

Original Article

Multiple Sequence Alignment using Modified Brain Storm Optimization Algorithm with new Mutant

Jeevana Jyothi Pujari¹, Kanadam Karteeka Pavan²

¹Research Scholar, Computer science and engineering, Acharya Nagarjuna University, AP, India

²professor, Computer science and engineering, RVR&JC college of engineering, Guntur, AP, INDIA.

ajeevanajyothi.pujari@email.com, bkarteeka@yahoo.com

Abstract - Multiple Sequence Alignment (MSA) is a challenging and computational task in bioinformatics and is a core and fundamental task for various biological analysis fields. Finding an optimized alignment is a very difficult task in sequence alignment problems. One of the new intelligence algorithms is the Brainstorm optimization Algorithm (BSO), which solves many optimization problems due to its unique capabilities. BSO can be trapped into local optima with successive iterations. To address this local optimum, we proposed a Modified Brain storm optimization algorithm with a new mutation operator (MBSO-Mu) to obtain more optimal alignments. This modified new mutant mechanism is incorporated into creating new ideas in BSO for enhancing search space capability by maintaining population diversity. The proposed Algorithm has been executed on various benchmark datasets to obtain the fitness score of alignments. The efficacy of the proposed with a mutant MBSO-Mu shows a more optimal and near-optimal alignment score in multiple sequences while compared to several evolutionary algorithms.

Keywords - MSA, MBSO-MU, BSO, Encoding candidate.

I. INTRODUCTION

Multiple Sequence Alignment (MSA) is a prominent and common task in Bioinformatics. MSA gains central importance due to its major application of detecting similar significant regions as motifs from the large collection of sequences composed of nucleic acids or proteins. MSA is a basis for constructing phylogenetic trees in depicting relations over several organisms. MSA is also used to reveal the constraints in protein families by characterizing the function and structure of sequences, detecting homologies, and predicting the secondary and tertiary structure of uncharacterized proteins [1]. MSA is an optimization constraint problem, and finding optimal alignment between sequences is still requires more computational time when the alignment sequences increase and hence shown as to be in the class of NP-complete problem [2], [3].

The extension version of pairwise sequence alignment (aligning two DNA/RNA/Protein sequences) is Multiple sequence alignment (aligning more than two sequences).

Most of the changes in MSA are because of mutation operations which are insertion, deletion and substitutions of letters in the sequences. There are three methods to solve the MSA problem such Exact, progressive and iterative techniques. Alignment can be local or global in local alignments identifying the similar regions within in the subsequences, whereas the whole sequences are aligned with the entire length in global alignment [4]. The optimal and exact alignment is achieved by Needleman and Wunsch technique which is a dynamic programming approach [5], whereas local alignment can be obtained by the Smith-Waterman technique [6]. However, these two techniques are suitable to align a pair of sequences optimally. It is unrealistic and computationally complex for aligning more than three sequences optimally [7]. To diminish the computational complexity of the dynamic programming approach, it is essential to evolve new techniques to align sequences and to solve Multiple Sequence Alignment problems optimally. There are various approaches to align multiple sequences, such as progressive alignment techniques [8]–[10] and iterative algorithms [11]–[13] are proposed. In the Progressive sequence alignment technique, initially, two sequences are aligned, then subsequently selecting and adding other sequences with the aligned one to produce consensus alignment. The family of Clustal alignments [8] and Kalign [11] are well known progressive alignment techniques developed to solve MSA. However, there exist some drawbacks in progressive alignment techniques, such as an error propagation that is once a sequence is aligned, it cannot be undone, and order dependencies among input sequences also lead to improper results [14]. To evade the greediness of sequences and their error propagation, iterative algorithms are designed. So as to correctly align the sequences, iterative algorithms attempt to optimize the score functions. Iterative algorithms are genetic algorithms [11]–[13], swarm intelligence algorithms [14]–[16] are deployed to produce optimal scores by iteratively aligning sequences until the constraint is met. Iterative techniques take a long time to converge towards a better optimal solution even though further enrichments are needed to obtain optimal alignment solutions for long length sequences.



