Multiple Sequence Alignment with Multiobjective Metaheuristics. A Comparative Study

Cristian Zambrano-Vega, Antonio J. Nebro, Juan J. Durillo,
José García-Nieto, José F. Aldana-Montes

Multiple sequence alignment (MSA) plays a core role in most bioinformatics studies and provides a framework for the analysis of evolution in biological systems. The MSA problem consists in finding an optimal alignment of three or more sequences of nucleotides or amino acids. Different scores have been defined to assess the quality of MSA solutions, so the problem can be formulated as a multiobjective optimization problem. The number of proposals focused on this approach in the literature is scarce, and most of the works take as base algorithm the NSGA-II metaheuristic. So, there is a lack of a study involving a set of representative multiobjective metaheuristics to deal with this complex problem. Our main goal in this paper is to carry out such study. We propose a biobjective formulation for the MSA and perform an exhaustive comparative study of six multiobjective algorithms. We have considered a number of problems taken from the benchmark BAliBASE (v3.0). Our experiments reveal that the classic NSGA-II algorithm and MOCell, a cellular metaheuristic, provide the best overall performance.


1. INTRODUCTION

The alignment of multiple DNA, RNA, and protein sequences (multiple sequence alignment, MSA) is a common task in bioinformatics[1] that plays a central role in the analysis of evolution in biological systems. The aim of MSA is comparing different sequences to extract their shared information and their significant differences. The accuracy of such alignments may influence the success of downstream analyses such as phylogenetic inference, protein structure prediction, and functional prediction. The alignment of pair of sequences can be achieved by using dynamic

programming techniques.[2] However, these strategies cannot be applied when dealing with three or more sequences, because the search space grows exponentially with the number of sequences and it is also dependent on the sequence lengths.[3] These reasons have led to the use of metaheuristics to deal with MSA problems.[4]

The basic alignment procedure is based on inserting gaps into the sequences to, first, make all of them have the same length, and second, to foster the alignment of columns of the sequences by manipulating gaps (inserting, deleting, shifting, grouping, etc). The number of gaps and their particular locations determine the quality of the final alignment.

A number of different methods to measure the accuracy of an alignment have been proposed, such as TC (percentage of aligned columns), NonGaps (percentage of nongaps), SOP (sum of pairs), STRIKE,[5] Entropy,[6] BAliScore[7] or MetAl.[8] However, there is still no consensus about which score is the most precise to measure the quality of an alignment. In this context, it does make sense to consider a multiobjective formulation of the problem, which would allow to get simultaneously results of two or more indicators, in order for the biologist to have a set of solutions that would provide her/him with the opportunity to choose the best trade-off solution. However, to the best of our knowledge, comparative studies dealing with MSA by means of multiobjective metaheuristics are scarce.

Therefore, our main motivation in this work is to fill this gap by presenting a comparative study involving a number of representative multiobjective metaheuristics of the state-of-the-art. In this regard, our first approximation to this idea was presented in a previous work,[10] where we compared five metaheuristics: NSGA-II,[9] SPEA2,[11] AbYSS,[12] MOCell,[13] and SMS-EMOA[14] in the scope of a summarized benchmark of five MSA instances. In this last study, NSGA-II outperformed the other techniques. Nevertheless, this result could be biased by the particular structure of the limited set of instances used and, consequently, an extensive experimentation should be performed to shedding light on which multiobjective metaheuristic shows a prominent performance for MSA in a wider and varied set of benchmarking instances.

In this regard, in this work we go one step beyond by offering an extension of Ref. 10, in which we include a series of new contributions, which are summarized as follows:

- The number of solved problems has been risen from 5 to 20 (taken from the benchmark BAliBASE 3.0).[7] This way, we aim at avoiding possible bias in general results as the set of instances is large and varied enough to cover, as much as possible, a significant variety of MSA problem structures.
- Two additional algorithms, MOEA/D[15] and GWASFGA,[16] have been incorporated. The former is a popular multiobjective metaheuristic that shows prominent results on complex structured optimization problems. The latter is a modern algorithm with promising performance whose behavior is still pending to be studied in real-world problems like MSA.
- A thorough comparative analysis is performed, to study the different learning procedures induced by selected algorithms when tackling the MSA problem.
- A rigorous analysis of the obtained results is performed by involving: different quality indicators for multiobjective approaches and statistical significant validation.

The remaining of this paper is structured as follows: Section 2 presents a review of related works in the current literature. In Section 3, the MSA problem is defined and the considered multiobjective approach is presented. After this, selected algorithms and their main features are described in Section 4. The experimental framework is detailed in Section 5, and performance comparisons are explained in Section 6. Finally, Section 7 presents concluding remarks and future work.

2. RELATED WORK

In this section, we review some multiobjective approaches published in the literature to solve the MSA problem using metaheuristic techniques.

One of the first approaches was presented by Seeluangsawatet al. in Ref. 17, where an algorithm called MOMSA was proposed. This technique considers two objectives, SOP and gap penalty, which are aggregated into a single function. In this early study, two point crossover and three mutation operators were used (move column, shift, and random shuffle). Nine data sets from BAliBASE 2.0 were used for performance assessment.

Ortuño et al.[18] implemented a multiobjective evolutionary algorithm based on NSGA-II, called MO-SAStrE, and applied it to optimize three objectives: STRIKE score, percentage of nongaps, and percentage of aligned columns (TC). The initial population was filled by applying a strategy based on taken precomputed alignments produced by others MSA tools, such as Muscle, ClustalW, Mafft, and T-Coffee. The authors proposed a novel encoding based on numeric strings, and the crossover and mutation operators were single point and gap shifting, respectively. The benchmark consisted of 218 problems included in the BAliBASE data set (v3.0),[7] and the Hypervolume[19] was used as a multiobjective quality indicator.

Soto and Becerra proposed in Ref. 6 a multiobjective evolutionary algorithm, also inspired in NSGA-II, to optimize prealigned sequences. They used the entropy and MetAl as objectives to be optimized. As variation operators they applied two-point crossover and random insertion and shift mutation. As in MO-SAStrE, the initial population was built using alignments produced by external MSA algorithms. To validate the predicted alignments, they considered the Sum-Of-Pairs (SOP), total columns (TC), MetAl, and Hypervolume metrics based on the BAliBASE (2.0) benchmark.

Other algorithm based on NSGA-II, called MSAGMOGA, was described by Kaya et al. in Ref. 20. In this work, three objectives were considered: similarity, affine gap penalty, and support. Two crossover operators (single- and two-point) and three mutation operators (random changing and shifts toward right and left) were applied to a problem data set taken from BAliBASE 2.0.

Parallel Niche Pareto AlineaGA (PNPAlineaGA) was proposed by da Silva et al.[21] It is based on a parallel island model used to solve a biobjective formulation of MSA: SOP and the total number of aligned columns. Three crossover and six mutation operators were applied in this approach with eight data sets from BAliBASE 2.0.

More recently, Abbasi et al.[22] introduced a local search method for multiobjective MSA, where SOP and minimizing the number of gaps were the goals. A subset of BAliBase 3.0 (the 38 instances in the set RV11) was studied.

This review shows a low number of works, most of them built around the NSGA-II algorithm. All these studies use part or the full BAliBASE sequences (in versions 2.0 or 3.0). Another conclusion is that there is not an agreement in the objectives to be optimized, although scores such as SOP and TC are among the most considered ones. From our perspective, it is clear that there is a lack of comparative studies involving current state-of-the-art multiobjective metaheuristic for solving the MSA. The purpose of this paper is therefore to fill this gap, aiming to serve also as reference work for further developments in this area.

