Applied Soft Computing 10 (2010) 1284–1292

Contents lists available at ScienceDirect

Applied Soft Computing

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

ARO: A new model-free optimization algorithm inspired from asexual reproduction

Alireza Farasat[a], Mohammad B. Menhaj[b], Taha Mansouri[c], Mohammad Reza Sadeghi Moghadam[d,*]

[a] *Department of Operation Research, University of Tehran, Tehran, Iran*
[b] *Department of Electrical Engineering, Amirkabir University, Tehran, Iran*
[c] *Department of Information Technology, University of Allame Tabatabaee, Tehran, Iran*
[d] *Department of production and Operation Management, University of Tehran, Tehran, Iran*

ARTICLE INFO

Article history:
Received 2 July 2009
Received in revised form 28 April 2010
Accepted 12 May 2010
Available online 20 May 2010

Keywords:
Model-free optimization
Asexual reproduction
Bio-inspired algorithms
Evolutionary computations
Particle swarm optimization

ABSTRACT

This paper proposes a new individual based optimization algorithm, which is inspired from asexual reproduction known as a remarkable biological phenomenon, called as Asexual Reproduction Optimization (ARO). ARO can be essentially considered as an evolutionary based algorithm that mathematically models the *budding* mechanism of asexual reproduction. In ARO, each individual produces an offspring called *bud* through a reproduction mechanism; thereafter parent and its offspring compete according to a performance index obtained from the underlying objective function of the given optimization problem. This process leads to the fitter individual. ARO's adaptive search ability and its strong and weak points are described in this paper. Furthermore, the ARO convergence to the global optimum is mathematically analyzed. To approve the effectiveness of the ARO performance, it is tested with several benchmark functions frequently used in the area of optimization. Finally, the ARO performance is statistically compared with that of Particle Swarm Optimization (PSO). Results of simulation illustrate that ARO remarkably outperforms PSO.

1. Introduction

In today's dynamic environment, decision making problems have emerged more complicated and they are limited by available information, available time, and the information-processing ability of the decision maker's mind; therefore, to make more appropriate and yet accurate decisions they better be modeled as optimization problems. As a result, a lot of methods have been proposed to tackle such problems. Among these developed algorithms, meta-heuristic optimization techniques have attracted increasing attentions in recent years for solving complex decision making problems. They are more robust than traditional methods based on formal logics or mathematical programming for many real world applications [1]. These techniques have obviously progressed and new ideas in the area of optimization algorithms have been developed during recent decades [63]. Although different meta-heuristic algorithms applied for various kinds of real world optimization exist, bio-inspired algorithms apparently stand out in comparison with others and construct a main category of meta-heuristic algorithms.

Biologically-inspired computation rapidly advanced over the past decade is a covering term for different computational methods that are based on principles or models of biological systems. This class of methods such as evolutionary algorithms (EAs), ant colony optimization (ACO), particle swarm optimization (PSO), artificial neural networks (ANNs), artificial immune systems (AISs) complements traditional techniques in the sense that the former can be applied to large-scale applications where little is known about the underlying problem and where the latter approaches encounter difficulties such as the risk of trapping in local minima. Therefore, bio-inspired methods are becoming increasingly important in face of the complexity of today's demanding applications, and accordingly they have successfully been used in various fields.

ACO algorithms introduced by Dorigo have been inspired by the behavior of real ants specially foraging. Moreover, they have successfully been applied to variety of problems, ranging from the classical traveling salesman problem (TSP) to routing in telecommunications network [2]. In addition, PSO developed by Kennedy and Eberhart [3] is a population based optimization algorithm modeling the exploration of a problem space by individuals called particles. Like ACO, PSO is also implemented for a number of optimization problems [4]. It should be noted that both ACO and PSO are considered as swarm intelligence. ANNs modeling the essence of the human neural system have been discussed for more than few decades [5]. Though extensively applied to various disciplines, ANNs have also been used for solving certain optimization

[*] Corresponding author.
E-mail address: REZASADEGHI@UT.AC.IR (M.R.S. Moghadam).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.05.011

problems [6]. More recently, AISs as adaptive systems inspired by immunology have emerged to tackle optimization problems [7].

The widespread term Evolutionary Computations (ECs), comprise a number of techniques modeling different aspects of evolution in the nature. Rechenberg first of all introduced Evolution Strategies (ESs) started with a population of two individuals, one parent and one offspring, the offspring being a mutated version of the parent [8,9]. This idea was further developed by Schwefel [10,11]. Fogel and his colleagues proposed Evolutionary Programming (EP), a technique in which candidate solutions to given tasks were represented as finite state machines [12]. Holland at the same time invented and developed Genetic Algorithms (GAs) casting a number of methods for moving from one population of chromosomes to a new population by using a kind of "natural selection" together with the genetics inspired operators of crossover, mutation, and inversion [13]. In contrast with ESs and EP, Holland's original goal was not to design algorithms to solve specific problems, but rather to formally study the phenomenon of adaptation as it takes place in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems [14]. Furthermore, Genetic Programming (GP), a form of the genetic algorithm to evolve Lisp programs to perform a variety of tasks was suggested by Koza [15].

In the recent years, various evolutionary models based on aforementioned approaches have been developed for multifarious applications [16]. Various genetic operators have been recommended in order to improve GAs performance for real world applications from the beginning of GA arising [17]. Another extention of canonical GAs named Co-evolution Genetic Algorithm (CGA) models multiple interacting subpopulations. Two different types of coevolution were simulated including compatitive coevolution which models the competition among subpopulations [18–22] and cooperative coevolution which evolves subcomponents that must cooperate in solving a given task, instead of evolving whole individuals [23–27]. An additional evolutionary based algorithm, the so-called Differential Evolution (DE) has extensively been used. DE is a simple yet efficient, direct-search algorithm for global optimization defined on totally ordered spaces, comprising spacially functions with real-valued parameters. It is also compared with other algorithms [28,29,64]. Furthermore, Memetic Algorithms (MAs) as a broad class of metaheuristics which combines a population-based search with other possible methods to employ all available knowledge about the problem [30] have fruitfully been applied in various optimization problems.