One of the swarm intelligence algorithms inspired by human beings' brainstorming behaviour is Brain Storm Optimization (BSO) Algorithm developed in 2011[17]. The brainstorming discussions are conducted by individuals to generate ideas that are novel and effective for solving many complex problems optimally. In BSO, new ideas are produced for each generation. Due to its simplicity, elaborated space for generating ideas and creative approach, BSO is an active algorithm for solving many real-world applications in a short period of time [18]. Due to the wide area of applications of BSO and its success, we proposed to solve the MSA problem optimally. In BSO, initially, random ideas are generated and are clustered together. Novel ideas are evolved from the assortment of current ideas from one or two groups; choosing current ideas is also impacted by three predefined probability variables. However, the BSO algorithm experiences an issue of local optima when solving more complex problems, which leads to the situation of premature convergence. So, various versions of BSO have been advanced to improve the attainment with respect to the efficacy of the BSO Algorithm [19]–[22] by balancing exploration and search exploitation abilities.

In the proposed work, to improve the optimization levels and efficacy of BSO for the optimal alignment of sequences, we proposed a new mutation operator, which can maintain more exploration and novelty in created ideas at each generation. This new mutation operator produces the ideas which are novel at each generation by maintaining a more diverged population. The results of the proposed MBSO-Mu have been verified on the benchmark sequence datasets. The alignment results show near-optimal performance when correlating to other evolutionary and the original BSO algorithms.

The Organization of the proposed work is as follows: section 2 describes multiple sequence alignment. Section 3 introduces about original BSO technique and its process. Section 4 concentrates on explaining the proposed MBSO-Mu algorithm, encoding problem, objective functions for evaluation. Section 5 detailed the compared results with the experiment conducted on different datasets. Conclusion and coming future improvement directions are listed in section 6.

II. MULTIPLE SEQUENCE ALIGNMENT: PROBLEM DESCRIPTION

Multiple Sequence Alignment (MSA) is aligning more than two sequences at a time. It reveals the biological information which is embedded in sequences. It is a prerequisite to many applications, which are identification of conserved regions or finding functionalities and structure of unknown proteins and estimation of divergence relationships in evolutionary analysis and sequence profiling. For a set of 'n' input sequences $S=\{s^1,s^2,s^3,\dots,s^n\}$ defined over an alphabet $\sum \cup \{-\}$. \sum consists of 4 or 20 elements from DNA or protein sequences, respectively. '-' is the symbol of a gap which is used to represent the insertion and deletion of mutations

in the sequences. $len=\{l^1,l^2,\dots,l^n\}$ is a vector of lengths of sequences in S respectively. To solve the MSA problem, scoring functions also are required to evaluate.

A. Brain Storm Optimization Algorithm (BSO)

BSO was developed in 2011 by the motivation of processes cultivated in human brains to develop creative ideas for solving complex problems. In BSO, a group of people can meet with their ideas and various backgrounds to generate a solution for a complex problem [17]. The steps in the BSO procedure involves initially creating N randomly generated ideas within the space of the solution set as the initial population. Each idea has evaluated by its objective function. The remaining steps in the evolutionary process involve grouping or clustering and updating ideas with new ideas, where a new population is generated based on the current ideas through various generations are discussed as follows.

Clustering individuals is the first step in BSO, where N randomly generated ideas are grouped into m user-specified number of clusters according to their objective function at each generation. The best-scored idea in each corresponding cluster is chosen as the centre cluster, which represents the entire cluster.

Replacing operation is the second step in BSO. Randomly choose one cluster centre with a chosen probability of Prreplace. The chosen cluster centre is replaced by a randomly selected cluster in the complete pool of individuals.

Selection of individuals all the selections are made based on the probability variables. Selecting one cluster or two clusters for creating new individuals can have estimated by the selected probability variable Prone. This means if $(rand < Prone)$ only one cluster is selected; otherwise, two clusters are selected for creating a new population. In the selected cluster(s), electing a cluster centre or individuals in the selected cluster is computed by the selected probabilities of Pronecenter and Prtwocenter. The chosen ideas are framed according to equation (1).

