Expert Systems with Applications 39 (2012) 11080–11088

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

A novel chemistry based metaheuristic optimization method for mining of classification rules

Bilal Alatas [*]

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

ARTICLE INFO

Keywords:
Artificial chemical reaction optimization algorithm
Data mining
Classification rules
Heuristic optimization
Performance
Uniform population

ABSTRACT

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 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. In this study, one of the first applications of this method has been performed in classification rule discovery field of data mining and efficiency has been demonstrated.

1. Introduction

Classical optimization algorithms are inflexible to adapt the solution procedure to an optimization problem. Generally a given problem is modeled in such a way that a classical algorithm can handle it. This generally requires making several assumptions and/or modifications which might not be easy to validate in many situations. These modifications and/or assumptions on the original problem parameters (rounding variables, softening constraints etc.) certainly affect the solution quality (Baykasoglu, Ozbakir, & Tapkan, 2007). They are insufficient if integer and/or discrete decision variables are required in optimization models (Baykasoglu et al., 2007). Solution strategies of classical optimization algorithms are generally depended on the type of objective and constraint functions (linear, non-linear etc.) and the type of variables used in the problem modeling (integer, real etc.). Their efficiency is also very much dependent on the size of the solution space, number of variables and constraints used in the problem modeling, and the structure of the solution space (convex, non-convex, etc.). Briefly, they do not offer general solution strategies that can be applied to problem formulations where, different type of variables, objective and constraint functions are used. However, most of the optimization problems require different types of variables, objective and constraint functions simultaneously in their formulation (Baykasoglu et al., 2007).

Inefficiency of classical optimization algorithms in solving larger scale and/or highly non-linear problems forced researchers to find more flexible and adaptable general purpose novel algorithms. Problem and model independent heuristic optimization algorithms have been proposed by researchers to overcome the drawbacks of the classical optimization procedures. 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.

Sometimes, some effective ideas are inspired by biological, physical, and social processing. In fact, human beings have often simulated natural phenomena to create new technologies resulting in created technologies different from the naturally and ecologically simulated phenomena. There are many computational methods inspired by the human beings and natural processes. Artificial neural networks (Haykin, 1999; Rupérez, Martín-Guerrero, Monserrat, & Alcañiz, 2012), evolutionary computation (de Jong, 2006), artificial immunology (de Castro & Timmis, 2002; Laurentys, Palhares, & Caminhas, 2010) are biologically based methods. Ant colony optimization (Dorigo, 2004; Xiao & Ping Li, 2011), bee colony optimization (Alatas, 2010; Bozorg et al., 2005), particle swarm optimization (Alatas, Akin, & Ozer, 2009; Kennedy & Eberhart, 1995), cat swarm optimization (Chu, Tsai, & Pan, 2006; Pradhan & Panda, 2012), bacterial foraging optimization (Chatzis & Koukas, 2011; Chen, Zhu, & Hu, 2009), glowworm swarm optimization (Krishnanand & Ghose, 2006; Wu, Qian, Ni, & Fan, 2012), artificial fish-swarm algorithm (Li, 2003), monkey search optimization (Mucherino & Seref, 2007), firefly algorithm (Horng & Liou, 2011; Yang, 2009) are swarm based optimization methods. Big bang – big crunch optimization (Alatas, 2011; Erol & Eksin, 2006), charged system search (Kaveh & Laknejadi, 2011; Kaveh & Talatahari, 2010), artificial physics optimization algorithm (Xie, Zeng, & Cui, 2009), intelligent water drops algorithm (Shah-Hosseini, 2008), electromagnetism-like algorithm (Birbil & Fang, 2003; Jamili, Shafia, & Tavakkoli-Moghaddam, 2011), particle collision algorithm (Sacco, 2005) central force optimization (Formato,

[*] Tel.: +90 428 213 17 94.
E-mail address: balatas@tunceli.edu.tr

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.03.066

2007; Green, Wang, & Alam, 2012), gravitational search algorithm (Rashedi, Nezamabadi-pour, & Saryazdi, 2009; Yin, Hu, Yang, Li, & Gu, 2011) are physics based methods. Tabu search (Aladag & Köksoy, 2011; Glover, 1989), imperialist competitive algorithm (Atashpaz-Gargari & Lucas, 2007; Behnamian & Zandieh, 2011), parliamentary optimization algorithm (Borji, 2007), and teaching-learning-based algorithm (Rao, Savsani, & Vakharia, 2012) are social based optimization methods. Some methods can be considered as hybrid.

Similar to these methods, chemistry based a new computational method, entitled as artificial chemical reaction optimization algorithm (ACROA), for searching and optimization problems based on the types and occurring of chemical reactions has been proposed (Alatas, 2011). There is only one paper considering the collision of particles and omitting the conservation of energy as chemistry based heuristics algorithm (Lam & Li, 2010). However, the proposed computational ACROA method is completely different from this paper in terms of reason and principle. Appropriate system can be selected according to the interested optimization problem and modifications may be added to the proposed ACROA. Molecules are encoded in ACROA using appropriate encoding scheme for the optimization problems. It seems robust and has less parameter than that of used in the literature. It also seems a strong algorithm with high in quality initial population generation method and chaotic properties of the operators.

Encoding of the reactants for ACROA depends on the interested problem. It can be binary, real, string, and 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 uniform 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).

In this study, one of the first applications of ACROA has been performed in classification rule discovery field of data mining. Databases have been conceived as search space and ACROA has been designed as search method for mining accurate and comprehensible classification rules.