3. MSA PROBLEM

In this section, we provide a formulation of MSA as a biobjective optimization problem. In addition, the adopted solution encoding strategy to tackle this problem is also described. Prior to these, we present a formal definition of the MSA problem as follows:

DEFINITION 1. *Given a finite alphabet set* Σ *and a set* $S = (s_1, s_2, ..., s_k)$ *of* k *sequences of varying length* l_1 *to* l_k *with* $s_i = s_{i1}s_{i2}, ..., s_{il_i} (1 \leq i \leq k)$*, where for DNA sequences,* Σ *consists of four characters of the nucleotides {A, T, G, C}, and for protein sequences* Σ *consists of 20 characters of the amino acids {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}; to find an optimal alignment* S' *of* S*, with regard of a scoring function* $f(S')$*, such that*

$$S' = (s'_{ij}),\ with\ 1 \leq i \leq k, 1 \leq j \leq l, max(l_i) \leq l \leq \sum_{i=1}^{k} l_i \tag{1}$$

satisfying

1. $s'_{ij} \in \Sigma \cup \{-\}$, where "-" denotes the gap character.
2. Each row $s'_i = s'_{i1}s'_{i2}, ..., s'_{il} (1 \leq i \leq k)$ of S' is exactly the corresponding sequence s_i if we eliminate all the gap symbols.
3. The length of the all the k sequences is exactly the same.
4. S' has no column, which only contains gaps.

According to Ref. 23, the complexity of finding an optimal alignment is $O(k2^k L^k)$, where k is the number of sequences and L is the $max\{l_1, l_2, ..., l_k\}$. An example of the MSA problem is shown in Figure 1, where the four sequences of the BB11001 instance of BAliBASE are depicted. A possible solution, obtained with the tool ClustalW,[24] is included in Figure 2, which contains four completely aligned columns marked with an asterisk (*). We can observe that both full and partially aligned columns are highlighted with a background color.

As it can be observed, the MSA problem is commonly tackled by inserting gaps in the proper places to maximize some scores. For example, in Ref. 18, two of the

Figure 1. Unaligned BAliBASE BB11001 instance.

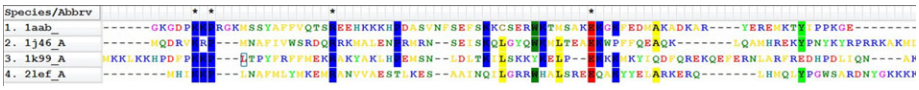

Figure 2. BAliBASE BB11001 instance aligned by ClustalW. Four columns, marked with an asterisk (*) and with a background color, are completely aligned.

considered objectives consisted in maximizing the percentage of nongaps and the percentage of completely aligned columns. These are very intuitive goals, but they are not fully contradictory from a multiobjective point of view: if after manipulating the sequences a column is full of gaps then it can be removed, thus improving the number of nongaps. However, this does not imply a worsening in the percentage of aligned columns.

The methods that have been proposed in the literature to evaluate the accuracy of alignments can be broadly grouped into two categories depending on whether they require to know the three-dimensional (3D) structure of the sequences or not. This way, algorithms such as STRIKE[5] are structured-based, whereas computing scores such as SOP, TC, or NonGaps, only need the amino acid sequences.

In this paper, we take as reference algorithm the metaheuristic called MOSAStrE.[18] This technique is an adaptation of the well-known NSGA-II algorithm to MSA, and it is applied to solve a triobjective formulation of the problem: STRIKE, TC, and NonGaps. However, this approach has some issues that should be dealt with. First, it is well known that NSGA-II is not well suited for problems having three or more objectives; second, TC and NonGaps are not strictly contradictory objectives as commented before (i.e., reducing gaps in a MSA also improves TC); finally, STRIKE requires at least one 3D known structure, which is not always available. These reasons led us to consider a biobjective approach taking as goals the optimization of SOP and TC. These two scores are in conflict with between them, as improving TC implies to add gaps, but gaps are penalized in the score matrix used to compute SOP. We give a definition of both objectives next.

The SOP score of an alignment, presented in Equation 2, is computed by adding all the scores of the pairwise comparisons between each character amino acid in each column of the alignment.

$$SOP(S) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Scoring\ Matrix(l_i, l_j) \tag{2}$$

A scoring matrix is needed to determine the cost of aligning a residue with another. Also, a gap penalty value must be settled for determining the cost of aligning

an amino acid with a gap. This penalty is only employed when aligning a residue with a gap. The alignment of two or more gaps is not penalized. We used the PAM250 scoring matrix[25] and a gap penalty of –10.

The percentage of aligned columns (TC) refers to the number of columns that are completely aligned with exactly the same compound. This objective function needs to be maximized to ensure more conserved regions within the alignment.

Consequently, the biobjective problem to be resolved can be defined as

$$maximize F(S) = \{f_1(S),\ f_2(S)\} \tag{3}$$

where $f_1(S)$ and $f_2(S)$ are the defined SOP and TC scores, respectively, and S is the alignment to be evaluated.

4. MULTIOBJECTIVE ALGORITHMS DESCRIPTION

In this section, we describe the six multiobjective algorithms, which are chosen for our study as representative techniques of the state-of-the-art. More specifically, these techniques are NSGA-II, SPEA2, MOCell, MOEA/D, SMS-EMOA, and GWASFGA. We also include details of the encoding, genetic operators, and the strategy used for initializing the initial population.

4.1. Algorithms

The algorithms we are comparing are as follows:

- NSGA-II[9] is a generational genetic algorithm based on generating new individuals from the original population by applying the typical genetic operators (selection, crossover, and mutation). A ranking procedure is applied to promote convergence, whereas a density estimator (the crowding distance) is used to enhance the diversity of the set of found solutions.
- SPEA2[11] is, as NSGA-II, a classic and very widely used algorithm. It is featured by using a population and an archive, and the strength raw fitness and the distance to the kth nearest neighbor are applied to foster convergence and diversity, respectively.
- MOCell[26] is a cellular multiobjective evolutionary algorithm that uses an external archive to store the nondominated solutions found during the search. As the archive is bounded in size, a density estimator (the same crowding distance used in NSGA-II) is applied to select which solution should be removed when the archive size is exceeded.
- MOEA/D[15] is based on decomposing a multiobjective optimization problem into a number of scalar optimization subproblems, which are optimized simultaneously, only using information from their neighboring subproblems. This algorithm also applies a mutation operator to the solutions.
- SMS-EMOA[14] is a steady-state evolutionary algorithm that uses a selection operator based on the hypervolume measure combined with the concept of nondominated sorting.
- GWASF-GA[16] is an evolutionary algorithm, which can find a Pareto front approximation by using a weighting achievement scalarizing function and two reference points (utopian and nadir).

These algorithms constitute a set of representative multiobjective evolutionary algorithms: classic (NSGA-II, SPEA2), cellular (MOCell), decomposition-based (MOEA/D and GWASF-GA), and indicator-based (SMS-EMOA).

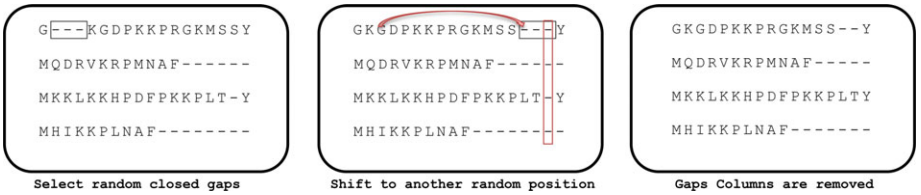

Figure 3. Closed gap shifting mutation operator: Closed gaps are randomly chosen and shifted to another position. Columns full of gaps are removed if they are found.

