computational procedures have been developed to solve MCDM problems (see, e.g., overviews by [4–8], and more recently, [9–11]). However, unlike single-objective optimizations, a characteristic feature of Pareto optimality is that the set of Pareto-optimal alternatives is typically large. In addition, all these Pareto-optimal alternatives must be considered mathematically equal (equally “good”). Correspondingly, the problem of choosing a specific Pareto-optimal alternative for implementation arises because the final decision must usually be unique. Hence, additional factors must be considered to aid decision-makers in selecting specific or more favorable alternatives from the set of Pareto-optimal solutions.

Therefore, we build a special score matrix for the MCDM problem, which allows us to construct the corresponding ranking for alternatives [3]. The score matrix can be built in different ways, but we use the simplest and most natural method. This study uses a scoring matrix calculating how many times one alternative is better than another according to the criteria. Hence, the proposed approach may yield an “objective” ranking method and provide an “accurate” ranking of the alternatives for MCDM. Correspondingly, a best-ranked alternative from the Pareto set is declared a “true” solution to the MCDM problem. The approach presented in this study for solving MCDM problems is useful when no decision-making authority is available or when the relative importance of various criteria has not been previously evaluated.

Finally, we demonstrate the possibilities of the proposed method in a case study based on the computational and experimental results for benchmarking differential evolution (DE) algorithms presented by Sala et al. [12]. Specifically, we benchmark nine DE algorithms on a set of 50 test problems, following the random sampling equivalent expected run time (ERTRSE) performance metric. By conducting a numerical investigation, we demonstrate that the solution results of the MCDM problem obtained using the methods proposed in this study are quite competitive.

1.1. Contributions.

This paper makes the following main contributions:

(1) The concept of the benchmarking context is introduced according to [2], and it is confirmed that a one-to-one correspondence exists between the set of benchmarking contexts and the set of MCDM problems

(2) The ranking-theory approach is proposed for solving MCDM problems corresponding to a given benchmarking context [3]

(3) The approach proposed in this article is tested on a known literature dataset for benchmarking DE algorithms (see [12]), and the possibility of effectively solving benchmarking problems is fully confirmed

1.2. Related Literature.

Without claiming to be a complete review, we present a brief overview of the literature on the benchmarking problem in the context of optimization problems. Generally, the consideration of a benchmarking problem is motivated by various reasons, such as selecting the best solver (algorithm, software, etc.) for some class of problems, testing the proposed novel solvers, and evaluating the solver performance for different option settings. For example, early contributions in the benchmarking of optimization algorithms are considered [13]. The results achieved at an early stage in the development of the subject can be judged according to work by the following researchers: Nash and Nocedal [14], Billups et al. [15], Conn et al. [16], Sandu et al. [17], Mittelmann [18], Vanderbei and Shanno [19], and Bondarenko et al. [20].

The beginning of a new stage of development is associated with research work of Dolan and Moré [21], in which a performance profile comparison technique was proposed. This technique is now prevalent (but see, e.g., Gould and Scott [22]). Along with the performance profile comparison method, other more direct approaches have also been used in modern research. An idea of the modern research in the area under consideration can be obtained from the following research examples: Moles et al. [23], Mittelmann [24], Benson et al. [25], Kämpf et al. [26], Foster et al. [27], Rios and Sahinidis [28], Weise et al. [29], Sala et al. [12], and Cheshmi et al. [30]. A critical overview of the current state in the subject area was provided by Beiranvand et al. [1].

At the end of this brief overview, this study focuses on benchmarking for solvers of only the optimization problem. However, the concept of benchmarking has a much broader context (see, e.g., https://en.wikipedia.org/wiki/Benchmarking). The approach proposed in this article is quite general and can also be applied in other areas, but we do not consider this possibility here.

1.3. Notation. Throughout the article, the following general notation is used: $\mathbb{N}$ is a set of natural numbers, and for a natural number $n \in \mathbb{N}$, we denote an n-dimensional vector space by $\mathbb{R}^n$ and $\|\cdot\|_p$ is the l_p − norm in $\mathbb{R}^n$. If not otherwise mentioned, we identify a finite set A with set $\mathbb{N}(n) = \{1, \ldots, n\}$, where $n = |A|$ is the capacity of set A. We also introduce the following notations for special vectors and sets: for any $n \in \mathbb{N}$: $0_n = \underbrace{(0, \ldots, 0)}_{n} \in \mathbb{R}^n$, $1_n = \underbrace{(1, \ldots, 1)}_{n} \in \mathbb{R}^n$, $\Delta_n = \{\xi \in \mathbb{R}^n_+ | \sum_{k=1}^n \xi_k\}$, $\Delta^\circ_n = \Delta_n \cap \text{int}\mathbb{R}^n_+$, and $\mathbb{R}^n_+ \subset \mathbb{R}^n$ is a positive orthant. By necessity, we also identify the matrix $\Pi \in \mathbb{R}^{n\times m}$ with the map Π: $\mathbb{N}(n) \times \mathbb{N}(m) \longrightarrow \mathbb{R}$. For a matrix $\Pi \in \mathbb{R}^{n\times m}$, we denote its transpose by $\Pi^T \in \mathbb{R}^{m\times n}$.

1.4. Outline. The remainder of this paper is structured as follows. In Section 2, all necessary theoretical preliminaries regarding the MCDM problem (Section 2.1) and ranking-theory methods for solving MCDM problems are presented (Section 2.2). Section 3 introduces the concept of benchmarking contexts, and its relationship with the MCDM problem is discussed. In Section 4, the case-study problem of DE algorithm benchmarking is investigated numerically. Finally, the conclusions are presented in Section 5.

2. Methodology

2.1. Multicriteria Decision-Making Problems. We use the following notation from the general theory of multicriteria optimization theory [31]. We consider the MCDM problem $\langle A, C\rangle$, where $A = \{a_1, \ldots, a_m\}$ is a set of alternatives, and $C = \{c_1, \ldots, c_n\}$ is a set of criteria, that is, c_i: $A \longrightarrow \mathbb{R}$, $i = 1, \ldots, n$. Hence, we introduce the following decision matrix:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}, \tag{1}$$

where $x_{ij} = c_j(a_i)$ is the performance measure of alternative $i \in \mathbb{N}(m)$ on criterion $j \in \mathbb{N}(n)$. Without loss of generality, we assume that the lower value is preferable for each criterion (i.e., each criterion is not beneficial; see [32]), and the goal of the decision-making procedure is to minimize all criteria simultaneously. Furthermore, A is the set of admissible alternatives, and map $\overrightarrow{c} = (c_1, \ldots, c_n)$: $A \longrightarrow \mathbb{R}^n$ is the criterion map (correspondingly, $\overrightarrow{c}(A) \subset \mathbb{R}^n$ is the set of admissible values of criteria). A point $\xi^I = (\xi^I_1, \ldots, \xi^I_n) \in \mathbb{R}^n$, where $\xi^I_j = \min_{a\in A} c_j(a)$, $j \in \mathbb{N}(n)$, is called the ideal point. An ideal point is considered attainable if alternative $a^I \in A$ exists such that $\xi^I = \overrightarrow{c}(a^I)$. The following concepts are also associated with the criterion map and set of alternatives. An alternative $a_* \in A$ is Pareto-optimal (efficient) if $a \in A$ exists such that $c_j(a) \le c_j(a_*)$ for all $j \in \mathbb{N}(n)$ and $c_k(a) < c_k(a_*)$ for some $k \in \mathbb{N}(n)$. The set of all efficient alternatives is denoted as A_e and is called the Pareto set. Correspondingly, $f(A_e)$ is called an efficient front.