The organization of this paper is as follows. In the Section 2, ACROA and the computational operators inspired by chemical reactions are presented. The application of this novel computational method in data mining and experimental results are presented in Section 3. Finally, Section 4 concludes this paper.

2. 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 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, and 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) (Alatas, 2011).

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. 1. 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. 2 and pseudo-code of which is shown in Fig. 3 consists of the following five steps:

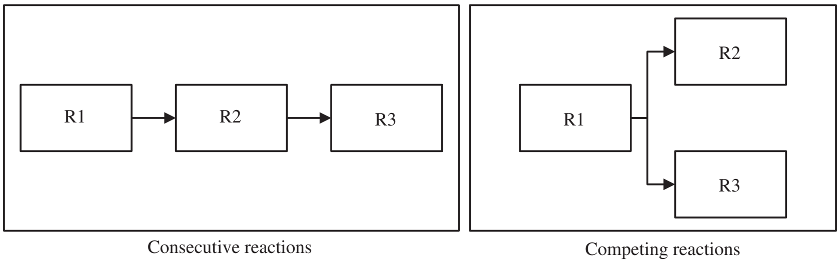
Step 1: Problem and algorithm parameter initialization.
Step 2: Setting the initial reactants and evaluation.
Step 3: Applying chemical reactions.
Step 4: Reactants update.
Step 5: Termination criterion check.

2.1. Problem and algorithm parameter initialization

The optimization problem is specified as follows:

Minimize $f(x)$ 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 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, and etc. The sole ACROA parameter, *ReacNum*, is also specified in this step.



Consecutive reactions

Competing reactions

Fig. 1. Consecutive and competing reactions.

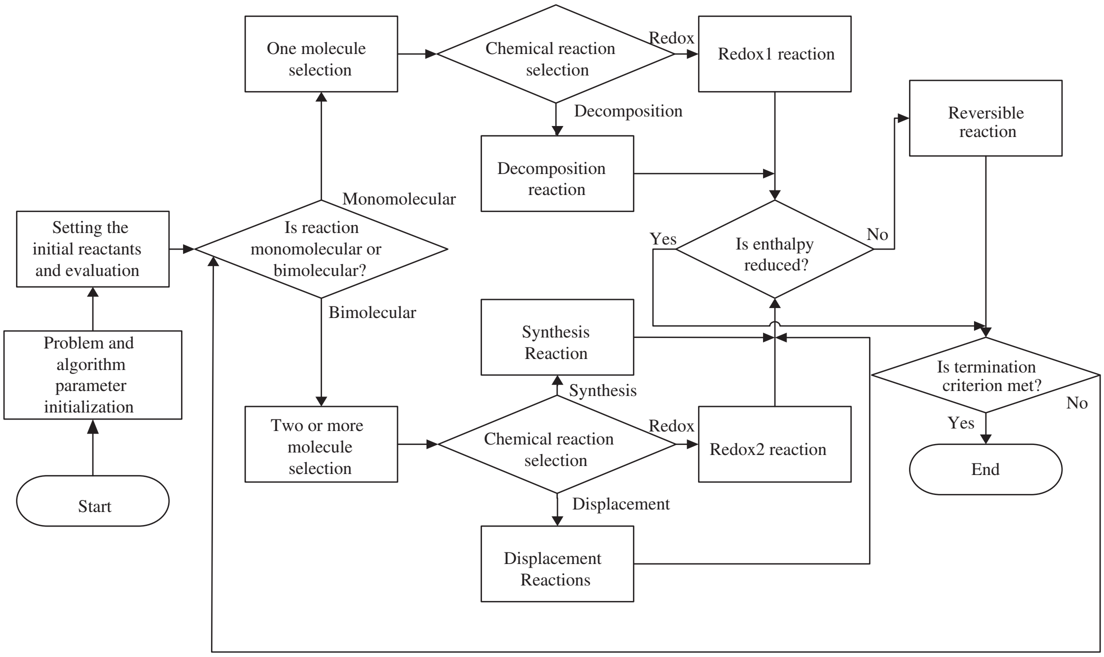
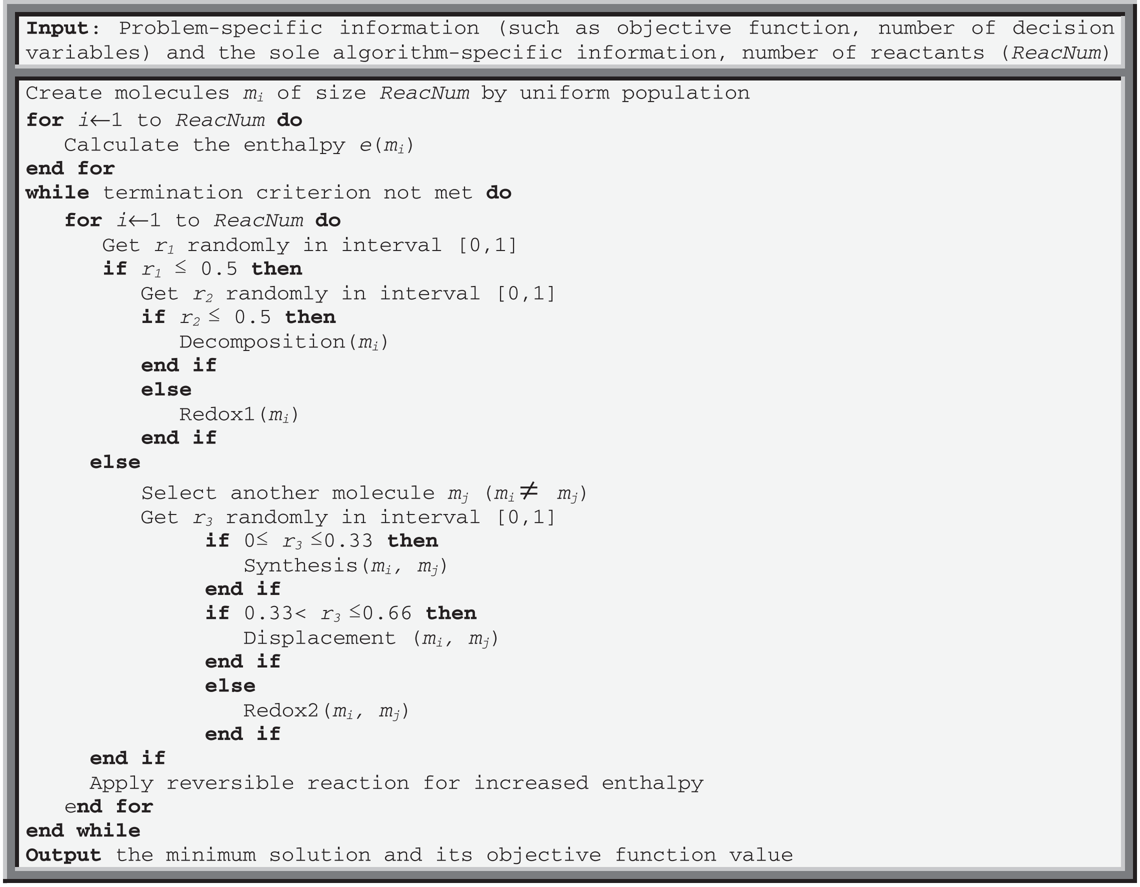

