Expert Systems with Applications 38 (2011) 13170–13180

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization

Bilal Alatas

Department of Computer Engineering, Tunceli University, TR62000 Tunceli, Turkey

ARTICLE INFO

Keywords:
Heuristic optimization
Computational method
Chemical reaction
Performance

ABSTRACT

Heuristic based computational algorithms are densely being used in many different fields due to their advantages. When investigated carefully, chemical reactions possess efficient objects, states, process, and events that can be designed as a computational method en bloc. In this study, a novel computational method, which is more robust and have less parameters than that of used in the literature, is intended to be developed inspiring from types and occurring of chemical reactions. The proposed method is named as Artificial Chemical Reaction Optimization Algorithm, ACROA. Applications to multiple-sequence alignment, data mining, and benchmark functions have been performed so as to put forward the performance of developed computational method.

1. Introduction

Heuristics algorithms are the algorithms that utilize a simple approach as a solution technique of search and optimization problems and are recently getting strong and becoming more popular (Baykasoglu, Ozbakir, & Tapkan, 2007). The reasons can be summarized as follows:

- They provide general solution strategies that can be applied to the problem in case of concurrent different decision variables, objective functions and constraints.
- They do not depend on the solution space type, the number of decision variables, and the number of constraints.
- They do not require very well defined mathematical models that are hard to organize for system modeling and objective function and, that cannot be used due to high solution time cost even the mathematical model has been organized.
- Their computation power is good and they do not require excessive computation time.
- Their transformations and adaptations are easy.
- They give efficacious solutions to the high-scale combinatorial and non-linear problems.
- They do not require the assumptions that are hard to be approved to adapt a solution algorithm to a given problem as done in classical algorithm.
- They do not require the alteration on the interested problem as done in the classical algorithms. They adapt themselves in order to solve different types of problems.

Due to these advantages, heuristic algorithms are densely being used in many different fields such as management science, engineering, computer, etc. and new versions of these algorithms have been proposed (Baykasoglu et al., 2007).

Many of the proposed versions are population based techniques that is, they begin to search the solution with multiple points. Many of them are inspired from nature. Some of them are physical based (multi point simulated annealing algorithm (Lamberti & Pappalettere, 2007), electromagnetism-like algorithm (Birbil & Fang, 2003), particle collision algorithm (Sacco & Cassiano, 2005), big bang–big crunch algorithm (Erol & Eksin, 2006), some of them are biology based (evolutionary algorithm (Alatas & Akın, 2006; Goldberg, 1989), ant colony algorithm (Alatas & Akın, 2005; Dorigo, Maziezzo, & Colorni, 1996), bee colony algorithm (Alatas, 2010; Karaboga & Basturk, 2007), artificial immune algorithms (Alatas & Akın, 2005; De Castro & Von Zuben, 2002), firefly algorithm (Yang, 2009), enzyme algorithm (Yang, 2008), saplings growing up algorithm (Karci, 2007; Karci & Alatas, 2006; Karci, Alatas, & Akin, 2006), invasive weed optimization (Mehrabian & Lucas, 2006), monkey search algorithm (Mucherino & Seref, 2007), bacterial foraging algorithm (Passino, 2002), some of the are social based (multi point tabu search algorithm (Niizuma, Yasuda, & Ishigame, 2006), imperialist competitive algorithm (Gargari & Lucas, 2007), some of them are social and biology based (particle swarm optimization (Alatas & Akin, 2008, 2009a, 2009b; Alatas, Akin, & Ozer, 2009; Kennedy & Eberhart, 1995), cat swarm optimization (Chu, Tsai, & Pan, 2006), some of them are biology and geography based (biogeography-based algorithm (Simon, 2008), and some of them are musical based (harmony search algorithm (Alatas, 2010; Lee & Geem, 2005). Some researchers have proposed their hybrid versions by combining two or more

E-mail addresses: bilalalatas@yahoo.com, balatas@tunceli.edu.tr

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.126

algorithms. These algorithms are efficient and flexible and they can be modified and/or adapted to suit specific problem requirements. Researches on these algorithms are still continuing all around the globe. Fig. 1 shows the classifications of the heuristic algorithms.

A molecule is composed of several atoms and characterized by the atom types, bond lengths, angles, and torsions. One molecule is separated from another when they contain different atoms and/or different number of atoms. In any chemical reaction, the initial reactants change to the products by the single-double formation, single-double destruction, and shifting of chemical bonds. The reactants normally change to a series of intermediate reactants in the transition states during elementary steps.

Enthalpy is the total energy associated with a chemical system which includes its internal energy and also energy due to environmental factors such as pressure-volume conditions. It is also called heat content of a chemical system (Wiese, Hendriks, & Deschênes, 2006). It is a state function and its value depends upon amount of the substance, chemical nature of the substance and conditions of temperature and pressure. Entropy is a measure of the randomness or disorder of components of a chemical system. It is also a state function. In general, entropy increases when gases are formed from liquids and solids, liquids or solutions are formed from solids, the number of gas molecules increases, the number of moles increases, and the temperature increases. Larger and more complex molecules have higher entropies.

According to the second law of thermodynamics, the entropy of the universe does not change for reversible processes and increases for spontaneous processes. A system tends toward the highest entropy and the lowest enthalpy. When investigated carefully, chemical reactions possess efficient objects, states, process, and events that can be designed as a computational method en bloc. Enthalpy or potential energy for minimization problem and entropy for maximization problem can be utilized as objective functions for the interested problem.

There are three systems according to matter and energy exchange with its surroundings. An open system can exchange both matter and energy with the surroundings. A closed system has a fixed amount of matter, but it can exchange energy with the surroundings. An isolated system has no contact with its surroundings (Nag, 2008). Appropriate system can be selected according to the interested optimization problem and modifications may be added to the proposed Artificial Chemical Reaction Optimization Algorithm (ACROA). Molecules are encoded in ACROA using appropriate encoding scheme for the optimization problems.

There is not a chemistry based computational method, which was developed inspiring from the occurring of the chemical reactions. There is only a technical report considering the collision of particles and omitting the conservation of energy as an idea (Lam & Li, 2009). However, the proposed computational method in this study is completely different from this technical report in terms of reason and principle. In this study, a novel computational method, which is more robust and have less parameters than that of used in the literature, is intended to be developed inspiring from types and occurring of chemical reactions. Its applications to multiple-sequence alignment, data mining, and benchmark functions have been performed so as to put forward the performance of developed method.

The organization of this paper is as follows. In the Section 2, the chemical reactions and their classification are described. In the Section 3, the novel computational method and its computational operators inspired by the chemical reactions are described. The applications of this novel computational method in various fields and experimental results are presented in Section 4. Finally, Section 5 concludes this paper.