Pareto optimality is an appropriate concept for solutions to MCDM problems in general. However, the set A_e of Pareto-optimal alternatives is very large, and all alternatives from A_e must be considered "equally good solutions." However, the final decision must be unique. Hence, additional factors must be considered to aid in selecting specific or more favorable alternatives from the set A_e. We cannot provide a detailed analysis of these methods; however, interested readers can become acquainted with them through overviews [4–8]. Furthermore, we consider only the method proposed by Gogodze [3] without diminishing the value of more classical methods.

2.2. Ranking Methods and Their Applications to MCDM Problems. This section provides a brief overview of the basic concepts of the ranking theory (e.g., see [33] for further details) and presents the necessary formal definitions. For a natural number N, the matrix $S = [S_{ij}]$, $1 \le i, j \le N$ is a score matrix if $S_{ij} \ge 0, S_{ii} = 0$, $1 \le i, j \le N$, and the pair $(\mathbb{N}(N), S)$ is the ranking problem. We assume (conditionally) that the elements of $\mathbb{N}(N)$ are athletes (or sports teams) who compete in matches between themselves. Moreover, $M(i, j)$ denotes a joint match for each pair of athletes (i, j), $1 \le i, j \le N$, and we interpret entry S_{ij}, $1 \le i, j \le N$, of matrix S as the total score of athlete i against athlete j in match $M(i, j)$. In addition, athlete i scored against athlete j in match $M(i, j)$ if $S_{ij} > 0$, and athlete i has beaten athlete j in match $M(i, j)$ if $S_{ij} > S_{ji}$. Based on the introduced notation, we define the following quantities:

$$\begin{aligned} G &= S + S^T, \text{i.e.}, G = G(S) = \left[G_{ij}(S)\right], \\ G_{ij}(S) &= S_{ij} + S_{ji}, \quad 1 \le i, j \le N; \\ g(S) &= (g_1(S), \ldots, g_N(S)), \\ g_i(S) &= \sum_{j=1}^{N} G_{ij}(S), \quad 1 \le i \le N. \end{aligned} \tag{2}$$

The weak order on the set $\mathbb{N}(N)$ is transitive and the complete relation $R \subset \mathbb{N}(N) \times \mathbb{N}(N)$. The relation $R_N(S) \subset \mathbb{N}(N) \times \mathbb{N}(N)$ is a ranking method if, for any given ranking problem $(\mathbb{N}(N), S), R_N(S)$ is a weak order on the set $\mathbb{N}(N)$. Any vector $r = (r_1, \ldots, r_N) \in \mathbb{R}^N$ can be considered a rating vector for elements of $\mathbb{N}(N)$, in the sense that each r_i, $1 \le i \le N$, can be interpreted as a measure of the performance of player $i \in \mathbb{N}(N)$. For the ranking problem $(\mathbb{N}(N), S)$, a ranking method $R_N(S)$ is induced by the rating vector $r \in \mathbb{R}^N$ if $(i, j) \in R_N(S)$ (i.e., $R_N(S)$ ranks i weakly above j) if and only if $r_i \ge r_j$.

For illustrative purposes, we consider only a few of the many ranking methods discussed in the literature. All of these methods are induced by their corresponding rating vectors. The considered ranking methods originate from different areas, such as athlete/team ranking in sports, citation indices, and website ranking. Hence, all of these reflect some (intuitive as a rule) human experience regarding the solution concept of the ranking problem. A brief overview of the ranking methods in this article is provided in Appendix.

We can unite all the information described above and demonstrate that, for any MCDM problem, we can construct the necessary matrices (e.g., S, $P(S)$, and $\mathfrak{A}(S)$) and, therefore, apply a suitable ranking method for the MCDM problem solution. To simplify the perception of the constructions described below, we use sports terminology. We assume that $\langle A, C\rangle$ is an MCDM problem (see Section 2.1) with a set of alternatives $A = \{a_1, \ldots, a_m\}$ and a set of nonbeneficial criteria $C = \{c_1, \ldots, c_n\}$ and that the decision-making goal is to minimize the criteria simultaneously. We imagine that the number of athletes is $N = m$ for constructing matrix S and that they are competing in an n-athlon (i.e., each match $M(i \cdot j)$ includes competitions in n different disciplines, $1 \le i, j \le n$). For illustrative purposes, we introduce the simplest method for score calculation:

$$S_{ij} = \sum_{k \in \mathbb{N}(n)} s_{ij}^k, \text{where } s_{ij}^k = \begin{cases} 1, & c_k(a_i) < c_k(a_j) \\ 0, & c_k(a_i) \ge c_k(a_j) \end{cases}, \ k \in \mathbb{N}(n),\ i, j \in \mathbb{N}(m). \tag{3}$$

Thus, for criterion $k \in \mathbb{N}(n)$, the equality $s_{ij}^k = 1$ means that $c_k(a_i) < c_k(a_j)$ and the alternative a_i (i.e., athlete $i \in \mathbb{N}(m)$) receives one point (i.e., athlete $i \in \mathbb{N}(m)$ wins the competition in discipline $k \in \mathbb{N}(n)$). Correspondingly, S_{ij} indicates the number of total wins of athlete $i \in \mathbb{N}(m)$ in match $M(i, j)$). Thus, $m \ge S_{ij} \ge 0, S_{ii} = 0, \forall i, j \in \mathbb{N}(m)$. An alternative a_i (athlete $i \in \mathbb{N}(m)$) has defeated an alternative a_j (athlete $j \in \mathbb{N}(m)$) if $S_{ij} > S_{ji}$. In addition, the result of match $M(i, j)$ is S_{ij} wins by athlete $i \in \mathbb{N}(m)$ (losses of athlete $j \in \mathbb{N}(m)$), S_{ji} wins of athlete $j \in \mathbb{N}(m)$ (losses of athlete $i \in \mathbb{N}(m)$), and the number of draws is $(m - S_{ij} - S_{ji})$. The constructed matrix $S = [S_{ij}]$, $1 \le i, j \le m$, is the score matrix for a set of alternatives.

Thus, we can define an auxiliary matrix $\Pi(S) = [\Pi_{ij}(S)]$, $1 \le i, j \le m$, where

$$\Pi_{ij}(S) = \begin{cases} S_{ij} + \dfrac{(m - S_{ij} - S_{ji})}{2}, & i \ne j; \\ 0, & i = j; \end{cases}, \quad \forall i, j \in \mathbb{N}(m). \tag{4}$$

Furthermore, using matrix $\Pi(S)$ and a well-known transformation, we can construct a (row) stochastic matrix $P = P(S)$ as follows: $P(S) = \Lambda(\Pi(S)1_m + \pi(S))^{-1}\Pi(S) + \pi(S)\xi^T$, where a vector $\xi \in \mathring{\Delta}_m$ (usually $\xi = (1/m)1_m$), and $\pi(S) = (\pi_1(S), \ldots, \pi_m(S))$ is a vector defined as follows:

$$\pi_i(S) = \begin{cases} 1, & \text{if } i^{\text{th}} \text{ row of } \Pi(S) \text{ is } 0_m, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The introduced matrix $\Pi(S)$ can be interpreted as an adjacency matrix for a directed graph $\Gamma(A, C)$ (associated with the MCDM problem $\langle A, C\rangle$), called the adjacency matrix for the MCDM problem $\langle A, C\rangle$. Correspondingly, matrix $P(S)$ can be interpreted as a transition probability matrix for the Markov chain determined by the graph $\Gamma(A, C)$. Moreover, we can construct a reciprocal matrix of pairwise comparisons $\mathscr{A}(S) = [\mathfrak{a}_{ij}]$, $i, j = 1, \ldots, m$, for the MCDM problem $\langle A, C\rangle$ as follows:

$$\mathfrak{a}_{ij} = \begin{cases} \exp\left(\Pi_{ij} - \dfrac{m}{2}\right) & i \ne j; \\ 1, & i = j; \end{cases}, \quad i, j = 1, \ldots, m. \tag{6}$$