Updating operation creating new individuals by adding random information to the chosen ideas depends on the use of probability variables. After that, the original ith idea is being updated with the generated new individual, only if the generated idea fitness is superior to the current idea. The new idea is created using equation (2):

$$Y_{sel} = \begin{cases} Y_i & , \text{one cluster} \\ rand * Y_{i1} + (1 - rand) * Y_{i2} & , \text{two clusters} \end{cases} \quad (1)$$

$$Y_{new} = Y_{sel} + \xi \cdot N(0,1) \quad (2)$$

$$\xi = \text{logsig}((0.5 * \max Itr - \text{Curt Itr})/k) \cdot \text{rand}(0,1) \quad (3)$$

Where Y_{new} and Y_{sel} represented the new population and selected population, respectively, $N(0,1)$ is a Gaussian

distribution function with mean 0 and variance 1. ξ is another coefficient used to add weight to Gaussian function. ξ is estimated as follows through equation (3). maxItr and CurrItr represent a maximum number of iterations and current iteration, respectively. $\text{logsig}()$ is the sigmoid function, and k is the slope adjustment. rand is the random value generator that lies between 0 and 1.

B. Modified Brain Storm Optimization Algorithm With New Mutation Operator(MBSO-MU)

This section describes the modified BSO algorithm with an introduced new Mutation operator for getting optimal sequence alignment score. Various versions of the BSO algorithm were developed to enhance their exploration and search exploitation capabilities [22], [23]. Encouraged by the above process, we proposed an enhanced version of BSO with a proposed novel mutant operator to obtain an optimal alignment score for aligning multiple sequences named as MBSO-MuOp for MSA. The new mutation operator improves the exploration capability by diversifying the novel ideas.

C. Encoding candidate Solution

In the development process of initialization, N ideas are randomly generated as candidate solutions for the given computational task. Initially, sequence lengths are determined and take the maximum length among sequences. In global alignment, all sequences are equal in length. So computation of a number of gaps is required for each sequence. To make it a global alignment, consider the largest sequence length and add 0-15% length to insert gaps in the largest sequence. The number of gaps and gap positions in each sequence varies according to their sequence length. The number of gaps can be chosen between the range and specified in equation (4).

$$\left(\text{rand}((\text{max}(\text{len}) - \text{len}_i), (\text{max}(\text{len}) + [15\% * \text{max}(\text{len}) - \text{len}_i])) \right) \quad (4)$$

$$\text{max}(\text{len}) \leq \text{TLen} \leq (\text{max}(\text{len}) * 15\%) \quad (5)$$

Where $i=\{1,2,3,\dots,n\}$ for n sequences, len_i length of the i th sequence. Each idea ID_i is encoded as matrix form with the gap positions to be inserted into the given sequences. The aligned sequences length should be the same, and it could lie between the given range specified through equation (5). This 15% of gap insertions takes lesser time to initialize the ideas and also helps to find optimal alignment results verified to 20% of gap insertions [24]. More gap insertions also affect the optimality of alignments. After initializing the gap positions, encode them into the sequences, and the aligned sequences will be undergoing for evaluation procedure for fitness calculation.

D. Scoring function

To estimate the optimality of the alignments requires primitive objective functions. Several fitness functions are developed for evaluating MSA problems. Those functions are sum-of-pair score(SOP), Match Score(MS), similarity

score(SS), column score(CS), weighted sum-of pair score(WSP) as well are developed [25]. The sum-of-pair score is the widely used fitness function to evaluate alignments. In this proposed work, our aim is to maximize this fitness function iteration wise. The sum-of-pair objective function is formulated as in equations (6), (7) and (8).

$$S_i = \sum_{j=1, j \neq k}^n \sum_{k=1}^n P_{ijk} \quad (6)$$

Subject to

$$P_{ijk} = \begin{cases} 1 & \text{if } S_{ij} = S_{ik} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

$$\text{SOP} = \frac{\sum_{i=1}^l S_i}{\sum_{i=1}^{lr} S_r} \quad (8)$$

Where the number of sequences represented by n . l and lr is the lengths of columns in the alignment and reference alignment, respectively.

