Computing and Informatics, Vol. 35, 2016, 963–985

EFFICIENT TWO-LEVEL SWARM INTELLIGENCE APPROACH FOR MULTIPLE SEQUENCE ALIGNMENT

Soniya Lalwani

Department of Mathematics, Malaviya National Institute of Technology
R & D, Advanced Bioinformatics Centre, Birla Institute of Scientific Research
Jaipur, India
e-mail: slalwani.math@gmail.com

Rajesh Kumar

Department of Electrical Engineering, Malaviya National Institute of Technology
Jaipur, India
e-mail: rkumar.ee@gmail.com

Nilama Gupta

Department of Mathematics, Malaviya National Institute of Technology
Jaipur, India
e-mail: guptanilama@gmail.com

Abstract. This paper proposes two-level particle swarm optimization (TL-PSO), an efficient PSO variant that addresses two levels of optimization problem. Level one works on optimizing dimension for entire swarm, whereas level two works for optimizing each particle's position. The issue addressed here is one of the most challenging multiple sequence alignment (MSA) problem. TL-PSO deals with the arduous task of determination of exact sequence length with most suitable gap positions in MSA. The two levels considered here are: to obtain optimal sequence length in level one and to attain optimum gap positions for maximal alignment score in level two. The performance of TL-PSO has been assessed through a comparative study with two kinds of benchmark dataset of DNA and RNA. The efficiency of the proposed approach is evaluated with four popular scoring schemes at spe-

cific parameters. TL-PSO alignments are compared with four PSO variants, i.e. S-PSO, M-PSO, ED-MPSO and CPSO-S_k, and two leading alignment software, i.e. ClustalW and T-Coffee, at different alignment scores. Hence obtained results prove the competence of TL-PSO at accuracy aspects and conclude better score scheme.

Keywords: Particle swarm optimization, multiple sequence alignment, average pairwise sequence identity, time complexity, scoring schemes

Mathematics Subject Classification 2010: 68T10, 92B20

1 INTRODUCTION

Multiple sequence alignment (MSA) is an intricate and challenging area of bioinformatics. It has wide applicability in discovering functional, structural and evolutionary information of biological sequences, species and the ancestors. MSA plays a cogent role in secondary and tertiary structure prediction, phylogenetic tree construction and conserved domain identification. It is a technique of arranging the sequences by inserting gaps in such a way that it may produce maximum number of matches in a column.

MSA is a NP-complete problem, therefore many algorithms have been developed to solve it. Current state-of-art algorithms are roughly classified in four categories: Progressive approach; Exact approach; Consistency based approach and Iterative approach. Progressive approach initially provides an alignment with most similar sequences and then gradually aligns lesser similar sequences. The most popular software based on this approach is ClustalW [1]. Progressive approaches are dependent on initial provided alignment and require appropriate scoring scheme, which is the drawback of these approaches. The basic example of exact approaches is dynamic programming (DP). In DP, a two-dimensional alignment path matrix is built for pairwise alignment, followed by filling the matrix by successive calculation of score values. Then, the optimal path for maximum score is traced for each pair of sequences so as to form MSA. This approach lacks efficiency as for time and accuracy. Consistency based approaches target to achieve the maximum consensus optimal pairwise alignment within the created library of alignments of provided sequences. T-Coffee [2] and DIALIGN [3] are the most popular softwares based on this approach. This approach has the drawback of being quite time consuming. Iterative approaches perform iterations to improve obtained solutions steadily. This approach includes hidden Markov model training [4], simulated annealing, evolutionary algorithms and swarm intelligence (SI) [5] techniques.

All the above discussed state-of-art approaches including the most popular sequence alignment tools based on these approaches, i.e. ClustalW and T-Coffee, have certain limitations, therefore there remains a scope of developing more heuristics which produce better alignments. Proposed heuristic is based on particle swarm

optimization (PSO), which belongs to fourth category, i.e. iterative approach, the most salient SI based approach. PSO has been proven to be a potent approach for MSA with numerous kinds of proposed variants discussed in [6].

The proposed approach two-level PSO (TL-PSO) has adopted the exponentially decreasing weight scheme by the conclusion of our previously developed MPSO (modified PSO) for MSA [7]. MPSO produced good results for S8 dataset (containing 5 sequences with maximum sequence length 10), but was not promising further, since the concept of random sequence length did not make any notable effect for complex problems. Due to random sequence length it was not able to converge towards any improved sequence length for long sequences. During literature survey regarding the determination of maximum allowed gap length, it was found that the approaches are varying. Some authors [8, 9, 10] proposed it to be 0.2 times the longest sequence, some [11, 12, 13] proposed it to be 0.5 times the length of the longest sequence, whereas some [14] proposed it to be 0.4 times the length of the longest sequence. Hence, this became a challenge to make the algorithm converge towards a suitable sequence length that produces optimal scores.

TL-PSO introduces a unique strategy of determining optimal alignment by employing PSO algorithm in two levels of the problem. The first level of TL-PSO determines the optimal sequence length which is transfered to the second level for finding out the optimal gap positions so as to produce a better alignment score. Both the parameters of both the levels get iteratively improved and optimize the alignment objective. Unlike ClustalW, TL-PSO is not dependent on the quality of initial alignment, besides this TL-PSO efficiently produces more accurate alignments than state-of-art approaches and family of PSO algorithms.

The rest of the paper is organized as follows: Section 2 presents the algorithm and concepts of PSO; Section 3 discusses all the concepts and problem formulation regarding MSA; Section 4 depicts proposed algorithm TL-PSO; Section 5 contains the details of experimental setup for benchmark dataset and for TL-PSO algorithm parameters; Section 6 presents the simulation results followed by the conclusions in Section 7.

2 PARTICLE SWARM OPTIMIZATION

PSO is a population-based heuristic optimization algorithm firstly introduced by Kennedy and Eberhart [15] for simulating social behaviour, such as embodiment of the movement of organisms in a bird flock or fish school. These birds (or fishes) are called particles and the group of particles is called swarm. These particles move in different directions in search for an optimal solution while communicating with each other and updating their positions and velocities through interaction for improving corresponding solutions. This communication of particles in a swarm follows certain topologies. These topologies present the behaviour of particles during social interaction and movement towards better particle. Present problem formulation is based on star topology, in which an entirely connected network offers each particle

to interact with the other particles. The procedure can be formulated as mentioned below:

The objective function for PSO is:

$$\min f(x) \qquad \text{s.t.} \qquad x \in S \subseteq R^D \tag{1}$$

where x is a matrix containing decision variables, composed of m vectors defined as $x = [\vec{x}^1, \vec{x}^2, \ldots, \vec{x}^m]$ with dimension D. S is the feasible solution space of the problem. For the i^{th} particle ($i = 1, 2, \ldots, m$), the three vectors governing the movement of the particles are as below:

1. Position of i^{th} particle can be presented by $\vec{x}^i$ $(x_1^i, x_2^i, \ldots, x_D^i)$, where each component of this vector denotes a decision variable of the problem.
2. Velocity of i^{th} particle can be presented by $\vec{v}^i$ $(v_1^i, v_2^i, \ldots, v_D^i)$, where each component of this vector presents an increment of the current position.
3. Each particle has its own best performance in the swarm defined by personal best i.e. $pbest^i$ $(p_1^i, p_2^i, \ldots, p_D^i)$.