One molecule selection
Chemical reaction selection
Redox
Redox1 reaction
Decomposition
Decomposition reaction
Reversible reaction
Setting the initial reactants and evaluation
Monomolecular
Is reaction monomolecular or bimolecular?
Bimolecular
Yes
Is enthalpy reduced?
No
Problem and algorithm parameter initialization
Synthesis Reaction
Synthesis
Is termination criterion met?
No
Yes
End
Two or more molecule selection
Chemical reaction selection
Redox
Redox2 reaction
Start
Displacement
Displacement Reactions

Fig. 2. 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 $i \leftarrow 1$ to ReacNum do
   Calculate the enthalpy $e(m_i)$
end for
while termination criterion not met do
   for $i \leftarrow 1$ to ReacNum do
      Get $r_1$ randomly in interval [0,1]
      if $r_1 \leq 0.5$ then
         Get $r_2$ randomly in interval [0,1]
         if $r_2 \leq 0.5$ then
            Decomposition($m_i$)
         end if
         else
            Redox1($m_i$)
         end if
     else
         Select another molecule $m_j$ ($m_i \neq m_j$)
         Get $r_3$ randomly in interval [0,1]
              if $0 \leq r_3 \leq 0.33$ then
                 Synthesis($m_i$, $m_j$)
              end if
              if $0.33 < r_3 \leq 0.66$ 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. 3. Pseudo-code of ACROA (Alatas, 2011).

2.2. Setting the initial reactants and evaluation

In this step initial reactants are evenly initialized in the feasible searching space. Uniform population method (Gundogan, Alatas, & Karci, 2004; Karci, 2007; Karci & Alatas, 2006; Karci & Arslan, 2002; Karci et al., 2006) 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

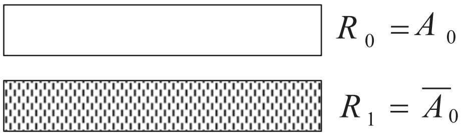


Fig. 4. Initial reactants.

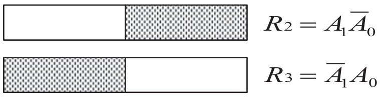


Fig. 5. New reactants with $k = 2$.

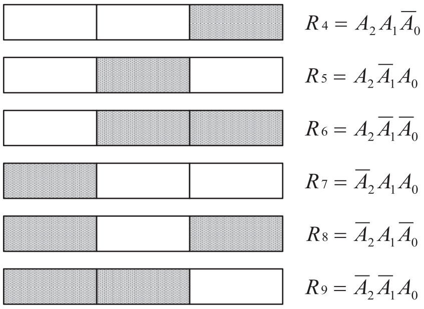

$R_4 = A_2 A_1 \overline{A}_0$
$R_5 = A_2 \overline{A}_1 A_0$
$R_6 = A_2 \overline{A}_1 \overline{A}_0$
$R_7 = \overline{A}_2 A_1 A_0$
$R_8 = \overline{A}_2 A_1 \overline{A}_0$
$R_9 = \overline{A}_2 \overline{A}_1 A_0$

Fig. 6. New reactants with $k = 3$.

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 (Gundogan et al., 2004; Karci, 2007; Karci & Alatas, 2006; Karci & Arslan, 2002; Karci et al., 2006).