III. PROPOSED MUTATION OPERATOR

In this proposed technique, we introduce a novel mutation operator into the BSO algorithm for solving the MSA problem effectively. In our MBSO-Mu algorithm, the new mutant operator generates new ideas to explore more search space and increases exploration capability by utilizing the global best solution and worst solution. The quality of the population can be increased with the new mutant by exchanging information from different clusters to best and worst populations. The creation operation of MBSO-Mu uses the selected idea from the randomly chosen cluster according to their probabilities Prone, Proncenter and Ptwocenter. The new individual idea Y_{new} established by taking the differential value from the worst idea Y_{worst} and selected Y_{sel} ideas to the global best idea among them. The new mutant operation (MBSO/selected/worst-to-Gbest) for updating the current individual is represented in equation (9).

$$Y_{\text{new}} = Y_{\text{Gbest}} + F(Y_{\text{Worst}} - Y_{\text{Sel}}) \quad (9)$$

Where F is a scaling factor generated between 0 and 1, which affects the differential value between two ideas, the steps in the MBSO-Mu technique for solving MSA are detailed in the Algorithm. Giving sequences as input for generating random ideas. The randomly generated ideas as solutions encode the gap position occurrences represented in terms of a matrix. All the candidate ideas are evaluated by the sum-of-pair scoring function. The next step is to disrupt all the population as clusters and make the maximum scored solution as the centre of the cluster. For each individual, updating operation can be done through the selection probabilities, and the new ideas are created using the proposed mutant operator described in equation (9), which enhances the exploration capability and minimizes the local optima. The updating of the current idea takes place when the fitness of the newly constructed idea is greater. The procedure is repetitive till it reaches

termination measures or specified iterations reached. With this new updating mutation operator, MBSO-Mu can productively evade premature convergence and attain more optimal alignments. The MBSO-MU algorithm is detailed as follows.

Algorithm: MBSO-Mu algorithm for Multiple Sequence Alignment

- 1) Read ‘n’ number of sequences as input
- 2) n_idea=N, n_cluster=m; maxIt /* initializing the population, clusters & termination criteria as number of iteration*/
- 3) N ideas are Initialized randomly
- 4) While (iter<= maxItr)
- 5) Do
- 6) Isolate N solutions into m clusters and keep the most fitted solutions as a centre of the clusters
- 7) evaluate individuals using the objective function
- 8) If(rand(0,1)<=Preplace)
- 9) select one cluster centre and replace it with the selected cluster individual
- 10) End if
- 11) For(i=1 to N)
- 12) If(rand(0,1)<Prone)
- 13) randomly choose one cluster among C clusters
- 14) If(rand(0,1)<Pronecenter)
- 15) generate new individual Ynew using the formula in equation (5)
- 16) Else
- 17) randomly select one individual idea from the cluster
- 18) generate new individual Ynew using the formula in equation (6)
- 19) End If
- 20) Else
- 21) select two clusters among m clusters randomly
- 22) If(rand(0,1)<Prtwocenter)
- 23) randomly select two cluster centres
- 24) generate new individual Ynew using the formula in equation (5) using selected centres
- 25) Else
- 26) Select two individual ideas from each of the selected clusters
- 27) generate new individual Ynew using the formula in equation (6) using selected individuals
- 28) End If
- 29) End If
- 30) eval_fit(Ynew) /*Evaluate the fitness of new idea
- 31) If fit_new>fit_old
- 32) replace the old idea with a newly created idea
- 33) Else
- 34) Keep old idea
- 35) End If
- 36) End for
- 37) End while

IV. RESULTS AND DISCUSSION

This present section discusses the benchmark dataset for alignment, parameter settings and optimal alignment score for the MBSO-Mu algorithm. The alignment results of the

proposed MBSO-Mu algorithm on MSA has been tested on the benchmark dataset is OXBench [26]. OXBench contains protein sequences of various lengths. The alignment results MBSO-Mu algorithm for MSA is demonstrated with the original Genetic algorithm(GA) and classical Brain Storm Optimization(BSO) Algorithm. The proposed MBSO-Mu algorithm was executed for 10 iterations for the datasets with 5 clusters in each iteration. MBSO-Mu got an optimal score at iterations 8,9, and 10; thus, the stopping criteria can be taken as maxitr=10.

Table 1. Best sop scores for OXBench sequences for MBSO-Mu along with original GA and BSO algorithms.