Subject to the facts presented in this section, the following procedure for solving the MCDM problem under consideration, $\langle A, C\rangle$, can be formulated:

(i) For the MCDM problem $\langle A, C\rangle$, the score matrix $S = [S_{ij}]$, $1 \le i, j \le m$, is constructed

(ii) Using the score matrix S, the alternatives from set A are ranked using a ranking method R ($R \in \{R^S, R^N, R^B, R^C, R^K, R^{PF}, R^{GM}, \ldots\}$; see, e.g., Appendix)

(iii) The alternative from the Pareto set, A_e, ranked best by method R is declared the R solution of the considered MCDM problem

3. Benchmarking Problem

We consider a set P of problems, a set S of solvers, and a function J: $S \times P \longrightarrow \mathbb{R}$, the assessment function (performance metric). The terms "solver," "problem," and "assessment function" are used conditionally only to simplify interpretation, although this is not generally necessary (and, as we observe below, can even lead to terminological inconsistency). Furthermore, we assume for definiteness that the high and low values of J correspond to the worst and best cases, respectively, and for convenience, we interpret $J(s, p)$ as the cost of solving the problem $p \in P$ by the solver $s \in S$. Moreover, the following conditions are assumed:

(i) Slover $s \in S$ solves problem $p \in P$ better than solver $s_* \in S$, if $J(s, p) < J(s_*, p)$

(ii) Problem $p \in P$ is easier for solver $s \in S$ than problem $p_* \in P$, if $J(s, p) < J(s, p_*)$

Thus, we can introduce the following definition, which is sufficient for many real-world applications.

Definition 1. A triple $\langle S, P, J\rangle$ is in the (solvers) benchmarking context if and only if S and P are the finite sets (called a set of solvers and set of problems, respectively), $J\colon S \times P \longrightarrow \mathbb{R}$ is a function (called the assessment function, or performance evaluation metric), and the following assumptions hold:

$$\begin{cases} (A0) & P = \mathbb{N}(n), S = \mathbb{N}(m), \\ (A1) & J(s, p) \geq 0, \forall (s, p) \in S \times P. \end{cases} \quad (7)$$

The presented concept is quite general and, as mentioned, emphasizes that the set of solvers, set of problems, and assessment function must be considered closely related objects for the benchmarking goal and not independently. Assumption $(A0)$ establishes that sets S and P have sizes $m, n \in \mathbb{N}$, respectively, and Assumption $(A1)$ establishes the nonnegativity of the assessment function. Moreover, because sets S and P are finite, Condition $(A1)$ does not limit the generality of our considerations. Generally, the selection of a benchmarking context $\langle S, P, J\rangle$ component is based on the research questions motivated by a benchmarking analysis goal. However, the choice of sets S and P is often a disputable issue in the practice of certain applications. In contrast, the situation is relatively straightforward in choosing the assessment function, J, at least in computer science (see, e.g., [34]). For example, the following indicators are often used in this case: running time (e.g., the CPU time [35]), reliability (i.e., the solver's ability to successfully solve several problems, such as the success rate [36]), and others. Moreover, the case when assessment, J, is a mapping in $\mathbb{R}^l$, where $l \in \mathbb{N}$ (i.e., it is a multiple criterion), also can be considered but we do not delve into this issue. Next, we consider the benchmarking context $\langle S, P, J\rangle$ as given, and introduce the following definition:

Definition 2. For a given (solver) benchmarking context $\langle S, P, J\rangle$, we define function $J^*\colon P \times S \longrightarrow \mathbb{R}$ as follows: $J^*(p, s) = J(s, p), \forall p \in P, \forall s \in S$. We call J^* the adjoint (to J) assessment function, and $\langle P, S, J^*\rangle$ is the adjoint to the $\langle S, P, J\rangle$ benchmarking context or problem benchmarking context (corresponding to the solver benchmarking context $\langle S, P, J\rangle$).

Definition 2 is easily validated as correct (i.e., J^* is the assessment function in the sense of Definition 1). Terminological inconsistency appears, as noted above. In the benchmarking context $\langle P, S, J^*\rangle$, the set of solvers is set P, which is the set of problems in the sense of the benchmarking context $\langle S, P, J\rangle$. We hope that this does not create any problems in understanding the text below.

We assume now that a benchmarking context $\langle S, P, J\rangle$ is given and build a corresponding MCDM problem $\langle A, C\rangle$ as follows: $A = S$ is a set of alternatives, and $C = \{c_p | p \in P\}, c_p(\cdot) = J(\cdot, p)\colon A \longrightarrow \mathbb{R}, \forall p \in P$ is a set of criteria. Hence, we define the decision matrix as a matrix whose elements exhibit the performance of different alternatives (i.e., solvers) with respect to various criteria (i.e., problems) through the assessment function. From Property $(A1)$, $c_p(s) \geq 0 \forall s \in S, 0 \forall p \in P$ In contrast, we assume that $\langle A, C\rangle$, where $A = \{a_1, \ldots, a_m\}$ and $C = \{c_1, \ldots, c_n\}$, is a given MCDM problem such that $c_k(a) \geq 0 \forall a \in A = \mathbb{N}(n)$ and $\forall k \in \mathbb{N}(n) = C$. Hence, for $P = \mathbb{N}(n), S = \mathbb{N}(m)$, and $J(i, k) = c_k(a_i) \forall i \in \mathbb{N}(n), \forall k \in \mathbb{N}(n)$, triplet $\langle S, P, J\rangle$ is a benchmarking context corresponding to the MCDM problem $\langle A, C\rangle$. The correspondences described above are one-to-one and reciprocal. Thus, we prove that the following proposition holds.

Proposition 1. *One-to-one mapping exists between the benchmarking contexts and MCDM problems with nonnegative criteria.*

To summarize the results of this section and achieve greater clarity in the presentation, we formulated the proposed approach to solving benchmarking problems in an algorithmic form. Furthermore, we assumed that the considered benchmarking problem has already been formalized as a benchmarking context $\langle S, P, J\rangle$, where S is a set of solvers, P is a set of problems, and J is an assessment function. The flowchart of the algorithm is presented in Figure 1. All elements of the Pareto set, Ae, are considered equally "good" solvers (in the sense of Pareto optimality). However, the R ranking allows detailed classification to define the "best of the good," "worst of the good," and other intermediate "good" solvers.

4. Case Study: Benchmarking Differential Evolution Algorithms

4.1. Data. In this section, we focus on an illustrative example of the proposed approach in Section 2. Our consideration is based on the results of numerical experiments borrowed from Sala et al. [12], where nine optimization DE algorithms and 25 test functions were considered (see Table 1) for a short description of the test functions. The set of solvers in the benchmarking problem under consideration is the set of the following algorithms:

(i) S_1-DE: rand/1/bin differential evolution [37],

(ii) S_2-DE2: best/2/bin differential evolution [38],

(iii) S_3-jDE: self-adapting differential evolution [39],

(iv) S_4-JADE: adaptive differential evolution [40],

(v) S_5-SaDE: strategy adaptation differential evolution [41],

(vi) $S6$-epsDE: ensemble parameter differential evolution [42],

(vii) S_7-CoDe: composite trial vector strategy differential evolution [43],

(viii) S_8-SQG: stochastic quasigradient search [44], and

(ix) S_9-SQG-DE: stochastic quasigradient-based differential evolution [12].