2. Chemical reactions

A chemical reaction is a process that leads to the transformation of one set of chemical substances to another. Chemical reactions take place in widely varying rates. Rates of various reactions vary from very slow to very fast. Multiple steps in chemical reactions are a very common occurrence. Two types of reactions are consecutive and competing reactions. Classically, chemical reactions encompass changes that strictly involve the motion of electrons in the forming and breaking of chemical bonds. The substance or substances initially involved in a chemical reaction

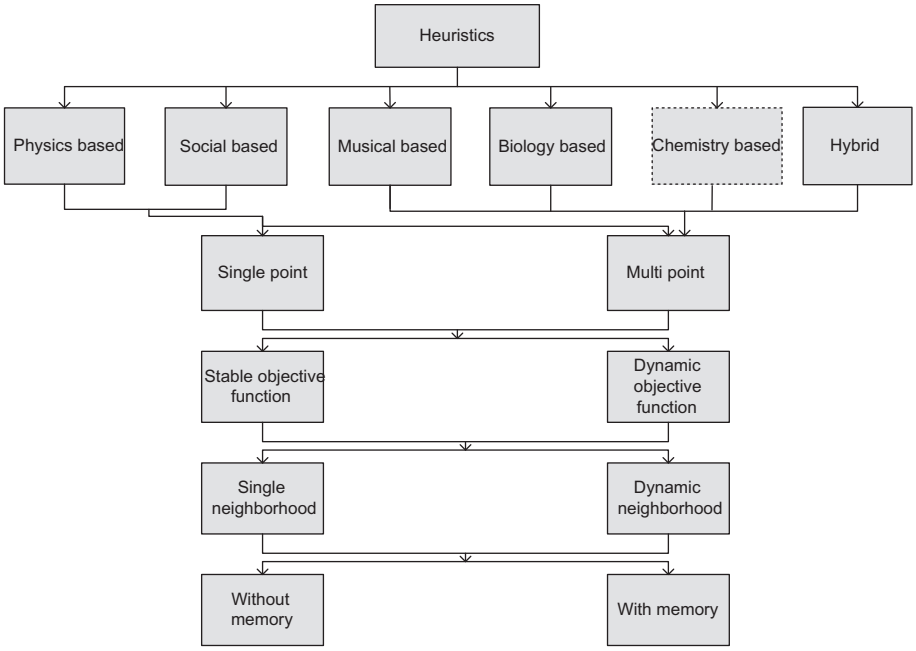

Fig. 1. Heuristic algorithms.

are called reactants. Chemical reactions are usually characterized by a chemical change. One or more products, which usually have properties different from the reactants, are produced. *N* different types of molecules or chemical species are involved in chemical processes. These molecules may take part in one or more of *M* types of chemical reactions. Reactions which are not strongly exergonic (energy liberating) or strongly endergonic (energy requiring) typically are spontaneous both forwards and backwards under typical physiological conditions. Many reactions are reversible, i.e. they can occur in either direction. At chemical equilibrium point the Rate Forward Reaction = Rate Reverse Reaction. The observable properties and concentrations of all participants (species) become constant when a chemical system reaches a state of equilibrium. The output of one reaction may be a reactant of other reactions. The schematic representations of chemical reactions may be summarized as in Fig. 2.

Chemical reactions can be classified into the following categories:

2.1. Synthesis reactions

A synthesis reaction is when two or more reactants combine to produce a single product or compound. In generic terms, synthesis reactions look like the following:

$$2H_2 + O_2 \rightarrow 2H_2O \text{ (Elements)},$$

$$6CO_2 + 6H_2O \rightarrow C_6H_{12}O_6 + 6O_2 \text{ (Compounds)}.$$

2.2. Decomposition reactions

A decomposition reaction is the opposite of a synthesis reaction. A single compound breaks down into two or more elements or new compounds. A decomposition reaction often requires an energy source, such as heat, light, or electricity, to occur. In generic terms, decomposition reactions look like the following:

$$2H_2O \rightarrow 2H_2 + O_2,$$

$$2KClO_3 \rightarrow 2KCl + 3O_2.$$

2.3. Single displacement reactions

In this reaction, one element trades places with another element in a compound. In generic terms, single replacement reactions look like the following:

$$Cl + 2KBr \rightarrow 2KCl + Br_2.$$

2.4. Double displacement reactions

This is when the anions and cations of two different molecules switch places, forming two entirely different compounds. In generic terms, double replacement reactions look like the following:

$$NaCO_3(aq) + BaCl_2(aq) \rightarrow 2NaCl(aq) + \downarrow BaCO_3(k).$$

2.5. Combustion reactions

A combustion reaction is when oxygen combines with a substance and releases energy in the form of heat and light. An example of this kind of reaction involves methane gas and oxygen. Methane is major component of natural gas.

$$CH_4 + O_2 \rightarrow CO_2 + H_2O.$$

2.6. Redox reactions

Redox reactions are the transfer (release and uptake) of electrons from one reactant to another. The chemical which gains electrons is reduced (reduces its valency) and is called the oxidising agent. The chemical which loses electrons, is oxidised (increases its valency) and is called the reducing agent.

$$Fe + Cu^{+2} \rightarrow Fe^{+2} + Cu.$$

Fe donates two electrons to the Cu^{2+} to form Cu (metal). The Fe lost 2 electrons, so is oxidised. The Cu^{2+} gained 2 electrons, so is reduced (in its valency).

2.7. Reversible reactions

In reversible reactions, products of certain reactions can be converted back to the reactants. Thus, in reversible reactions the products can react with one another under suitable conditions to give back the reactants. In other words, in reversible reactions the reaction takes place in both the forward and backward directions. Reaction can also be called as elastic.

$$CaCO_3 \leftrightarrow CaO + CO_2.$$

3. Artificial Chemical Reaction Optimization Algorithm (ACROA)

Atoms and molecules move and collide in a continuous manner in a viscous fluid filling a 2D cellular space. Atoms are elementary particles possessing a type, mass, radius, charge, orientation, position, and velocity. A molecule is a set of atoms connected by bonds. Chemical reactions are mappings of discrete cellular configurations

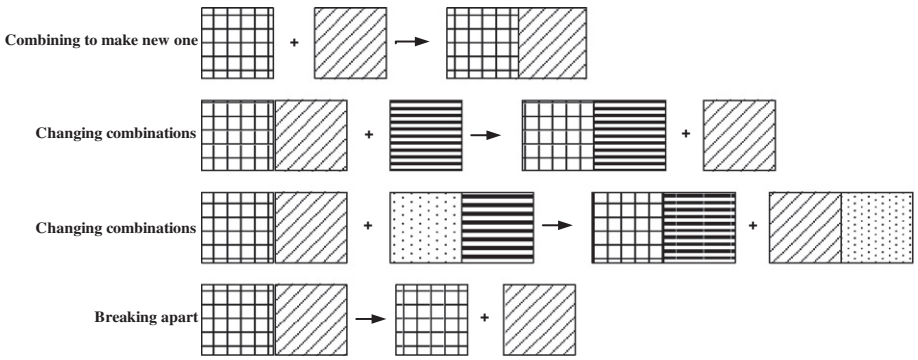

Fig. 2. Schematic representations of types of chemical reactions.

to parameterized actions on atoms. Actions allow atom creation and destruction, bonding and un-bonding to make and break molecules, orientation, type change, and propulsion. Time proceeds in discrete steps. Briefly reactions result from the interaction of atoms (Portegts, 2004).

