Modified Off-lattice AB Model for Protein Folding Problem Using the Vortex Search Algorithm

Berat Doğan and Tamer Ölmez

***Abstract*—The energy function of the off-lattice AB model has a number of deep valleys and hills which usually leads the search algorithms to trap into a local minimum point. Existing studies usually performs algorithmic improvements on the well-known search methods to avoid from these local minimum points. However, these algorithmic improvements further increase the computational time which is not desired for the protein folding problem. In this study, it is aimed to smooth the energy landscape of this energy function and thus, to find a near optimal or optimal configuration without performing algorithmic improvements on the search methods. This is achieved by adding an additional term to the original energy function by which a hydrophobic core is formed and a near optimal and optimal configuration is found easily. In the experiments, a newly proposed optimization algorithm, the Vortex Search (VS) algorithm, is used to minimize both the original and modified energy functions. Experimental results showed that, the modified energy function helps the VS algorithm to find the desired configurations much more easier than the original function when the maximum number of iterations is kept equal for both cases.**

***Index Terms*—Off-lattice AB model, protein folding, vortex search algorithm.**

I. Introduction

The protein folding problem is one of the most widely studied optimization problem which is known to be NP-complete. Once the proteins are synthesized, they fold in a unique three dimensional structure that makes them functional or biologically active. This physical process is known as the protein folding process. The mechanism behind the protein folding process is still unknown. However, there are some mathematical models proposed to simulate the protein folding process and to find the correct fold of an amino-acid sequence [1].

Existing mathematical models for the protein folding problem (or process) can be categorized into two different groups. The first group includes the all atom models in which all atomic details of a protein along with the physical interactions such as bond angle, torsion angle, van-der Waals forces, electrostatic interactions, charge transfer etc. are considered. These models are usually computationally expensive and utilize molecular dynamic (MD) simulations. The second group includes the simplified coarse-grained methods which are emerging as a practical alternative to all-atom models. In the coarse-grained methods, each amino-acid of a chain is represented in a binary form. Perhaps the most widely studied model is the so called HP model [2], in which each amino-acid in a protein chain is considered either hydrophobic or polar. In the HP model, high resolution lattice models are used to accurately model the protein structure and to retain the computational efficiency of lattice models as well [3]. In lattice models, each amino-acid is mapped to a particular lattice point to form a continuous and self-avoiding amino-acid chain with fix bond lengths between successive amino-acid pairs. The lattice models benefits greatly from the discretization of protein phase space; however, it also suffers from this strategy. The discrete nature of the model surely affects the folding behaviors, especially the dynamics of the system [3]. To overcome this problem off-lattice model (or toy model) was proposed [4]. In the off-lattice model each amino-acid in a protein chain is considered either A (hydrophobic) or B (polar or hydrophilic) as in HP model. In this model, again the amino-acids are linked up with a fixed bond length, but different from the HP model the backbone can continuously bend between any pair of successive links. Additionally, in this model nonconsecutive amino-acids interact through a modified Leonard-Jones potential and there is an energy contribution from each bond angle between successive bonds. Therefore, when compared to the HP model, the off-lattice AB model is much more realistic.

Although, it is a more realistic model of the protein folding problem, even the simplified off-lattice AB model is far to be solved in polynomial time (it is NP complete). In literature, a number of studies have been proposed to solve the protein folding problem [5]-[9] by using off-lattice AB model. These studies, mainly utilizes well-known optimization algorithms or their extensions. When the proposed studies are compared to each other, it can be shown that, the improvements from one study over another mainly arise in terms of the fitness value reached by each method. The computational efficiency or the convergence behaviors of the used methods are usually not compared.

In this study, it is aimed to increase the convergence speed of the algorithms to a near optimal or an optimal protein fold by modifying the off-lattice AB model energy function. In their initial study, Stillinger et. al. pointed out that, the given energy function of the off-lattice AB model makes no mention of solvent [4]. They also stated that, one could implicitly modify the energy function to include a solvent effect. Thus, with a small modification on the energy function of the off-lattice AB model, it is shown that, the energy surface of this function can be smoothed and thus, the convergence speed of the algorithms can be significantly improved. In our simulations, a newly proposed metaheuristic, the Vortex Search (VS) algorithm [10], is used. However, any other optimization algorithm could also be

Manuscript received December 26, 2014; revised February 26, 2015.

The authors are with the Istanbul Technical University, Istanbul, SARIYER 34469, Turkey (e-mail: bdogan@ itu.edu.tr, olmezt@itu.edu.tr).

used in the experiments.