Thus, $S = \{S_1, \ldots, S_9\}$ is a set of solvers. The set of problems $P = \{P_1, \ldots, P_{50}\}$ comprises 50 problems, and each problem is defined by the dimension indicator $d = 30, 50$ and by the test function types $F_1, \ldots, F_{25}$, as listed in Table 2.

A description of the assessment function used by Sala et al. [12] is as follows: first, the expected running time (ERT), a widely used performance metric for optimization algorithms, is defined as follows:

Input: Benchmarking context <S, P, J>.

Initialization: The MCDM problem <$A(S)$, $C(P, J)$>, where $A(S) = S$ and

$C(P, J) = \{c_p \mid p \in P\}$, $c_p(\cdot) = J(\cdot, p) : A \rightarrow \mathbb{R}$, $\forall\, p \in P$, is introduced.

Step 1: For the MCDM problem <$A(S)$, $C(P, J)$>, the Pareto-set, $A_e \subset A(S)$ isdefined.

Step 2: For the MCDM problem <$A(S)$, $C(P, J)$>, the score matrix $S = [Sij]$, $1 \leq i, j \leq m$, is constructed.

Step 3: An appropriate ranking method R is chosen (see, e.g., Annex A).

Step 4: Using the score matrix S, the alternatives from set A are ranked using ranking method R.

Output: The alternatives from the Pareto set, A_e, ranked using method R are declared the R solutions of the benchmarking problem <$A(S)$, $C(P, J)$>.

Figure 1: Algorithm flowchart for solving a benchmarking problem.

Table 1: Test functions.

	Short description
F_1	Shifted sphere function
F_2	Shifted Schwefel's problem 1.2
F_3	Shifted rotated high conditioned elliptic function
F_4	Shifted Schwefel's problem 1.2 (with noise in fitness function)
F_5	Schwefel's problem 2.6 with the global optimum on the bounds
F_6	Shifted Rosenbrock's function
F_7	Shifted rotated Griewank's function
F_8	Shifted rotated Ackley's function (with the global optimum on the bounds)
F_9	Shifted Rastrigin's function
F_{10}	Shifted rotated Rastrigin's function
F_{11}	Shifted rotated Weierstrass function
F_{12}	Schwefel's problem 2.13
F_{13}	Expanded extended Griewank's plus Rosenbrock's function
F_{14}	Shifted rotated expanded Schaffer's F_6
F_{15}	Hybrid composition function
F_{16}	Rotated hybrid composition function
F_{17}	Rotated hybrid composition function (with noise in fitness function)
F_{18}	Rotated hybrid composition function
F_{19}	Rotated hybrid composition functions (with the global optimum on the bounds)
F_{20}	Rotated hybrid composition function (with a narrow basin for the global optimum)
F_{21}	Rotated hybrid composition function
F_{22}	Rotated hybrid composition function (with a high condition number matrix)
F_{23}	Noncontinuous rotated hybrid composition function
F_{24}	Rotated hybrid composition function
F_{25}	Rotated hybrid composition function

Note: $F_1, \ldots, F_{25}$ are the test functions. Source: Sala et al. [12].

Table 2: Problem description.

Problem	Dim-on	Test function	Problem	Dim-on	Test function	Problem	Dim-on	Test function	Problem	Dim-on	Test function	Problem	Dim-on	Test function
P_1	30	F_1	P_{11}	30	F_{11}	P_{21}	30	F_{21}	P_{31}	50	F_6	P_{41}	50	F_{16}
P_2	30	F_2	P_{12}	30	F_{12}	P_{22}	30	F_{22}	P_{32}	50	F_7	P_{42}	50	F_{17}
P_3	30	F_3	P_{13}	30	F_{13}	P_{23}	30	F_{23}	P_{33}	50	F_8	P_{43}	50	F_{18}
P_4	30	F_4	P_{14}	30	F_{14}	P_{24}	30	F_{24}	P_{34}	50	F_9	P_{44}	50	F_{19}
P_5	30	F_5	P_{15}	30	F_{15}	P_{25}	30	F_{25}	P_{35}	50	F_{10}	P_{45}	50	F_{20}
P_6	30	F_6	P_{16}	30	F_{16}	P_{26}	50	F_1	P_{36}	50	F_{11}	P_{46}	50	F_{21}
P_7	30	F_7	P_{17}	30	F_{17}	P_{27}	50	F_2	P_{37}	50	F_{12}	P_{47}	50	F_{22}
P_8	30	F_8	P_{18}	30	F_{18}	P_{28}	50	F_3	P_{38}	50	F_{13}	P_{48}	50	F_{23}
P_9	30	F_9	P_{19}	30	F_{19}	P_{29}	50	F_4	P_{39}	50	F_{14}	P_{49}	50	F_{24}
P_{10}	30	F_{10}	P_{20}	30	F_{20}	P_{30}	50	F_5	P_{40}	50	F_{15}	P_{50}	50	F_{25}

$$\mathrm{ERT}(\tau) = \mathrm{mean}\left(M_\tau\right) + \frac{1-q}{q} N_{\max},$$
$$q = \frac{N_{\mathrm{succes}}}{N_{\mathrm{total}}}, \tag{8}$$

where τ indicates a reference threshold value, M_τ is the number of function evaluations required to reach an objective value better than τ (e.g., successful runs), $N_{\max}$ denotes the maximum number of function evaluations per optimization run,N_{succes} represents the number of successful runs, N_{total} is the total number of runs, and q denotes the named success rate [45]. The ERT is interpreted as the expected number of function evaluations of an algorithm to reach an objective function threshold for the first time. A threshold or success criterion is required for the ERT performance measure. However, unlike conventional optimization problems (where the ERT criterion is usually related to reaching the value of the known global optimum within a specified tolerance), the probability of coming close to the global optimum is negligible for difficult optimization problems, and a more acceptable alternative success criterion is required. Moreover, all compared algorithms must meet the success criterion a few times to compare qualitative performance using ERT for difficult optimization problems. Correspondingly, Sala et al. [12] used the success criterion to reach the target value corresponding to the expected value of the best objective function value from the uniform random sampling (1000 samples). Next, the estimation of the expected objective value $E_{\mathrm{RSE}}(f)$ for test function f is based on 100 repetitions. Finally, the ERT with respect to this objective function value limit was referred to as $\mathrm{ERT}_{\mathrm{RSE}}$ for test function f. The dataset of $\mathrm{ERT}_{\mathrm{RSE}}$ estimations [12] for the above-described solvers and problems is presented in Table 3.

Thus, the benchmarking context $\langle S, P, J\rangle$, where $S = \{s_1, \ldots, s_9\}, P = \{p_1, \ldots, p_{50}\}$, and J is the $\mathrm{ERT}_{\mathrm{RSE}}$ assessment, is fully defined. Hence, in Section 3, the MCDM problem associated with the benchmarking problem under consideration is fully defined with a set of alternatives $A = S = \mathbb{N}(9)$ a set of (nonbeneficial) criteria $C = P = \mathbb{N}(50)$, and a primary decision matrix obtained by transposing the matrix/table presented in Table 3, which is the transposed primary decision matrix $Z = [z_{ij}], i \in \mathbb{N}(9), j \in \mathbb{N}(50)$, for writing convenience. Hence, the MCDM problem associated with the benchmarking context $\langle S, P, J\rangle$ (i.e., the solver benchmarking problem) is fully defined. The benchmarking context $\langle P, S, J^*\rangle$ is analogously defined, where the assessment function J^* is obtained based on the decision matrix Z^* (which is the transpose of the decision matrix Z defined above). Hence, the MCDM problem associated with this benchmarking context (i.e., the benchmark problem for the problems) is also fully defined.