All the algorithms and the MSA problem are implemented with jMetal,[27] a Java framework for multiobjective optimization with metaheuristics. In concrete, we have used jMetal 5.[28a]

4.2. Encoding

Choosing the representation or encoding of the individuals is a key issue in evolutionary algorithms, mainly because the variation operators that can be used are directly dependent on the codification scheme. We adopt here a similar codification that has been used in previous studies,[21,29] according to which a MSA is implemented as a list of strings, each of them representing a particular sequence.

The possible values for each character of the individual are the nucleotides {A, T, G, C} for DNA sequences or the 20 characters of the amino acids {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W and Y} for protein sequences. The symbol "-" is used to denote the each gap in the sequence.

4.3. Genetic Operators

All the evolutionary algorithms in our study share the same mutation and crossover operators, which are the ones used in MO-SAStrE.[18]

The mutation operator is closed gap shifting, where a random set of closed gaps are shifted to another random position in a sequence. The aim is to reduce the number of gaps in the MSA when columns only having gaps are detected. This operator is illustrated in Figure 3.

The crossover operator is the single-point crossover over alignments proposed in Ref. 4, which works as depicted in Figure 4. The operator randomly selects a position from a parent by splitting it into two blocks (let us refer to them P1a and P1b). The same selected positions are found in the other parent (but not necessarily in the same column) and is tailored so that the right piece can be joined to the left piece of the first parent and vice versa (P2a and P2b). Finally, the selected blocks are crossed between these two parents, generating two new individuals with the combination of the blocks: [P1a + P2b] and [P1a + P1b]. With the aim to be

[a]The source code of this work will be freely available at `https://github.com/jMetal/jMetalMSA` if the paper is accepted for publication.

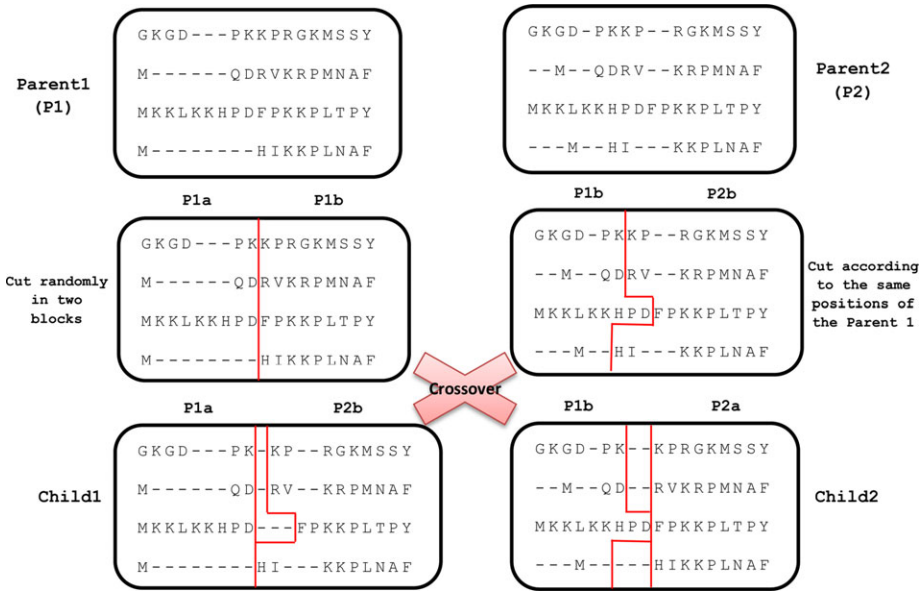

Figure 4. Single point crossover operator: The first parent is cut straight at a randomly chosen position. The second one is tailored so that the right piece can be joined to the left piece of the first parent and vice versa.

assured that the obtained children do not alter their sequences, any empty space that appears at the junction point is filled with gaps.

4.4. Strategy for Initializing the Initial Population

The usual approach to create the initial population in evolutionary algorithms is to fill it with individuals which are randomly initialized. However, in the case of the MSA, with the aim of accelerating the convergence of the search, a commonly applied strategy is to take a number of precomputed alignments and use them as the starting point of an initialization scheme.

We follow again the ideas presented in Ref. 18, so we have generated, for every data set, a number of alignments by using these tools: ClustalW, MUSCLE, Kalign, Mafft, RetAlign, TCOFFEE, ProbCons, and FSA (see Table I). The eight obtained alignments are then added to the initial population, and the remaining individuals are created by applying the crossover operator to pairs of randomly selected solutions which are taken from the same initial population that is being created.

5. EXPERIMENTATION

In this section, we describe the parameter settings of the metaheuristics we have selected, the benchmark problems, and the experimentation methodology.

Table I. Methods used to generate the initial population of the algorithms.

Tool	Version	Type
ClustalW[24]	2.0.10	Progressive
MUSCLE[30]	3.8.31	Progressive
Kalign[31]	2.04	Progressive
Mafft[32]	6.85	Progressive
RetAlign[33]	1.0	Progressive
TCOFFEE[29]	8.97	Consistency based
ProbCons[34]	1.12	Consistency based
FSA[35]	1.15.5	Consistency based

5.1. Parameter Setup

We have configured the algorithms with the same parameter settings with the aim of making a fair comparison. All of them use the single-point crossover and closed gap shifting mutation operators described in Section 4.3. Both operators are applied with probabilities of 0.8 and 0.2. The population size is 100, and the stopping condition is set to compute a total number of 50,000 function evaluations.[18]

The particular control parameters of the algorithms are detailed next. MOCell uses an external archive of size 100; the neighbor size, number of replaced solutions, and the neighborhood selection probability in MOEA/D are, respectively, 20, 2, and 0.9.

5.2. Test Benchmark: BAliBASE

Currently, a large amount of data sets and techniques have been designed to standardize the comparison of sequence alignment results. Examples are OXBench,[36] HOMSTRAD,[37] or Prefab.[30] In this work, we have selected a subset of the BAliBASE (v3.0) benchmark;[7] in particular, we have chosen the first 10 data sets from the RV11 and RV12 families, summing up 20 problem instances.

5.3. Methodology

We have used three quality indicators to assess the performance of the algorithms. The first one is the Hypervolume (I_{HV}),[19] a Pareto compliant indicator which takes into account both the convergence and the diversity of the Pareto front approximations. The additive Epsilon (I_{E^+})[38] and the Spread or Δ (I_Δ) indicators are used as a complement to measure the degree of convergence and diversity, respectively.

For each combination of algorithm and problem, we have made 20 independent runs, and we report the median, $\tilde{x}$, and the interquartile range, IQR, as measures of location (or central tendency) and statistical dispersion, respectively, for every considered indicator. When presenting the obtained values in tables, we emphasize with a dark grey background the best result for each problem, and a clear grey background is used to indicate the second best result; this way, we can see at a glance the most salient algorithms.

To provide the obtained results with statistical confidence, a series of non-parametric statistical tests have been applied, as in several cases the distributions of results did not follow the conditions of normality and homoskedasticity.[39] A confidence level of 95% (i.e., significance level of 5% or p-value under 0.05) has been used in all cases, meaning that the differences are unlikely to have occurred by chance with a probability of 95%. Therefore, analyses and comparisons focus on the entire distribution of binding energies, although they pay particular attention to the median values, for the 20 tackled instances. In particular, Friedman's ranking and Holm's post hoc multicompare tests[39] have been applied to know which algorithms are statistically worse than the control one (the algorithm with the best ranking).