The remaining part of this paper is organized as follows. In Section-2, the off-lattice AB model is mentioned and then the modified energy function for the off-lattice AB model is introduced. Section-3 covers the recently proposed Vortex Search algorithm. In Section-4, experimental results and discussions are given. Finally, Section-5 concludes the work.

II. The Off-lattice AB Model

A. The Original Energy Function

The off-lattice AB model was proposed by Stillinger et. al. [4]. In this model, amino-acids are linked by rigid unit-length bonds to form linear unoriented polymers that reside in two dimensions. A configuration of n-mer sequence is defined by n angles $\theta_2,...,\theta_{n-1}$,where $-\pi \le \theta_i \le \pi$. A sample configuration is shown in Fig.1. It is obvious that $\theta_i = 0$ corresponds to linearity of successive bonds, and positive angles indicate counterclockwise rotation.

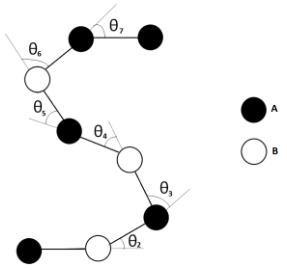

Fig. 1. A sample configuration of off-lattice AB model in two dimensional space.

The free energy function Φ of a configuration is defined as in Eq. (1).

$$\Phi = \sum_{i=2}^{n-1} \frac{1-\cos\theta_i}{4} + 4\sum_{i=1}^{n-2}\sum_{j=i+2}^{n}\left[r_{ij}^{-12} - C\left(\xi_i,\xi_j\right)r_{ij}^{-6}\right] \quad (1)$$

The first component of the energy function models the backbone bend potentials and it is independent of the A,B sequence, while the second one models the non-bonded interactions and it depends on the A,B sequence and receives a contribution from the each amino-acid pairs that are not directly attached by a backbone bond. Thus, $C(\xi_i,\xi_j) = 1$ for AA pairs, $C(\xi_i,\xi_j) = -0.5$ for AB or BA pairs, and $C(\xi_i,\xi_j) = 0.5$ for BB pairs. r_{ij} represents the distance between the amino-acid i and the amino-acid j and can be computed by using Eq. (2).

$$r_{ij} = sqrt\left(\left[1 + \sum_{k=i+1}^{j-1}\cos\left(\sum_{l=i+1}^{k}\theta_l\right)\right]^2 + \left[\sum_{k=i+1}^{j-1}\sin\left(\sum_{l=i+1}^{k}\theta_l\right)\right]^2\right) \quad (2)$$

Stillinger *et al.* pointed out that, the above given energy function of the off-lattice AB model makes no mention of solvent [4]. They also stated that, one could implicitly modify the energy function to include a solvent effect. In the following subsection, the proposed modified energy function that includes the solvent effect is introduced.

B. The Modified Energy Function

Protein-solvent interactions and the influence of the solvent on protein's thermodynamic properties is in fact very complex and a simulation with a real solvent medium is therefore computationally expensive. Thus, in this study instead of a real solvent medium, the solvent effect is implicitly modeled by an hydrophobic core. It is accepted that the first-order driving force of the protein folding is due to a "hydrophobic collapse" in which those amino-acids which prefer to be shielded from water are driven to the core of the protein, while those which interact more favorably with water remain on the outside of the protein [11]. Hence, the energy function of the off-lattice AB model is modified to force the amino-acid chain to form a hydrophobic core which in turn favors hydrophobic-hydrophobic interactions. To achieve this, a simple modification is performed on the original energy function. The modified energy function includes an additional term h which is a function of the distance for non-consecutive AA pair of amino-acids, while for other pairs it is independent of the distance (Eq. (3)).

$$h = \sum_{i=1}^{n-2}\sum_{j=i+2}^{n} P_{ij} \quad (3)$$

In Eq. (3), $P_{ij} = \frac{C\left(\xi_i,\xi_j\right)}{r}$ for AA pairs, and $P_{ij} = C\left(\xi_i,\xi_j\right)$ for other pairs, n represents the total number of amino-acids for a given amino-acid chain. In this manner, AA pairs are forced to form a hydrophobic core. Because the neighbor BB pairs are also welcomed, they are also included in the additional term but they are independent of the distance to prevent a possible formation of a hydrophilic core. Both the AB and BA pairs are also selected to be independent of distance. Otherwise, they form a high energy barrier during search process which prevents the algorithm to visit other configurations. The additional term h helps the algorithm to form a hydrophobic core during the search process. However, using this term alone causes to algorithm to form configurations that have high AA pair interactions by suppressing the effect of the bend potentials. The desired case is to have configurations in which both the bend potential effects and the hydrophobic effects are balanced. Thus, the Eq. (3) is compensated by an additional term which balances these two forces and forms more realistic configurations (Eq. (4)).