At t^{th} iteration the previous velocity $v^i(t)$ and position $x^i(t)$ are updated as follows:

$$v^i(t+1) = wv^i(t) + c_1 r_1 \left(pbest^i(t) - x^i(t)\right) + c_2 r_2 \left(gbest(t) - x^i(t)\right), \tag{2}$$

$$x^i(t+1) = x^i(t) + v^i(t+1), \tag{3}$$

$$\text{with } x^i(0) \sim U\left(x_{\min}, x_{\max}\right).$$

In Equation (2) several parameters are introduced that play a significant role in particle's movement. The velocity v^i is monitored by employing velocity clamping over a range between lower and upper bound, i.e. $[v_{\min}, v_{\max}]$, where $v_{\min} = -v_{\max}$. This clamping is essential for limiting the particle from getting accelerated out of control and decreasing the potential divergent behaviour. The initial approximation of velocity vectors is randomly generated within predetermined $v_{\min}$ and $v_{\max}$. Inertia weight w (generally, $0 \leq w \leq 1$) plays the role of scaling factor over the previous velocity which results in either acceleration or deceleration on trajectory of particle. A study on impact of dynamically changing inertia weights performed in [7] depicts the effect on the convergence of particles. c_1 is the cognitive acceleration coefficient, which represents the confidence in the particle's own experience, whereas c_2 is the social acceleration coefficient representing the confidence in the neighborhood's experience. If $c_1 > c_2$, then the particle is biased towards own best position, whereas if $c_1 < c_2$ then the particle is attracted more towards best position of neighborhood. Usually c_1 and c_2 are set equal with the constraint $c_1 + c_2 \leq 4$. r_1 and r_2 are uniform random numbers in range $[0, 1]$.

Besides these, the number of particles and number of iterations are also significantly influencing factors for performance of PSO. Large number of particles have the advantage of getting spread fast; hence algorithm explores more solutions and

finds optimal solution in small number of iterations. But, large number of particles increase complexity of the algorithm. Small number of particles have advantage of being less computationally complex, but less assured to achieve optimal solution. In the similar way, small number of iterations may not attain the optimal solution, whereas, large number of iterations may be computationally expensive. If the solution can be achieved in very early iterations and the stopping criteria is not defined, then the large number of iterations result in waste of time and computation memory.

Equation (3) shows the position update $x^i(t)$ at iteration t for i^{th} particle, performed by adding the velocity to its current position, whereas the initial approximate of $x^i(0)$ is randomly generated within the predetermined search domain $[x_{\min}, x_{\max}]$. The initial approximation of $pbest^i$ is generally set as the initial current position vector. *pbest* at iteration $t+1$ is updated using the following equation:

$$pbest^i(t+1) = \begin{cases} pbest^i(t) & \text{if } f\left(x^i(t+1)\right) \geq f\left(pbest^i(t)\right), \\ x^i(t+1) & \text{if } f\left(x^i(t+1)\right) < f\left(pbest^i(t)\right). \end{cases} \tag{4}$$

The best of the positions, i.e. *gbest*, found among all particles from the entire swarm, is updated as follows:

$$\begin{aligned} gbest(t) &= x_k \in \left\{pbest^1(t), pbest^2(t), \ldots, pbest^m(t)\right\} \\ \text{where } f(x_k) &= \min\left\{f\left(pbest^1(t)\right), f\left(pbest^2(t)\right), \ldots, f\left(pbest^m(t)\right)\right\}. \end{aligned} \tag{5}$$

The main features of the PSO algorithm are: simple concept, easily implementable, robust to control parameters and better computational efficiency comparative to many other mathematical algorithms and heuristic optimization techniques. PSO is applicable to nonlinear and non-continuous optimization problems as well [16].

3 PROBLEM DESCRIPTION AND OBJECTIVE

The objective taken here is to obtain optimal MSA at maximum alignment score. To quantify the quality of an alignment, score schemes are applied, which basically play to obtain the maximization of similarity and minimization of gaps at optimum sequence length. Sequence length determines the number of maximum allowed gaps to enhance the maximization of symbol similarity score minus gap score. This section presents the sequence length schemes, score schemes and gap penalty schemes.

3.1 Sequence Length

Optimum sequence length is the parameter which decides the optimum number of gaps allowed. The generalized maximum allowed length of the sequence ψ can be formulated as:

$$\psi = \xi + \kappa \tag{6}$$

where ξ is length of the longest sequence and κ is the maximum allowed gap length.

3.2 Scoring Schemes

Numerous scoring schemes are available to obtain the alignment score of MSA. The four most popular approaches for obtaining alignment score are depicted here. First two approaches, i.e. similarity score (SS) [17] and match score (MS) [18], can be applied when reference alignment is not available, whereas next two approaches, i.e. Sum-of-Pairs (SoP) score and column match (CM) score [19], can be applied only when the reference alignment is available. The similarity score (SS) scheme can be formulated as:

$$\max(\text{SS}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} score(S_i, S_j) \tag{7}$$

subject to:

$$score(S_i, S_j) = \begin{cases} a & \text{if } S_i = S_j, \\ b & \text{if } S_i \neq S_j \text{ and } S_i \neq \text{'-'} \ \& \ S_j \neq \text{'-'}, \\ c & \text{if } S_i \neq S_j \text{ and } S_i = \text{'-'} \text{ or } S_j = \text{'-'} \end{cases} \tag{8}$$

where n is the number of sequences; '-' represents a gap; a, b and c are the scores assigned to match, mismatch and gaps respectively. Scores a and b are determined with the score schemes, whereas c is determined with the gap penalty model described in next subsection. A number of score matrices are available for determining match and mismatch score. The mostly used matrices for Protein sequences are PAM and BLOSUM series; for DNA are IUB matrices. ClustalW uses CLUSTAL matrix in scoring MSA, which is determined by the best fit according to the parameters given.

Second approach known as the match score (MS) scheme is formulated as:

$$\max(\text{MS}) = \sum_{i=1}^{l} M_i \left\{ 1 + \frac{M_i}{n} \right\} \tag{9}$$

where M_i is the number of matches in the i^{th} column and l is the length of sequence.