Algorithm can be considered as a simulation of reactants in a vessel. Suppose a fixed volume vessel containing a spatially uniform mixture of N chemical reactants (species) interacting through specific chemical reaction channels. Let R_i $(1 \leqslant i \leqslant N)$ be the list of chemical species, and suppose these species can interact through M specified chemical reaction channels. Encoding of the reactants for ACROA depends on the interested problem. It can be binary, real, string, etc. These encoding schemes plays role in the formation of reaction rules. Reaction rules define the interaction among one or two reactants which may lead to production of a new reactant. ACROA begins with set of initial reactants in a solution. Then reactants are consumed and produced via chemical reactions. Algorithm is terminated when the termination criterion is met similar to the state when no more reactions can take place (inert solution).

Reactants are selected for reactions probabilistically based on their concentrations and potentials. Furthermore, two main types of reactions are called as consecutive and competing reactions and are shown in Fig. 3. In consecutive type, chemical reactions joined together serially such as $A + B + C \leftrightarrow AB + C \leftrightarrow ABC \leftrightarrow A + BC$. In competing type, different products may occur according to the special conditions. The output of one reaction may be a reactant of other reactions. In fact, many factors effect the reactions however. In the ACROA, very simple idea is utilized by an equal probability for monomolecular or bimolecular reactions and their variants.

According to the above algorithm concept, the ACROA flow chart of which is depicted in Fig. 4 and pseudo-code of which is shown in Fig. 5 consists of the following five steps:

Step 1: Problem and algorithm parameter initialization.
Step 2: Setting the initial reactants and evaluation.
Step 3: Applying chemical reactions.
Step 4: Reactants update.
Step 5: Termination criterion check.

3.1. Problem and algorithm parameter initialization

The optimization problem is specified as follows:

$$\text{Minimize } f(x) \text{ subject to } x_j \in X_j = 1, 2, \ldots, N,$$

where $f(x)$ is an objective function; x is the set of each decision variable x_j; N is the number of decision variables, X_j is the set of the

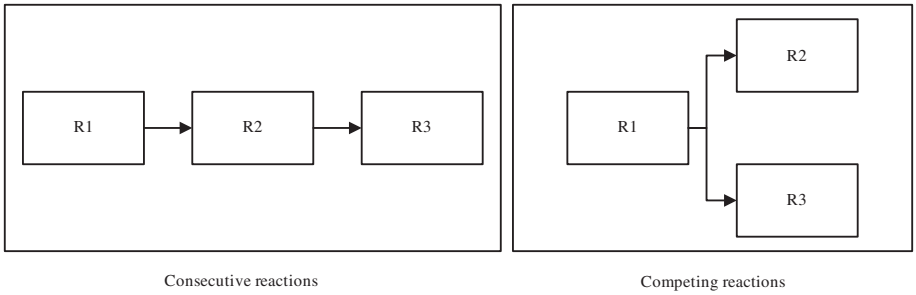

Fig. 3. Consecutive and competing reactions.

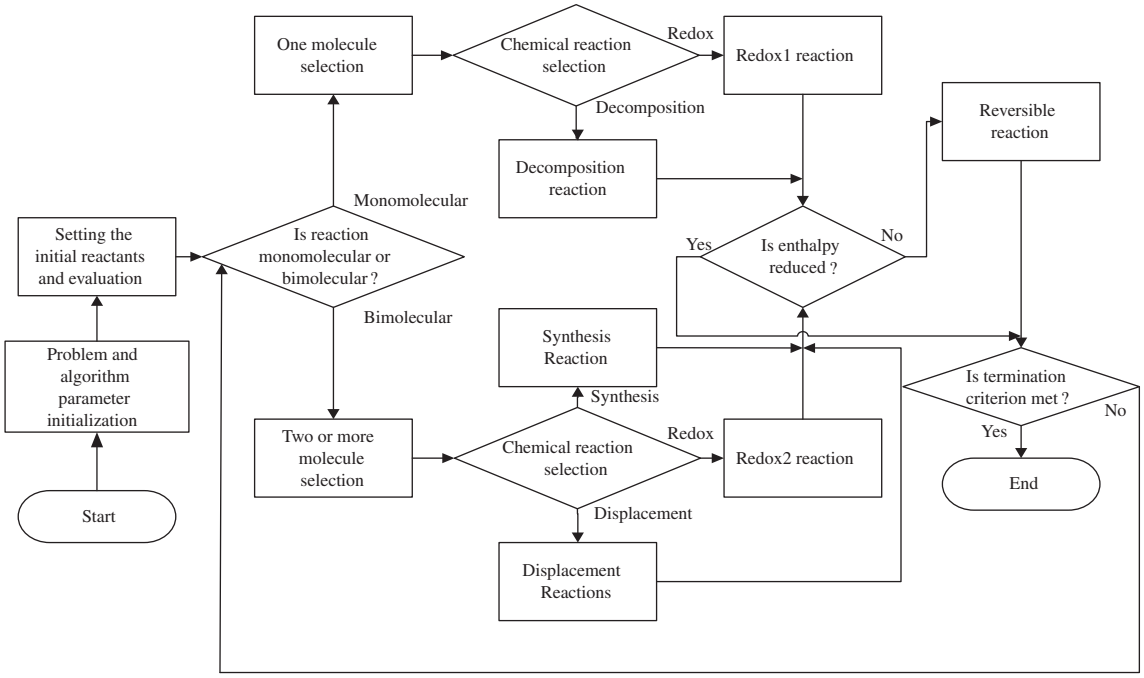

Fig. 4. Flow chart of ACROA.

```
Input: Problem-specific information (such as objective function, number of decision
variables) and the sole algorithm-specific information, number of reactants (ReacNum)

Create molecules m_i of size ReacNum by uniform population
for i←1 to ReacNum do
   Calculate the enthalpy e(m_i)
end for
while termination criterion not met do
   for i←1 to ReacNum do
      Get r_1 randomly in interval [0,1]
      if r_1 ≤ 0.5 then
         Get r_2 randomly in interval [0,1]
         if r_2 ≤ 0.5 then
            Decomposition(m_i)
         end if
         else
            Redox1(m_i)
         end if
     else
         Select another molecule m_j (m_i ≠ m_j)
         Get r_3 randomly in interval [0,1]
              if 0≤ r_3 ≤0.33 then
                 Synthesis(m_i, m_j)
              end if
              if 0.33< r_3 ≤0.66 then
                 Displacement (m_i, m_j)
              end if
              else
                 Redox2(m_i, m_j)
              end if
     end if
     Apply reversible reaction for increased enthalpy
   end for
end while
Output the minimum solution and its objective function value
```

Fig. 5. Pseudo-code of ACROA.

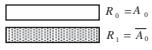

Fig. 6. Initial reactants.

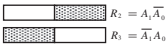

Fig. 7. New reactants with $k = 2$.

possible range of values for each decision variable, that is $x_j^{\min}$ and $x_j^{\max}$ are the lower and upper bound of the jth decision parameter respectively for real-values encoding. The problem may require different types of encoding such as binary, real, permutation, etc. The sole ACROA parameter, *ReacNum*, is also specified in this step.