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. 4). 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. 5).

$$R_2 = \{r^*u_1, r^*u_2, \ldots, r^*u_n/2, r^*l_{n/2+1}, r^*l_{n/2+2}, \ldots, r^*l_n\} \quad \text{and}$$
$$R_3 = \{r^*l_1, r^*l_2, \ldots, r^*l_{n/2}, r^*u_{n/2+1}, r^*u_{n/2+2}, \ldots, r^*u_n\}$$

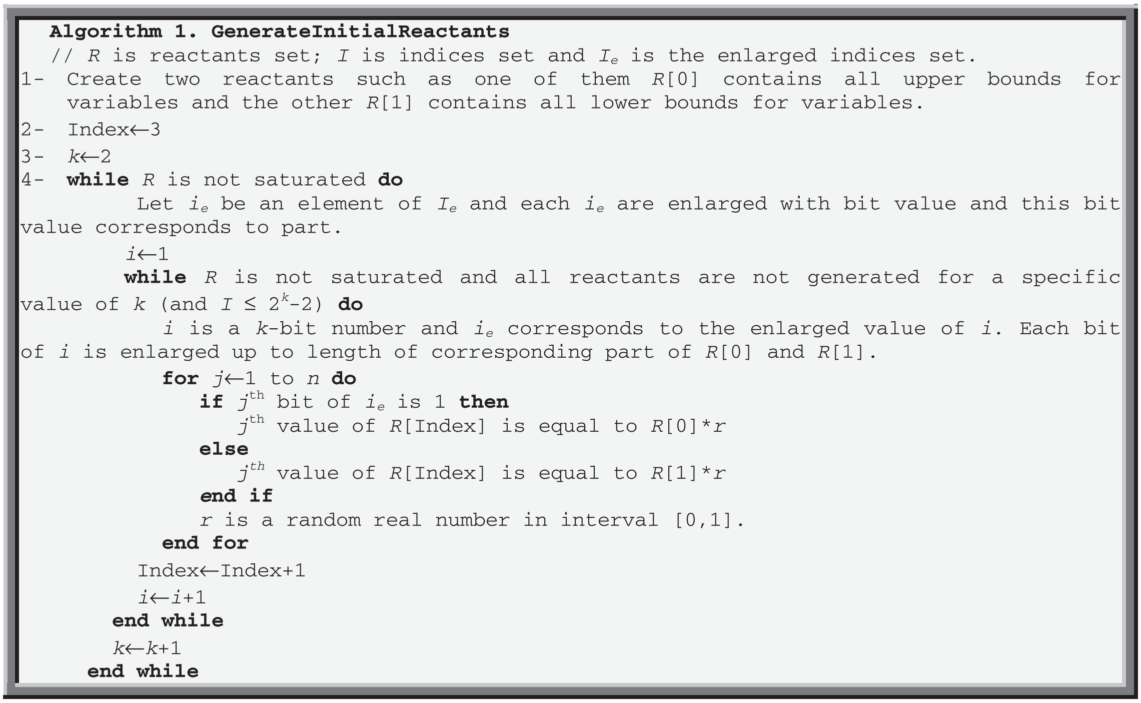
where r is a random number such as $0 \leqslant r < 1$.

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. 6).

$$R_4 = \{r^*u_1, r^*u_2, \ldots, r^*u_{2n/3}, r^*l_{2n/3+1}, r^*l_{2n/3+2}, \ldots, r^*l_n\}$$
$$R_5 = \{r^*u_1, r^*u_2, \ldots, r^*u_{n/3}, r^*l_n/3+1, \ldots, r^*l_{2n/3}, r^*u_{2n/3+1}, \ldots, r^*u_n\}$$
$$R_6 = \{r^*u_1, r^*u_2, \ldots, r^*u_{n/3}, r^*l_{n/3+1}, \ldots, r^*l_n\}$$
$$R_7 = \{r^*l_1, r^*l_2, \ldots, r^*l_{n/3}, r^*u_{n/3+1}, \ldots, r^*u_n\}$$
$$R_8 = \{r^*l_1, r^*l_2, \ldots, r^*l_{n/3}, r^*u_{n/3+1}, \ldots, r^*u_{2n/3}, r^*l_{2n/3+1}, \ldots, r^*l_n\}$$
$$R_9 = r^*l_1, r^*l_2, \ldots, r^*l_{2n/3}, r^*u_{2n/3+1}, \ldots, r^*u_n$$