S.No.	Sequence Number	GA	BSO	MBSO-Mu
1	_75	131	194	222
2	_9	156	375	396
3	_8	38	326	358
4	_14	75	95	100
5	_19	24	116	130
6	_46	12	45	47
7	_54	33	52	54
8	_60	34	44	55
9	_71	23	47	46
10	_99	133	169	178
11	_4t2	128	156	143
12	_4t3	156	209	213
13	_8t2	231	256	255
14	_9t3	364	431	412
15	_93	456	885	893

Table 2. Average sop scores for OXBench sequences for MBSO-Mu along with original GA and BSO algorithms

S.No.	Sequence Number	GA	BSO	MBSO-Mu
1	_75	116.5	153.9	212.3
2	_9	135.6	341.2	375.4
3	_8	32.6	304.8	334.3
4	_14	56.2	82.5	89.9
5	_19	23.1	101.8	124.5
6	_46	11.5	41.5	45.5
7	_54	32.4	46.3	48.9
8	_60	31.4	38.9	48.9
9	_71	19.3	43.5	41.2
10	_99	435.1	823.6	888.2
11	_4t2	112.5	145.3	174.2
12	_4t3	122.4	148.4	139.8
13	_8t2	136.9	174.5	207.2
14	_9t3	225.8	228.2	226.9
15	_93	358.2	401.6	354

All the above-specified algorithms are compared for the randomly selected datasets from the OXBench dataset. Table 1 and Table 2 summarize and show the sequence wise comparative results achieved for the GA, BSO and

MBSO-Mu algorithms. The best and average sop scores iterated by 10 runs individually are obtained for the datasets are tabulated in Table 1 and Table 2. From Table 1 and Table 2, it is observed that MBSO-Mu gave better and optimal results for 11 sequences and for the remaining 4 sequences proposed, one gave a near-optimal score. In the other cases, BSO gave better results.

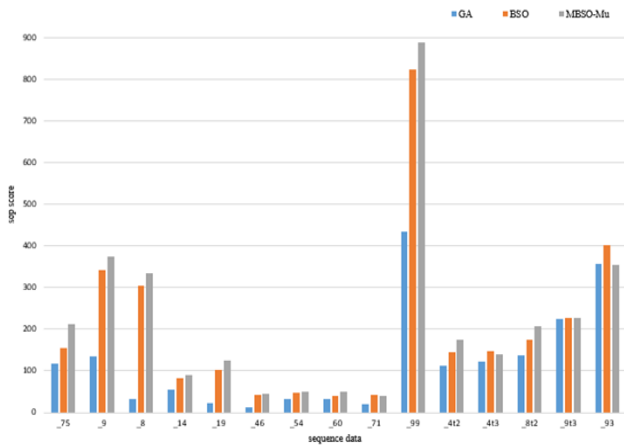


Fig 1. Avg sop score performance comparisons of GA, BSO and MBSO-Mu algorithms for the OXBench datasets

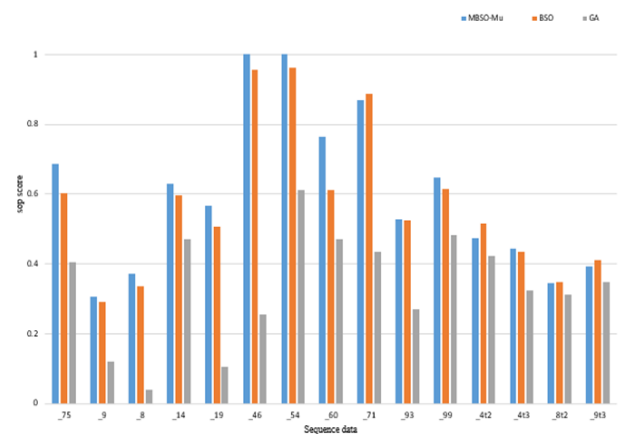


Fig 2. Avg sop score performance comparisons of GA, BSO and MBSO-Mu algorithms for the OXBench datasets