$$f(h,\Phi) = h \cdot \frac{n_A - \Phi}{n_A} \quad (4)$$

In Eq. (4), n_A represents the total number of A type amino-acids for a given amino-acid chain, Φ is the original energy function. Because the energy value of the original function can never exceed the $-n_A$ value, a normalization can be performed by using the total number of A type amino-acids. This normalization process balances the effect of the terms which is required to find desired configurations.

Then the modified off-lattice AB model energy function is given as in Eq. (6).

$$\Phi_m = \Phi - f(h,\Phi) \tag{5}$$

The modified energy function is thought to have a more smoothed energy surface when compared to the original function which has many local minimum points with deep valleys and hills. Thus, it is much easier for algorithms to converge the optimum or a near optimal point during the search process. This assumption is verified with the experimental results given in Section IV.

III. The Vortex Search Algorithm

The Vortex Search (VS) algorithm is a recently proposed metaheuristic approach which is shown to be an effective method to perform numerical function optimization [10]. In order to achieve a good balance between the exploration and the exploitation, the search behavior of the VS algorithm is modeled as a vortex pattern.

Let us consider a two-dimensional optimization problem. In a two dimensional space, a vortex pattern can be modeled by a number of nested circles. Here, the outer (largest) circle of the vortex is first centered on the search space, where the initial center μ_0 can be calculated using Eq. (6).

$$\mu_0 = \frac{upLim + lowLim}{2} \tag{6}$$

where *upLim* and *lowLim* are $d \times 1$ vectors that define the bound constraints of the problem in *d* dimensional space. Then a number of neighbor solutions $C_t(s)$ (*t* represents the iteration index and initially $t = 0$) are randomly generated around the initial center μ_0 in the *d*-dimensional space by using a Gaussian distribution. Here, $C_0(s) = \{s_1, s_2, ..., s_k\}$, $k = 1, 2, ..., n$ represents the solutions, and *n* represents the total number of candidate solutions. In Eq. 7, the general form of the multivariate Gaussian distribution is given.

$$p(x \mid \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d \Sigma}} \exp\left\{\frac{-1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\} \tag{7}$$

where *d* represents the dimension, *x* is the $d \times 1$ vector of a random variable, μ is the $d \times 1$ vector of the sample mean (center), and Σ is the covariance matrix. If the diagonal elements (variances) of the values of Σ are equal and if the off-diagonal elements (covariance) are zero (uncorrelated), then the resulting shape of the distribution will be spherical (which can be considered circular for a two-dimensional problem, as in our case). Thus, the value of Σ can be computed using equal variances with zero covariance, as in Eq. (8).

$$\Sigma = \sigma^2 \times [I]_{d \times d} \tag{8}$$

where σ^2 represents the variance of the distribution and *I* represents the $d \times d$ identity matrix. The initial standard deviation σ_0 of the distribution can be calculated by using Eq. (9).

$$\sigma_0 = \frac{\max(upLim) - \min(lowLim)}{2} \tag{9}$$

Here, σ_0 can also be considered as the initial radius (r_0) of the outer circle for a two-dimensional optimization problem. Because a weak locality is required in the initial phases, r_0 is chosen to be a large value. Then in the selection phase, a solution (which is the best one) $s' \in C_0(s)$ is selected and memorized from $C_0(s)$ to replace the current circle center μ_0. Prior to the selection phase, the candidate solutions must be ensured to be inside the search boundaries, as in Eq. (10).

$$\begin{aligned} &\text{if } s_k^i < lowLim^i,\ s_k^i = rand \cdot (upLim^i - lowLim^i) + lowLim^i \\ &\text{if } s_k^i > lowLim^i,\ s_k^i = rand \cdot (upLim^i - lowLim^i) + lowLim^i \end{aligned} \tag{10}$$

where $k = 1, 2, ..., n$, $i = 1, 2, ..., d$ and *rand* is a uniformly distributed random number. Next, the memorized best solution s' is assigned to be the center of the second circle (the inner one). In the generation phase of the second step, the effective radius (r_1) of this new circle is reduced, and then, a new set of solutions $C_1(s)$ is generated around the new center. In the selection phase of the second step, the new set of solutions $C_1(s)$ is evaluated to select a solution $s' \in C_1(s)$. If the selected solution is better than the best solution found so far, then this solution is assigned to be the new best solution and it is memorized. Next, the center of the third circle is assigned to be the memorized best solution found so far. This process iterates until the termination condition is met. A description of the VS algorithm is also provided in Fig. 2.