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

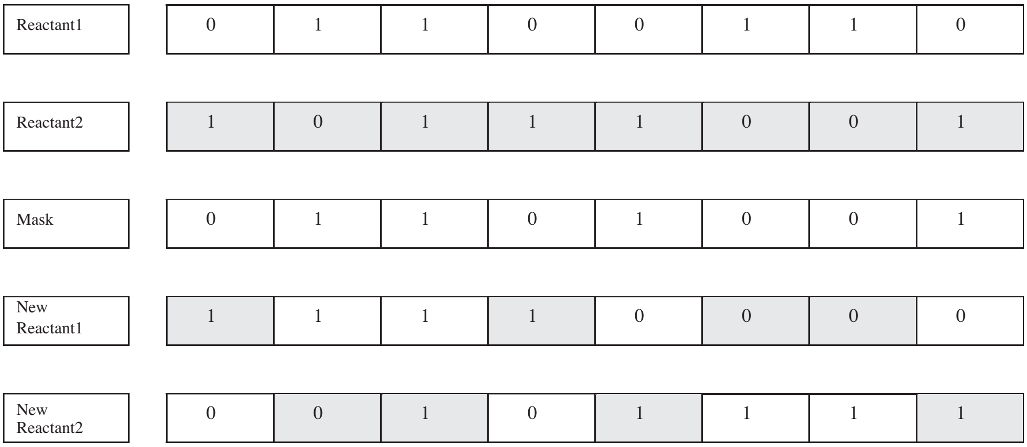


```
 Algorithm 1. GenerateInitialReactants
 // R is reactants set; I is indices set and I_e is the enlarged indices set.
 Create two reactants such as one of them R[0] contains all upper bounds for
 variables and the other R[1] contains all lower bounds for variables.
 Index←3
 k←2
 while R is not saturated 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[0]*r
          else
            j^th value of R[Index] is equal to R[1]*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. 7. Steps of generating the initial reactants (adapted from Karci and Alatas (2006), Karci et al. (2006) and Karci (2007)).

Reactant1	0	1	1	0	0	0	1	1
Reactant2	1	0	1	1	1	0	0	1
Non-matching bits position	*	*		*	*		*	
New Reactant	0	0	1	0	1	1	1	1

Fig. 8. Synthesis reaction operation representation for binary encoding.



Reactant1	0	1	1	0	0	1	1	0
Reactant2	1	0	1	1	1	0	0	1
Mask	0	1	1	0	1	0	0	1
New Reactant1	1	1	1	1	0	0	0	0
New Reactant2	0	0	1	0	1	1	1	1

Fig. 9. Displacement reaction operation representation for binary encoding.

Fig. 7 depicts the steps of generating the initial reactants for ACROA.

2.3. Applying chemical reactions

2.3.1. 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.

2.3.1.1. 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. 8.

2.3.1.2. Synthesis reaction for real encoding. A new reactant is obtained as

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

where

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

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).

2.3.1.3. 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. 9.

2.3.1.4. Displacement reaction for real encoding. Two new reactants are obtained as

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

where

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

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

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

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

td is increased by 1 when this reaction is performed.

2.3.1.5. 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. 10 depicts an example of Redox2 reaction for binary encoding.

2.3.1.6. Redox2 reaction for real encoding. If R_1 is the reactant with better objective function (enthalpy) then,

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

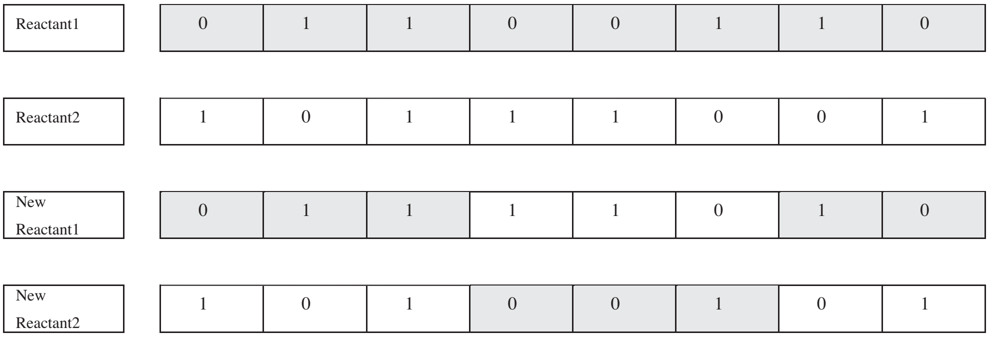


Fig. 10. Redox2 reaction operation representation for binary encoding.

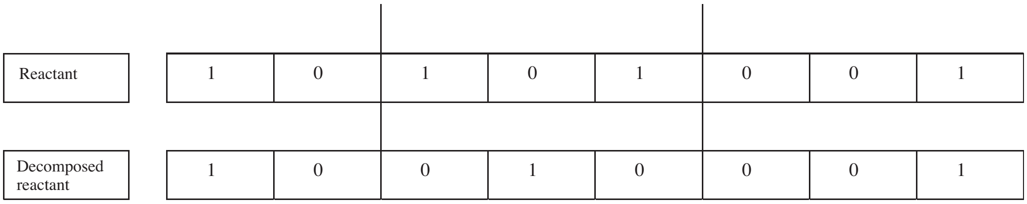


Fig. 11. Decomposition reaction operation representation for binary encoding.

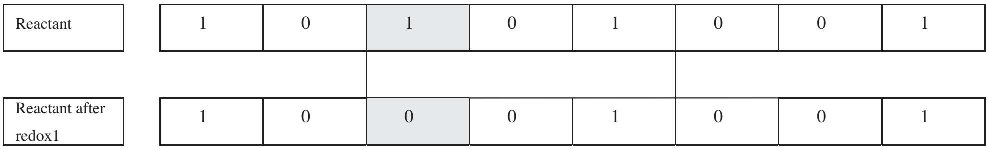


Fig. 12. Redox1 reaction operation representation for binary encoding.

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

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

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

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.

2.3.2. Monomolecular reactions

2.3.2.1. 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. 11.

2.3.2.2. Decomposition reaction 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 $[l_i, u_i]$.

2.3.2.3. Redox1 reaction decomposition reaction for binary encoding.
2.3.2.3.1. For binary encoding. A randomly selected bit is changed from one to zero or vice versa as shown in Fig. 12.

2.3.2.4. Redox1 reaction decomposition reaction for real encoding.

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

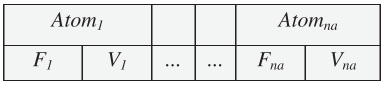


Fig. 13. Rule representation in the reactant.

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.

2.4. 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.

2.5. 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.

3. Application of ACROA

Heuristics search algorithms are popular due to the strength and good ability to search in huge and complex search space in data mining applications (Alatas & Akin, 2005; Alatas & Akin,

Table 1
Mined rules by ACROA within the Zoo data set.

Class	Mined rule antecedent	Predictive accuracy	Comprehensibility
1	If (eggs = 0) ∧ (venomous = 0) ∧ (domestic = 0)	0.91	0.87
2	If (hair = 0) ∧ (feathers = 1) ∧ (venomous = 0) ∧ (legs = 2) ∧ (domestic = 0)	1.00	0.73
3	If (eggs = 1) ∧ (predator = 1) ∧ (toothed = 1) ∧ (catsize = 0)	1.00	0.80
4	If (aquatic = 1) ∧ (breathes = 0) ∧ (tail = 1)	0.80	0.87
5	If (airborne = 0) ∧ (aquatic = 1) ∧ (toothed = 1) ∧ (catsize = 0)	1.00	0.80
6	If (airborne = 1) ∧ (fins = 0)∧ (tail = 0)	0.83	0.87
7	If (predator = 1) ∧ (breathes = 0)∧ (tail = 0) ∧ (domestic = 0)	0.88	0.80

Table 2
Mined rules by GA within the Zoo data set.

Class	Mined rule antecedent	Predictive accuracy	Comprehensibility
1	If (eggs = 0) ∧ (venomous = 0) ∧ (domestic = 0)	0.91	0.87
2	If (feathers = 1) ∧ (breathes = 1) ∧ (domestic = 0)	0.93	0.87
3	If (eggs = 1) ∧ (predator = 1) ∧ (catsize = 0)	0.99	0.87
4	If (aquatic = l) ∧ (breathes = 0)∧ (tail = l)	0.80	0.87
5	If (airborne = 0) ∧ (aquatic = 1)∧ (toothed = 1) ∧ (catsize = 0)	1.00	0.80
6	If (airborne = 1) ∧ (fins=0)∧ (tail = 0)	0.83	0.87