In comparison with bio-inspired algorithms, other metaheuristic optimization methods remarkably implemented for hard optimization problems were developed. Some of the best known and most widely applied techniques not related to biological systems are simulated annealing (SA) [31], tabu search (TS) [32,33], guided local search (GLS) [34], greedy randomized adaptive search procedures (GRASP) [35] and iterated local search (ILS) [36]. All of the aforementioned methods have in common that they strive to evade the generation of poor quality solutions by applying general mechanisms that extend problem-specific, single run algorithms [2].

All of abovementioned algorithms possess both advantages and disadvantages which are comparatively associated with the number of elements participating in the algorithm construction. Regardless to inspire by biological systems or not, some algorithms are population based while some others are individual based. Because of the number of individuals, population based algorithms (like GAs, ACO, PSO, etc.) are potentially able to escape from local optima though they impose additional computation costs. In other words, population based algorithms beget to increase run times since they make troubles in real time applications. In contrast with population based algorithms, individual based algorithms (i.e. SA, TS, etc.) are quick enough to implement in real time, however they do not guarantee to find the optimum solution and they probably trap in the local optimum. Although it is proved to converge to the global optimum for some of these algorithms (like SA), they require a prolonged time to converge. Another problem that all algorithms deal with is parameters tuning because most of them have a number of parameters to be set.

As a result, we propose a model free optimization algorithm inspired by one of the astounding biological phenomenon, asexual reproduction; hence we entitle it Asexual Reproduction Optimization (ARO) algorithm. ARO is an individual based algorithm which intelligently guides the search process and it can reach to the global optimum in an astonishing time possessing advantages of both population and individual based algorithms. It meanwhile escapes from local optima by adaptively exploration and exploitation as inspired by the biological model.

The rest of the paper has been organized as follows. Section 2 is assigned to biological concepts and foundations of the asexual reproduction. ARO is scrutinized in Section 3 and its convergence is analyzed in Section 4. In Section 5, we examine the ARO performance in the optimization of benchmark functions. In Section 6, ARO is compared with other frequently used optimization algorithms. Finally, Section 7 concludes the paper.

2. Biological foundations of ARO

2.1. Asexual reproduction

As stated above, ARO is inspired by asexual reproduction. Asexual reproduction is a method of reproduction where a 1N (chromosome number) cell produces two to four cells with the same chromosome number. This can be done by binary fission of a motile stage or a non-motile stage. In many armoured dinokonts the original cell divides along predetermined sutures and then each half produces a new half with new thin plates [37]. Asexual reproduction involves only one parent passing on the genetic information to their offspring. This sharing of genetic information makes the offspring identical to the parent [38].

Many species reproduce successfully in the absence of sex [39]. Asexual reproduction is common among organisms like bacteria, rapidly reproducing to generate large populations. In these large populations, mutation can provide considerable genetic variations, so sex may be less important in producing genetic diversity within the population [40]. Although they do not sexually reproduce, there are ways in which genes get transferred between asexual individuals [41]. Some complex organisms, such as plants and some lizards, may reproduce asexually.

There are different types of asexual reproduction like "binary fission" [42], "asexual spore production" [43], "plants asexual reproduction" [44] and "budding" [45]. In the binary fission only single-icelled organisms reproduce. The cell duplicates its contents, including its nucleus and other organelles and then splits into two cells with each one being identical (bacteria, amoeba, algae) [46]. In Asexual Spore Production, spores are similar to seeds, but are produced by the division of cells on the parent, not by the union of two cells. One parent may produce many spores; each of this will grow into a new individual, identical to its parent (fungi, green algae, moulds, ferns). Many spores are produced to ensure that at least some of the individual organisms will survive. Zoospores can also be produced by some fungi and green algae. They move using tail-like flagella [43].

A plant continues to grow throughout its life. The rapidly growing tips of roots and stems contain specialized reproductive cells called meristem. At a certain time these cells will be diverted into cells that make up roots, stems and leaves. If parts of the plant are damaged, the meristem cells make repairs. Clones can be made from cuttings of a plant, because the meristem cells can specialize to reproduce the different parts needed to make a new plant. Asexual reproduction can produce many plants very quickly. This is an advantage in places where the environment doesn't change very much (bacteria). By building a large population of organisms very quickly the species is able to thrive. The great disadvantage is that when the environment changes, all of the organisms will die, if they do not have the ability to adapt to the change [44].

Eventually in the budding mathematically modeled by ARO, the parent organism produces a bud (a smaller version of itself), which eventually detaches itself from the parent and becomes a self-sufficient individual—identical to the parent. Coral also reproduces in this way, but do not detach themselves (hydra, yeast, coral, sea sponge) [47].

Asexual reproduction is a significant adaptation to specific environments and biological conditions where the cost of sexual reproduction to a species is considerable [48,65]. Asexuality can be seen as the reverse side of the coin "why sex?", which has been questioned among scientists over the history of biology [49]. According to Ghiselin [49], the question should be directed primarily to the genetic consequences of sex. Awareness of the genetic consequences of reproductive modes forms the basis of the background to the evolutionary approach.

2.2. Asexual versus sexual reproduction

Sexual reproduction usually involves two individual organisms. The offspring that are produced from this union have a mix of characteristics, half from one parent and the other half from the other parent. It should be considered that most of evolutionary algorithms model sexual reproduction. Sexual reproduction does not always involve male and female parents, however they can have specialized *gametes* (reproductive cells that have only one role—to join with another gamete during reproduction). Many organisms are capable of both sexual and asexual reproduction, like some moulds, such as *Rhizopus*, which produce spores. They can also produce *zygospores*, enabling them to reproduce sexually as well.

Sexual reproduction has the advantage of providing lots of variations within a species, helping it to survive when the environment changes. The main disadvantage is that this process takes a lot of energy. This means that they can only produce small populations.

There has been considerable discussion of the advantages of sexual and asexual reproduction in the life history of a species (e.g. [50,51]). In general, sexual propagules are widely dispersed during periods of environmental change or uncertainty, while asexual reproduction occurs close to the parent during more benign periods [50]. In addition, asexual reproduction often occurs in physically disturbed or isolated habitats where a single asexual individual can establish a new colony. These habitats also harbor relatively few competing, predatory or pathogenic species [51–53]. Sex and genetic recombination, in contrast, may be necessary adaptations of a species living in a biologically complex environment, where there is a continuous appearance of qualitatively new selective forces [51]. This biological complexity may be best developed in physically uniform environments [51,54]. No single model accommodates these various correlations and ideas [50,51]. The relation between asexual reproduction and the physical and biological environment is poorly understood in marine animals. There are few comparative studies of asexual populations within a single species or of asexual reproduction in closely related species (e.g. [55,56]). In particular, asexual budding, fission, and fragmentation among the large group of polychaete worm's show little pattern of geographical occurrence [57].