The third approach is much similar to SS approach defined in Equation (7); the difference lies in the requirement of the reference alignment. For a test alignment of n sequences consisting of l columns the SoP score is defined as:

$$\max(\text{SoP}) = \frac{\sum_{i=1}^{l} S_i}{\sum_{r=1}^{q} S_r} \tag{10}$$

where q is the number of columns in the reference alignment, S_r is the score for the r^{th} column in the reference alignment which may be written as the product of q and ${}^{n}C_2$. S_i is the score for the i^{th} column of tested alignment defined as:

$$S_i = \sum_{j=1,\, j\neq k}^{n} \sum_{k=1}^{n} p_{ijk}. \tag{11}$$

The i^{th} column in the alignment is represented by S_{i1}, S_{i2}, ..., S_{in} with the condition:

$$p_{ijk} = \begin{cases} 1 & \text{if residues } S_{ij} \ \& \ S_{ik} \text{ are aligned in the reference alignment,} \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

The fourth approach applied for scoring MSA is to obtain maximum column match score (CM) formulated as:

$$\max(\text{CM}) = \frac{\sum_{r=1}^{l} N_r}{r} \tag{13}$$

subject to:

$$N_r = \begin{cases} 1 & \text{if residues in the } r^{\text{th}} \text{ column are aligned in the reference alignment,} \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

3.3 Gap Penalty Scheme

The alignment of the sequences is performed by introducing some gaps at specific positions so as to obtain maximum number of matches and maximum similarity score. For each introduced and extended gap, some gap penalty is deducted from the score. The most popular gap penalty models are linear gap penalty and affine gap penalty. Linear gap penalty model does not concern with gap length, i.e. the penalty of gap opening and gap extension remains same, as formulated below:

$$\mu = \omega * t \tag{15}$$

where μ stands for the gap penalty for linear gap penalty model, ω stands for constant gap penalty value and t is the total number of gaps.

Affine gap penalty model concerns with the number of opening and extended gaps, as formulated below:

$$\eta = \alpha + (t-1) * \beta \tag{16}$$

where η stands for the gap penalty for affine gap penalty model, α for the gap open penalty and β for the gap extension penalty. The gap length t stands for a single string of gaps, whereas in an alignment there could be a number of gap openings, hence number of gap extensions. All these penalties are added to obtain the total gap penalty of the alignment.

3.4 Final Fitness Function

The final fitness function from above scoring schemes and gap penalty schemes becomes:

$$F = \lambda_1(\textit{Alignment Score}) - \lambda_2(\textit{Gap Score}) \tag{17}$$

where F is the final fitness value of an alignment, which serves as the final objective function to be maximized. The maximization problem is converted to minimization problem by $\max(F) = \min(-F)$. Equation (17) represents a multi-objective problem converted to a single objective problem using weighted aggregation method. The independent weights λ_1 and λ_2 are defined for conflicting objectives alignment score and gap score respectively. Here weights λ_1 and λ_2 are given the values of one-one as the common practice in sequence alignment. Alignment score can be obtained by any method depicted by Equations (7), (9), (10) and (13), whereas gap score can be obtained by any method presented by the formulae from Equations (15) and (16). Hence, the resultant objective function becomes:

$$\min(f) = \textit{Gap Score} - \textit{Alignment Score}. \tag{18}$$

4 TWO-LEVEL PARTICLE SWARM OPTIMIZATION

The objective of the proposed work is to develop an algorithm serving objective of Equation (18), by quantifying suitable sequence length formulated by Equation (6). For the purpose, TL-PSO algorithm is proposed as depicted by Figure 1.

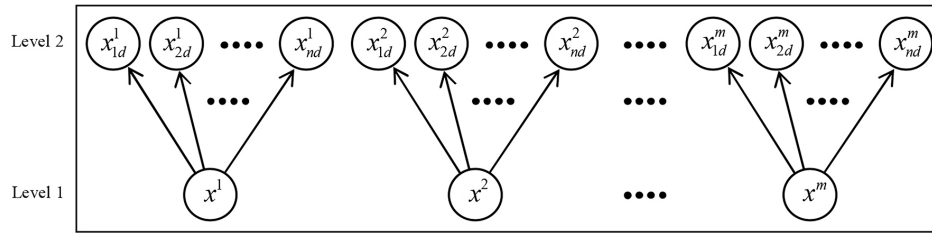


Figure 1. The structure of proposed TL-PSO

Level one determines the maximum allowed sequence length, whereas, level two works on the obtained sequence length of level one. Separate PSO runs for both the levels. As evident from Figure 1, the two levels of the algorithm are defined for MSA as follows: first level is to optimize ψ represented by x^i for i^{th} swarm and second level is to optimize f at predetermined value of ψ for particles x^i_{jd}, $\forall j = 1, 2, \ldots, n$; $\forall d = 1, 2, \ldots, D$ for optimal gap positions. Algorithm 1 presents the outline of the pseudo code for MSA using TL-PSO.

```
Load n sequences in fasta format and save all in S
Initialize matrices for sequence length and gap position for each particle, i.e.
 position matrices; velocity matrices; particle best matrices and global best
 matrices
While(not terminated)
 Do {
  For all particles i ∈ {1, 2, ..., m} do:
      For all sequences j ∈ {1, 2, ..., n} with dimension D
        Update velocity for sequence length by Equation (27) for each swarm
        Update sequence length by Equation (28) for each swarm
        Update velocity for gap position matrix at the dimension determined
        by sequence length ∀j ∈ {1, 2, ..., n}, by Equation (29) for each
        particle
        Update gap position matrix, by Equation (30) for each particles
        Incorporate all gap positions of matrix in binary matrix G
        Incorporate gaps in S from G
        Evaluate the fitness value of all particles from Equation (18)
        Update particle best and global best using Equations (31)–(32) for
        sequence length and gap position both
      end For
  end For
 }
```

Algorithm 1: Pseudo code for MSA using TL-PSO

Level one is defined on each swarm $i = 1, 2, \ldots, m$, whereas level two is defined on the entire population of every swarm. The algorithm is defined as follows:

Step 1. Parameter determination:
Set number of particles, swarm size, number of iterations and TL-PSO parameters (w, c_1, c_2).

Step 2. Initialization:

1. Determine sequence length for all particles of each swarm as:

$$x^i_{1l}(0) = \xi * \{1 + int(0.3 * rand * \rho)\} \qquad \forall i = 1, 2, \ldots, m. \tag{19}$$

   where *int* stands for output in round off form to the nearest integer value, *rand* stands for a random number in range $[0, 1]$ and ρ is determined by Equation (35).
2. Generate randomly the initial velocity v^i_{1l}, $\forall i = 1, 2, \ldots, m$ for particles of level one by:

$$v^i_{1l}(0) = int\left(rand * \left(u^l - l^l\right)\right) \tag{20}$$

where u^l is the upper bound of the length which is equal to ψ and l^l is the lower bound of the length which is equal to ξ. For sequence length, upper bound of velocity v_{max} is $(u^l - l^l)$ and lower bound v_{min} is $-(u^l - l^l)$.