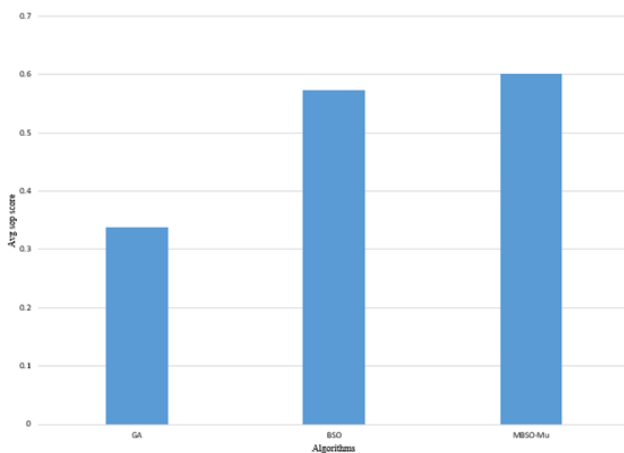


Fig 3. average performance comparison of GA, BSO and MBSO-Mu algorithms

Fig.1 and Fig.2 depicts the average sop score for the alignment of the proposed GA and BSO algorithm for the protein sequences. The proposed MBSO-Mu technique shows better performance or near equal in other cases compared to GA and BSO algorithms which shows that MBSO-Mu lessens the chances of getting into local optima. Fig.3 depicts the overall comparative results for the algorithms. In this scenario, also MBSO-Mu achieves better performance for getting optimal alignment of sequences.

IV. CONCLUSIONS

The novelty in the proposed MBSO-Mu algorithm is incorporating mutation operator into the BSO technique for the computation of multiple sequence alignment problems. MBSO-Mu increase the quality of aligned sequence solutions in the search space and also decreases the chances of getting into the situation of local optima. The demonstrated aligned results show that the MBSO-Mu algorithm guides the search towards achieving more optimal alignment results for solving the MSA problem. The proposed Algorithm yields a better alignment score compared to the GA and BSO algorithms. In future, performance comparison of computational complications among algorithms is also a remarkable area. Improving the efficiency of the Algorithm in aligning lengthy sequences with an optimal score is also the future scope.

REFERENCES

- [1] C. Notredame, Recent progress in multiple sequence alignment, A survey, *Pharmacogenomics*, 3(1) (2002) 131–144, doi, 10.1517/14622416.3.1.131.
- [2] P. Bonizzoni and G. Della Vedova, The complexity of multiple sequence alignment with SP-score that is a metric, *Theor. Comput. Sci.*, 259(1–2) (2001) 63–79, doi, 10.1016/S0304-3975(99)00324-2.
- [3] L. Wang and T. Jiang, On the Complexity of Multiple Sequence Alignment, *J. Comput. Biol.*,1(4) (1994) 337–348, doi, 10.1089/cmb.1994.1.337.
- [4] J. D. Thompson, F. Plewniak, and O. Poch, A comprehensive comparison of multiple sequence alignment programs, *Nucleic Acids Res.*, 27(13) (1999) 2682–2690, doi, 10.1093/nar/27.13.2682.
- [5] Saul B.NeedlemanChristian D.Wunsch, A general method applicable to search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, (1970) 443–453, 1970, doi, doi.org/10.1016/0022-2836(70)90057-4.
- [6] L. Santulli et al., Epilessia generalizzata idiopatica con stati di male mioclonici palpebrali e duplicazione 7q31 coinvolgente KCND2, *Boll. - Lega Ital. contro l'Epilessia*, 144 (2012) 173–175.
- [7] J. Stoye, S. W. Perrey, and A. W. M. Dress, Improving the divide-and-conquer approach to sum-of-pairs multiple sequence alignment, *Appl. Math. Lett.*, vol. 10, no. 2, pp. 67–73, 1997, doi, 10.1016/S0893-9659(97)00013-X.
- [8] M. A. Larkin et al., Clustal W and Clustal X version 2.0, *Bioinformatics*, 23(21) (2007) 2947–2948, doi, 10.1093/bioinformatics/btm404.
- [9] D.-F. Feng and R. F. Doolittle, Progressive sequence alignment as a prerequisite to correct phylogenetic trees, *J. Mol. Evol.*, 25(4) (1987) 351–360, doi, 10.1007/BF02603120.
- [10] J. Armstrong et al., Progressive Cactus is a multiple-genome aligner for the thousand-genome era, *Nature*, 587(7833) (2020) 246–251, doi, 10.1038/s41586-020-2871-y.
- [11] T. Lassmann, Kalign 3, Multiple sequence alignment of large datasets, *Bioinformatics*, 36(6) (2020) 1928–1929, doi, 10.1093/bioinformatics/btz795.
- [12] J. T. Horng, L. C. Wu, C. M. Lin, and B. H. Yang, A genetic algorithm for multiple sequence alignment, *Soft Comput.*, . 9(6) (2005) 407–420, doi, 10.1007/s00500-004-0356-9.