4.2. Calculation Results. In this section, we present a brief description of the calculation results (all calculations related to the case study were calculated in the MATLAB environment using standard equipment: laptop with 2.59 GHz, 8 GB RAM, and a 64 bit operation system and required a few seconds (4.87 s for the solver benchmarking and 5.04 s for the problem benchmarking for calculating all considered rankings without special code optimization measures). First, we consider the solver benchmarking problem and explain the construction of the normalized decision matrix by transforming the primary dataset (see, e.g., [32]).

For the primary decision matrix $Z = [z_{ij}]$, we define the normalized decision matrix $X = [x_{ij}]$ as $x_{ij} = (z_{ij} - l_j)/(u_j - l_j)$ where $u_j = \max_{i \in \mathbb{N}(9)}\{z_{ij}\}, l_j = \min_{i \in \mathbb{N}(9)}\{z_{ij}\}$, $j \in \mathbb{N}(50)$. For the solver benchmarking problem, we consider all criteria to be nonbeneficial (i.e., minimizable). We consider a solver to be better if it solves a given problem in less time $(\mathrm{ERT}_{\mathrm{RSE}})$.

To illustrate this, we present the score matrix for the solver benchmarking problem in Table 4. Table 5 presents the obtained $R^S, R^N, R^B, R^C, R^K, R^{PF}$, and R^{GM} ranks for the solver benchmarking problem.

Analogously, we consider the problem benchmarking but define the normalized decision matrix $X = [x_{ij}]$ as follows: $x_{ij} = (z^*_{ij} - l^*_j)/(u^*_j - l^*_j)$ where $u^*_j = \max_{i \in \mathbb{N}(50)}\{z^*_{ij}\}, l^*_j = \min_{i \in \mathbb{N}(50)}\{z^*_{ij}\}$, and $Z^* = [z^*_{ij}]$, is the corresponding primary decision matrix. For the benchmarking problem, we also assume that all criteria are nonbeneficial (i.e., minimizable). Again, a problem is better (i.e., easier) for a given solver if it is solved in less time $(\mathrm{ERT}_{\mathrm{RSE}})$ by this solver. Table 6 presents the R^S, R^N, R^B, R^C, R^K, R^{PF}, and R^{GM} ranks for the problem benchmarking (the score matrix for the problem benchmarking is not presented).

4.3. Discussion. As Table 5 indicates, the results of solver ranking using the considered methods $(R^S, R^N, R^B, R^C, R^K, R^{PF}$, and $R^{GM})$ are somewhat similar. This observation was confirmed quantitatively by considering the Spearman correlations between ranks (Table 7), where the correlations of the solver ranks for the $R^S, R^N, R^B, R^C, R^K, R^{PF}$, and R^{GM} rankings are presented. As Table 7 demonstrates, the $R^S, R^N, R^B, R^C, R^K, R^{PF}$, and R^{GM} ranks are strongly correlated with each other. Analogously, Table 8 reflects the interrelation between ranks for problem benchmarking. In particular, $R^S, R^N, R^B, R^C, R^K, R^{PF}$, and R^{GM} ranks are strongly correlated with each other.

Regarding the results of the correlation analysis, the observed similarity of the ranking results for $R^S, R^N, R^B, R^C, R^K, R^{PF}$, and R^{GM} ranking methods appears very intriguing, given that these methods have completely different areas of origin and underlying ideas (see the corresponding scholium in Appendix). It is interesting to consider the Pareto optimization results (see the solvers and problems marked in gray in Table 5 and 4, respectively). In particular, from Table 5, all considered solvers were Pareto-optimal (i.e., they are considered “equally good” in the considered benchmarking context). We believe that this is due to the large (compared to the number of solvers) number of problems (i.e., too many criteria exist in the

Table 3: $\mathrm{ERT}_{\mathrm{RSE}}$ metric.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
P_1	11657	24049	491	264	329	904	606	202	168
P_2	7401	11567	739	309	379	894	3653	137	318
P_3	6536	32763	606	356	519	1148	1045	135	210
P_4	6182	13326	570	285	342	662	1712	694	310
P_5	4342	4766	536	311	527	871	564	459	160
P_6	7404	11728	491	234	315	807	603	124	168
P_7	256	124	482	310	596	879	169	32365	107
P_8	2877	3011	1783	2084	2673	2720	2082	132	2771
P_9	15735	24058	467	269	380	818	726	126	193
P_{10}	11658	24054	474	256	320	850	630	201	172
P_{11}	2414	1555	1967	1502	1321	1541	1979	368	1949
P_{12}	1072	934	509	342	416	931	594	128	225
P_{13}	15709	10150	524	224	231	723	898	161	289
P_{14}	7497	19174	2735	1735	1509	2366	6379	6899	1658
P_{15}	437	373	668	366	565	857	458	1016	185
P_{16}	2999	11671	490	335	615	689	616	219	179
P_{17}	6925	11670	514	362	572	845	693	9096	178
P_{18}	1017	1036	501	314	460	741	341	263	143
P_{19}	1271	1045	533	317	551	836	389	259	148
P_{20}	1190	1098	491	315	545	1039	384	284	154
P_{21}	9466	24251	498	253	327	901	597	301	175
P_{22}	2433	2889	489	281	389	730	602	13320	202
P_{23}	13650	24194	474	254	325	802	586	664	181
P_{24}	7785	4449	544	272	376	911	573	10183	187
P_{25}	220	115	445	304	554	855	167	5321	109
P_{26}	19030	9054	439	205	254	674	579	192	178
P_{27}	9055	8145	572	316	388	908	2135	127	296
P_{28}	3819	10317	514	289	455	764	566	121	145
P_{29}	19059	10175	538	341	434	920	2931	1332	310
P_{30}	9173	19040	568	294	366	964	1418	451	214
P_{31}	11557	13327	491	235	244	831	609	132	182
P_{32}	394	187	443	289	536	884	231	24041	109
P_{33}	1732	2347	2098	1792	2724	2522	2740	117	1960
P_{34}	15719	19060	482	245	295	845	738	123	194
P_{35}	24071	24059	456	235	306	739	661	169	188
P_{36}	2024	2568	2187	1324	1814	1825	1675	347	1331
P_{37}	1183	1263	560	340	418	1132	774	119	224
P_{38}	7381	9057	514	227	193	604	1139	145	336
P_{39}	7476	15824	1792	1310	1198	1296	3482	4706	1808
P_{40}	341	291	557	354	556	852	331	205	173
P_{41}	7506	11568	495	320	487	738	593	260	156
P_{42}	11585	32369	640	327	515	794	809	5743	192
P_{43}	10264	19135	631	319	380	1071	892	352	226
P_{44}	24351	7387	511	290	389	866	728	307	210
P_{45}	19307	13407	533	321	372	915	794	380	227
P_{46}	13571	9089	486	243	320	871	605	325	176
P_{47}	3311	5068	564	291	431	755	724	19048	153
P_{48}	7468	7544	483	248	333	880	590	537	170
P_{49}	15709	13335	498	231	245	784	688	8170	201
P_{50}	304	163	488	315	552	1038	210	13368	109

Note: $S_1, \ldots, S_9$ are solvers, $P_1, \ldots, P_{50}$ are problems, and $\mathrm{ERT}_{\mathrm{RSE}}$ metric is assessment function (as described in the main text of the article, see Section 4.1). Source: Sala et al. [12].

corresponding MCDM problem) and, accordingly, each solver is good in "its own way." However, ranking methods enable the establishment of an appropriate hierarchy among solvers. Analogously, Table 6 demonstrates that Pareto-optimal problems are allocated to different groups or clusters, indicating similar problems belonging to the same

Table 4: Score matrix for the solver benchmarking problem.