3. Determine initial personal best sequence length for particles of each swarm by:
$$x_{1l}^{pbest(i)}(0) = x_{1l}^{i}(0) \qquad \forall i = 1, 2, \ldots, m. \tag{21}$$
Move to level two.
4. Generate randomly the initial gap positions for the particles by:
$$x_j^i(0) = int(rand * x_{1l}^i). \tag{22}$$
The position matrix is defined by:
$$x_j^i = \{x_{j1}^i, x_{j2}^i, \ldots, x_{jd}^i\} \qquad \forall i = 1, 2, \ldots, m; \quad \forall j = 1, 2, \ldots, n.$$
5. Generate randomly the initial velocity for the particles by:
$$v_j^i(0) = int(rand * (u^p - l^p)) \tag{23}$$
where u^p is the upper limit of gap position which is equal to x_{1l}^i and l^p is the lower limit of gap position which is equal to 1. For gap position, velocity upper bound v_{max} is $(u^p - l^p)$ and lower bound v_{min} is $-(u^p - l^p)$. The velocity is defined by:
$$v_j^i = \left\{v_{j1}^i, v_{j2}^i, \ldots, v_{jd}^i\right\} \qquad \forall i = 1, 2, \ldots, m; \quad \forall j = 1, 2, \ldots, n.$$
6. Determine initial personal best gap position of each particle by:
$$x_j^{pbest(i)}(0) = x_j^i(0) \qquad \forall i = 1, 2, \ldots, m; \quad \forall j = 1, 2, \ldots, n. \tag{24}$$
7. Determine global best gap position of entire swarm by:
$$x_{2l}^{gbest}(0) = x_{gbest}^q(0) \tag{25}$$
$$\text{where } x_{gbest}^q(0) = \arg\min_{i=1}^{m} f\left(x^{pbest(i)}(0)\right), \qquad q \in (1, 2, \ldots, m).$$
8. Determine global best sequence length of the entire swarm by:
$$x_{1l}^{gbest}(0) = x_{1l}^q(0) \tag{26}$$
where q is determined by Equation (25). Global best sequence length belongs to level 1, but is determined after evaluation of entire swarm obtained as the output of level 2 with respect to objective.

Step 3: Set $t = 0$.

Step 4: Update velocity and position (sequence length) vector for level one:

$$v_{1l}^{i}(t+1) = wv_{1l}^{i}(t) + c_1 r_1 \left(x_{1l}^{pbest(i)}(t) - x_{1l}^{i}(t)\right) + c_2 r_2 \left(x_{1l}^{gbest}(t) - x_{1l}^{i}(t)\right), \quad \forall i = 1, 2, \ldots, m, \tag{27}$$

$$x_{1l}^{i}(t+1) = x_{1l}^{i}(t) + v_{1l}^{i}(t+1), \qquad \forall i = 1, 2, \ldots, m. \tag{28}$$

Step 5: Update the velocity and gap position of the particles for level two as:

$$v_{j}^{i}(t+1) = wv_{j}^{i}(t) + c_1 r_1 \left(x_{j}^{pbest(i)}(t) - x_{j}^{i}(t)\right) + c_2 r_2 \left(x_{2l}^{gbest}(t) - x_{j}^{i}(t)\right), \quad \forall i = 1, 2, \ldots, m; \quad \forall j = 1, 2, \ldots, n, \tag{29}$$

$$x_{j}^{i}(t+1) = x_{j}^{i}(t) + v_{j}^{i}(t+1), \qquad \forall i = 1, 2, \ldots, m; \quad \forall j = 1, 2, \ldots, n. \tag{30}$$

Hence obtained gap positions matrix is used to create a binary matrix G, showing 1 for nucleotide and 0 for gap.

Step 6: Incorporate the gap positions from matrix G in sequence and calculate objective f from Equation (18).

Step 7: Update the personal best and global best gap position as:

$$x_{j}^{pbest(i)}(t+1) = \begin{cases} x_{j}^{pbest(i)}(t) & \text{if } f\left(x_{j}^{i}(t+1)\right) \geq f\left(x_{j}^{pbest(i)}(t)\right), \\ x_{j}^{i}(t+1) & \text{if } f\left(x_{j}^{i}(t+1)\right) < f\left(x_{j}^{pbest(i)}(t)\right), \end{cases} \quad \forall i = 1, 2, \ldots, m; \forall j = 1, 2, \ldots, n, \tag{31}$$

$$x_{2l}^{gbest}(t+1) = x_{gbest}^{q}(t+1), \tag{32}$$

$$\text{where } x_{gbest}^{q}(t+1) = \arg\min_{i=1}^{m} f\left(x^{pbest(i)}(t+1)\right), \qquad q \in (1, 2, \ldots, m). \tag{33}$$

Step 8: Extract the corresponding sequence lengths regarding Equations (31) and (32). These lengths will be respective *pbest* and *gbest* sequence lengths represented by $x_{1l}^{pbest(i)}(t+1)$ and $x_{1l}^{gbest}(t+1)$, respectively.

Step 9: $t = t + 1$ until the stopping criteria is met.

Step 10: The optimal solution is the final $x_{gbest}^{q}(t)$, which represents the gap position matrix at maximum alignment score and minimum objective function (Equation (18)).

5 EXPERIMENTAL SETUP

5.1 Benchmark Data Set

The performance of TL-PSO has been tested on two types of data sets:

1. Sequence sets tested in Zablocki FBR [20];
2. Sequence sets from BRAlibase database [21].

5.1.1 Benchmark Data Set 1

This group of dataset consists of eight sequence sets S1 to S8 of DNA and RNA sequences. Here, S8 is an artificially generated sequence set, whereas S1-S7 are real sequence datasets with varying complexities. The details of these datasets are listed in Table 1.

ID	Type	n	$l_{avg}(l_{max}, l_{min})$	Sequence Similarity (%)
S1	DNA	10	212 (211, 212)	92.2
S2	DNA	5	1 780 (1 775, 1 782)	63.0
S3	DNA	21	122 (122, 122)	95.6
S4	DNA	8	1 437 (1 356, 1 485)	44.4
S5	DNA	8	1 680 (1 680, 1 680)	95.4
S6	mRNA	6	1 456 (1 430, 1 463)	77.7
S7	rRNA	8	457 (457, 457)	99.3
S8	DNA	5	8 (7, 10)	52.5

Table 1. The eight benchmark dataset

5.1.2 Benchmark Data Set 2

The RNA sequences with APSI score more than 75 % are selected from k5 and k7 dataset of BRAlibase database as listed in Table 2.