Selective advantage of asexual reproduction is that the organism can reproduce without finding a partner. The disadvantage of strictly asexual reproduction is that genetic variability decreases, and the population is less able to survive changes in the environment. Some species use the advantages of both asexual and sexual reproduction. In these species, most reproduction is asexual, or vegetative, but occasionally, often during periods of environmental stress, sexual reproduction occurs.

Asexual reproduction yields clones, offspring completely identical to the parent. If a starfish is cut into five pieces, each arm can regenerate an entire starfish. Although a few animals can regenerate in this fashion, in general plants are better candidates for cloning. Tissues of many plants retain totipotency, the ability to regenerate the entire plant structure. A single cell from a carrot root can divide and differentiate to form an embryo and then an adult plant. Even humans have cells that are totipotent early in their development. It is these totipotent cells that surround the controversy of stem cells.

Because the offspring are identical to the parent, asexual reproduction is often used to propagate desirable horticultural varieties. Many plants reproduce asexually under natural conditions. Extensive stands of bracken fern or entire forests of alder may be genetically identical; essentially one huge plant that has spread over a large area. Like microorganisms, plants make good subjects for the investigation of asexual reproduction.

3. ARO algorithm description

Before ARO algorithm description, all symbols used in the paper, are listed in Table 1.

3.1. Main structure of ARO

As discussed above, there are different types of asexual reproduction such as binary composition, asexual reproduction of spore, asexual reproduction of plants and budding process. The main goal here is to formulate an algorithm to model the budding mechanism. In the proposed algorithm, called as ARO, each individual is represented by a binary string. On the other hands, a solution of the given problem, e.g. $X = (x_1, x_2, \ldots, x_n)$; $X \in \Re^n$, is in fact a parent with some chromosomes composed of several bits called genes. A chromosome with the length of L is made of the following bits:

the first bit represents the sign,
the next l_1 bits represent the integer part of the chromosome,
the final l_2 bits represent the decimal part of the chromosome.

Therefore, $L = l_1 + l_2 + 1$, and the total length of the parent is n^*L. Fig. 1 shows a typical ARO chromosome.

It is assumed that each solution in the search space (S) is indeed an organism in its own environment. It is further assumed that the existing resources in the environment are limited; hence, merely the deserving individual can survive. To start the algorithm, an individual is first generated randomly in the distinctive domain of S, then the offspring (*bud*) is produced through a mechanism called reproduction which will be fully discussed later. The parent and its offspring compete to survive according to a performance index or a fitness function. If the parent wins the competition; the offspring

Table 1
Some important notations used in the paper.

Symbol	Description	Symbol	Description
S	Search space	K	Maximum iterate number
L	Length of chromosome	Ω	State space of X
N	Number of variables	Φ	State space of Y
G	Number of selected bits	X^*	Optimal solution
X_{k_t}	t-th bud produced by k-th parent	Y_k	k-th parent
$P_{Y_k X_{k_m}}$	Transforming probability	$Pa_{X_{k-1_m} Y_k}$	Acceptance probability

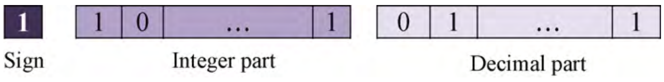

Fig. 1. A typical ARO chromosome.

will be discarded totally and vice versa. The algorithm repeats steps described in Table 1 until the stopping criteria are met.

It is clear that choosing an appropriate reproduction operator is critical. Despite the fact that many of evolutionary algorithms apply multiple operators in order to search in the search space, ARO uses only one operator to exploit available information according to the conventional control theory. In order to reproduce an offspring, a substring of length g bits uniformly distributed in [1,L], g~Uniform [1, L], is randomly selected for each chromosome. Afterwards 1's in this segment of bits are replaced by 0 and vice versa; i.e., the bits in the substring mutate. Table 2 illustrates ARO pseudo code.

In fact, each substring (called *Larva*) which is a mutated form of its parent is consequently selected. According to the optimization theory, even though both exploration and exploitation mechanisms are necessary, mutated form of its parent, only does larva explore the search space. To develop the algorithm capability, an exploitation mechanism is also added to it in which the parent and its offspring can share their information by *merging*. In other words, during the mutation process, the crossover implicitly takes place. Fig. 2 illustrates the reproduction mechanism.

After reproduction, the bud fitness is calculated according to the performance index. As shown in Table 1, bud fitness is compared with that of parent. At last, the most deserved individual is permitted to reproduce.

3.2. ARO adaptive search ability

In the first ARO simulation, the reproduction operator was implemented as follows. After larva was produced by its parent, a random number uniformly distributed in [0,1] is generated. If this number is less than 0.5, the bit will be selected from the parent otherwise it will be chosen from larva till bud is completed. It means that merging is definitely performed. The number of bits which alters in the substring, g, is a random number. When g is large, more exploration is expected and vice versa, while the exploitation applied is done based on the aforementioned procedure; this means that the amount of exploration is merely controlled by g. Similarly exploitation rate is handled by the probability of merging the larva with its parent. Consequently, we may employ a procedure to adaptively set the rate of exploration and exploitation. As a consequence, when the rate of exploration, g, is large, the exploitation rate decreases and vice versa. To achieve this goal, a function, p, calculating the probability of merging is defined. Therefore, ARO can adaptively shrink or widen the search space.

$$p = \frac{1}{1 + ln(g)} \tag{1}$$

It is obvious that when g increases, p decreases and vice versa.

3.3. ARO strong and weak points

ARO has several advantages which makes it different from other meta-heuristic optimization algorithms. At first, as described above, ARO is an individual-based algorithm; therefore in spite of other population-based algorithms which take a lot of time to evolve, ARO utilizes a little energy; consequently due to its fast convergence it can be a very good candidate for online (real-time) optimization problems in which there is not enough importation about the systems under study. This property of ARO makes it very popular for real word problems especially for real time control and decision making [58], signal processing, data mining, etc.