0	30	8	5	7	7	2	9	1
20	0	7	5	7	7	8	8	1
42	43	0	1	10	46	40	17	2
45	45	49	0	46	49	44	24	9
43	43	40	4	0	49	39	20	6
43	43	4	1	1	0	12	15	3
48	42	10	6	11	38	0	14	1
41	42	33	26	30	35	36	0	17
49	49	48	41	44	47	49	33	0

Table 5: Solver ranking using the proposed methods.

	Ranking						
	R^S	R^N	R^B	R^C	R^K	R^{PF}	R^{GM}
S_1	8	9	8	8	8	8	8
S_2	9	8	9	9	9	9	9
S_3	5	5	5	5	5	5	5
S_4	2	2	2	3	3	2	2
S_5	4	4	4	4	4	3	4
S_6	7	7	7	7	7	7	7
S_7	6	6	6	6	6	6	6
S_8	3	3	3	2	2	4	3
S_9	1	1	1	1	1	1	1

Note: the Pareto-optimal solvers are gray marked.

Table 6: Problem ranking using the proposed methods.

	Ranking						
	R^S	R^N	R^B	R^C	R^K	R^{PF}	R^{GM}
P_1	24	25	25	25	27	25	24
P_2	38	37	37	40	38	37	38
P_3	42	43	42	43	43	41	42
P_4	34	34	34	35	34	35	34
P_5	26	26	26	27	26	26	25
P_6	4	4	4	7	6	3	4
P_7	7	7	7	5	5	7	7
P_8	45	44	45	48	45	46	45
P_9	25	24	24	24	23	23	26
P_{10}	18	16	18	12	14	18	18
P_{11}	47	47	47	47	47	45	47
P_{12}	22	23	22	28	25	21	22
P_{13}	20	20	20	18	17	22	20
P_{14}	50	50	50	50	50	50	50
P_{15}	33	32	33	31	32	32	33
P_{16}	27	27	27	23	24	28	27
P_{17}	37	38	38	36	36	38	37
P_{18}	3	3	3	4	4	4	3
P_{19}	9	12	9	15	13	11	9
P_{20}	13	15	13	14	15	15	13
P_{21}	29	30	29	26	30	29	29
P_{22}	14	14	14	22	16	13	14
P_{23}	23	21	23	21	22	24	23
P_{24}	31	31	31	32	31	33	31
P_{25}	2	2	2	2	2	2	2
P_{26}	1	1	1	1	1	1	1
P_{27}	36	36	36	38	37	36	36
P_{28}	6	6	6	6	7	6	6
P_{29}	48	48	48	44	48	48	48

Table 6: Continued.

	Ranking						
	R^S	R^N	R^B	R^C	R^K	R^{PF}	R^{GM}
P_{30}	39	39	39	37	39	39	39
P_{31}	8	8	8	11	10	8	8
P_{32}	5	5	5	3	3	5	5
P_{33}	44	40	44	49	42	42	44
P_{34}	19	19	19	19	19	20	19
P_{35}	17	13	17	8	11	17	17
P_{36}	46	46	46	45	46	47	46
P_{37}	32	33	32	33	33	27	32
P_{38}	11	9	11	9	8	14	11
P_{39}	49	49	49	46	49	49	49
P_{40}	16	18	16	20	20	10	15
P_{41}	21	22	21	16	21	19	21
P_{42}	41	41	41	42	41	43	41
P_{43}	43	45	43	41	44	44	43
P_{44}	35	35	35	34	35	34	35
P_{45}	40	42	40	39	40	40	40
P_{46}	15	17	15	17	18	16	16
P_{47}	28	28	28	30	29	31	28
P_{48}	10	11	10	13	12	12	10
P_{49}	30	29	30	29	28	30	30
P_{50}	12	10	12	10	9	9	12

Note: the Pareto-optimal problems are gray marked.

Table 7: Spearman correlations for the solver benchmarking problem.

Ranks	R^S	R^N	R^B	R^C	R^K	R^{PF}	R^{GM}
R^S	1						
R^N	0.983	1					
R^B	0.999	0.983	1				
R^C	0.983	0.967	0.983	1			
R^K	0.983	0.967	0.983	0.999	1		
R^{PF}	0.983	0.967	0.983	0.950	0.950	1	
R^{GM}	0.999	0.983	0.999	0.983	0.983	0.983	1

Table 8: Spearman correlations for the benchmarking problem.

Ranks	R^S	R^N	R^B	R^C	R^K	R^{PF}	R^{GM}
R^S	1						
R^N	0.996	1					
R^B	0.999	0.996	1				
R^C	0.977	0.980	0.977	1			
R^K	0.990	0.996	0.991	0.989	1		
R^{PF}	0.993	0.989	0.993	0.968	0.982	1	
R^{GM}	0.999	0.995	0.999	0.977	0.990	0.993	1

clusters. Ranking methods also make it possible to establish an appropriate hierarchy among the problems.

Summarizing the results of the case-study investigation, we conclude the following:

(i) The results of the calculations (Table 5) confirm that the SQG-DE algorithm (solver S_9) is the best in the considered benchmarking context (for comparison, see [12]), and this conclusion is correct for all rankings used in this study, despite their quite different natures. Moreover, the worst results are DE2 (solver S_2) according to all considered ranking methods, excluding Neustadt's method, and DE (solver S_1) according to Neustadt's ranking method.

(ii) Unlike Sala et al. [12], where the analysis of the problems was not carried out, our calculations also indicate (Table 6) that the best problems in the considered benchmarking context (in the sense of a lower value of the considered metric) are the shifted sphere function in Dimension 50 (problem 26) and the rotated hybrid composition function in

Dimension 30 (problem 25). Accordingly, the worst ones are the shifted rotated expanded Schaffer's function 6 in Dimension 30 (problem P_{14}), and the next worst ones are the shifted rotated Ackley's function in Dimension 50 (problem P_{33}) (by the Colley ranking method) and the shifted rotated expanded Schaffer's function 6 in Dimension 50 (problem P_{39}) (for all other considered ranking methods).

We stress that these results were obtained using only the ranking-theory methods without an analysis of any statistical indicators of the assessment function values, as currently practiced (see, e.g., the related literature overview in the Introduction section).

5. Conclusions

In this study, we presented a new MCDM technique for solving decision-making problems for benchmarking. Our investigation was based on the concept of a benchmarking context, presented in detail, and the observation that a benchmarking problem is an MCDM problem. Correspondingly, to solve benchmarking problems successfully, an extensive array of MCDM methods can be used. We also presented a new approach to the MCDM problem solution based on the ranking-theory methods. The corresponding ranks are obtained by constructing a special score matrix. We emphasize that this method defines the appropriate ranks directly from the decision matrix and does not use preliminary assessments conducted by external experts or other methods. Therefore, the technique presented in this study is useful when the relative importance of various criteria has not been evaluated in advance. As a case study, the benchmarking problem of DE algorithms was considered based on the data presented by Sala et al. [12]. A detailed numerical investigation was conducted using various ranking methods. Moreover, these ranks were also correspondingly compared for solvers and problems. The results demonstrate that the method presented in this study is competitive and generates relevant solutions.