Group	Family	APSI (%)	l_{max}	Seq	Role
S	HIV_GSL3	82	83	k5	Directs specific packaging of HIV-1 genomic RNA
		83	83	k7	
M	Retroviral_psi	89	118	k5	An element identified in the genomes of the retroviruses HIV
		90	117	k7	
L	IRES_Picorna	83	251	k5	Used in dicistronic or multicistronic vectors in gene therapy, virus replicon systems and analysis of IRES function [22]
		83	252	k7	

Table 2. The six benchmark dataset from BRAlibase dataset

RNA sequences with average pairwise sequence identity (APSI) score higher than 55 % can be aligned by sequence alignment programs, but for the APSI score lesser than 55 %, sequence structure alignment is recommended [21]. For APSI score greater than 75 %, gap penalty parameters have lesser impact. Hence the RNA sequences from BRAlibase dataset with APSI score more than 75 % are selected. Here S stands for small, i.e. sequence length < 100 nucleotides, M stands for medium i.e., sequence length between 100 and 200 nucleotides, L stands for long i.e., sequence length more than 200 nucleotides. The accession numbers are mentioned in Appendix A.

5.2 Experiment Settings

5.2.1 Parameter Setting of TL-PSO

As discussed earlier, exponentially decreasing weight strategy provides the best results at time and convergence criteria than other six inertia weight schemes, i.e. linear (decreasing and increasing), non-linear (decreasing and increasing), increasing exponential and constant weight schemes [7]. The stopping criteria of the algorithm is: either the maximum number of iteration has been reached; or the output has not improved till 50 iterations.

We used the following parameters for the TL-PSO in all experiments performed in MATLAB programming environment:

- Number of particles $m = 30$
- Number of iterations $(T) = 2\,000$
- Number of simulation run for each sequence set = 30
- Neighbourhood topology = *gbest*
- Cognitive coefficient $c_1 = 1.496180$
- Social coefficient $c_2 = 1.496180$
- Inertia Weight (w) = By exponentially decreasing weight strategy as formulated below:

$$w = \theta - \frac{(\theta - \phi)}{\{-\exp(t/T)\}} \tag{34}$$

where $\theta = 0.9$, $\phi = 0.4$, t = iteration number and T is the total number of iterations with the condition $0.4 \leq w \leq 0.9$.

5.2.2 Other Parameter Settings

Score schemes depicted by Equations (7) and (9) are applied to find the scores for dataset 1. The gap penalty scheme from Equation (16) is adopted. Present work follows the same scoring scheme with the same gap penalty for this dataset (as that of [20]). The score for match is 2, for mismatch is 0 and for gap is -1. The gap

opening penalty is -2 and gap extension penalty is -1. The maximum number of allowed gaps κ is determined by:

$$\kappa = \rho * (0.3 * \xi) \tag{35}$$

where ρ is determined by:

$$\rho = \frac{(100 - \chi)}{100} \tag{36}$$

where χ is the percent similarity of that specific sequence set being aligned and ξ is the length of longest sequence. This value of κ is applied in Equation (6) so as to obtain the maximum allowed sequence length.

Because of the availability of reference alignment, the scoring scheme of Equations (10) and (13) are applied for benchmark dataset 2, with all the alignment parameters binary by default.

6 SIMULATION RESULTS

6.1 Results for Dataset 1

Tables 3 and 4 summarize the simulation results for dataset 1. The TL-PSO results are compared with family of PSO algorithms, i.e. S-PSO (standard PSO), M-PSO (mutating PSO), CPSO-S_k (cooperative split PSO) and our previously developed variant MPSO with decreasing exponential weight. We write MPSO as ED-MPSO (exponentially decreasing modified PSO) so as to avoid the confusion with other compared variant M-PSO. ED-MPSO was tested by us on S8 dataset containing 5 sequences with maximum sequence length 10 [7]. The gapped sequence length was taken as a random number. MPSO alignment results were tested against standard PSO and ClustalW. All the parameters presented in Tables 3 and 4, i.e. number of matches, number of full column matches and number of gaps, are counted for the alignment producing minimum value of objective function from Equation (18). Table 3 presents the results of applying SS method depicted by Equation (7). SS for sequence set S1 to S7 could not be compared because the score scheme is different than the mentioned one (since S8 was easily simulated), hence the other three parameters, i.e. total number of matches, number of gaps and number of full column matches are compared to avail the comparison. For S8 sequence set, a few results which are not available are marked as NA in Tables 3–6.

As evident from Tables 3 and 4, the performance of TL-PSO for sequence set S1, S3, S5, S7 and S8 remained same for all the approaches compared. The reason is, quite apparently, that since the sequence sets S1, S3, S5 and S7 have more than 90 % similarity score, hence with reference to [21] any common alignment program can easily obtain optimal alignments for these sequence sets. For sequence set S8, the small length of the sequences and small number of sequences makes all the PSO variants perform well for optimal alignment.

Table 3 shows that for the SS method TL-PSO performed well for S4 and S6 comparative to the other PSO based approaches but performance for S2 is a bit lesser than one other PSO based strategy, i.e. CPSO-S_k. The results with MS method from Table 4 show that TL-PSO is able to perform better if applied with match method. The score of TL-PSO is always greater than or equal to the score of other PSO variants compared whereas the performance of ED-MPSO started getting weak as soon as the length of sequences increases and sequence similarity decreases.

It can be observed from Table 5 that performance-wise TL-PSO always exists at first rank except for sequence set S2 with score scheme SS. TL-PSO is unanimously on the first rank with score scheme MS for all the sequence sets as shown by Table 6. Here TL-PSO is a better performer for sequence set S2 as well. The reason behind this is the lesser complexity of MS method as compared to SS method.

Figure 2 presents the comparison between score schemes SS and MS depicted by Equations (7) and (9), respectively, for alignment quality and time taken at the same PSO parameters, same number of swarms and iterations for the artificially generated sequence set S8. In this figure NM stands for number of matches, NG for number of gaps and NCM for number of column matches. The time taken by MS is very less than SS, whereas the score quality remains same as evident from NM and NC. The reason is evident that SS aligns the pairs of two sequences at a time and calculates the alignment score for all ${}^nC_2$ pairs and then adds them, whereas match score method takes all the n sequences at a time to calculate the alignment score. The time complexity of similarity score method is $O(\ln^2)$, whereas for match score method it is $O(\ln)$ for n sequences with sequence length l.