Secondly, knowing the fact that the recently developed algorithms like GAs, PSO, CGA, TS and SA might face with some troubles due to lack of optimum parameters setting, ARO does not require any parameter tuning. For example in PSO, if its main parameters are not well set, it cannot appropriately work [59].

Furthermore, no selection mechanism is required in ARO. Selection mechanisms have been fully discussed for more than two decades [14]. Choice of inappropriate selection mechanisms (e.g. Roulette-Wheel), leads to some problems like the selective pressure and the premature convergence.

Finally, in contrast with other bio-inspired algorithms which are applicable for some special optimization problems like ACO that is

Table 2
ARO pseudo code.

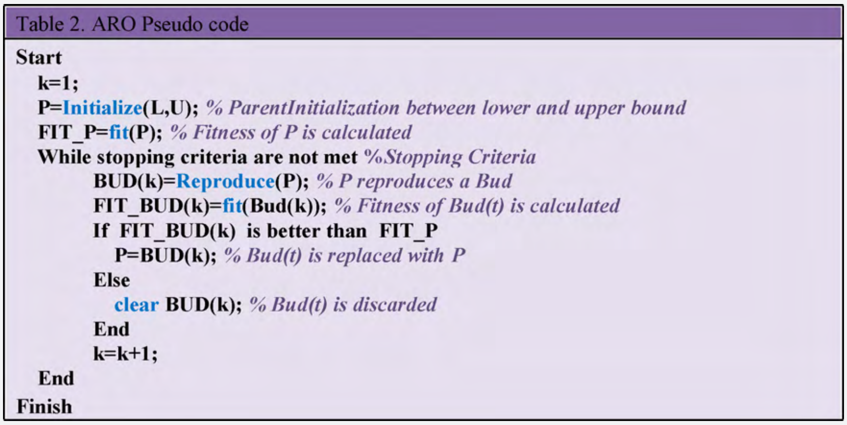

Table 2. ARO Pseudo code

```
Start
  k=1;
  P=Initialize(L,U); % ParentInitialization between lower and upper bound
  FIT_P=fit(P); % Fitness of P is calculated
  While stopping criteria are not met %Stopping Criteria
      BUD(k)=Reproduce(P); % P reproduces a Bud
      FIT_BUD(k)=fit(Bud(k)); % Fitness of Bud(t) is calculated
      If FIT_BUD(k) is better than FIT_P
        P=BUD(k); % Bud(t) is replaced with P
      Else
        clear BUD(k); % Bud(t) is discarded
      End
      k=k+1;
  End
Finish
```

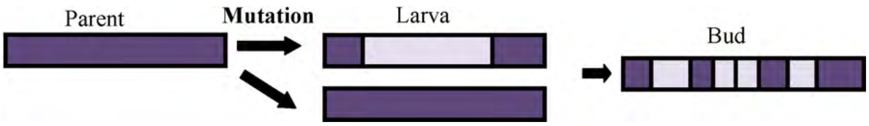

Fig. 2. The ARO reproduction mechanism.

limited to discrete optimization [2], ARO is a model-free algorithm able to be applied for a range of optimization and decision making problems. Therefore, it can be utilized for graph optimization, integer and combinatorial optimization, constrained optimization and a lot more.

On the contrary, a predictable weak point of ARO is possibly its adaptation with the turbulent environment; hence genetic diversity becomes low as it is indeed in its corresponding biological model. Even if the aforementioned problem exists, it can be resolved by applying particular reproduction operators as we did to explore throughout the search space.

4. Real representation version of ARO

Representation methods, including binary and real encodings, in evolutionary algorithms have frequently argued during the past decades. Binary encodings (i.e., bit strings) are the most common encodings for a number of reasons and they have several advantages [14]. In spite of these advantages, binary encodings are censured for many problems (i.e. hamming cliff) discussed in evolutionary computation literatures [60]. In contrast with the binary representation, real encoding, mostly used in recent years, seems more appropriate for some applications [14].

Like other evolutionary algorithms, we utilized both binary and real representation in ARO. In the real encoding of ARO, each variable is represented by a string containing both integer and decimal parts. As a consequence, a bunch of reproduction operators, corresponding to the real encoding, is used and other steps of the algorithm are the same. When the real encoding is used, more mathematical operators for merging are accordingly applicable. For example, for two real numbers x and y as a parent and larva, a linear combination can be applied such as $\lambda x+(1-\lambda)y$.

Fig. 3 illustrates that different types of the arithmetic merging mechanism may lead to reproduce a bud somewhere in an ellipse as shown in the above figure. This claim has been approved through several different simulation runs.

5. ARO convergence analysis

In this section, the ARO convergence is analyzed. Firstly *state* and *state space* are described:

5.1. State

In this paper, a state is indeed a situation that the algorithm can be. When ARO starts the search from an initial parent denoted by X_0, two sequences such as $\{X_0, X_1, X_2, \ldots\}$ and $\{Y_0, Y_1, Y_2, \ldots\}$ will be generated. As stated before, ARO starts with an initial parent and frequently generates new buds using reproduction operator until one of these buds excels the parent. In order to acquire a better objective function value, the bud is replaced with its parent thereby creating a new parent. In the search process, a finite sequence $\{X_{k_1}, X_{k_2}, \ldots, X_{k_m}\}$ is used for all buds generated by exploiting the k-th parent. It means that X_{k_t} is the t-th bud produced by using the k-th parent. Similarly, in the finite sequence $\{Y_0, Y_1, \ldots, Y_K\}$, Y_k is the k-th parent generated by the algorithm.

5.2. State space

State space of random variables $\{X_{k_t}, k_t \in K_T\}$ includes all possible values for X_{k_t} denoted by Ω. A state space of random variables $\left\{Y_k, k \in K\right\}$ indicated by Φ is represented by all the possible values of Y_k.

We consider three conditions that will be shown to be sufficient to guarantee the ARO convergence.