To compute the quality indicators, particularly I_{E^+} and I_Δ, it is necessary to know the Pareto front of the problems. As they are unknown in the case of the MSA problems we are considering, we have adopted the solution of generating a reference front. This is the result of combining into a single front all the nondominated solutions produced by all the algorithms in all the independent runs for each problem. This strategy allows to make a relative performance assessment of the metaheuristics, because if the behavior of all the compared techniques is poor we know which of them yields the best fronts, but we not know if they are near or far from the true Pareto front.

6. RESULTS

We present and analyze here the obtained results. We start with the values of the three quality indicators, which are presented in Tables II–IV. We have to note that in the case of I_{HV}, the higher the value the better, and the opposite is applied in the other indicators.

The I_{HV} figures in Table II show that NSGA-II is the metaheuristic providing the best overall performance, because it obtains the best values in 15 of the 20 considered MSA data sets. MOCell and GWASFGA are the techniques that obtain the best front approximations in the remaining five problems.

We pay attention now to the results provided by the I_{E^+} indicator, which measures the convergence of the obtained fronts. The values, included in Table III, show that there are three algorithms (NSGA-II, MOCell, and GWASFGA) with a similar performance if we merely consider the number of best and second best results. As in the case of the I_{HV}, SMS-EMO and MOEA/D have shown a poor behavior.

The last quality indicator we have used, I_Δ, gives a value of the degree of diversity of the fronts. In this case (see Table IV), MOCell is clearly the algorithm yielding the best values in all the problems.

At a first glance, it may seem that the results provided by the three quality indicators are contradictory, because NSGA-II and MOCell would be ranked the first ones according to the I_{HV} and I_Δ indicators, respectively, and I_{E+} does not allow to declare a clear outstanding algorithm. This is explained because we must consider that the indicators use different strategies to measure convergence and diversity. Thus, I_{HV} is based on adding volumes, I_{E+} measures distances between

Table II. Median and interquartile range of the I_{HV} indicator values.

	NSGAII	SPEA2	MOCell	SMS-EMOA	MOEA/D	GWASFGA
BB11001	$1.40e-01_{1.6e-01}$	$4.34e-02_{1.4e-01}$	$1.34e-01_{5.4e-02}$	$1.04e-01_{1.3e-01}$	$0.00e+00_{2.8e-02}$	$4.85e-02_{1.2e-01}$
BB11002	$1.98e-01_{1.6e-01}$	$1.79e-01_{1.7e-02}$	$1.71e-01_{1.4e-01}$	$1.73e-01_{1.3e-01}$	$1.62e-02_{1.7e-02}$	$0.00e+00_{3.2e-02}$
BB11003	$4.02e-01_{1.4e-01}$	$3.42e-01_{1.2e-01}$	$3.64e-01_{7.7e-02}$	$3.47e-01_{9.0e-02}$	$4.25e-02_{6.0e-02}$	$3.26e-01_{9.4e-02}$
BB11004	$5.48e-01_{1.5e-01}$	$4.62e-01_{7.2e-02}$	$4.46e-01_{1.1e-01}$	$4.73e-01_{1.1e-01}$	$5.12e-02_{4.0e-02}$	$4.87e-01_{9.6e-02}$
BB11005	$1.61e-01_{9.8e-03}$	$1.57e-01_{1.6e-02}$	$1.45e-01_{1.0e-02}$	$1.15e-01_{7.3e-03}$	$0.00e+00_{0.0e+00}$	$0.00e+00_{0.0e+00}$
BB11006	$3.60e-01_{2.6e-02}$	$3.35e-01_{1.9e-02}$	$3.35e-01_{8.2e-02}$	$3.20e-01_{1.8e-02}$	$0.00e+00_{0.0e+00}$	$1.00e-01_{3.8e-02}$
BB11007	$2.70e-01_{4.3e-02}$	$2.60e-01_{3.4e-02}$	$2.63e-01_{2.7e-02}$	$2.60e-01_{2.9e-02}$	$6.63e-02_{3.4e-02}$	$1.20e-01_{8.0e-02}$
BB11008	$4.94e-01_{1.4e-01}$	$4.68e-01_{1.0e-01}$	$3.35e-01_{9.4e-02}$	$4.69e-01_{1.5e-01}$	$1.04e-01_{1.3e-01}$	$2.97e-01_{6.0e-02}$
BB11009	$2.28e-01_{1.2e-01}$	$2.26e-01_{1.2e-01}$	$3.12e-01_{9.4e-02}$	$2.43e-01_{1.2e-01}$	$0.00e+00_{1.6e-02}$	$1.53e-01_{1.5e-01}$
BB11010	$2.72e-01_{1.3e-01}$	$3.17e-01_{7.5e-02}$	$2.12e-01_{1.2e-01}$	$2.77e-01_{7.3e-02}$	$1.45e-02_{5.6e-02}$	$3.53e-01_{7.9e-02}$
BB12001	$4.22e-01_{4.8e-02}$	$4.02e-01_{4.0e-02}$	$4.00e-01_{5.4e-02}$	$4.11e-01_{1.4e-02}$	$1.76e-01_{2.6e-02}$	$1.59e-01_{4.7e-02}$
BB12002	$1.62e-01_{2.1e-01}$	$2.19e-01_{1.7e-01}$	$2.63e-01_{2.2e-01}$	$1.01e-01_{1.7e-01}$	$0.00e+00_{8.0e-03}$	$2.50e-01_{2.7e-01}$
BB12003	$2.20e-01_{6.2e-02}$	$2.16e-01_{6.2e-02}$	$2.35e-01_{4.6e-02}$	$1.86e-01_{5.9e-02}$	$1.56e-01_{5.6e-02}$	$2.34e-01_{2.9e-02}$
BB12004	$4.62e-01_{6.0e-02}$	$3.98e-01_{5.2e-02}$	$4.40e-01_{7.3e-02}$	$3.78e-01_{8.7e-02}$	$3.52e-01_{5.5e-02}$	$4.58e-01_{2.6e-02}$
BB12005	$4.86e-01_{6.3e-02}$	$4.59e-01_{1.0e-01}$	$4.48e-01_{7.3e-02}$	$4.29e-01_{4.9e-02}$	$2.57e-01_{1.7e-02}$	$3.32e-01_{6.5e-02}$
BB12006	$3.50e-01_{2.2e-01}$	$3.57e-01_{1.8e-01}$	$2.31e-01_{2.1e-01}$	$4.14e-01_{2.2e-01}$	$2.52e-01_{1.2e-01}$	$4.57e-01_{2.7e-01}$
BB12007	$5.27e-01_{1.9e-02}$	$4.88e-01_{2.5e-02}$	$4.82e-01_{2.0e-02}$	$5.03e-01_{2.7e-02}$	$1.88e-01_{4.2e-02}$	$2.01e-01_{6.6e-02}$
BB12008	$1.56e-01_{1.9e-02}$	$1.32e-01_{8.2e-03}$	$1.37e-01_{4.0e-02}$	$1.25e-01_{1.1e-02}$	$1.13e-01_{1.9e-02}$	$1.43e-01_{1.1e-02}$
BB12009	$5.15e-01_{3.7e-02}$	$4.89e-01_{4.6e-02}$	$4.91e-01_{6.8e-02}$	$4.83e-01_{5.2e-02}$	$1.78e-01_{2.4e-02}$	$4.24e-01_{5.1e-02}$
BB12010	$5.46e-01_{3.1e-02}$	$5.33e-01_{3.9e-02}$	$5.05e-01_{4.8e-02}$	$5.23e-01_{6.3e-02}$	$2.35e-01_{2.1e-01}$	$3.57e-01_{6.5e-02}$

Dark and light grey background cells represent the best and second best results, respectively.