Referring to the analysis presented in this study, we conclude the following:

(i) The results of applying MCDM methods to aid benchmarking problem solutions based on the proposed approach are encouraging.

(ii) The proposed approach provides a constructive view of the benchmarking problem solution, identifying the "best" and "worst" cases and ordering all intermediate cases.

(iii) The proposed approach is easily implementable because of its simplicity and flexibility. Moreover, the approach is sufficiently general and can be successfully used to investigate benchmarking problems in other application areas.

However, this study has limitations because we provided a tool for benchmarking only in the case in which the benchmarking context is given (i.e., when the sets of solvers (problems), problems (solvers), and performance metrics are given). However, issues regarding selecting benchmarking context components remain unresolved. The literature does not contain clear and direct recommendations regarding the correct selection of solvers, problems, and performance metrics. Hence, further investigation in this direction will be helpful.

Appendix

A. Ranking Methods

Let us assume that the ranking problem $(\mathbb{N}(N), S)$ is given and considered as examples the following rating methods (see notations and concepts in Section 2.2) for it.

A.1. Score Method. The rating vector for the score method $r^S = (r_1^S, \ldots, r_N^S) \in \mathbb{R}^N$ is defined as the average score:

$$r_i^S = \sum_{j=1}^{N} \frac{S_{ij}}{g_i(S)}, \quad 1 \leq i \leq N, \tag{A.1}$$

and ranking defined by rating vector r^S will be call R^S rank.

A.2. Neustadt's Method. Neustadt's rating vector $r^N \in \mathbb{R}^N$ is defined by the equality $r^N = \overline{S} r^S$, where $\overline{S} = [\overline{S}_{ij}]$, $\overline{S}_{ij} = (S_{ij}/g_i(S))$, $1 \leq i, j \leq N$ and ranking defined by rating vector r^N will be call R^N rank.

A.3. Buchholz's Method. Buchholz's rating vector (the score, Neustadt, and Buchholz methods were used in practice of chess tournament, and go back to the investigations of H. Neustadt, E. Zermelo, and B. Buchholz, for more details see [33]) $r^B \in \mathbb{R}^N$ is defined by the equality

$$r^B = \left[\overline{G}(S) + E_N\right] r^S, \tag{A.2}$$

where $\overline{G}(S) = [\overline{G}_{ij}(S)], \overline{G}_{ij}(S) = (G_{ij}(S)/g_i(S))$, $1 \leq i, j \leq N$, and ranking defined by rating vector r^B will be call R^B rank.

A.4. Colley Method. We describe the Colley method (Colley, W., Colley's college football ranking method, https://www.colleyrankings.com/, (accessed 12.08.2020)) as follows: let N be the number of athletes/teams, w_i is the number of wins for athletes/team i, l_i is the number of losses for athlete/team i, n_i is the total number of games played by athletes/team i, and n_{ij} is the number of times athlete/teams i, j played, $i, j = 1, \ldots, N$. From these data, the following objects can be introduced: the Colley matrix, $C = [C_{ij}]_{i,j=1,\ldots,N}$, and the Colley vector $v^C = (v_1^C, \ldots, v_N^C)$, where

$$C_{ij} = \begin{cases} 2 + n_i, & i = j, \\ -n_{ij}, & i \neq j, \end{cases}$$
$$v_i^C = 1 + \frac{1}{2}(w_i - l_i), \quad i, j = 1, \ldots, N. \tag{A.3}$$

Now, using score matrix $S = [S_{ij}]$, $1 \le i, j \le N$, we define quantities w_i, l_i, n_i, n_{ij} as follows:

$$w_i = \sum_{j=1}^{N} \chi\left(S_{ij} - S_{ji}\right), l_i = \sum_{j=1}^{N} \chi\left(S_{ji} - S_{ij}\right), \quad n_{ij} = \chi\left(S_{ij} - S_{ji}\right) + \chi\left(S_{ji} - S_{ij}\right). \tag{A.4}$$

$1 \le i, j \le N$ and, obviously, $n_i = w_i + l_i = \sum_{j=} 1^N n_{ij}$, $1 \le i, j \le N$. The Colley rating vector, r^C, is obtained as a solution of the equation $Cr^C = v^C$, and the ranking defined by rating vector r^C is called R^C rank.

A.5. Keener Method. We describe the Keener method [46] as follows: let N be the number of athletes/teams and $S = [S_{ij}]$, $i, j = 1, \dots, N$ is the corresponding score matrix. Keener matrix $K = [K_{ij}]_{i.j=1,\dots,N}$ is defined as follows:

$$K_{ij} = \begin{cases} \kappa\left(\dfrac{1 + S_{ij}}{2 + S_{ij} + S_{ji}}\right) & \text{if teams } i \text{ and } j \text{ played each other} \\ 0 & \text{otherwise} \end{cases}, \quad i, j = 1, \dots, N, \tag{A.5}$$

where

$$\kappa(x) = \frac{1}{2} + \frac{1}{2}\operatorname{sgn}\left(x - \frac{1}{2}\right)\sqrt{|2x - 1|}. \tag{A.6}$$

Correspondingly, the rating vector for the Keener method r^K is obtained as a solution of the eigenvalue problem $Kr^K = \lambda r^K$ and the ranking defined by rating vector r^K is called R^K rank.

A.6. Analytical Hierarchy Process. The analytical hierarchy process (AHP) is a well-known decision-making method [47]. Many modifications of this method exist, but we restrict ourselves to considering only two of them: AHP Perron–Frobenius version (AHP$_{PF}$) and AHP geometric mean version (AHP$_{GM}$), which are briefly described below. A main problem related to AHP is the inconsistency problem (of a pairwise comparison matrix). We will not discuss this problem here because of its technical nature. Therefore, we consider AHP only as a procedure for constructing a rating vector. Let us assume again that N is the number of athletes/teams, which should be ranked based on the score matrix $S = [S_{ij}]$, $i, j = 1, \dots, N$. We also assume that the score matrix S allows the construction of a matrix $\mathfrak{A} = \mathfrak{A}(S)$ which is the reciprocal matrix of pairwise comparisons. Recall that matrix $\mathfrak{A} = [\mathfrak{a}_{ij}]$, $i, j = 1, \dots, N$, is called the reciprocal matrix of pairwise comparisons if it has the following properties: $\mathfrak{a}_{ij} > 0, \mathfrak{a}_{ii} = 1, \mathfrak{a}_{ij} = \mathfrak{a}_{ji}^{-1}$, $\forall i, j \in \{1, \dots, m\}$. Note also that for a positive reciprocal matrix $\mathfrak{A}$, its principal eigenvalue $\lambda_{\max}$ has following properties: $\lambda_{\max} \ge n$ and if $\lambda_{\max} \ne n$ we have an inconsistency problem. The *AHP$_{PF}$* rating vector r^{PF} is defined as the solution to the eigenvalue problem: $\mathfrak{A}r^{PF} = \lambda_{\max} r^{PF}$, with the principal eigenvalue $\lambda_{\max}$, and the corresponding ranking is called R^{PF} rank. At the other hand, the AHP$_{GM}$ rating vector $r^{GM} = (r_1^{GM}, \dots, r_m^{GM})$ is defined as follows:

$$r_i^{GM} = \sqrt[n]{\prod_{j=1}^{m} \mathfrak{a}_{ij}}, \quad 1 \le i \le m, \tag{A.7}$$

and corresponding ranking will be call R^{GM} rank.

Data Availability

The data of Sala et al. [12] were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest regarding this article.

References

[1] V. Beiranvand, W. Hare, and Y. Lucet, "Best practices for comparing optimization algorithms," *Optimization and Engineering*, vol. 18, no. 4, pp. 815–848, 2017.