C1: We assume that the state space S is a bounded and closed subset of $\Re^n$ and there exists an X^* such that $\forall X \in S \Rightarrow f(X^*) \le f(X)$.
C2: For every $\varepsilon > 0$, the set $S_\varepsilon = \{X \in S : \left|f^* - f(X)\right| < \varepsilon\}$ is defined as the optimum set and members of this set symbolized by X_ε which is an optimal set with ε value of error. When finding an $X \in X_\varepsilon$, the algorithm stops.
C3: In the proposed algorithm, each offspring is reproduced by using a function H entitled *reproduction* function, i.e. $X_{k_{t+1}} = H(Y_k)$. Reproduction function is delineated by mutation and merging. This function is continues and its image $\Re_H$ is a sub set of S, $\Re_H \subseteq S$. H is able to produce all points of S and adaptively explore and exploit. If the algorithm is stuck in a local optimum, H increases exploration to expand the search space and narrows it by exploitation when needed.

To analyze the ARO convergence, some lemmas are required which are described as follows.

Lemma 1. *The stochastic sequence* $\left\{f(Y_k), k \in K\right\}$ *is a non-increasing sequence.*

Proof. In each iteration of the algorithm, Y_k is calculated as:

$$Y_k = \begin{cases} X_{k-1_t} & \text{if } f(X_{k-1_t}) < f(Y_{k-1}) \\ Y_{k-1} & o.w \end{cases} \tag{2}$$

Thus, it is obvious that:

$$\forall k \in K \Rightarrow f(Y_k) \le f(Y_{k-1})$$

Hence, the sequence $\left\{f(Y_k), k \in K\right\}$ is a non-increasing sequence and the proof of Lemma 1 is completed. It is required to show whether the sequence $\{X_{k_t}, k_t \in K_T\}$ is finite or not. The following lemma deals with this question and states that the number of required iterations to achieve a local optimum is limited.□

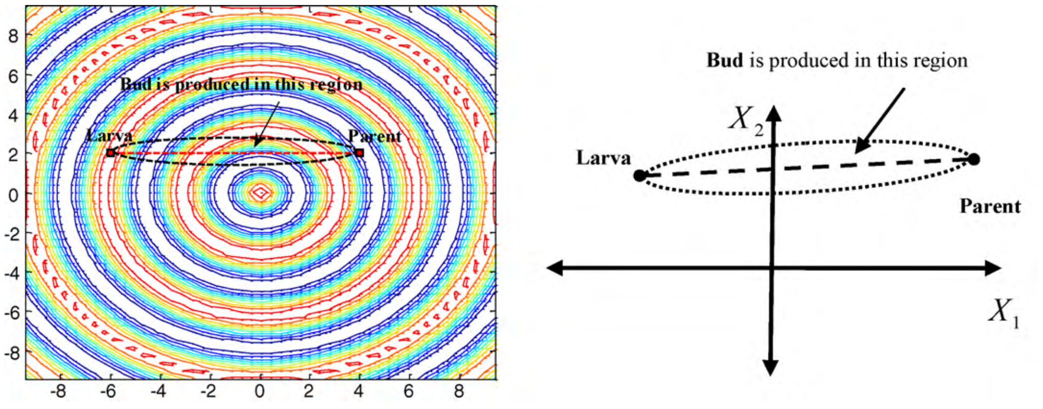

Fig. 3. Region in which bud may be generated by using different reproduction operators.

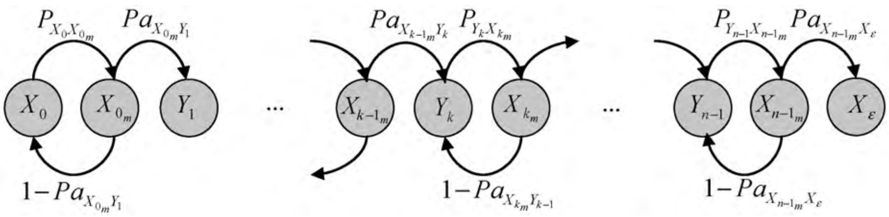

Fig. 4. Markov chain of states generated by ARO.

Lemma 2. *For each $Y_k \notin S_\varepsilon$, the sequence $\left\{X_{k_t}, k_t \in K_T\right\}$ will be a finite stochastic sequence*

Proof. To prove Lemma 3, we employ reductio ad absurdum. Suppose in order to reach to a parent better than Y_k, infinite numbers of X_{k_t} should be generated. Because the sequence $\left\{X_{k_t}, k_t \in K_T\right\}$ is infinite, according to *C1* and *C3* there is no Y_{k+1} in S which excels Y_k. Consequently, Y_k must be in S_ε; this contradicts the assumption of Lemma 2, therefore the sequence $\left\{X_{k_t}, k_t \in K_T\right\}$ is finite.□

Lemma 3. *The sequence of stochastic variables $\left\{Y_k, k \in K\right\}$ is a stochastic sequence modeled by Markov chain.*

Proof. According to the chromosome length in the binary ARO, we can construct hyper-cubes mapping domain of the objective function, S so that by increasing the chromosome length, the mapping is more accurate (it should be considered that this mapping is as well plausible in real representation because each real number is corresponding to a binary number). In fact, points of each hypercube map solutions in the main domain of the function. As a result, by connecting these points to each other a fully connected, undirected graph is created so that points are graph nodes and connections are graph edges. Now we can begin a *special* random walk on this graph in which a node is randomly selected to move. Therefore, if we do not situate in a state with lower energy level, we'll return to the previous situation. Hence, by starting from $Y_0 = \omega_0$, $\left\{Y_k, k \in K\right\}$ a stochastic sequence is created. To demonstrate $\left\{Y_k, k \in K\right\}$ is a Markov process, we firstly define $P_{\omega_{k+1}|\omega_t}$ describing the probability of $Y_{k+1} = \omega_{k+1}$ given to $Y_{k1} = \omega_k$:

$$P_{\omega_{k+1}|\omega_t} = P\left\{Y_{k+1} = \omega_{t+1} | Y_k = \omega_t\right\} \tag{3}$$