3.2. Setting the initial reactants and evaluation

In this step initial reactants are evenly initialized in the feasible searching space. Uniform population method (Karci, 2007; Karci & Alatas, 2006; Karci et al., 2006; Karci & Arslan, 2002) proposed for initial population generating can be used for creating initial reactants. In general, all vectors in a space can be obtained in a linear combination of elements of base set. If one of elements in the base set is absent, then the dimension corresponding to this element may be vanished. That is why, it is important that initial reactants must contain reactants which must hold each element of base set.

By considering regularity case and base set, the initial reactants must be regular and also hold base set. The proposed method in this paper satisfies both cases. Generating initial reactants based on divide-and-generate paradigm is a method to generate reactants of good quality. This method works as follows.

Initially, two reactants R_0, R_1 are set where $R_0 = \{u_1, u_2, \ldots, u_n\}$, $R_1 = \{l_1, l_2, \ldots, l_n\}$, n is the length of reactant and this case is considered as $k = 1$ (Fig. 6). Then a dividing factor, k, is determined. Firstly, $k = 2$ and two extra R_2, R_3 reactants are derived from R_0 and R_1 (Fig. 7)

$$R_2 = \{r^*u_1, r^*u_2, \ldots, r^*u_{n/2}, r^*l_{n/2+1}, r^*l_{n/2+2}, \ldots, r^*l_n\} \quad \text{and}$$

$$R_3 = \{r^*l_1, r^*l_2, \ldots, r^*l_{n/2}, r^*u_{n/2+1}, r^*u_{n/2+2}, \ldots, r^*u_n\}$$

where r is a random number such as $0 \leqslant r < 1$.

Let us consider the population P size as $|P|$ and the number of elements in the set of generated reactants R as $|R|$. So, if $|R| < |P|$, then the value of k is increased by 1, and $2^3 - 2 = 8 - 2 = 6$ reactants can be derived from R_0 and R_1, which are not in R, since R_0 and R_1 are divided into three parts. These reactants can be listed as below (Fig. 8)

$$R_4 = \{r^*u_1, r^*u_2, \ldots, r^*u_{2n/3}, r^*l_{2n/3+1}, r^*l_{2n/3+2}, \ldots, r^*l_n\}$$

$$R_5 = \{r^*u_1, r^*u_2, \ldots, r^*u_{n/3}, r^*l_{n/3+1}, \ldots, r^*l_{2n/3}, r^*u_{2n/3+1}, \ldots, r^*u_n\},$$

$$R_6 = \{r^*u_1, r^*u_2, \ldots, r^*u_{n/3}, r^*l_{n/3+1}, \ldots, r^*l_n\},$$

$$R_7 = \{r^*l_1, r^*l_2, \ldots, r^*l_{n/3}, r^*u_{n/3+1}, \ldots, r^*u_n\},$$

$$R_8 = \{r^*l_1, r^*l_2, \ldots, r^*l_{n/3}, r^*u_{n/3+1}, \ldots, r^*u_{2n/3}, r^*l_{2n/3+1}, \ldots, r^*l_n\},$$

$$R_9 = \{r^*l_1, r^*l_2, \ldots, r^*l_{2n/3}, r^*u_{2n/3+1}, \ldots, r^*u_n\},$$

where r is a random number such as $0 \leqslant r \leqslant 1$.The reactant $R_4, \ldots, R_9$ are added to R and if $|R|$ is still less than $|P|$, then the value of k is increased by 1. In this case, the reactants R_0 and R_1 are divided into four parts, and the derivation similar to derivation for $k = 3$ is applied and 14 reactants are derived and added to R. If $|R|$ is still less than $|P|$, then the value of k is increased by 1 and

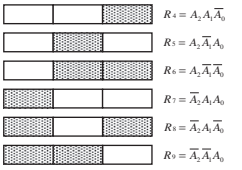

Fig. 8. New reactants with $k = 3$.

derivation is again applied. This process goes on until $|R| \geqslant |P|$ (Gundogan, Alatas, & Karci, 2004; Karci, 2007; Karci & Alatas, 2006; Karci & Arslan, 2002; Karci et al., 2006). Hereafter, the first $|P|$ elements of the set R are taken as reactants.

Fig. 9 depicts the steps of generating the initial reactants for ACROA.

3.3. Applying chemical reactions

3.3.1. Bimolecular reactions

Let $R_1 = (r_1^1, \ldots, r_n^1)$ and $R_2 = (r_1^2, \ldots, r_n^2)$ be two reactants that will act a part in a bimolecular reaction. Below, types of bimolecular reaction operations used in ACROA are described. Reaction operation for string encoding is similar to the binary encoding. For ordered encoding, crossover and mutation types used in genetic algorithms may be easily used for ACROA.

3.3.2. Synthesis reaction

For binary encoding:

Non-matching bits of two reactants are determined. Then, one bit from the non-matching bit of the first reactant and one bit from the non-matching bit of the second reactant are consecutively selected to form a new reactant. Representation of this operation is depicted in Fig. 10.

For real encoding:

A new reactant is obtained as

$R = (r_1, \ldots, r_i, \ldots, r_n)$where

$$r_1 = r_i + \lambda_i(r_i^2 - r_i^1), \tag{1}$$

where λ_i is a randomly chosen value in the interval $[-0.25, 1.25]$. This is similar to extended line crossover operator proposed in Muhlenbein and SchlierkampVoosen (1993).

3.3.3. Displacement reaction

For binary encoding:

Each bit position of the two reactants strings are considered for information swapping based on a randomly generated mask similar to the mask used in uniform crossover used in genetic algorithms (Sywerda, 1989). If the mask value for the corresponding bit location in the string is 1, then the bits of the reactants are not swapped. However, if it is 0, the bits at the corresponding locations are exchanged as depicted in Fig. 11.

For real encoding:

Two new reactants are obtained as

$$R_k = (r_1^k, \ldots, r_i^k \ldots, r_n^k), k = 1, 2,$$

where

$$r_i^1 = \lambda_{td} r_i^1 + \left(1 - \lambda_{td} r_i^2\right), \tag{2}$$

$$r_i^2 = \lambda_{td} r_i^2 + \left(1 - \lambda_{td} r_i^1\right), \tag{3}$$

where $\lambda_{td} \in [0, 1]$ and

$$\lambda_{td+1} = 2.3(\lambda_{td})^{2\mathrm{Sin}(\pi\lambda_{td})}, \tag{4}$$

td is increased by 1 when this reaction is performed.

3.3.4. Redox2 reaction

For binary encoding:

Two points are randomly selected and bits of the reactants between the two points are exchanges similar two 2-point crossover used in genetic algorithms. Fig. 12 depicts an example of Redox2 reaction for binary encoding.