7	If (predator = 1) ∧ (breathes = 0)∧ (domestic = 0)	0.88	0.87

Table 3
Mined rules by ACROA within the Nursery data set.

Class	Mined rule antecedent	Predictive accuracy	Comprehensibility
NR	If (parents = pretentious) ∧ (children = 3) ∧ (housing = convenient) ∧ (health = not_recom)	0.76	0.50
R	If (has_nurs = proper) ∧ (finance = convenient)	0.75	0.75
VR	If (housing = less_conv) ∧ (finance = inconv) ∧ (social = slightly_prob)	0.90	0.63
P	If (parents = usual) ∧ (housing = less_conv) ∧ (social = problematic)	0.78	0.63
SP	If (parents = usual) ∧ (has_nurs = very_crit) ∧ (form = more)	0.79	0.63

Table 4
Mined rules by GA within the Nursery data set.

Class	Mined rule antecedent	Predictive accuracy	Comprehensibility
NR	If (parents = usual) ∧ (housing = less_conv) ∧ (social = slightly_prob) ∧ (health = not_recom)	0.63	0.5
R	If (has_nurs = proper) ∧ (finance = convenient) ∧ (health = recommended)	0.81	0.63
VR	If (housing = less_conv) ∧ (finance = inconv) ∧ (social = slightly_prob)	0.90	0.63
P	If (parents = great_pret) ∧ (children = 3) ∧ (social slightly_prob) ∧ (health = recommended)	0.69	0.50
SP	If (parents = usual) ∧ (has_nurs = very_crit) ∧ (form = more)	0.79	0.63

Table 5
Average performance from the Zoo and the Nursery data sets.

Data set algorithm	Zoo		Nursery	
	Predictive accuracy	Comprehensibility	Predictive accuracy	Comprehensibility
ACROA	0.92	0.82	0.80	0.63
GA	0.91	0.86	0.76	0.58

2006; Alatas & Akin, 2008). An application of ACROA that searches the comprehensible classification rules within data sets has been performed. This application discovers comprehensible IF-THEN rules within two public domain real-world data sets, Zoo and Nursery (they have more than one potential class attribute), obtained from UCI Machine Learning Repository (Blake, 1998). The discovered rules can be evaluated according to several criteria, such as the degree of confidence in the prediction, classification accuracy rate on unknown-class examples, comprehensibility, etc. These rules must be ultimately comprehensible since most decision makers that use the results of mining are not data mining experts, ideally the mined rules should be comprehensible and understandable to human beings.

3.1. Rule representation

Let *na* be the number of predicting attributes in the data being mined. Then a reactant is composed of *na* atoms, where each atom corresponds to a condition containing one attribute. Each *i*th atom is has into two fields: flag (F_i) and value (V_i) as shown in Fig. 13. A reactant corresponds to the entire IF part of the rule and each atom corresponds to one condition in this IF part. The atoms do not involve the class predicted by a rule. In a given run of the ACROA, all reactants are searching for rules predicting the same class. ACROA is run at least once for each class (value of the goal attribute)

The flag field (F_i) is a binary-valued variable in the range. This variable indicates whether or not the corresponding attribute is

Table 6
Predictive accuracy (%) in the Zoo data set.

Class	ACROA	GA
1	100 ± 0.0	100 ± 0.0
2	100 ± 0.0	100 ± 0.0
3	92 ± 12.8	0.0 ± 0.0
4	100 ± 0.0	100 ± 0.0
5	100 ± 0.0	100 ± 0.0
6	90 ± 10.0	90 ± 10
7	83.9 ± 10.2	85.5 ± 0.0

involved in the rule. 1 shows that the corresponding condition will be involved in the rule while 0 shows the condition will be removed from the rule. The value (V_i) field involves one of the values belonging to the domain of the attribute A_i.

3.2. Rule evaluation

Let a rule be of such form “**If A Then C”** where A is rule antecedent that is a conjunction of attributes and C is rule consequent that is the prediction class. The objective function combines two objectives namely the comprehensibility and predictive accuracy.

$$\text{Comprehensibility} = 1 - (\#\ \text{attributes in the reactant})/(\#\ \text{predicting attribute}) \tag{10}$$

$$\text{Predictive accuracy} = \frac{|A + C| - 1/2}{|A|} \tag{11}$$

$|A + C|$ is the number of instances that satisfy both the rule antecedent and the rule consequent and $|A|$ is the number of instances that satisfy only the rule antecedent. Finally the objective function is given by

$$\text{Objective function} = w_1 \times \text{Comprehensibility} + w_2 \times \text{Predictive accuracy} \tag{12}$$

w_1 and w_2 are user-defined weights and in the experiments they are set to 0.3 and 0.7, respectively. Other criterion such as interestingness can also be included in objective function.

3.3. Experimental results

The Zoo data set contains 101 instances and 18 attributes. In this data set, the class attribute is ‘type’ and there are seven classes. The attribute containing the name of the animal that has no generalization capability was removed as a preprocessing. The Nursery data set contains 12960 instances and 9 attributes. In this data set, the class attribute is ‘recommendation’ and there are five classes. They are Not_recom (NC), Recommend (R), Very_recom (VR), Priority (P), and Spec_prior (SP).