Additionally, as cited before, the reproduction operator in the algorithm is constructed by mutation and merging. If we assume two predetermined operators (i.e. uniform random mutation and arithmetic merging in real representation) are utilized in the algorithm, we have:

$$X'_{k_{t+1}} = Y_k + \gamma;\ \gamma \sim U[-a, a] \tag{4}$$

$$\begin{aligned} X_{k_{t+1}} &= \alpha X'_{k_{t+1}} + (1-\alpha)Y_k = \alpha[Y_k + \gamma] + (1-\alpha)Y_k \Rightarrow X_{k_{t+1}} \\ &= Y_k + \alpha\gamma \end{aligned} \tag{5}$$

$$Y_{k+1} = \begin{cases} X_{k_{t+1}} & \mathit{if}\, f(X_{k_{t+1}}) < f(Y_k) \\ Y_k & o.w \end{cases} \tag{6}$$

It should be noted that the assumptions, cited above, does not restrict the proof. However, the value of random variable Y_{k+1} merely depends on the value of Y_k and it is not related to previous situations. Therefore, as a result, the following Eq. (7) is obtained:

$$\begin{aligned} &P\left\{Y_{k+1} = \omega_{t+1} | Y_k = \omega_t, Y_{k-1} = \omega_{t-1}, \ldots, Y_0 = \omega_0\right\} \\ &\quad = P\left\{Y_{k+1} = \omega_{t+1} | Y_k = \omega_t\right\} \end{aligned} \tag{7}$$

Eq. (7) illustrates that the stochastic sequence $\left\{Y_k, k \in K\right\}$ is a Markov chain and this completes the proof of Lemma 3.□

Lemma 4. *For every $X_0 \in S$ and $\varepsilon > 0$ there exists an integer N such that for any $\delta > 0$:*

$$P\left\{Y_n \notin S_\varepsilon | X_0 = \omega_0\right\} < \delta \quad \forall n > N$$

Proof. To ascertain Lemma 4, it is adequate to demonstrate that N is a finite integer number. For that reason, a Markov chain as illustrated below is used:

In Fig. 4, the non-integrated Markov chain illustrates each state, which is defined above. There are two classes of transmission probability stated as:

$P_{Y_k X_{k_m}}$: The probability of transferring from the current state Y_k a perturbed state X_{k_m} (Probability of variation).
$Pa_{X_{k-1_m} Y_k}$: The acceptance probability of X_{k-1_m} as a new parent Y_k; this probability is non-zero when $f(X_{k-1_m}) < f(Y_{k-1})$ (Probability of acceptance).

With attention to the type of reproduction operator used in the binary representation, probability of variation is calculated as follows:

$$P_{Y_k X_{k_m}} = 1 - \left(\frac{1}{2}\right)^g \left[1 - \frac{1}{1 + ln(g)}\right] \forall g \leq g_{\max} \tag{8}$$

As stated, g denotes the number of mutated genes and $1 - (1/(1 + \ln(g)))$ indicates the probability of crossover. $(1/2)^g$ is related to the uniform merging we used and finally $g_{\max}$ represents the maximum length of a chromosome. Indeed, $(1/2)^g[1 - (1/(1 + \ln(g)))]$ denotes probability of not having any changes. If g enlarges to $g_{\max}$ then:

$$\lim_{g \to g_{\max}} P_{Y_k X_{k_m}} \to 1$$

Furthermore, acceptance probability of the new parent is equivalent to the following probability:

$$Pa_{X_{k-1_m} Y_k} = P\left\{f(X_{k-1_m}) < f(Y_{k-1})\right\} \tag{9}$$

As Lemma 2 states, $\left\{X_{k_t}, k_t \in K_T\right\}$ represents a finite stochastic sequence therefore there is a natural number M, so that for any $m \geq M$, $P\left\{f(X_{k-1_m}) < f(Y_{k-1})\right\} = 1$. This definitely implies that:

$$\lim_{t \to M} P\left\{f(X_{k-1_t}) < f(Y_{k-1})\right\} = 1 \tag{10}$$

Hence, after generating n parents $n \geq N$, a lower bound N exists so that:

$$P\left\{Y_n \notin S_\varepsilon | X_0 = \omega_0\right\} < \delta \quad \forall n > N$$

This completes the proof.□

Lemma 5. *In the aforementioned Markov chain, all Y_k and X_{k_m} represent transient states.*

Proof. To prove Lemma 5 η_i and $\eta_i^{(n)}$ are defined as follows:

$\eta_i^{(n)}$: Probability of the first return to the state i after passing n states starting from it.

$$\eta_i = \sum_{n=1}^{\infty} \eta_i^{(n)} \tag{11}$$

It is a well known fact that for in the case of $\eta_i = 1$, state i becomes recurrent and it becomes transient if $\eta_i < 1$ ([61] and [62]).

Now we proceed as follows.

For any state, Y_k we can easily have the following:

$$\begin{gathered}
\eta_{Y_k}^{(1)} = 0 \\
\eta_{Y_k}^{(2)} = P_{Y_k X_{km}}(1 - Pa_{X_{km} Y_{k+1}}) \\
\eta_{Y_k}^{(3)} = 0 \\
\cdot \\
\cdot \\
\eta_{Y_k}^{(n)} = \begin{cases} [P_{Y_k X_{km}}(1 - Pa_{X_{km} Y_{k+1}})]^n & n \in E \\ 0 & n \in O \end{cases} \\
\cdot \\
\cdot
\end{gathered} \tag{12}$$

And finally obtain:

$$\begin{aligned}
\eta_{Y_k} &= \sum_{n=1}^{\infty} \eta_{Y_k}^{(n)} = P_{Y_k X_{km}}(1 - Pa_{X_{km} Y_{k+1}}) + [P_{Y_k X_{km}}(1 - Pa_{X_{km} Y_{k+1}})]^2 \\
&\quad + \cdots + [P_{Y_k X_{km}}(1 - Pa_{X_{km} Y_{k+1}})]^n + \cdots
\end{aligned} \tag{13}$$

η_{Y_k} indicates a geometric series with $P_{Y_k X_{km}}(1 - Pa_{X_{km} Y_{k+1}}) < 1$. As a consequence, the sum of this series is equal to:

$$\eta_{Y_k} = \sum_{n=1}^{\infty} \eta_{Y_k}^{(n)} = \frac{P_{Y_k X_{km}}(1 - Pa_{X_{km} Y_{k+1}})}{1 - P_{Y_k X_{km}}(1 - Pa_{X_{km} Y_{k+1}})} \tag{14}$$

According to Lemma 4, $\lim\limits_{t \to M} Pa_{X_{km} Y_{k+1}} = 1$ and hence $\eta_{Y_k} = 0$ implying that Y_k is a transient state. For X_{km}, if $f(X_{km}) < f(Y_k)$ then it will be accepted as a new parent else it will be discarded; hence the return probability to this state in long term equals zero and the proof is completed.□

Lemma 6. *In the abovementioned Markov chain, X_ε is a recurrent state.*

Proof. Being situated in X_ε state, the process abandons its state with probability $P_{X_\varepsilon X_{\varepsilon m}}$, and there is no any state like X in S such that $X \notin S_\varepsilon$; hence $Pa_{X_{\varepsilon m} Y} = 0$. As a result when departing from X_ε with probability $P_{X_\varepsilon X_{\varepsilon m}}$, the process returns to X_ε with probability of one and Lemma 6 is easily done.