Table III. Median and interquartile range of the I_{E^+} indicator values.

	NSGAII	SPEA2	MOCell	SMS-EMOA	MOEA/D	GWASFGA
BB11001	$5.56e-01_{3.2e-01}$	$7.67e-01_{2.7e-01}$	$5.17e-01_{1.5e-01}$	$6.41e-01_{2.8e-01}$	$9.66e-01_{2.7e-01}$	$6.29e-01_{3.0e-01}$
BB11002	$4.93e-01_{8.1e-02}$	$5.19e-01_{6.9e-02}$	$5.26e-01_{6.7e-02}$	$5.55e-01_{4.8e-02}$	$8.99e-01_{2.8e-01}$	$6.43e-01_{1.7e-01}$
BB11003	$4.18e-01_{5.8e-02}$	$4.43e-01_{1.4e-01}$	$3.68e-01_{8.9e-02}$	$4.14e-01_{4.9e-02}$	$7.60e-01_{2.2e-01}$	$4.07e-01_{1.3e-01}$
BB11004	$3.06e-01_{1.2e-01}$	$3.16e-01_{6.1e-02}$	$4.02e-01_{1.6e-01}$	$3.25e-01_{7.0e-02}$	$8.47e-01_{1.1e-01}$	$2.87e-01_{7.4e-02}$
BB11005	$5.04e-01_{4.1e-02}$	$5.09e-01_{6.7e-02}$	$5.75e-01_{4.8e-02}$	$7.17e-01_{3.0e-02}$	$1.00e+00_{0.0e+00}$	$1.00e+00_{0.0e+00}$
BB11006	$3.94e-01_{8.0e-02}$	$4.17e-01_{5.6e-02}$	$3.47e-01_{6.7e-02}$	$4.73e-01_{3.6e-02}$	$1.00e+00_{0.0e+00}$	$3.76e-01_{3.4e-02}$
BB11007	$5.86e-01_{2.9e-02}$	$6.28e-01_{3.1e-02}$	$6.01e-01_{6.7e-02}$	$6.26e-01_{3.0e-02}$	$8.03e-01_{1.2e-01}$	$3.72e-01_{1.1e-01}$
BB11008	$2.27e-01_{1.7e-01}$	$2.43e-01_{1.4e-01}$	$3.80e-01_{1.2e-01}$	$2.67e-01_{1.7e-01}$	$5.75e-01_{1.1e-01}$	$4.07e-01_{4.9e-02}$
BB11009	$3.96e-01_{1.3e-01}$	$4.27e-01_{9.2e-02}$	$3.68e-01_{1.1e-01}$	$4.20e-01_{1.3e-01}$	$1.08e+00_{3.1e-01}$	$5.04e-01_{6.6e-02}$
BB11010	$6.46e-01_{1.8e-01}$	$5.68e-01_{1.2e-01}$	$7.30e-01_{1.7e-01}$	$6.31e-01_{1.3e-01}$	$5.99e-01_{1.2e-01}$	$3.65e-01_{1.6e-01}$
BB12001	$3.03e-01_{3.3e-02}$	$3.74e-01_{3.1e-02}$	$3.40e-01_{5.1e-02}$	$3.64e-01_{2.0e-02}$	$6.39e-01_{8.3e-02}$	$4.00e-01_{4.0e-02}$
BB12002	$5.96e-01_{1.9e-01}$	$6.25e-01_{1.9e-01}$	$4.64e-01_{2.3e-01}$	$6.22e-01_{1.7e-01}$	$1.02e+00_{1.7e-01}$	$4.62e-01_{3.1e-01}$
BB12003	$6.53e-01_{1.2e-01}$	$6.54e-01_{1.0e-01}$	$5.87e-01_{7.2e-02}$	$7.13e-01_{1.1e-01}$	$6.42e-01_{1.2e-01}$	$4.29e-01_{6.9e-02}$
BB12004	$4.53e-01_{7.8e-02}$	$5.31e-01_{8.0e-02}$	$4.72e-01_{8.2e-02}$	$5.65e-01_{9.9e-02}$	$4.67e-01_{1.2e-01}$	$2.80e-01_{4.2e-02}$
BB12005	$2.81e-01_{5.7e-02}$	$3.08e-01_{6.8e-02}$	$3.25e-01_{3.6e-02}$	$3.27e-01_{6.8e-02}$	$4.66e-01_{4.3e-02}$	$3.81e-01_{5.4e-02}$
BB12006	$4.03e-01_{1.6e-01}$	$4.25e-01_{1.6e-01}$	$5.16e-01_{2.1e-01}$	$3.80e-01_{1.9e-01}$	$5.50e-01_{9.9e-02}$	$3.11e-01_{2.9e-01}$
BB12007	$1.83e-01_{6.8e-02}$	$2.85e-01_{6.1e-02}$	$2.23e-01_{6.0e-02}$	$2.70e-01_{5.5e-02}$	$4.12e-01_{3.5e-02}$	$4.34e-01_{4.7e-02}$
BB12008	$7.98e-01_{3.3e-02}$	$8.33e-01_{1.1e-02}$	$8.24e-01_{6.2e-02}$	$8.45e-01_{1.3e-02}$	$8.04e-01_{8.8e-02}$	$7.60e-01_{4.8e-02}$
BB12009	$3.57e-01_{5.9e-02}$	$3.63e-01_{7.3e-02}$	$2.84e-01_{2.5e-01}$	$3.86e-01_{7.6e-02}$	$5.02e-01_{5.0e-02}$	$2.96e-01_{3.9e-02}$
BB12010	$2.62e-01_{5.3e-02}$	$2.97e-01_{5.6e-02}$	$2.81e-01_{9.9e-02}$	$3.18e-01_{1.1e-01}$	$4.29e-01_{1.7e-01}$	$3.53e-01_{4.5e-02}$

Dark and light grey background cells represent the best and second best results, respectively.

Table IV. Median and interquartile range of the I_Δ indicator values.