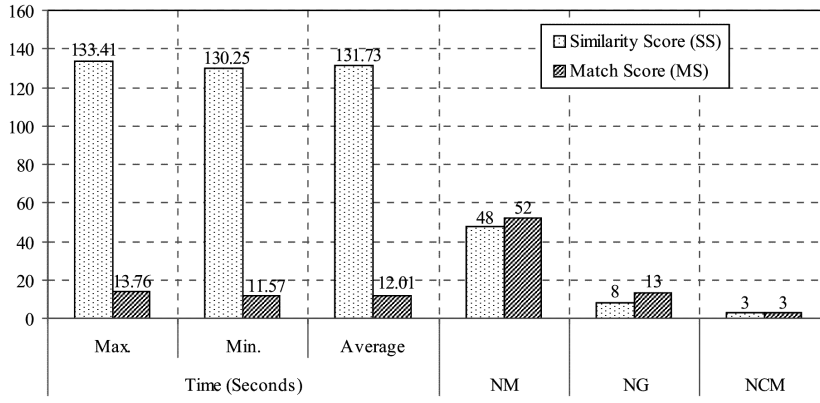


Figure 2. Time and accuracy comparison between both the score schemes for S8

ID	MSA Approach	Total No. of Matches	No. of Gaps	No. of Full Column Matches
S1	TL-PSO	9 392	1	198
	S-PSO	9 392	1	198
	M-PSO	9 392	1	198
	ED-MPSO	9 392	1	198
	CPSO-S_k	9 392	1	198
S2	TL-PSO	15 308	38	1 188
	S-PSO	13 443	13	782
	M-PSO	14 452	28	1 073
	ED-MPSO	14 618	43	1 055
	CPSO-S_k	15 380	68	1 251
S3	TL-PSO	24 503	0	109
	S-PSO	24 503	0	109
	M-PSO	24 503	0	109
	ED-MPSO	24 503	0	109
	CPSO-S_k	24 503	0	109
S4	TL-PSO	28 985	941	519
	S-PSO	17 821	381	242
	M-PSO	16 027	381	137
	ED-MPSO	20 476	965	131
	CPSO-S_k	21 344	461	302
S5	TL-PSO	44 910	0	1 440
	S-PSO	44 910	0	1 440
	M-PSO	44 910	0	1 440
	ED-MPSO	44 910	0	1 440
	CPSO-S_k	44 910	0	1 440
S6	TL-PSO	17 993	124	904
	S-PSO	17 920	43	887
	M-PSO	17 699	43	866
	ED-MPSO	17 783	139	755
	CPSO-S_k	17 934	109	843
S7	TL-PSO	12 713	0	449
	S-PSO	12 713	0	449
	M-PSO	12 713	0	449
	ED-MPSO	12 713	0	449
	CPSO-S_k	12 713	0	449
S8	TL-PSO	48	8	3
	S-PSO	48	8	3
	M-PSO	NA	NA	NA
	ED-MPSO	48	8	3
	CPSO-S_k	NA	NA	NA

Table 3. Performance comparison of TL-PSO with S-PSO, M-PSO, ED-MPSO, CPSO-S_k for optimizing similarity score for dataset 1

ID	MSA Approach	Max. Score	Total No. of Matches	No. of Gaps	No. of Full Matches	Average Match Score
S1	TL-PSO	51 157.2	9 392	1	198	51 157.2
	S-PSO	51 157.2	9 392	1	198	51 157.2
	M-PSO	51 157.2	9 392	1	198	51 157.2
	ED-MPSO	51 157.2	9 392	1	198	51 157.2
	CPSO-S_k	51 157.2	9 392	1	198	51 157.2
S2	TL-PSO	49 957.6	16 975	38	1 592	35 126.8
	S-PSO	36 254.0	13 463	108	929	28 083.5
	M-PSO	40 802	14 640	38	1 120	31 307.7
	ED-MPSO	41 475.4	14 875	53	1 137	30 014.8
	CPSO-S_k	38 842.4	14 296	368	1 086	32 326.9
S3	TL-PSO	263 570.1	24 503	0	109	263 570.1
	S-PSO	263 570.1	24 503	0	109	263 466.1
	M-PSO	263 570.1	24 503	0	109	263 466.1
	ED-MPSO	263 570.1	24 503	0	109	263 570.1
	CPSO-S_k	263 570.1	24 503	0	109	263 466.1
S4	TL-PSO	113 661.5	29 750	941	548	76 123.5
	S-PSO	49 217.0	17 180	613	83	40 677.7
	M-PSO	57 188.3	18 848	453	92	39 234.9
	ED-MPSO	61 384.4	20 271	973	103	39 076.3
	CPSO-S_k	58 503.4	19 021	981	206	43 390.5
S5	TL-PSO	197 274.0	44 910	0	1 440	197 274.0
	S-PSO	187 769.4	43 481	64	1 307	175 396.6
	M-PSO	197 274.0	44 910	0	1 440	197 274.0
	ED-MPSO	197 274.0	44 910	0	1 440	197 274.0
	CPSO-S_k	195 467.4	44 663	88	1 420	191 293.2
S6	TL-PSO	61 776.5	18 805	127	1 006	50 062.7
	S-PSO	49 759.2	16 131	151	691	29 457.9
	M-PSO	55 641.2	17 409	37	853	41 596.0
	ED-MPSO	54 752.5	17 389	133	766	32 658.1
	CPSO-S_k	56 975.3	17 938	277	845	49 386.9
S7	TL-PSO	57 038.6	12 713	0	449	57 038.6
	S-PSO	57 038.6	12 713	0	449	57 038.6
	M-PSO	57 038.6	12 713	0	449	57 038.6
	ED-MPSO	57 038.6	12 713	0	449	57 038.6
	CPSO-S_k	57 038.6	12 713	0	449	57 038.6
S8	TL-PSO	108	52	13	3	106.2
	S-PSO	108	52	13	3	104.5
	M-PSO	NA	NA	NA	NA	NA
	ED-MPSO	108	52	13	3	105.7
	CPSO-S_k	NA	NA	NA	NA	NA

Table 4. Performance comparison of TL-PSO with S-PSO, M-PSO, ED-MPSO, CPSO-S_k for optimizing match score for dataset 1

PSO Variant	Rank for Sequence Set							
	S1	S2	S3	S4	S5	S6	S7	S8
TL-PSO	1	2	1	1	1	1	1	1
S-PSO	1	5	1	4	1	3	1	1
M-PSO	1	4	1	5	1	5	1	NA
ED-MPSO	1	3	1	3	1	4	1	1
CPSO-S_k	1	1	1	2	1	2	1	NA

Table 5. Rank wise performance comparison between PSO variants with SS method

PSO Variant	Rank for Sequence Set							
	S1	S2	S3	S4	S5	S6	S7	S8
TL-PSO	1	1	1	1	1	1	1	1
S-PSO	1	5	1	5	3	5	1	1
M-PSO	1	3	1	4	1	3	1	NA
ED-MPSO	1	2	1	2	1	4	1	1
CPSO-S_k	1	4	1	3	2	2	1	NA

Table 6. Rank wise performance comparison between PSO variants with MS method

6.2 Results for Dataset 2

Table 7 shows the simulation results of applying SoP score depicted by Equation (10) and CM score presented by Equation (13), respectively. The scores are compared with the optimal alignments of ClustalW and T-Coffee. It can be observed from Table 7 that TL-PSO is equally efficient and somewhere better performer than ClustalW and T-Coffee. Match method was applied for alignment, due to its better performance for dataset 1. The number of gaps in benchmark sequences is 5, 7, 21, 38, 16 and 30, respectively, in the order of their appearance in the table. For k5 short and k7 short sequences it is equally good as the other two MSA strategies (ClustalW and T-Coffee), for both SoP score and CM score. For SoP score the performance of TL-PSO is found to be better than T-Coffee and ClustalW for sequence sets k5 medium, k7 medium, k5 long and k7 long. For CM score, TL-PSO scored equally good as did T-Coffee for k7 medium and better than ClustalW. For k5 medium, k5 long and k7 long it scored better than both the MSA strategies. Hence we found TL-PSO to be the better performer for medium and long sequences, i.e. k5 medium, k7 medium, k5 long and k7 long, whereas TL-PSO is found to be equally efficient for short sequences, i.e. k5 short and k7 short.