To conclude this section, the following theorem that theoretically demonstrates the ARO convergence to a global optimum in a finite time based on aforementioned lemmas is stated.□

Theorem. Let f be a function defined on a subset $S \subset \Re^n$ and $\left\{Y_1, Y_2, Y_3, \ldots\right\}$ be the sequence of states generated by ARO. Assume that conditions *C1* to *C3* hold. Then for every initial state X_0, the sequence of function values $\left\{f(Y_k), k \in K\right\}$ for finite values of K converges to f^*.

Proof. As stated by the above lemmas, if the algorithm begins from any random solution $X_0 \in S$, after a finite number of iterations as indicated by Lemma 2 and Lemma 4, it will get to X_ε and stays there forever due to Lemmas 5 and 6 stating that all sates are transient excluding X_ε which is an absorbing state. In other words, after a finite number of iterations the algorithm converges to X_ε. The proof of theorem is consequently accomplished.□

6. ARO simulation and comparison with PSO

In this section, ARO simulation results are reported. Indeed, ARO was applied to optimize a number of benchmark functions listed in Table 5. In order to test ARO in different situations, various functions are accordingly selected. We examined ARO for three different dimensions including dimension 2, 5 and 10. To satisfy central limit theorem for statistical analysis, ARO run 30 times for each function. To simulate, MATLAB 7.4 and a computer with following properties were used:

Core2Du with 2.8 GHz duration and 1 GB RAM.

Additionally, PSO (Kennedy & Eberhart's version) with the same abovementioned conditions was used to compare with ARO. Tables 3–5 illustrates ARO and PSO simulation results for dimensions 2, 5 and 10. In these tables, the error goal is the criterion selected for algorithm to stop. Furthermore, average and standard deviation of run times of each algorithm, both in seconds, are informed.

Table 3
Comparison between run time of ARO and PSO in dimension 2.

Function	Goal of function	ARO		PSO	
		Average	S.D.	Average	S.D.
1	10e-30	0.1373	0.064	1.1594	0.491
2	10e-30	0.1433	0.078	0.9201	0.377
3	10e-30	4.7674	4.681	3.325	3.522
4	10e-20	0.089	0.098	1.343	0.8179
5	10e-30	0.13956	0.192	0.205	0.075
6	10e-20	0.075	0.067	21.917	0.0627
7	10e-20	0.4726	0.4220	4.67843	7.814
8	10e-20	0.10356	0.064	0.216	0.0250
9	10e-30	0.03608	0.0444	0.142	0.013
10	−0.99999	0.8267	2.707	14.676	9.599

Table 4
Comparison between run time of ARO and PSO in dimension 5.

Function	Goal of function	ARO		PSO	
		Average	S.D.	Average	S.D.
1	10e-20	1.823	0.735	37.1835	14.13
2	10e-20	1.944	0.853	41.211	11.208
3	10e-20	39.5638	10.61	47.612	5.6222
4	10e-10	9.592	6.70	43.129	9.535
5	10e-20	2.571	2.575	2.0642	1.2397
6	10e-10	1.016	0.653	45.408	0.1050
7	10e-10	9.604	1.567	45.3458	0.081
8	10e-10	1.109	0.616	2.657	0.331
9	10e-10	4.273	6.16	8.6203	3.566
10	−0.999	19.0372	5.338	45.423	0.103

Table 5
Comparison between run time of ARO and PSO in dimension 10.

Function	Goal of function	ARO		PSO	
		Average	S.D.	Average	S.D.
1	10e-15	19.93	7.631	37.1835	14.13
2	10e-15	17.23	1.4307	85.520	0.586
3	10e-15	73.991	0.17	88.77	0.3157
4	10e-5	84.893	2.077	85.4955	0.2594
5	10e-15	9.405	8.014	55.537	6.945
6	10e-5	11.504	2.425	84.917	0.167
7	10e-5	59.95	1.23	84.956	0.263
8	10e-5	8.04	2.82	7.039	0.64
9	10e-5	10.779	1.02	9.608	0.9444
10	−0.99	49.754	0.174	84.963	0.217

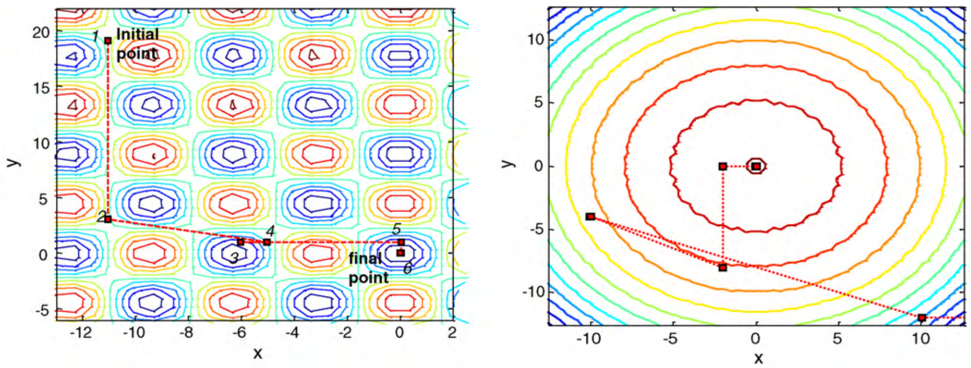

Fig. 5. Trajectory from the initial point to the optimal point for functions 9 and 10.