Data: Initial center μ_0 calculated using Eq. 6
Initial radius r_0 (or the standard deviation, σ_0) is computed using Eq. 9
Fitness of the best solution found so far $f(s_{best}) = inf$
Iteration index $t = 0$;
while $t \leq MaximumNumberofIterations$ **do**
  /* Generate candidate solutions by using Gaussian distribution around the center μ_t with a standard deviation (radius) r_t */
  $Generate\,(C_t(s))$
  if *exceeded* **then**
    Shift the $C_t(s)$ values into the boundaries as in Eq. 10
  end
  if $f(s') < f(s_{best})$ **then**
    $s_{best} = s'$
    $f(s_{best}) = f(s')$
  else
    keep the best solution found so far s_{best}
  end
  /* Center is always shifted to the best solution found so far */
  $\mu_{t+1} = s_{best}$
  /* Decrease the standard deviation (radius) for the next iteration */
  $r_{t+1} = Decrease(r_t)$
  $t = t + 1$
end

Fig. 2. A description of the vortex search algorithm.

The radius decrement process given in Fig. 2 can be considered as a type of adaptive step-size adjustment process, which is also used in RS (Random Search) algorithms [12]. This process should be performed in such a way that allows the algorithm to behave in an explorative manner in the initial steps and in an exploitative manner in the latter steps. In the VS algorithm, the inverse incomplete gamma function is used to decrease the value of the radius during each iteration pass [10]. The incomplete gamma function used in the VS

algorithm is not very successful on the protein folding problem because of the shape of the energy landscape. The energy landscape of the protein folding problem has a funnel like shape which requires a sharp decrease in the initial steps and a less slope in the latter steps. Thus, in this study a piece-wise linear function is used to achieve this type of behavior. In Eq. (11) required equations are given to form the piece-wise liner function that is used during the radius decrement process. a_t represents the function value at each iteration pass and a_0 is selected as $a_0 = 1$ to ensure full coverage of the search space at the first iteration, t is the iteration index, and *MaxItr* represents the maximum number of iterations. Fig. 3 shows the change of the a value with respect to the iteration number.

$$a_t = \begin{cases} a_{t-1} - \dfrac{9}{MaxItr}, & if\ t < 0.1 \times MaxItr \\ a_{t-1} - \dfrac{1}{20 \times MaxItr}, & if\ t > 0.8 \times MaxItr \\ a_{t-1} - \dfrac{9}{70 \times MaxItr}, & otherwise \end{cases} \quad (11)$$

The initial radius r_0 can be calculated using Eq. (12). Because $a_0 = 1$, $r_0 = \sigma_0$ as indicated before.

$$r_t = \sigma_0 \cdot a_t \quad (12)$$

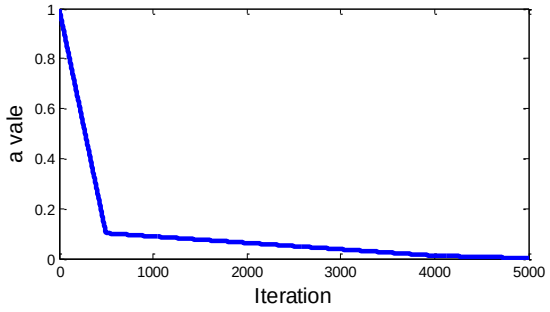

Fig. 3. Change of the piece-wise linear function (a value) for 5000 iterations.

IV. Experimental Results and Discussion

In Table I, computational results of 50 different trials are listed for 6 different benchmark amino-acid chains. The maximum number of iterations is selected as 5000 for the VS algorithm. Note that, the values listed in the column named "VS with modified energy function" are corresponding original energy function values of the found configurations by using the modified energy function with VS algorithm. As it can be shown from this table, for short amino-acid chains, original energy function performs better than the modified one in terms of the reached fitness value. However, note that, as stated in the first section, aim of this study is not to find the known ground state energy configurations of these benchmark sequences, but the aim is to find a near optimal solution as quickly as possible. Although, there is a small difference in the reached best energy values, the resulting configurations are quite similar. For short sequences, it is trivial to find a near optimal or optimal configuration. But when it comes to longer chains, the original energy function fails to find the near optimal configurations and performs worst than the modified energy function. As stated before, by using the original energy function, usually algorithmic improvements are required to find the optimal configuration. Even these improvements, do not guarantee the optimal configuration and they are usually computationally expensive.