Fifty reactants were generated as the initial population for Zoo and Nursery data set. ACROA was terminated when the best objective function value did not change continually throughout 10 generations. For each classification rule, the ACROA was run 3 times, and the rules with generation numbers were obtained. Approximately, this process took 60 generations for Zoo data set and 200 for Nursery data set. Each data set was randomly partitioned into two parts: the training set (1/3 of the instances) and the test set (2/3 of the instances).

The rules obtained by the proposed algorithm are compared with the Genetic Algorithm (GA). The same encoding for chromosomes and the same function for fitness have been used for GA. Cross-over rate was 0.8 and mutation rate was 0.03 for both data sets. Tournament selection with tournament size 10 has been used. It took approximately 100 iteration for Zoo data set and 450 for Nursery data set to reach the termination criteria for GA.

Table 7
Predictive accuracy (%) in the Nursery data set.

Class	ACROA	GA
NR	33.8 ± 10.2	12.8 ± 9.8
R	0.0 ± 0.0	0.0 ± 0.0
VR	100 ± 0.0	100 ± 0.0
P	0.0 ± 0.0	0.0 ± 0.0
SP	100 ± 0.0	100 ± 0.0

Tables 1 and 2 show the results obtained by ACROAA and GA respectively from Zoo data set. Tables 3 and 4 show the results obtained by ACROA and GA respectively from Nursery data set. From these tables it can be observed that ACROA is competitive with the GA, a well known evolutionary computation technique. Predictive accuracies obtained by ACROA are higher than or equal to the accuracies obtained by GA. Table 5 shows the average performance of the methods. Although ACROA is a new technique, it has good performance in classification rule discovery field of data mining within the used data sets.

A 10-fold cross-validation process was also run for each data set to evaluate the quality of the rules discovered by two methods. The only selected quality measure is predictive accuracy. Tables 6 and 7 show the results for the Zoo and Nursery data set, respectively. When all results with respect to the predictive accuracy completely considered, ACROA has somewhat better performance than GA.

4. 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. With the chaotic properties of ACROA operators, being trapped in local solutions has been prevented.

ACROA has been designed for classification rule discovery field of data mining. Efficiency of this new algorithm in the selected problem has been demonstrated. If all reactants are used for the solution, they can be used for multi-objective optimization which is a desired property in data mining where mined rules are wanted to be interesting, comprehensible, accurate, and etc.

The algorithm seems to be a potential effective search and optimization method for different types of problems. Especially, it can easily be designed for a search method for clustering rules mining, sequential pattern mining, and other task of data mining.

Acknowledgements

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

References

Aladag, C. H., & Köksoy, O. (2011). A tabu search meta-heuristic approach to the dual response systems problem. *Expert Systems with Applications, 38*(12), 15370–15376.

Alatas, B. (2010). Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications, 37*(8), 5682–5687.

Alatas, B. (2011). Uniform big bang-chaotic big crunch optimization. *Communications in Nonlinear Science and Numerical Simulation, 16*(9), 3696–3703.

Alatas, B. (2011). ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Systems with Applications, 38*(10), 13170–13180.

Alatas, B., & Akin, E. (2005). Mining fuzzy classification rules using an artificial immune system with boosting. *Lecture Notes in Computer Science Springer-Verlag, 3631*, 283–293.

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

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

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

Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. *IEEE Congress on Evolutionary Computation, 7*, 4661–4666.

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.), *Swarm intelligence: Focus on ant and particle swarm optimization* (pp. 113–144). I-tech Education and Publishing.

Behnamian, J., & Zandieh, M. (2011). A discrete colonial competitive for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties. *Expert Systems with Applications, 38*(12), 14490–14498.

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

Blake, C., Merz, C. J. (1998). UCI Repository of machine learning databases <www.ics.uci.edu/~mlearn/MLRepository.html>.

Borji, A. (2007). A new global optimization algorithm inspired by parliamentary political competitions. *Lecture Notes in Computer Science Springer-Verlag, 4827*, 61–71.

Bozorg, H. O., Afshar, A., Mariño, M. A. (2005). Honey bees mating optimization algorithm (HBMO): A new heuristic approach for engineering optimization. In *1st international conference on modeling, simulation & applied, optimization*.

Chatzis, S. P., & Koukas, S. (2011). Numerical optimization using synergetic swarms of foraging bacterial populations. *Expert Systems with Applications, 38*(12), 15332–15343.