For real encoding:

If R_1 is the reactant with better objective function (enthalpy) then,

$$r_i = \lambda_{tr}(r_i^1 - r_i^2) + r_i^1, \tag{5}$$

where $\lambda_{tr} \in [0, 1]$ and

$$\lambda_{tr+1} = \begin{cases} 0 & , \lambda_{tr} = 0 \\ 1/\lambda_{tr} \bmod (1) & , \lambda_{tr} \in (0, 1), \end{cases} \tag{6}$$

$$1/\lambda_{tr} \bmod(1) = \frac{1}{\lambda_{tr}} - \left\lfloor \frac{1}{\lambda_{tr}} \right\rfloor \tag{7}$$

```
Algorithm 1. GenerateInitialReactants
// R is reactants set; I is indices set and I_e is the enlarged indices set.
1- Create two reactants such as one of them R[1] contains all upper bounds for
   variables and the other R[2] contains all lower bounds for variables.
2- Index←3
3- k←2
4- while R is not saturated do
      Let i_e be an element of I_e and each i_e are enlarged with bit value and this bit
value corresponds to part.
     i←1
     while R is not saturated and all reactants are not generated for a specific value
of k (and I ≤ 2^k-2) do
        i is a k-bit number and i_e corresponds to the enlarged value of i. Each bit of
i is enlarged up to length of corresponding part of R[0] and R[1].
        for j←1 to n do
           if j^th bit of i_e is 1 then
              j^th value of R[Index] is equal to R[1]*r
           else
              j^th value of R[Index] is equal to R[2]*r
           end if
           r is a random real number in interval [0,1].
        end for
      Index←Index+1
      i←i+1
    end while
    k←k+1
  end while
```

Fig. 9. Steps of generating the initial reactants (adapted from Karci and Alatas (2006), Karci et al. (2006) and Karci (2007)).

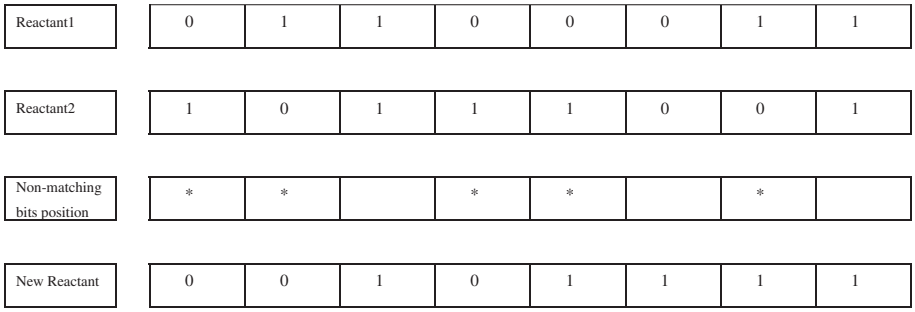

Fig. 10. Synthesis reaction operation representation for binary encoding.

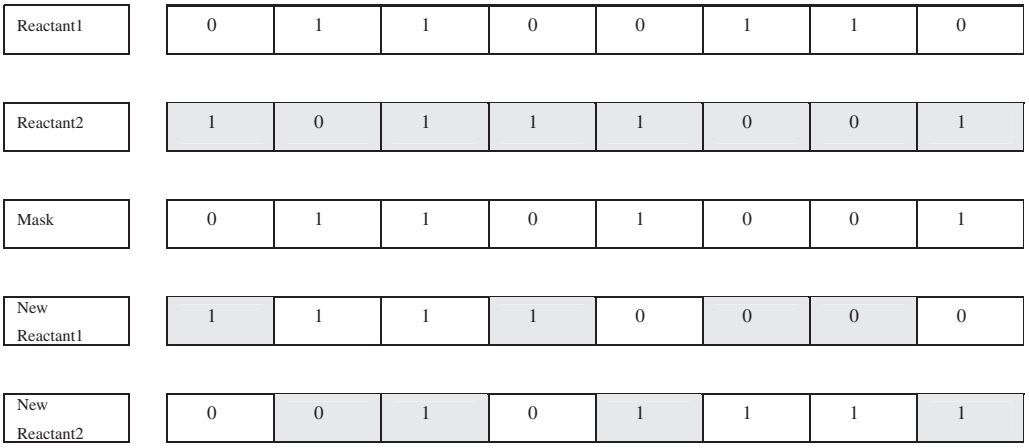

Fig. 11. Displacement reaction operation representation for binary encoding.

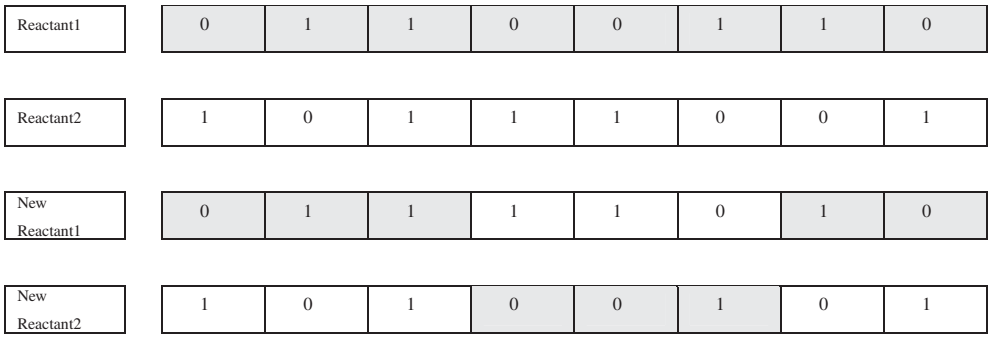

Fig. 12. Redox2 reaction operation representation for binary encoding.

and $\lfloor z \rfloor$ denotes the largest integer less than z and acts as a shift on the continued fraction representation of numbers. *tr* is increased by 1 when this reaction is performed.

3.3.5. Monomolecular reactions

3.3.5.1. Decomposition reaction. For binary encoding:

Two random points in the reactant string are selected and the bits between those points are reversed. The schematic representation is depicted in Fig. 13.

For real encoding:

Let $R = (r_1, \ldots, r_i, \ldots, r_n)$ be the reactant and $r_i \in [l_i, u_i]$ be an atom or a property that will be act a part in a monomolecular reaction. The new atom or a different property of this molecule r'_i is a random number from the domain $[l_i, u_i]$.

3.3.5.2. Redox1 Reaction. For binary encoding:

A randomly selected bit is changed from one to zero or vice versa as shown in Fig. 14.

For real encoding:

$$r'_i = l_i + \lambda_t(u_i - l_i), \tag{8}$$

where $\lambda_t \in [0, 1]$ under the conditions that the initial $\lambda_0 \in (0, 1)$ and that $\lambda_0 \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$ and

$$\lambda_{t+1} = 4\lambda_t(1 - \lambda_t), \tag{9}$$

t is increased by 1 when this reaction is performed.

3.4. Reactants update

In this step, chemical equilibrium test is performed. If the newly generated reactants give better function value, the new reactant set is included and the worse reactant is excluded similar to reversible chemical reactions.

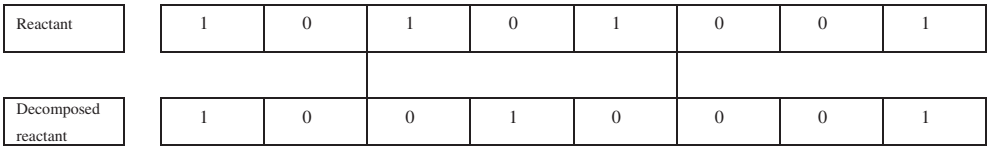

Fig. 13. Decomposition reaction operation representation for binary encoding.