- [13] Z. J. Lee, S. F. Su, C. C. Chuang, and K. H. Liu, Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment, *Appl. Soft Comput. J.*, vol. 8, no. 1, pp. 55–78, 2008, doi, 10.1016/j.asoc.2006.10.012.
- [14] N. Moustafa, M. Elhosseini, T. H. Taha, and M. Salem, Fragmented protein sequence alignment using two-layer particle swarm optimization (FTLPSO), *Fragmented protein sequence alignment, J. King Saud Univ. - Sci.*, 29(2) (2017) 191–205, doi, 10.1016/j.jksus.2016.04.007.
- [15] C. Öztürk and S. Aslan, A new artificial bee colony algorithm to solve the multiple sequence alignment problem, *Int. J. Data Min. Bioinform.*, 14(4) (2016) 332–353, doi, 10.1504/IJDMB.2016.075823.
- [16] P. Jeevana Jyothi, K. P. K., S. M. Raiyyan, and T. Rajasekhar, MSA, An Application of Brain Storm Optimization Algorithm, *SSRN Electron. J.*, (2020), doi, 10.2139/ssrn.3646174.
- [17] Y. Shi, Brainstorm optimization algorithm, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 6728(1) (2011) 303–309, doi, 10.1007/978-3-642-21515-5_36.
- [18] S. Cheng et al., A comprehensive survey of brainstorm optimization algorithms, *2017 IEEE Congr. Evol. Comput. CEC 2017 - Proc.*, (2017) 1637–1644, doi, 10.1109/CEC.2017.7969498.
- [19] Z. Song, J. Peng, C. Li, and P. X. Liu, A Simple Brain Storm Optimization Algorithm with a Periodic Quantum Learning Strategy, *IEEE Access*, 6 (2017) 19968–19983, doi, 10.1109/ACCESS.2017.2776958.
- [20] Z. Cao, X. Hei, L. Wang, Y. Shi, and X. Rong, An Improved Brain Storm Optimization with Differential Evolution Strategy for Applications of ANNs, *Math. Probl. Eng.*, 2015 (2015), doi, 10.1155/2015/923698.
- [21] X. Xue and J. Lu, A Compact Brain Storm Algorithm for Matching Ontologies, *IEEE Access*, 8 (2020) 43898–43907, doi, 10.1109/ACCESS.2020.2977763.
- [22] A. Cervantes-Castillo and E. Mezura-Montes, A modified brain storm optimization algorithm with a special operator to solve constrained optimization problems, *Appl. Intell.*, 50(12) (2020) 4145–4161, 2020, doi, 10.1007/s10489-020-01763-8.
- [23] Y. Sun, J. Wei, T. Wu, K. Xiao, J. Bao, and Y. Jin, Brainstorm optimization using a slight relaxation selection and multi-population based creating ideas ensemble, *Appl. Intell.*, 50(10) (2020) 3137–3161, doi, 10.1007/s10489-020-01690-8.
- [24] M. Kumar and H. Om, *A Hybrid Bio-Inspired Algorithm*. Springer International Publishing.
- [25] S. Lalwani, H. Sharma, M. K. Mohan, and K. Deep, An efficient Bi-level discrete PSO variant for multiple sequence alignment, vol. 741. Springer Singapore, (2019).
- [26] G. P. S. Raghava, S. M. J. Searle, P. C. Audley, J. D. Barber, and G. J. Barton, OXBench, A benchmark for evaluation of protein multiple sequence alignment accuracy, *BMC Bioinformatics*, 4 (2003) 1–23, doi, 10.1186/1471-2105-4-47.