[2] J. Gogodze, "Benchmarking of problems and solvers: a game-theoretic approach," *Foundations of Computing and Decision Sciences*, vol. 44, 2019.

[3] J. Gogodze, "Ranking-theory methods for solving multicriteria decision-making problems," *Advances in Operations Research*, vol. 2019, Article ID 3217949, 2019.

[4] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[5] E. K. Zavadskas and Z. Turskis, "Multiple criteria decision making (Mcdm) methods in economics: an overview/daugiatiksliai sprendimų priėmimo metodai ekonomikoje: apžvalga," *Technological and Economic Development of Economy*, vol. 17, no. 2, pp. 397–427, 2011.

[6] D. Stanujkic, B. Djordjevic, and M. Djordjevic, "Comparative analysis of some prominent MCDM methods: a case of ranking Serbian banks," *Serbian Journal of Management*, vol. 8, no. 2, pp. 213–241, 2013.

[7] M. Velasquez and P. T. Hester, "An analysis of multi-criteria decision-making methods," *International Journal of Operations Research*, vol. 10, no. 2, pp. 56–66, 2013.

[8] R. K. Gavade, "Multi-criteria decision making: an overview of different selection problems and methods," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp. 5643–5646, 2014.

[9] J. Dong, S. Wan, and S.-M. Chen, "Fuzzy best-worst method based on triangular fuzzy numbers for multi-criteria decision-making," *Information Sciences*, vol. 547, pp. 1080–1104, 2021.

[10] S.-p. Wan, Z.-h. Chen, and J.-y. Dong, "An integrated interval type-2 fuzzy technique for democratic-autocratic multi-criteria decision making," *Knowledge-Based Systems*, vol. 214, Article ID 106735, 2021.

[11] S. Wan, J. Dong, and S. M. Chen, "Fuzzy best-worst method based on generalized interval-valued trapezoidal fuzzy numbers for multi-criteria decision-making," *Information Sciences*, 2021.

[12] R. Sala, N. Baldanzini, and M. Pierini, "SQG-Differential Evolution for difficult optimization problems under a tight function evaluation budget," in *International Workshop on Machine Learning, Optimization, and Big Data*, pp. 322–336, Springer, Cham, Switzerland, 2017.

[13] M. J. Box, "A comparison of several current optimization methods, and the use of transformations in constrained problems," *The Computer Journal*, vol. 9, no. 1, pp. 67–77, 1966.

[14] S. G. Nash and J. Nocedal, "A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization," *SIAM Journal on Optimization*, vol. 1, no. 3, pp. 358–372, 1991.

[15] S. C. Billups, S. P. Dirkse, and M. C. Ferris, *A Comparison of Algorithms for Large Scale Mixed Complementarity Problems*, University of Colorado, Denver, CO, USA, 1995.

[16] A. R. Conn, N. Gould, and P. L. Toint, "Numerical experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization," *Mathematical Programming*, vol. 73, no. 1, pp. 73–110, 1996.

[17] A. Sandu, J. G. Verwer, M. Van Loon et al., "Benchmarking stiff ode solvers for atmospheric chemistry problems-I. implicit vs explicit," *Atmospheric Environment*, vol. 31, no. 19, pp. 3151–3166, 1997.

[18] H. D. Mittelmann, "Benchmarking interior point LP/QP solvers," *Optimization Methods and Software*, vol. 11, no. 1–4, pp. 655–670, 1999.

[19] R. J. Vanderbei and D. F. Shanno, "An interior-point algorithm for nonconvex nonlinear programming," *Computational Optimization and Applications*, vol. 13, no. 1–3, pp. 231–252, 1999.

[20] A. S. Bondarenko, D. M. Bortz, and J. J. Moré, *COPS: Large-Scale Nonlinearly Constrained Optimization Problems*, Argonne National Lab., IL, USA, 2000.

[21] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[22] N. Gould and J. Scott, "A note on performance profiles for benchmarking software," *ACM Transactions on Mathematical Software (TOMS)*, vol. 43, no. 2, pp. 1–5, 2016.

[23] C. G. Moles, P. Mendes, and J. R. Banga, "Parameter estimation in biochemical pathways: a comparison of global optimization methods," *Genome Research*, vol. 13, no. 11, pp. 2467–2474, 2003.

[24] H. D. Mittelmann, "An independent benchmarking of SDP and SOCP solvers," *Mathematical Programming*, vol. 95, no. 2, pp. 407–430, 2003.

[25] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei, "Interior-point methods for nonconvex nonlinear programming: jamming and numerical testing," *Mathematical Programming*, vol. 99, no. 1, pp. 35–48, 2004.

[26] J. H. Kämpf, M. Wetter, and D. Robinson, "A comparison of global optimization algorithms with standard benchmark functions and real-world applications using EnergyPlus," *Journal of Building Performance Simulation*, vol. 3, no. 2, pp. 103–120, 2010.

[27] J. D. Foster, A. M. Berry, N. Boland, and H. Waterer, "Comparison of mixed-integer programming and genetic algorithm methods for distributed generation planning," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 833–843, 2013.

[28] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013.

[29] T. Weise, R. Chiong, J. Lassig et al., "Benchmarking optimization algorithms: an open source framework for the traveling salesman problem," *IEEE Computational Intelligence Magazine*, vol. 9, no. 3, pp. 40–52, 2014.

[30] K. Cheshmi, D. M. Kaufman, S. Kamil, and M. M. Dehnavi, "NASOQ: numerically accurate sparsity-oriented QP solver," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 96–101, 2020.

[31] M. Ehrgott, *Multicriteria Optimization*, Springer Science & Business Media, Berlin, Germany, 2005.

[32] R. T. Marler and J. S. Arora, "Function-transformation methods for multi-objective optimization," *Engineering Optimization*, vol. 37, no. 6, pp. 551–570, 2005.

[33] J. González-Díaz, R. Hendrickx, and E. Lohmann, "Paired comparisons analysis: an axiomatic approach to ranking methods," *Social Choice and Welfare*, vol. 42, no. 1, pp. 139–169, 2014.

[34] P. Hansen, B. Jaumard, and S. H. Lu, "Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison," *Mathematical Programming*, vol. 55, no. 1–3, pp. 273–292, 1992.

[35] C. C. McGeoch, "Experimental analysis of algorithms," in *Handbook of Global Optimization*, R. Horst and P. M. Pardalos, Eds., Springer Science & Business Media, Berlin, Germany, 2013.

[36] R. S. Barr, B. L. Golden, J. P. Kelly, M. G. C. Resende, and W. R. Stewart, "Designing and reporting on computational experiments with heuristic methods," *Journal of Heuristics*, vol. 1, no. 1, pp. 9–32, 1995.

[37] R. M. Storn and K. Price, "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces," *Technical report*, International Computer Science Institute, Berkeley, CA, USA, 1995.

[38] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[39] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[40] Z. J. Jingqiao and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[41] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2008.

[42] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with

ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

[43] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.

[44] Y. M. Ermol'ev, "Methods of solution of nonlinear extremal problems," *Cybernetics*, vol. 2, no. 4, pp. 1–4, 1966.

[45] A. Auger and N. Hansen, "Performance evaluation of an advanced local search evolutionary algorithm," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, Edinburgh, UK, September 2005.

[46] J. P. Keener, "The perron-frobenius theorem and the ranking of football teams," *SIAM Review*, vol. 35, no. 1, pp. 80–93, 1993.

[47] T. Saaty, *The Analytic Hierarchy Process*, McGraw-Hill, New York, NY, USA, 1980.