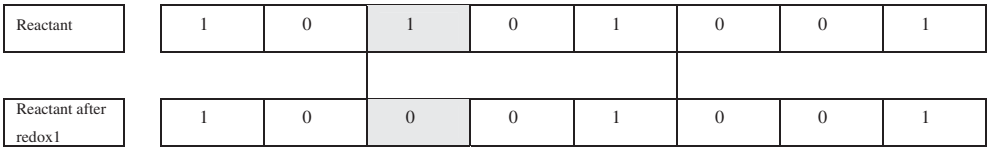

Fig. 14. Redox1 reaction operation representation for binary encoding.

3.5. Termination criterion check

The ACROA is terminated when the termination criterion (e.g. maximum number of iterations) has been met. Otherwise, Steps 3.3 and 3.4 are repeated.

4. Applications of ACROA

4.1. Multiple sequence alignment

The multiple sequence alignment problem searches the homology in three or more sequences by simultaneously aligning. Multiple sequence alignment is very useful in finding motifs or conserved domains, doing phylogenetic analysis and estimating evolutionary distance, characterizing protein families, prediction of the secondary and tertiary structures of new sequences, molecular evolution analysis using phylogenetic methods for constructing phylogenetic trees, highlighting conserved and/or variable sites/regions, uncovering changes in gene structure, and summarizing information.

Multiple sequence alignments are computationally difficult to produce and most formulations of the problem lead to NP-complete combinatorial optimization problems (Wang & Jiang, 1994). Time complexity is given in Eq. (10). In this equation, L is the length of the sequences and N is the number of sequences

$$T(L,N) = L^N. \tag{10}$$

In order to code this problem for the ACROA, a molecule is considered as knowledge string. So a molecule is needed to express the appropriate alignment. A possible sequence alignment can be considered string consists of N numbers. N represents the number of aligned sequence. A number in the string shows the sliding amount of its sequence. Fig. 15 depicts a sequence sliding process of a molecule string. First number of the string shows that the first sequence will be slid two characters right, second number shows that the second sequence will be slid one character right and, etc. The string 32,351 means the same alignment. That is why, the smallest number in the string is subtracted from all the numbers in the string and minimum number is always zero.

Objective function value can be calculated by adding one when getting the same values in each column of the sequences. For example the objective function value of the molecule above is four.

Six different sequences consist of $\{A, B, C, D\}$ have been generated as shown in Fig. 16. The number of possible alignment of these sequences is 64,000,000. After 100 iterations, the obtained molecule is shown in Fig. 17. Objective function value of this problem is seven as obtained via ACROA. The last molecule string is 403,166.

4.2. Association rules mining

Mining of association rules, one of the important data mining technologies, has been originally proposed for Market Basket data to study consumer-purchasing patterns in retail stores. However it has potential applications in many areas. An association rule tells us about the association between two or more items. The major task is to discover dependences among data. It is a relationship of the form $X \Rightarrow Y$ where X is the antecedent itemset and Y is the consequent itemset. Most of the algorithms for mining association rules are based on the rule induction paradigm, where the algorithm usually performs a kind of local search. Generally, association rules mining problem generate all association rules that have support and confidence greater than or equal to user-specified minimum support and minimum confidence thresholds (Agrawal, Imielinski, & Swami, 1993). There are also automated approaches for mining association rules.

Association rules may also be useful within bio-medical data. For example, protein–protein interaction data or gene expression profiling data are potential data for mining useful and interesting association rules (Li, Liu, Tung, & Wong, 2004). It is important to understand general rules about protein interacting with each other such as "the protein having a feature A interacts with the protein having the feature B", "this domain interacts with this domain" and so on. Biologically relevant association between different genes or between environmental effects and gene expression may be discovered using association rules mining techniques. These rules help relating the expression of genes to their cellular environment (Li et al., 2004, chap. 3).

In order to test the ACROA for mining association rules, a reduced protein database is selected for simplicity. Let $F = \{A, B, C, D, E\}$ be the features and $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}$ be proteins where

$p_1 = \{ABD\}$, $p_2 = \{ABC\}$, $p_3 = \{ABCDE\}$, $p_4 = \{CDE\}$, $p_5 = \{ABDE\}$, $p_6 = \{CDE\}$, $p_7 = \{ABC\}$, $p_8 = \{CDE\}$, $p_9 = \{CDE\}$, $p_{10} = \{CD\}$.

Support is 0.5 and confidence is selected as 0.8.

Proteins in the database are

$$P = \{\{11010\}, \{11100\}, \{11111\}, \{00111\}, \{11011\}, \{00111\}, \{11100\}, \{00111\}, \{00111\}, \{00110\}\}.$$

Association rule mining using ACROA requires representation of rules and evaluating the objective function values for these rules. The simplest form of representation is to consider each item in binary domain $\{0, 1\}$. Each molecule consists of fixed length binary string representing some subset of the given item set. Molecule is encoded with 5-bit binary code and these bits represent five items, respectively A, B, C, D, E. The zeros in the code represent the elimination and the ones represent the

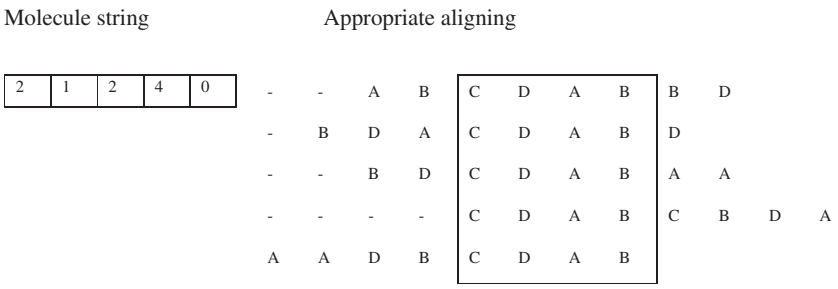

Fig. 15. Encoding the molecule and decoding for evaluating.

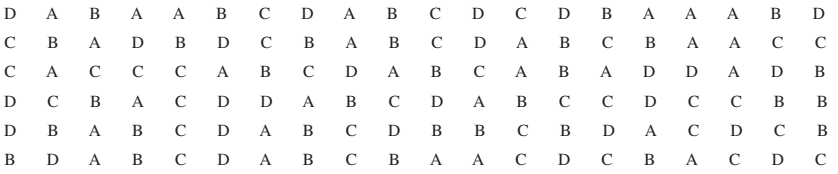

Fig. 16. The used sequences in the simulation.

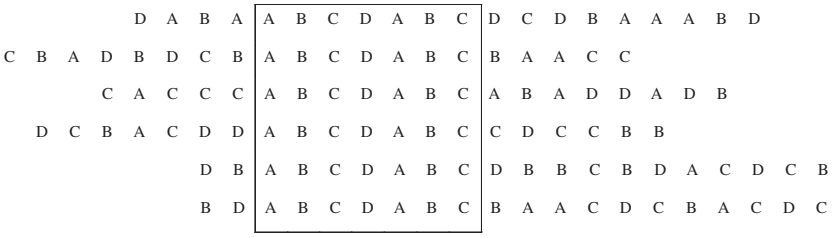

Fig. 17. Result of sequence alignment.