	NSGAII	SPEA2	MOCell	SMS-EMOA	MOEA/D	GWASFGA
BB11001	$1.28e+00_{6.6e-02}$	$1.24e+00_{5.5e-02}$	$7.37e-01_{1.6e-01}$	$1.27e+00_{8.5e-02}$	$1.32e+00_{1.2e-01}$	$1.38e+00_{7.1e-02}$
BB11002	$7.75e-01_{7.7e-02}$	$8.47e-01_{1.1e-01}$	$7.55e-01_{1.3e-01}$	$8.46e-01_{1.6e-01}$	$1.31e+00_{6.5e-03}$	$1.53e+00_{9.3e-02}$
BB11003	$1.03e+00_{1.5e-01}$	$1.02e+00_{1.3e-01}$	$8.19e-01_{9.7e-02}$	$1.04e+00_{1.1e-01}$	$1.27e+00_{7.2e-02}$	$1.41e+00_{1.4e-01}$
BB11004	$9.21e-01_{1.3e-01}$	$9.17e-01_{9.8e-02}$	$8.40e-01_{6.2e-02}$	$9.64e-01_{1.7e-01}$	$1.32e+00_{3.9e-02}$	$1.47e+00_{1.8e-01}$
BB11005	$6.82e-01_{3.7e-02}$	$7.20e-01_{6.5e-02}$	$5.74e-01_{5.8e-02}$	$9.14e-01_{7.7e-02}$	$1.00e+00_{4.1e-03}$	$1.00e+00_{0.0e+00}$
BB11006	$6.74e-01_{5.4e-02}$	$7.14e-01_{3.4e-02}$	$5.58e-01_{5.4e-02}$	$6.62e-01_{7.1e-02}$	$1.08e+00_{8.8e-02}$	$1.75e+00_{1.4e-01}$
BB11007	$7.74e-01_{5.3e-02}$	$7.78e-01_{5.0e-02}$	$6.77e-01_{7.2e-02}$	$7.26e-01_{7.7e-02}$	$1.13e+00_{4.0e-02}$	$1.59e+00_{1.7e-01}$
BB11008	$7.13e-01_{2.8e-02}$	$6.39e-01_{6.9e-02}$	$5.86e-01_{6.6e-02}$	$5.92e-01_{9.1e-02}$	$1.47e+00_{4.8e-02}$	$1.59e+00_{1.1e-01}$
BB11009	$8.26e-01_{8.2e-02}$	$8.58e-01_{1.3e-01}$	$7.96e-01_{1.1e-01}$	$8.67e-01_{1.3e-01}$	$1.17e+00_{1.9e-01}$	$1.56e+00_{6.1e-02}$
BB11010	$1.02e+00_{1.0e-01}$	$1.01e+00_{1.4e-01}$	$8.76e-01_{9.8e-02}$	$1.00e+00_{1.2e-01}$	$1.41e+00_{1.1e-01}$	$1.38e+00_{9.4e-02}$
BB12001	$6.48e-01_{5.4e-02}$	$6.26e-01_{2.9e-02}$	$4.94e-01_{6.3e-02}$	$5.59e-01_{3.4e-02}$	$1.28e+00_{3.5e-02}$	$1.68e+00_{1.3e-01}$
BB12002	$1.24e+00_{2.6e-02}$	$1.24e+00_{4.6e-02}$	$7.62e-01_{8.6e-02}$	$1.25e+00_{6.6e-02}$	$1.30e+00_{5.1e-02}$	$1.41e+00_{1.8e-01}$
BB12003	$1.26e+00_{2.0e-02}$	$1.26e+00_{2.7e-02}$	$6.99e-01_{3.9e-02}$	$1.25e+00_{2.9e-02}$	$1.58e+00_{1.0e-01}$	$1.49e+00_{1.5e-01}$
BB12004	$8.84e-01_{6.2e-02}$	$9.14e-01_{1.1e-01}$	$7.43e-01_{8.5e-02}$	$9.44e-01_{1.3e-01}$	$1.28e+00_{7.2e-02}$	$1.54e+00_{7.0e-02}$
BB12005	$8.77e-01_{8.8e-02}$	$8.94e-01_{1.3e-01}$	$7.34e-01_{5.5e-02}$	$8.59e-01_{1.1e-01}$	$1.38e+00_{5.4e-02}$	$1.49e+00_{1.1e-01}$
BB12006	$1.29e+00_{1.2e-01}$	$1.34e+00_{1.1e-01}$	$7.79e-01_{5.7e-02}$	$1.33e+00_{1.4e-01}$	$1.36e+00_{1.8e-01}$	$1.43e+00_{1.4e-01}$
BB12007	$6.04e-01_{6.3e-02}$	$5.80e-01_{6.3e-02}$	$4.11e-01_{5.7e-02}$	$4.76e-01_{4.5e-02}$	$1.53e+00_{2.6e-02}$	$1.73e+00_{7.4e-02}$
BB12008	$8.63e-01_{2.4e-02}$	$8.45e-01_{1.8e-02}$	$7.92e-01_{3.2e-02}$	$8.34e-01_{1.5e-02}$	$1.05e+00_{5.1e-02}$	$1.28e+00_{8.6e-02}$
BB12009	$1.00e+00_{8.7e-02}$	$9.90e-01_{8.1e-02}$	$5.96e-01_{9.6e-02}$	$1.01e+00_{8.6e-02}$	$1.59e+00_{9.0e-02}$	$1.55e+00_{1.3e-01}$
BB12010	$8.15e-01_{5.9e-02}$	$7.88e-01_{1.5e-01}$	$5.21e-01_{9.7e-02}$	$8.51e-01_{1.7e-01}$	$1.57e+00_{1.3e-01}$	$1.62e+00_{9.6e-02}$

Dark and light grey background cells represent the best and second best results, respectively.

Table V. Average Friedman's rankings with Holm's Adjusted p-values ($\alpha = 0.05$) of compared algorithms for the test set of RV11 and RV12 instances.

Hypervolume (HV)			Epsilon ($I_{\epsilon+}$)			Spread I_Δ		
Algorithm	Fri_{Rank}	$Holm_{Ap}$	Algorithm	Fri_{Rank}	$Holm_{Ap}$	Algorithm	Fri_{Rank}	$Holm_{Ap}$
NSGA-II	1.65	-	*NSGA-II	2.19	-	*MOCell	1.00	
MOCell	3.07	3.71e–01	MOCell	2.75	8.61e–01	SMS-EMOA	2.95	1.45e–01
SPEA2	3.15	3.71e–01	GWASFGA	2.92	8.61e–01	SPEA2	3.00	1.45e–01
SMS-EMOA	3.37	3.71e–01	SPEA2	3.75	3.87e–01	NSGA-II	3.05	1.45e–01
GWASFGA	3.92	7.84e–02	SMS-EMOA	4.05	2.29e–01	MOEA/D	5.17	3.31e–04
MOEA/D	5.82	3.41e–04	MOEA/D	5.32	2.66e–02	GWASFGA	5.82	3.43e–05

Global Ranking: **MOCell (5)**, NSGA-II (6), SPEA2 (10), SMS-EMOA (11), GWASFGA (14), MOEA/D (17)

Symbol * indicates the control algorithm and column at right contains the overall ranking of positions with regard to I_{HV}, $I_{\epsilon+}$, and I_Δ.

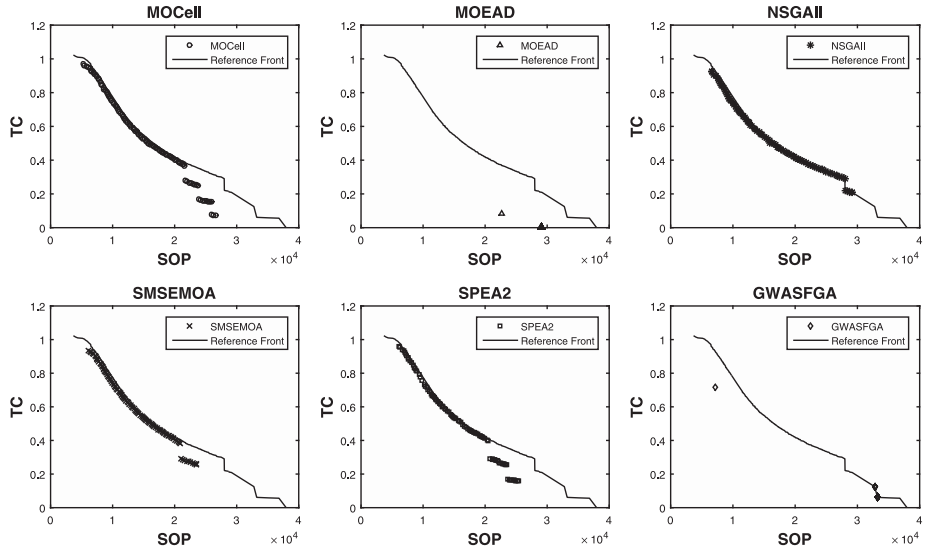

Figure 5. Pareto front approximations with the best I_{HV} values obtained by all the algorithms (MOCell, MOEA/D, NSGAII, SMSEMOA, SPEA2, and GWASFGA) on BAliBase instance BB11006 over 20 independent runs.

the obtained fronts and the Pareto front (or the reference front), and I_{Δ} needs to know the extreme points of the Pareto front.