Tables 8 and 9 conclude the performance-wise ranking of TL-PSO along with ClustalW and T-Coffee. It can be observed that for k5 short all the alignment strategies exist at first rank for both SoP score and CM score schemes. For k5 medium, TL-PSO exists at first position followed by T-Coffee and ClustalW respectively for both SoP score and CM score. For k7 medium, T-Coffee is a better performer than ClustalW. TL-PSO is on the first rank with T-Coffee for CM scores, whereas T-Coffee is on second rank for SoP scores. For k5 long, TL-PSO is again on

S.No.	MSA Approach	No. of Gaps	Best SoP Score	Best CM Score
k5 short	TL-PSO	0	0.9600	0.9176
	ClustalW	0	0.9600	0.9176
	T-Coffee	0	0.9600	0.9176
k7 short	TL-PSO	0	0.9681	0.9176
	ClustalW	0	0.9681	0.9176
	T-Coffee	0	0.9681	0.9176
k5 medium	TL-PSO	21	0.9402	0.9180
	ClustalW	31	0.9115	0.8852
	T-Coffee	21	0.9098	0.8771
k7 medium	TL-PSO	38	0.9241	0.9024
	ClustalW	31	0.9051	0.8861
	T-Coffee	31	0.9210	0.9024
k5 long	TL-PSO	16	0.9545	0.9333
	ClustalW	11	0.9396	0.9137
	T-Coffee	21	0.9502	0.9137
k7 long	TL-PSO	30	0.9524	0.918
	ClustalW	30	0.9224	0.8789
	T-Coffee	23	0.9381	0.8906

Table 7. Comparison between MSA approaches for SoP score and CM score on dataset 2

MSA Approaches	Rank for Sequence Set					
	k5 short	k7 short	k5 medium	k7 medium	k5 long	k7 long
TL-PSO	1	1	1	1	1	1
ClustalW	1	1	2	3	3	3
T-Coffee	1	1	3	2	2	2

Table 8. Rank wise performance comparison of TL-PSO with ClustalW and T-Coffee for SoP score

MSA Approaches	Rank for Sequence Set					
	k5 short	k7 short	k5 medium	k7 medium	k5 long	k7 long
TL-PSO	1	1	1	1	1	1
ClustalW	1	1	2	2	2	3
T-Coffee	1	1	3	1	2	2

Table 9. Rank wise performance comparison of TL-PSO with ClustalW and T-Coffee for SoP score

the first position for both scores while for CM score T-Coffee and ClustalW share the same position showing equal efficiency. For k5 and k7 long, TL-PSO exists at the first rank dominating the scores of T-Coffee and ClustalW at the second and third positions, respectively. Hence TL-PSO is an efficient performer even when the sequence size increases.

7 CONCLUSIONS

The objective for the proposed approach TL-PSO is to maximize alignment score while addressing two important alignment issues of multiple sequence alignment, i.e. optimal sequence length and optimal gap positions. Both the issues are addressed at two different levels of the algorithm by employing separate PSO for each. For better convergence speed exponentially decreasing inertia weight strategy is applied. The testing is performed on two kinds of datasets containing sequence sets of different complexities.

For dataset 1, TL-PSO is compared with four other PSO variants, i.e. S-PSO, M-PSO, ED-MPSO and CPSO-S_k by evaluating alignment scores at two popular scoring schemes, i.e. similarity score and match score. For dataset 2, performance of TL-PSO is compared with popular alignment softwares: ClustalW and T-Coffee. For dataset 2, two popular scoring schemes are adapted, i.e. column match and sum of pair.

TL-PSO is found competitive and generally a better strategy for MSA than compared PSO variants. For dataset 1, match score method is able to perform faster than similarity score method. TL-PSO is found more potent than compared PSO variants. For dataset 2, TL-PSO is found to be outperforming ClustalW and T-Coffee at both the scoring methods.

Further improvements in TL-PSO can be in direction of decrement in time complexity of scoring scheme for similarity score. TL-PSO can be implemented to multi-objective optimization by considering gap minimization and score maximization as two objectives as shown by Equation (17). Sequence structure alignment can be a very effective area for obtaining optimal alignment of RNA sequences. Sequence structure alignment is again a multi-objective problem and TL-PSO can be tested for such kind of problems as well.

Acknowledgements

The authors wish to thank the Executive Director, Birla Institute of Scientific Research for the support given during this work. We are thankful to Dr. Krishna Mohan for his valuable suggestions throughout the work. We gratefully acknowledge financial support by BTIS-sub DIC (supported by DBT, Government of India) to one of us (S.L.) and Advanced Bioinformatics Centre (supported by Govt. of Rajasthan) at Birla Institute of Scientific Research for infrastructure facilities for carrying out this work.

Appendix A

Family	Seq.	Accession No.
HIV_GSL3	k5	AY178916; AF418322; AY093605; L20571; AF391259
	k7	AF418330; AB081221; AF386047; AY339787; AF377957; U56898; L20571
Retroviral_psi	k5	AF450098; AF408626; AF075701; AF443107; AF538302
	k7	AJ237565; AF193275; AF110974; AY161886; AF110978; AB078005; AF286236
IRES_Picorna	k5	L76393; AY055130; AJ430385; L76401; AB081362
	k7	AF363455; AJ293918; AF541919; X89533; AJ295198; AB059821; AY055171

REFERENCES

[1] Thompson, J. D.—Higgins, D. G.—Gibson, T. J.: CLUSTAL W: Improving the Sensitivity of Progressive Multiple Sequence Alignment Through Sequence Weighting, Position-Specific Gap Penalties and Weight Matrix Choice. Nucleic Acids Research, Vol. 22, 1994, No. 22, pp. 4673–4680.

[2] Notredame, C.—Higgins, D. G.—Heringa, J.: T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment. Journal of Molecular Biology, Vol. 302, 2000, No. 1, pp. 205–217.