Table 1
Initial reactants and objective function values,

Reactants	00000	11111	00011	11100	00101	00110	00111	11000	11001	11110
Objective function	0	1	6	3	5	5	5	5	2	1

inclusion of the features. For example "10010" indicates the inclusion of the first and the fourth features and elimination of the second, the third and the fifth features. Objective function value is evaluated based on the number of occurrences of each molecule with respect to the proteins in the database. Molecules that have unique 1 in their code can have highest objective function value but this does not mean anything about association rule, therefore in the case of meeting like this situation, one of the 0 bits is converted to 1.

The initial reactants and objective function values are in Table 1. After 10 iterations the obtained molecules and their objective function values are demonstrated in Table 2.

From Table 3, it can be seen that hidden information are:

$$F = \{\{11000\}, \{00110\}, \{00101\}, \{00011\}, \{00111\}\}$$

or,

$$F = \{\{AB\}, \{CD\}, \{CE\}, \{DE\}, \{CDE\}\}$$

This hidden information is frequent feature sets and can be used to derive association rules. Table 3 shows the rules with computed confidence values to be checked dependent upon the given minimum confidence threshold. Support threshold is 0.5 and confidence threshold is 0.8. Association rules are easily derived as shown below:

$AB : A \Rightarrow B$ is if

$$\frac{\text{support}(A \cup B)}{\text{support}(A)} = \frac{5}{5} = 1 \geqslant 0.8$$

$BA : B \Rightarrow A$ is if

$$\frac{\text{support } (A) \cup B)}{\text{support}} \ (B) = \frac{5}{5} = 1 \geqslant 0.8.$$

All derived association rules can be seen in Table 4. $A \Rightarrow B$ means that every protein containing A also contains B meeting support and confidence thresholds. Others rules are similarly interpreted.

The proposed method seems not to be scalable for binary transactions; however the effectiveness of the method may be seen in quantitative data. This is only a new approach for association rule mining.

4.3. Numerical test problems

Well-defined benchmark functions which are based on mathematical functions can be used as objective functions to measure and test the performance of optimization methods. The nature, complexity and other properties of these benchmark functions can be easily obtained from their definitions. The difficulty levels of most benchmark functions are adjustable by setting their parameters. From the standard set of benchmark problems available in the

Table 2
Molecules in the last iteration.

Reactants	11000	00110	00101	00011	00111	00110	00011	01100	11100	01100
Objective function	5	6	5	6	5	6	6	6	3	3

Table 3
Rules to be checked.

Test rule	$A \Rightarrow B$	$C \Rightarrow D$	$C \Rightarrow E$	$D \Rightarrow E$	$C \Rightarrow DE$	$D \Rightarrow CE$	$E \Rightarrow CD$
Confidence	1.000	0.750	0.625	0.750	0.625	0.625	0.8333
Test rule	$B \Rightarrow A$	$D \Rightarrow C$	$E \Rightarrow C$	$E \Rightarrow D$	$DE \Rightarrow C$	$CE \Rightarrow D$	$CD \Rightarrow E$
Confidence	1.000	0.750	0.833	1.000	0.8333	1.000	0.8333

literature, three important functions one of which is unimodal (containing only one optimum) ant two of which are multimodal (containing many local optima, but only one global optimum) are considered to test the efficacy of the proposed methods. Table 5 shows the main properties of the selected benchmark functions used in the simulations.

All compared algorithms were initialized in regions that include the global optimum for a fair evaluation. The algorithms were run for 100 times. Selected three benchmark problems are solved by simulating the newest metaheuristics Artificial Bee Colony (ABC) algorithm Karaboga & Basturk, 2007, Biogeography-based Optimization (BBO) algorithm Simon, 2008, and ACROA. Two criteria are applied to terminate the simulation of the algorithms: reaching maximum number of iterations which is set to a constant number and the second criterion is getting a minimum error.

Simulation has been performed not to find the global optimum values but to find out the potential of the algorithms. Algorithm success rate defined in Eq. (11) has been used for comparison of the results obtained from the simulated algorithms.

$$S = 100\frac{NT_{successful}}{NT_{all}}|Q_{level} \tag{11}$$

$NT_{successful}$ is the number of trials, which found the solution on the Q_{level} in the allowable maximum iteration. NT_{all} is the number of all trials. Q_{level} is the end condition to stop the algorithm, when it converges into Q_{level} tolerance.

All algorithms were initialized in regions that include the global optimum for a fair evaluation. The algorithms were run for 100 times to catch their stochastic properties. In this simulation, maximum iteration number was set to 500.

The colony size for the ABC algorithm was selected as 20. Limit parameter for ABC was selected as 40. The number of food sources equals the half of the colony size (20/2 = 10).

The number of islands for BBO algorithm was selected as 20. Mutation probability was 0.005 and elitism parameter was 1 for BBO algorithm. Elitism parameter shows how many of the best habitats will be kept from one generation to the next. Habitat modification probability was selected as 1. ACROA was started to be

Table 4
Obtained association rules.

Mined association rules							
$A \Rightarrow B$	$B \Rightarrow A$	$E \Rightarrow C$	$E \Rightarrow D$	$DE \Rightarrow C$	$CE \Rightarrow D$	$E \Rightarrow CD$	$CD \Rightarrow E$

Table 5
Properties of test problems (*lb* indicates lower bound, *ub* indicates upper bound, *opt* indicates optimum point).

Function no.	Function name	Definition	lb	ub	opt	Dimension (N)	Property
1	Griewangk	$F_1(x) = \sum_{i=1}^{N}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	−600	600	0	10	Multimodal
2	Rastrigin	$F_2(x) = 10 \times N + \sum_{i=1}^{N}\left(x_i^2 - 10 \cdot \cos(2\pi x_i)\right)$	−5.12	5.12	0	10	Multimodal
3	Rosenbrock	$F_3(x) = \sum_{i=1}^{N} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	−2.048	2.048	0	2	Unimodal

Table 6
Success rates of the used algorithms for the selected test functions (Best results have been written in bold).

Q_{level}	Griewangk			Rastrigin			Rosenbrock		
	ABC	BBO	ACROA	ABC	BBO	ACROA	ABC	BBO	ACROA
1.e−5	13	0	**32**	15	70	**85**	8	**31**	10
1.e−6	13	0	**29**	60	67	**79**	8	**30**	9

implemented with 20 reactants. The simulation results have been depicted in Table 6. Although this was the first implementation of ACROA, it has shown better performance within two of three benchmark functions. Especially, in multi-modal functions it has been shown better performance. ACROA is better than ABC algorithm within all selected benchmark functions and better than BBO algorithm in Griewangk and Rastrigin functions.