Because of the complexity of the energy landscape created by the original energy function, even very similar configurations can have very distinct energy values. These means the original energy function have a number of deep valleys and hills around the ground state configuration. Thus, a more smoothed energy landscape is required for an algorithm to find the optimal or near optimal configuration easily. The small modification performed on the original energy function smoothes the energy landscape and thus allows the algorithms to find a near optimal or optimal configuration as quickly as possible. In Fig. 4, for the amino-acid chain BABABBABABBABBABABBAB, the best energy value achieved at each iteration pass is shown for the modified energy function and it is compared to the corresponding original energy value. As it can be shown from the Fig. 4, the modified energy function has a smooth energy landscape. But in contrast, the corresponding original energy value highly varies towards the near optimal configuration found by the algorithm.

In Fig. 5 optimal and near optimal configurations found by the proposed method are given. These configurations are quite similar to the known ground state configurations [5]-[9].

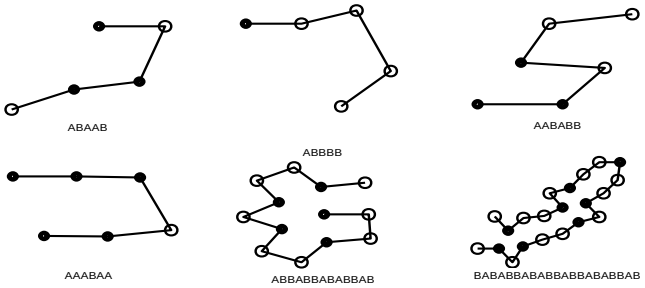

Fig. 5. Best configurations found by modified energy function and VS algorithm.

TABLE I: Computational Results of the Proposed Method

Sequence	n	VS with original energy function	VS with modified energy function
ABAAB	5	Mean: -1.37647 Std: 0 Best: -1.37647	Mean: -1.3631 Std: 2.41e-4 Best: -1.3637
ABBBB	5	Mean: -0.06596 Std: 0 Best: -0.06596	Mean: -0.06596 Std: 0 Best: -0.06596
AABABB	6	Mean: -1.3497 Std: 0.0155 Best: -1.36198	Mean: -1.3288 Std: 0.0128 Best: -1.3406
AAABAA	6	Mean: -3.5939 Std: 0.1234 Best: -3.6975	Mean: -3.5433 Std: 0.1332 Best: -3.6757
ABBABBABABBAB	13	Mean: -1.7590 Std: 0.3996 Best: -2.42	Mean: -2.4335 Std: 0.5966 Best: -3.2522
BABABBABABBABB ABABBAB	21	Mean: -3.2622 Std: 0.7744 Best: -5.3137	Mean: -3.4190 Std: 0.7965 Best: -5.8249

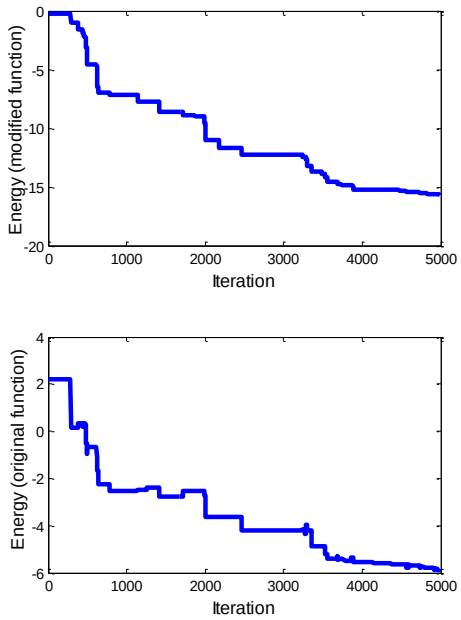

Fig. 4. Iteration number vs. energy value plot of amino-acid chain BABABBABABBABBABABBAB with modified energy function and its corresponding original energy function plot.