Chen, H., Zhu, Y., & Hu, K. (2009). Cooperative bacterial foraging optimization. *Discrete Dynamics in Nature and Society, 2009*, 1–17.

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

de Castro, L. N., & Timmis, J. I. (2002). *Artificial immune systems: A new computational intelligence approach*. London: Springer-Verlag.

de Jong, K. (2006). *Evolutionary computation: A unified approach*. MIT Press.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Stützle: MIT Press.

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

Formato, R. A. (2007). Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Progress in Electromagnetics Research, PIER, 77*, 425–491.

Glover, F. (1989). Tabu search – Part 1. OSRA. *Journal on Computing, 1*(2), 190–206.

Green, R. C., Wang, L., & Alam, M. (2012). Training neural networks using central force optimization and particle swarm optimization: Insights and comparisons. *Expert Systems with Applications, 39*(1), 555–563.

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 Co, 12*(1), 43–52.

Haykin, S. (1999). *Neural networks*. New York: Macmillan.

Horng, M.-H., & Liou, R.-J. (2011). Multilevel minimum cross entropy threshold selection based on the firefly algorithm. *Expert Systems with Applications, 38*(12), 14805–14811.

Jamili, A., Shafia, M. A., & Tavakkoli-Moghaddam, R. (2011). A hybridization of simulated annealing and eectromagnetism-like mechanism for a periodic job shop scheduling problem. *Expert Systems with Applications, 38*(5), 5895–5901.

Karci, A. (2007). Theory of saplings growing up algorithm. *Lecture Notes in Computer Science Springer-Verlag, 4431*, 450–460.

Karci, A., & Alatas, B. (2006). Thinking capability of saplings growing up algorithm. *Lecture Notes in Computer Science Springer-Verlag, 4224*, 386–393.

Karci, A., & Arslan, A. (2002). Uniform Population in Genetic algorithms, I.U.. *Journal of Electrical & Electronics, 2*(2), 495–504.

Karci, A., Alatas, B., Akin, E. (2006). Saplings growing up algorithm (Turkish). *ASYU-INISTA*, pp. 57–61.

Kaveh, A., & Laknejadi, K. (2011). A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization. *Expert Systems with Applications, 38*(12), 15475–15488.

Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: Charged system search. *Acta Mechanica, 213*(3–4), 267–289.

Kennedy, J., Eberhart, R. C. (1995). Particle swarm optimization. In *IEEE conf. on neural networks IV* (pp. 1942–1948).

Krishnanand, K. N., & Ghose, D. (2006). Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. *Multi-agent and Grid Systems, 2*(3), 209–222.

Lam, A. Y. S., & Li, V. O. K. (2010). Chemical-reaction-inspired metaheuristic for optimization. *IEEE Transactions on Evolutionary Computation, 14*(3), 381–399.

Laurentys, C. A., Palhares, R. M., & Caminhas, W. M. (2010). Design of an artificial immune system based on Danger Model for fault detection. *Expert Systems with Applications, 37*(7), 5145–5152.

Li X. L. (2003). New ıntelligent optimization-artificial fish swarm algorithm. PhD thesis, Zhejiang University, China.

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

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

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

Pradhan, P. M., & Panda, G. (2012). Solving multiobjective problems using cat swarm optimization. *Expert Systems with Applications, 39*(3), 2956–2964.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2012). Teaching-learning-based optimization: An optimization method for continuous non-linear large scale problems. *Inform Sciences, 183*(1), 1–15.

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Inform Sciences, 179*(13), 2232–2248.

Rupérez, M. J., Martín-Guerrero, J. D., Monserrat, C., & Alcañiz, M. (2012). Artificial neural networks for predicting dorsal pressures on the foot surface while walking. *Expert Systems with Applications, 39*(5), 5349–5357.

Sacco, W. F., de Oliveira, C. R. E. (2005). A new stochastic optimization algorithm based on a particle collision metaheuristic. In *6th world congresses of, structural and multidisciplinary optimization*.

Shah-Hosseini, H. (2008). The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm. *International Journal of Bio-Inspired Computation, 1*(1/2), 71–79.

Sywerda, G. (1989). Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms, USA*.

Wu, B., Qian, C., Ni, W., & Fan, S. (2012). The improvement of glowworm swarm for continuous optimization problems. *Expert Systems with Applications, 39*(7), 6335–6342.

Xiao, J., & Ping Li, L. (2011). A hybrid ant colony optimization for continuous domains. *Expert Systems with Applications, 38*(9), 11072–11077.

Xie, L., Zeng, J., & Cui, Z. (2009). General framework of artificial physics optimization algorithm. *IEEE Nature & Biologically Inspired Computing*, 1321–1326.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. Stochastic algorithms: Foundations and applications. *Lecture Notes in Comput Science, 5792*, 169–178.

Yin, M., Hu, Y., Yang, F., Li, X., & Gu, W. (2011). A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering. *Expert Systems with Applications, 38*(8), 9319–9324.