5. Conclusions

A novel, chemistry inspired computational method has been developed by inspecting the efficient objects, states, process, and events en bloc in the different types of chemical reactions. This method includes both global and local search ability and does not need a local search method to refine the search. This method does not use an extra function or relation such as fitness function for determination of quality of reactants. Only objective function is needed. Furthermore, this method does not need many parameters that must be specified by the users a priori. Only number of initial reactants is enough for the algorithm implementation. The initial reactants are distributed over feasible search region. That is why; optimal or near-optimal solutions can be obtained in shorter time.

It can be easily adapted to multi-objective optimization problems. Parallel or distributed versions may also be considered for further implementations.

Acknowledgements

This study is supported by Turkish Scientific and Technical Research Council – TUBITAK (Project Number: MAG-110M309).

References

Agrawal, R., Imielinski, T., & Swami, A. N. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on management of data, Washington, DC* (pp. 207–216).

Alatas, B. (2010). Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications, 37*(8), 5682–5687. doi:10.1016/j.eswa.2010.02.042.

Alatas, B. (2010). Chaotic harmony search algorithms. *Applied Mathematics and Computation, 216*(9), 2687–2699.

Alatas, B., & Akın, E. (2005). *Mining fuzzy classification rules using an artificial immune system with boosting ADBIS 2005. Lecture notes in computer science* (Vol. 3631). Springer-Verlag, pp. 283–293.

Alatas, B., & Akın, E. (2005). *FCACO: Fuzzy classification rules mining algorithm with ant colony optimization, ICNC 2005. Lecture notes in computer science* (Vol. 3612). Springer-Verlag, pp. 787–797.

Alatas, B., & Akın, E. (2006). An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Computing, 10*(3), 230–237.

Alatas, B., & Akin, E. (2008). Rough particle swarm optimization algorithm and ıts applications in data mining. *Soft Computing, 12*(2), 1205–1218.

Alatas, B., & Akin, E. (2009a). Chaotically encoded particle swarm optimization algorithm and its applications. *Chaos Solitons & Fractals, 41*(2), 939–950.

Alatas, B., & Akin, E. (2009b). Multi-objective rule mining using a chaotic particle swarm optimization algorithm. *Knowledge-Based Systems, 22*, 455–460.

Alatas, B., Akin, E., & Ozer, A. B. (2009). Chaos embedded particle swarm optimization algorithms. *Chaos Solitons & Fractals, 40*(4), 1715–1734.

Baykasoglu, A., Ozbakir, L., & Tapkan, P. (2007). Artificial bee colony algorithm and its application to generalized assignment problem. In F. T. S. Chan & M. K. Tiwari (Eds.), *Chapter 8 of swarm intelligence: Focus on ant and particle swarm optimization* (pp. 532). Itech Education and Publishing.

Birbil, S. I., & Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization, 25*(3), 263–282.

Chu, S. C., Tsai, P. W., & Pan, J. S. (2006). Cat swarm optimization. *Lecture Notes in Computer Science, 4099/2006*, 854–858.

De Castro, L. N., & Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transaction on Evolutionary Computation, 6*(3), 239–251.

Dorigo, M., Maziezzo, V., & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating ants. *IEEE Transactions on Systems, Man and Cybernetics B, 26*(1), 29–41.

Erol, O. K., & Eksin, I. (2006). A new optimization method: Big bang-big crunch. *Advances in Engineering Software, 37*, 106–111.

Gargari, E. A., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In *IEEE congress on evolutionary computation, Singapore* (pp. 4661–4667).

Goldberg, D. E. (1989). *Genetic algorithms in search. Optimization and machine learning*. Boston, MA: Kluwer Academic Publishers.

Gundogan, K. K., Alatas, B., & Karci, A. (2004). Mining classification rules by using genetic algorithms with non-random initial population and uniform operator. *Turkish Journal of Electrical Engineering and Computer Sciences (ELEKTRİK), 12*(1), 43–52.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471.

Karci, A. (2007). Theory of saplings growing up algorithm. *Lecture Notes in Computer Science, 31*, 450–460.

Karci, A., & Alatas, B. (2006). *Thinking capability of saplings growing up algorithm iDEAL 2006. Lecture notes in computer science* (Vol. 4224). Springer-Verlag, pp. 386–393.

Karci, A., & Arslan, A. (2002). Uniform population in genetic algorithms İU. *Journal of Electrical and Electronics, 2*(2), 495–504.

Karci, A., Alatas, B., & Akin, E. (2006). *Fidan Gelişim Algoritması (Turkish)*, ASYU-INISTA 2006 (pp. 57–61).

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks*, Australia (Vol. 4, pp. 1942–1948).

Lam, A. Y. S., & Li, V. O. K. (2009). *Chemical-reaction-inspired metaheuristic for optimization*. Technical Report TR-2009-003.

Lamberti, L., & Pappalettere, C. (2007). Weight optimization of skeletal structures with multi-point simulated annealing. *Computer Modeling in Engineering and Sciences, 18*, 183–221.

Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continues engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering, 194*, 3902–3933.

Li, J., Liu, H., Tung, A., & Wong, L. (2004). Data mining for the practical bioinformatician. In L. Wong (Ed.), *The practical bioinformatician* (pp. 35–70). World Scientific.

Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics, 1*(4), 355–366.

Mucherino, A., & Seref, O. (2007). Monkey search: A novel meta-heuristic search for global optimization. In *AIP conference proceedings 953, data mining, system analysis and optimization in biomedicine* (pp. 162–173).

Muhlenbein, H., & SchlierkampVoosen, D. (1993). Predictive models for the breeder genetic algorithm I. Continuous parameter optimization. *Evolutionary Computation, 1*, 25–49.

Nag, P. K. (2008). *Engineering thermodynamics* (4th ed.). McGraw-Hill, pp. 9–10.

Niizuma, D., Yasuda, K., & Ishigame, A. (2006). Multi-point Tabu search for traveling salesman problems. *IEEJ Transactions on Electrical and Electronic Engineering, 1*(1), 126–129.

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine, 22*(3), 52–67.

Portegts, T. E. (2004). An evolvable artificial chemistry featuring continuous physics and discrete reactions. In *Alife IX proceedings* (pp. 484–488).

Sacco, W. F., & Cassiano, R. E. (2005). De Oliveira A new stochastic optimization algorithm based on a particle collision metaheuristic. In *World congress of structural and multidisciplinary optimizaion*.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation, 12*, 702–713.

Sywerda, G. (1989). Uniform crossover in genetic algorithms. In *Proceedings of the 3rd international conference on genetic algorithms, USA*.

Wang, L., & Jiang, T. (1994). On the complexity of multiple sequence alignment. *Journal of Computational Biology, 1*, 337–348.

Wiese, K. C., Hendriks, A., Deschênes, A. (2006). Analysis of thermodynamic models and performance in RnaPredict – An evolutionary algorithm for RNA folding. In *IEEE CIBCB 2006* (pp. 1–9).

Yang, X. S. (2008). *Nature-inspired metaheuristic algorithms*. Luniver Press, p. 128.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *Stochastic algorithms: Foundations and applications SAGA 2009. Lecture notes in computer sciences* (Vol. 5792, pp. 169–178).