V. Conclusion

The off-lattice AB model for the protein folding problem is one of the most widely studied models within computer science community. This model has a very complex energy function which makes difficult the problem to be solved within a reasonable time. Existing studies usually performs algorithmic improvements on the well-known search methods to find a near optimal or optimal configuration. However, these attempts further complicates the problem because an improvement on a search algorithm usually requires additional computational time which is not desired for the protein folding problem. Different from the existing studies, in this study, it is aimed to smooth the energy landscape of the off-lattice AB model energy function. By this way, even simple algorithms could find a near optimal or optimal configuration quickly without trapping into a local minimum point. For this purpose, a simple modification is performed on the original energy function by an additional term which helps the algorithm to form a hydrophobic core during the search process. This modification must be perform in such a way that will provide a balance between the original function and the additional term. Experiments showed that, the modified energy functions performs well on the chains up to 21 amino-acids.

In the future studies, a more efficient energy function will be searched to find the near optimal or optimal configurations for longer amino-acid chains.

References

[1] B. Doğan and T. Ölmez, "A novel state space representation for the solution of 2D-HP protein folding problem using reinforcement learning methods," *Applied Soft Computing*, vol. 26, pp. 213-223, January 2015

[2] K. A Dill, "Theory for the folding and stability of globular proteins," *Biochemistry*, vol. 24, no. 6, pp. 1501-1509, 1985

[3] J. Zhang *et al.*, "Protein folding simulations: From coarse-grained model to all-atom model," *IUBMB Life*, vol. 61, no. 6, pp. 627-643, 2009

[4] F. H. Stillinger, T. H. Gordon, and C. L. Hirshfeld, "Toy model for protein folding," *Physical Review*, vol. 48, no. 2, p. 1469, 1993

[5] L. Bai, Y. Li, and L. Gong, "Protein secondary structure optimization using an improved artificial bee colony algorithm based on AB off-lattice model," *Engineering Applications of Artificial Intelligence* vol. 27, pp. 70-79, 2014

[6] L. Jingfa *et al.*, "Heuristic-based tabu search algorithm for folding two-dimensional AB off-lattice model proteins," *Computational Biology and Chemistry*, vol. 47, pp. 142-148, 2013

[7] R. S. Parpinelli and S. L. Heitor, "An ecology-based evolutionary algorithm applied to the 2D-AB off-lattice protein structure prediction problem," in *Proc. the 2013 Brazilian Conference on Intelligent Systems (BRACIS), IEEE*, 2013.

[8] D. H. Kalegari and S. L. Heitor, "An improved parallel differential evolution approach for protein structure prediction using both 2D and 3D off-lattice models," in *Proc. the 2013 IEEE Symposium on Differential Evolution (SDE), IEEE*, 2013.

[9] L. Jingfa *et al.*, "Structure optimization of the two-dimensional off-lattice hydrophobic–hydrophilic model," *Journal of Biological Physics*, vol. 35, no. 3, pp. 245-253, 2009

[10] B. Doğan and T. Ölmez, "A new metaheuristic for numerical function optimization: Vortex search algorithm," *Information Sciences*, vol. 293, pp. 125-145, Feb. 2015

[11] S. E. Decatur, "Protein folding in the generalized hydrophobic-polar model on the triangular lattice," *Technical M emo M It-Lc S*, 1996

[12] L. A. Rastrigin, "Convergence of random search method in extremal control of many-parameter system," *Automation and Remote Control*, vol. 24, no. 11, p. 1337, 1964

Berat Doğan was born in Malatya, Turkey in 1985. He received his B.Sc. degree in electronics engineering from Erciyes University, Kayseri, Turkey in 2006, and M.Sc. degree in biomedical engineering from Istanbul Technical University, Istanbul, Turkey in 2009. He is currently studying his PhD in electronics engineering at Istanbul Technical University. Istanbul, Turkey. He worked as a software engineer at Nortel Networks Netaş, Turkey, between the 2008 and 2009. Since 2009, he is working as a research assistant at Istanbul Technical University, Istanbul, Turkey. His research interests include, pattern recognition, signal and image processing, nature inspired optimization and bioinformatics. He has a number of journal papers and conference proceedings.

Tamer Ölmez received his B.Sc. degree in electrical and electronics engineering in 1985, M.Sc. degree in computer engineering in 1988, and PhD degree in electrical and electronics engineering in 1995, from Istanbul Technical University, Istanbul, Turkey. He worked as a research engineer at Teletaş, Turkey, between the 1985 and 1988. Until the end of 1989 he worked at the scientific and technical research council of Turkey as a research engineer. Since then he has been with the Department of Electrical and Electronics Engineering at Istanbul Technical University, Turkey, where at present he is a professor. His research interests include, pattern recognition, machine learning, biomedical signal processing, image processing, computer vision, neural networks, genetic algorithms, real-time signal processing applications based on microprocessors.