Only reporting the values of central tendency and dispersion can lead to wrong conclusions, so the results have been statistically tested by means of Friedman and Holm's post hoc tests. Slightly different remarks can be extracted from them. In this regard, as shown in Table V, NSGA-II reaches the best ranking value (Friedman) with 1.65 for the HV indicator, followed by MOCell, SPEA2, SMS-EMOA, GWASFGA, and MOEA/D. Therefore, NSGA-II is established as the control algorithm for HV in the *post hoc* Holm test, which is compared with the remaining algorithms.

The adjusted p-values ($Holm_{Ap}$ in Table V) resulting from these comparisons are, for the last algorithm (MOEA/D), lower than the confidence level, meaning that NSGA-II is statistically better than this algorithm. However, it cannot be stated that NSGA-II shows statistically better performance than MOCell, SPEA2, SMS-EMOA, and GWASFGA.

In the case of $I_{\epsilon+}$, a similar ranking is computed, meaning that NSGA-II is the best ranked (control algorithm) with 2.16 for the HV indicator, followed by MOCell, GWASFGA, SPEA2, SMS-EMOA, and MOEA/D. Again, only MOEA/D is statistically worse that the control algorithm for this quality indicator, since its adjusted p-value ($Holm_{Ap}$) resulted lower than 0.05 (confidence level).

For the third quality indicator (I_{Δ}), MOCell is the best ranked (control algorithm) with 1.00, followed by SMS-EMOA, SPEA2, NSGA-II, MOEA/D, and GWASFGA. In this case, MOEA/D and GWASFGA, are statistically worse that the control algorithm (MOCell) for this quality indicator, since their adjusted p-value ($Holm_{Ap}$) resulted lower than 0.05 (confidence level).

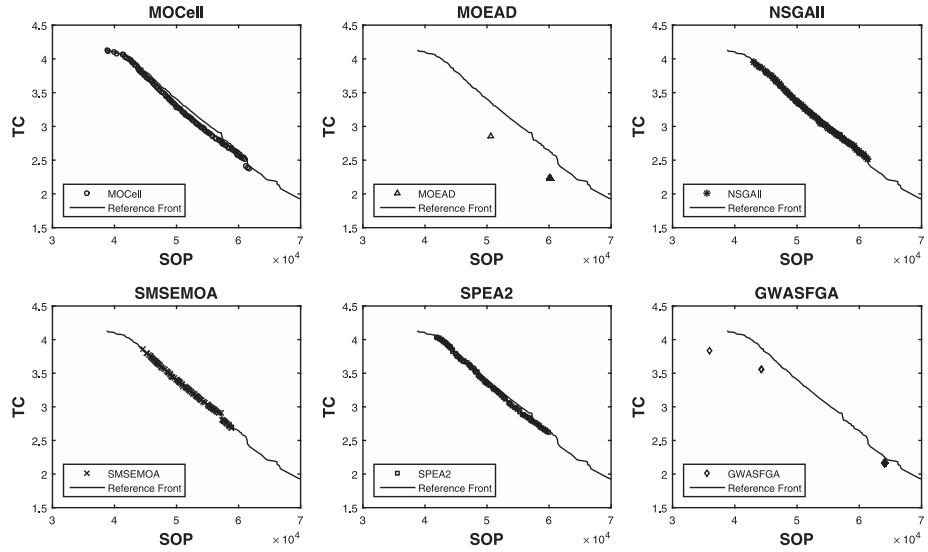

Figure 6. Pareto front approximations with the best I_{HV} values obtained by all the algorithms (MOCell, MOEA/D, NSGAII, SMSEMOA, SPEA2, and GWASFGA) on BAliBase instance BB12001 over 20 independent runs.

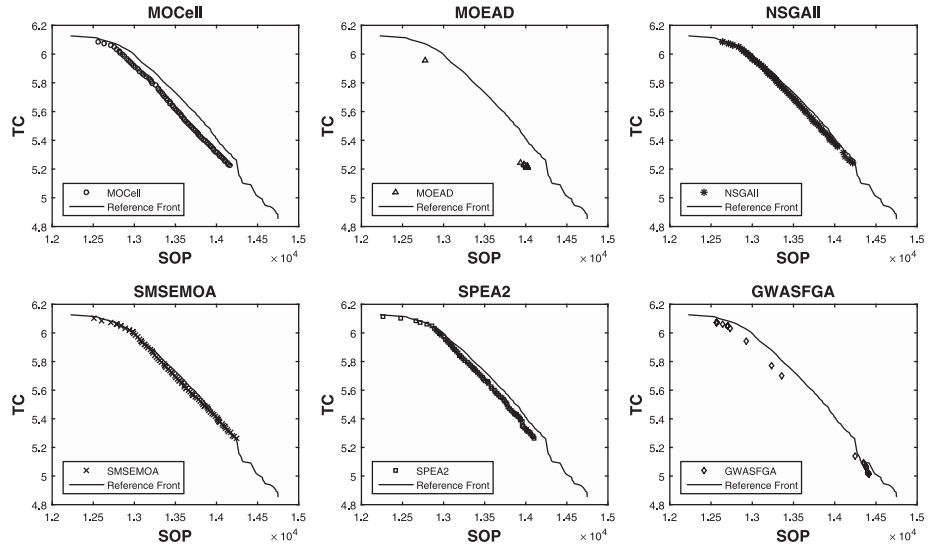

Figure 7. Pareto front approximations with the best I_{HV} values obtained by all the algorithms (MOCell, MOEA/D, NSGAII, SMSEMOA, SPEA2, and GWASFGA) on BAliBase instance BB12010 over 20 independent runs.

Summing up all ranking positions (as shown in the last row of Table V), we can observe that MOCell and NSGA-II show the overall best balance for the three quality indicators, followed by SPEA2, SMS-EMOA, GWASFGA, and MOEA/D. In general, it can be observed that classic multiobjective approaches obtain better performance than modern ones.

These results are graphically supported by providing some examples of the fronts produced by the algorithms. We include the fronts with the best I_{HV} values on three instances (BB11006, BB12001, and BB12010) in Figures 5–7.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have performed a comparative study of six multibjective metaheuristics representative of the state-of-the-art when dealing with the MSA problem. We have worked on a biobjective formulation of the problem, by considering the scores sum of pairs and the percentage of aligned columns as the functions to optimize. The benchmark has been composed of 20 problems taken from the BAliBASE library. Three quality indicators for measuring the convergence and diversity properties have been used for performance assessment.

Our study reveals that in the context of the chosen algorithms, the adopted parameter settings, the experimentation methodology, and the solved problems, MOCell and NSGA-II are the metaheuristics providing the best overall performance, whereas modern metaheuristics such as SMS-EMOA, MOEA/D, and GWASFGA have encountered difficulties to find accurate Pareto front approximations.

As a matter of future work, we are currently working on solving all the 218 data sets of BAliBASE to confirm if the performance of the algorithms is the same when dealing with a larger set of problems. Some research lines that remain open are to make a parameter sensitivity study (including the use of different mutation operators) and to consider a structure-based score (e.g., STRIKE) and a distance metric such as MetAl as goals to be optimized.