[3] Subramanian, A. R.—Menkhoff, J. W.—Kaufmann, M.—Morgenstern, B.: DIALIGN-T: An Improved Algorithm for Segment-Based Multiple Sequence Alignment. BMC Bioinformatics, Vol. 6, 2005, Art. No. 66.

[4] Durbin, R.—Eddy, S.—Krogh, A.—Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, United Kingdom. 1998.

[5] Blum, C.—Li, X.: Swarm Intelligence in Optimization. In: Blum, Ch., Merkle, D. (Eds.): Swarm Intelligence, Springer-Verlag Berlin Heidelberg, 2008, pp. 43–85.

[6] Lalwani, S.—Kumar, R.—Gupta, N.: A Review on Particle Swarm Optimization Variants and Their Applications to Multiple Sequence Alignment. Journal of Applied Mathematics and Bioinformatics, Vol. 3, 2013, No. 2, pp. 87–124.

[7] Lalwani, S.—Kumar, R.—Gupta, N.: A Study on Inertia Weight Schemes with Modified Particle Swarm Optimization Algorithm for Multiple Sequence Alignment. Proceedings of 6th IEEE International Conference on Contemporary Computing, Noida, India, 2013, pp. 283–288.

[8] Long, H. X.—Xu, W. B.—Sun, J.—Ji, W. J.: Multiple Sequence Alignment Based on a Binary Particle Swarm Optimization Algorithm. Proceedings of Fifth International Conference on Natural Computation, Tianjin, Vol. 3, 2009, pp. 265–269.

[9] Lee, Z. J.—Su, S. F.—Chuang, C. C.—Liu, K. H.: Genetic Algorithm with Ant Colony Optimization (GA-ACO) for Multiple Sequence Alignment. Applied Soft Computing, Vol. 8, 2008, No. 1, pp. 55–78.

[10] Lei, X. J.—Sun, J. J.—Ma, Q. Z.: Multiple Sequence Alignment Based on Chaotic PSO. In: Cai, Z., Li, Z., Kang Z., Liu, Y. (Eds.): Computational Intelligence and Intelligent Systems, Springer-Verlag Berlin Heidelberg, Vol. 51, 2009, pp. 351–360.

[11] Gondro, C.—Kinghorn, B. P.: A Simple Genetic Algorithm for Multiple Sequence Alignment. Genetics and Molecular Research, Vol. 6, 2007, No. 4, pp. 964–982.

[12] Thomsen, R.—Fogel, G. B.—Krink, T.: A Clustal Alignment Improver Using Evolutionary Algorithms. Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, HI, Vol. 1, 2002, pp. 121–126.

[13] Lai, C. C.—Wu, C. H.—Ho, C. C.: Using Genetic Algorithm to Solve Multiple Sequence Alignment Problem. International Journal of Software Engineering and Knowledge Engineering, Vol. 19, 2009, No. 6, pp. 871–888.

[14] Xu, F.—Chen, Y.: A Method for Multiple Sequence Alignment Based on Particle Swarm Optimization. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (Eds.): 5th International Conference on Intelligent Computing (ICIC 2009), Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence. Springer-Verlag Berlin Heidelberg, Lecture Notes in Computer Science, Vol. 5755, 2009, pp. 965–973.

[15] Kennedy, J.—Eberhart , R. C.: Particle Swarm Optimization. Proceedings IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.

[16] Wang, D.—Liu, L.: Improved Particle Swarm Optimization and Its Application. Proceedings of the 3rd International Conference on Innovative Computing Information and Control, Dalian, Liaoning, 2008.

[17] Setubal, J. C.—Meidanis, J.: Introduction to Computational Biology. Brooks/Cole, Pacific Grove, California, 1997.

[18] Chellapilla, K.—Fogel, G. B.: Multiple Sequence Alignment Using Evolutionary Programming. Proceedings of the 1999 Congress on Evolutionary Computation, Washington DC, Vol. 1, 1999, pp. 445–452.

[19] Thompson, J. D.—Plewniak, F.—Poch, O.: A Comprehensive Comparison of Multiple Sequence Alignment Programs. Nucleic Acids Research, Vol. 27, 1999, No. 13, pp. 2682–2690.

[20] Zablocki, F. B. R.: Multiple Sequence Alignment Using Particle Swarm Optimization. Master's Dissertation, University of Pretoria, 2007.

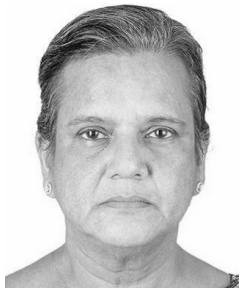
[21] Wilm, A.—Mainz, I.—Steger, G.: An Enhanced RNA Alignment Benchmark for Sequence Alignment Programs. Algorithms for Molecular Biology, Vol. 1, 2006, Art. No. 19.

[22] Junemann, C.—Song, Y.—Bassili, G.—Goergen, D.—Henke, J.—Niepmann, M.: Picornavirus Internal Ribosome Entry Site Elements Can Stimulate

Translation of Upstream Genes. The Journal of Biological Chemistry, Vol. 282, 2007, No. 1, pp. 132–141.

Soniya Lalwani is Research Scholar in the Department of Mathematics at Malaviya National Institute of Technology (MNIT), India. Besides this, she is working at R & D, Bioinformatics Centre at Birla Institute of Scientific Research (BISR), Jaipur, India from 2007. She has been a university rank holder during her academics and Gold Medalist in M.Phil. (Applied Mathematics) from MDS University, India. Her area of research includes particle swarm optimization, sequence analysis, multi-objective optimization, sequence-structure alignment and hidden Markov model. She is Life Member of ORSI, RGP, ISCA, IMS, IACSIT and IAENG-SB.

Rajesh Kumar is Associate Professor in the Department of Electrical Engineering at the Malaviya National Institute of Technology (MNIT), India. He was Post Doctorate Research Fellow in the Department of Electrical and Computer Engineering at the National University of Singapore (NUS), Singapore, from 2009 to 2011. He has been active in the research and development of intelligent systems and applications more than ten years, and is internationally known for his work in this area. He has published over two hundred articles on the theory and practice of intelligent control, evolutionary algorithms, bio and nature inspired algorithms, fuzzy and neural methodologies, power electronics, electrical machines and drives, applications of AI to image processing and bioinformatics. He received the Career Award for Young Teachers in 2002 from Government of India. He has been a member of editorial board and reviewer of many journals. He also served on many program committees and conducted conferences and workshops in the area. He is a Senior Member IEEE, Member IE (India), Fellow Member IETE, Senior Member IEANG and Life Member ISTE.



Nilama Gupta is Associate Professor in the Department of Mathematics at Malaviya National Institute of Technology (MNIT), India. She received her M.Sc. in 1982, followed by Ph.D. in 1985 from Washington State University, Washington in the area of non-linear optimization. She has authored many books and research papers on linear, non-linear optimization and linear fractional programing.