Acknowledgments

This work was partially supported by the Secretaría Nacional de Educación Superior Ciencia y Tecnología SENESCYT from Ecuador, and it is partially funded by grants TIN2014-58304-R (Ministerio de Ciencia e Innovación) and P11-TIC-7529 and P12-TIC-1519 (Plan Andaluz de Investigación, Desarrollo e Innovación). José García-Nieto is recipient of a Post-Doctoral fellowship of "Captación de Talento para la Investigación" at Universidad de Malaga.

References

1. Pei J. Multiple protein sequence alignment. Curr Opin Struct Biol 2008;18(3):382–386. Nucleic acids/sequences and topology.
2. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 1970;48(3):443–453.
3. Elias I. Settling the intractability of multiple alignment. J Comput Biol 2016;13(7):1323–1339.

4. da Silva FJM, Sánchez Pérez JM, Gómez Pulido JA, Vega Rodríguez MA. AlineaGA - a genetic algorithm with local search optimization for multiple sequence alignment. Appl Intell 2010;32(2):164–172.
5. Kemena C, Taly JF, Kleinjung J, Notredame C. Strike: evaluation of protein msas using a single 3d structure. Bioinformatics 2011;27(24):3385–3391.
6. Soto W, Becerra D. A multi-objective evolutionary algorithm for improving multiple sequence alignments. In: Campos S, editor. Advances in bioinformatics and computational biology, volume 8826 of Lecture Notes in Computer Science, Berlin: Springer; 2014. pp 73–82.
7. Thompson JD, Koehl P, Poch O. Balibase 3.0: latest developments of the multiple sequence alignment benchmark. Proteins 2005;61:127–136.
8. Blackburne BP, Whelan S. Measuring the distance between multiple sequence alignments. Bioinformatics 2012;28(4):495–502.
9. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–197.
10. Nebro AJ, Cristian Zambrano-Vega, Durillo JJ, Aldana-Montes José F. A study of multiple sequence alignment with multi-objective metaheuristics. In: Kózcy L, Medina J, editors. 7th European Symposium on Computational Intelligence and Mathematics (ESCIM 2015); Universidad de Cádiz (Dept. Matemáticas), Cádiz, Spain; 2015. pp 156–161.
11. Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength pareto evolutionary algorithm. In: Giannakoglou K, Tsahalis D, Periaux J, Papailou P, Fogarty T, editors. EUROGEN 2001: Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems; Athens, Greece, 2002. pp 95–100.
12. Nebro AJ, Luna F, Alba E, Dorronsoro B, Durillo JJ, Beham A. AbYSS: adapting scatter search to multiobjective optimization. IEEE Trans Evol Comput 2008;12(4):439–457.
13. Nebro AJ, Durillo JJ, Luna F, Dorronsoro B, Alba E. Design issues in a multiobjective cellular genetic algorithm. In: Obayashi S, Deb K, Poloni C, Hiroyasu T, Murata T, editors. 4th Int Conf on Evolutionary Multi-Criterion Optimization (EMO 2007), volume 4403 of Lecture Notes in Computer Science. Berlin: Springer; 2007. pp 126–140.
14. Emmerich M, Beume N, Naujoks B. An emo algorithm using the hypervolume measure as selection criterion. In: Coello CA, Hernández A, Zitler E, editors. Third Int Conf on Evolutionary MultiCriterion Optimization (EMO 2005), volume 3410 of Lecture Notes in Computer Science. Springer; 2005. pp 62–76.
15. Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 2007;11(6):712–731.
16. Ruiz AB, Saborido R, Luque M. Global WASF-GA: an evolutionary algorithm in multiobjective optimization to approximate the whole pareto optimal front. Evol Comput (accepted).
17. Seeluangsawat P, Chongstitvatana P. A multiple objective evolutionary algorithm for multiple sequence alignment. In: Proc 7th Annual Conf on Genetic and Evolutionary Computation (GECCO '05)., New York: ACM; 2005. pp 477–478.
18. Ortuño FM, Valenzuela O, Rojas F, Pomares H, Florido JP, Urquiza JM, Rojas I. Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns. Bioinformatics (Oxford, England) 2013;29(17):2112–2121.
19. Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 1999;3(4):257–271.
20. Kaya M, Sarhan A, Abdullah R. Multiple sequence alignment with affine gap by using multi-objective genetic algorithm. Comput Methods Prog Biomed 2014;114(1):38–49.
21. José Mateus da Silva F, Sánchez Pérez JM, Gómez Pulido JA, Miguel a Vega Rodríguez. Parallel niche Pareto AlineaGA—an evolutionary multiobjective approach on multiple sequence alignment. J Integr Bioinform 2011;8(3):174.
22. Abbasi M, Paquete L, Pereira FB. Local search for multiobjective multiple sequence alignment. In: Ortuño F, Rojas I, editors. Bioinformatics and biomedical engineering, volume 9044 of Lecture Notes in Computer Science. Berlin: Springer; 2015. pp 175–182.

23. Doğan H, Otu HH. Objective functions. In: Russell DJ, editor. Multiple sequence alignment methods, volume 1079 of Methods in Molecular Biology. New York: Humana Press, 2014. pp 45–58.
24. Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res 1994;22(22):4673–4680.
25. Dayhoff MO, Schwartz RM, Orcutt BC. A model of evolutionary change in proteins. Atlas Prot Seq Struct 1978;5:345–352.
26. Nebro AJ, Durillo JJ, Luna F, Dorronsoro B, Alba E. Mocell: A cellular genetic algorithm for multiobjective optimization. Int J Intell Syst 2009;24(7):723–725.
27. Durillo JJ, Nebro AJ. jMetal: a java framework for multi-objective optimization. Adv Eng Softw 2011;42(10):760–771.
28. Nebro AJ, Juan Durillo J, Vergne M. Redesigning the jmetal multi-objective optimization framework. In: Proc Companion Publication of the 2015 Annual Conf on Genetic and Evolutionary Computation (GECCO Companion '15). New York: ACM; 2015. pp 1093–1100.
29. Notredame C, Higgins DG, Heringa J. T-coffee: a novel method for fast and accurate multiple sequence alignment. J Mol Biol 2000;302(1):205–217.
30. Edgar RC. Muscle: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res 2004;32(5):1792–1797.
31. Lassmann T, Sonnhammer ELL. Kalign–an accurate and fast multiple sequence alignment algorithm. BMC Bioinform 2005;6(1):1–9.
32. Katoh K, Misawa K, Kuma K, Miyata T. Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. Nucleic Acids Res 2002;30(14):3059–3066.
33. Szabó A, Novák A, Miklós I, Hein J. Reticular alignment: a progressive corner-cutting method for multiple sequence alignment. BMC Bioinform 2010;11(1):1–19.
34. Do CB, Mahabhashyam MSP, Brudno M, Batzoglou S. Probcons: Probabilistic consistency-based multiple sequence alignment. Genome Res 2005;15(2):330–340.
35. Bradleyand RK, Roberts A, Smoot M, Juvekar S, Do J, Dewey C, Holmes I, Pachter L. Fast statistical alignment. PLoS Comput Biol 2009;5(5):e1000392, 05.
36. Raghava GPS, Stephen MJ Searle, Audley PC, Barber JD, Barton GJ. Oxbench: A benchmark for evaluation of protein multiple sequence alignment accuracy. BMC Bioinform 2003;4(1):1–23.
37. de Bakker PIW, Bateman A, Burke DF, Miguel RN, Mizuguchi K, Shi J, Shirai H, Blundell TL. Homstrad: adding sequence information to structure-based alignments of homologous protein families. Bioinformatics 2001;17(8):748–749.
38. Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG. Performance assessment of-multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 2003;7(2):117–132.
39. Sheskin DJ. Handbook of parametric and nonparametric statistical procedures. London: Chapman & Hall/CRC; 2007.