Results stated in Tables 3–5 demonstrate that according to acquired times, ARO considerably operates in real time applications when the dimension of the problem increases. In addition, contours of functions 9 and 10 are depicted in Fig. 4 illustrating the trajectory that the parent passes through search space to reach the global optimum.

7. Conclusion

In this paper, ARO inspired from asexual reproduction is proposed as a new individual based algorithm to tackle both optimization and decision-making issues. Although there are different asexual reproduction mechanisms such as budding, binary composition, asexual reproduction of spore, asexual reproduction of plants, the ARO models the budding mechanism. The proposed ARO has several advantages which make it very appropriate device to solve the real time decision-making problems. These advantages which facilitate the ARO to effectively explore the search space and exploit from available information include but are not limited to the fast convergence time, no parameter setting, being model-free and biological foundations of ARO. Moreover, both real and binary versions of ARO are presented in the paper. In addition, the ARO convergence is comprehensively analyzed and a theorem is presented that proves the algorithm converges to the global optimum in a limited time. To prove its performance, ARO is tested through several benchmark functions with 3 different dimensions. Simulation results approve the ARO's remarkable performance as compared with PSO. For the future, other abovementioned types of asexual reproduction mechanisms can be applied to develop a new model-free optimization algorithm. Additionally, ARO should be compared with other meta-heuristic optimization algorithms like GAs, CGAs, DE, etc. Finally, ARO can be used for real time applications such as neural networks training, control and filtering applications, combinatorial optimization problems, data mining and so on which some of them are currently under investigation and will be reported later (Fig. 5).

References

[1] R. Sarker, M. Mohamadian, X. Yao, Evolutionary Optimization, Kluwer Academic Publishers, 2003.

[2] M. Dorigo, T. Stützle, Ant Colony Optimization, Prentice-Hall of India, New Delhi, 2005.

[3] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE international conference on neural networks, vol. 4, IEEE Press, Piscataway, NJ, 1995, pp. 1942–1948.

[4] D.Y. Sha, C.Y. Hsu, A hybrid particle swarm optimization for job shop scheduling problem, Computers & Industrial Engineering 51 (2006) 791–808.

[5] L. Fausett, Fundamental of Neural Networks, Prentice-Hall, 1994.

[6] A. Agarwal, H. Pirkul, V.S. Jacob, Augmented neural networks for task scheduling, European Journal of Operational Research 151 (2003) 481–502.

[7] D. Dasgupta, Artificial Immune Systems and Their Applications, Springer, Berlin, 1999.

[8] I. Rechenberg, Cybernetic Solution Path of an Experimental Problem, Ministry of Aviation, Royal Aircraft Establishment, UK, 1965.

[9] I. Rechenberg, Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution, Frommann–Holzboog, Stuttgart, 1973.

[10] H.-P. Schwefel, Evolutionsstrategie und numerische Optimierung, Ph.D. thesis, Technische Universität, Berlin, 1975.

[11] H.-P Schwefel, Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie, Birkhäuser, Basel, 1977.

[12] L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial Intelligence through Simulated Evolution, Wiley, 1966.

[13] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.

[14] M. Mitchel, An Introduction to Genetic Algorithms, MIT Press, 1999.

[15] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.

[16] D. Kalyanmoy, A.R. Reddy, Reliable classification of two-class cancer data using evolutionary algorithms, BioSystems 72 (2003) 111–129.

[17] J. Dr'eo, A. Pĭetrowski, Metaheuristics for Hard Optimization, Springer–Verlag, Berlin, Heidelberg, 2006.

[18] R. Tanese, Distributed genetic algorithms, Proceedings of the third international conference on Genetic Algorithms, 1989.

[19] D. Whitlely, T. Starkwether, Genitor II: a distributed genetic algorithm, Journal of Experimental and Theoretical Artificial intelligence 2 (1990) 189–214.

[20] J. Paredis, Coevolutionary constraint satisfaction, in: Proceedings of the third international conference on parallel problem solving from nature, vol. 866, Springer–Verlag, 1994, pp. 46–55.

[21] J.D. Lohn, W.F. Kraus, D.S. Linden, S.P. Colombano, Evolutionary optimization of yagi-uda antennas, in: Y. Liu, K. Tanaka, M. Iwata, T. Higuchi, M. Yasunaga (Eds.), Proceedings of the Fourth International Conference on Evolvable Systems, Tokyo, October 3–5, Springer–Verlag, 2001, pp. 236–243.

[22] J.D. Lohn, W.F. Kraus, G.L. Haith, Comparing a co-evolutionary genetic algorithm for multi-objective optimization, in: Proceedings of the 2002 IEEE congress on Evolutionary Computation, 2002 May, pp. 1157–1162.

[23] D.W. Hills, Co-evolving parasites improve simulated evolution as an optimization procedure, Artificial Life II (1990) 313–324.

[24] P. Husbands, F. Mill, Simulated co-evolution as the mechanism for emergent planning and scheduling, Proceeding of the third International Conference on Genetic Algorithms, 1991, pp. 264–270.

[25] A. Giordana, F. Neri, Search-intensive concept induction, Evolutionary Computation 3 (4) (1996) 375–416.

[26] D.E. Moriarty, R. Miikkulainen, Efficient reinforcement learning through symbiotic evolution, Machine Learning 22 (1996) 11–32.

[27] M.A. Potter, K.A. de Jong, Cooperative coevolution: an architecture for evolving coadapted subcomponents, Evolutionary Computation 8 (1) (2000) 1–29.

[28] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic strategy for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.

[29] J. Vesterstrom, R. Thomsen, A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems Evolutionary Computation, McGraw-Hill, vol. 2, 2004, pp. 1980–1987.

[30] D Corne, M. Dorigo, F. Golver, New Ideas in Optimization, McGraw-Hill International (UK) Limited, 1999.

[31] V. Cerný, A thermo-dynamical approach to the traveling salesman problem, Journal of Optimization Theory and Applications (1985) 41–51.

[32] F. Glover, Tabu search—part I, ORSA Journal of Computing 1 (3) (1989) 190–206.

[33] F. Glover, Tabu Search—Part II, ORSA Journal of Computing 2 (1) (1990) 4–32.

[34] C. Voudouris, E. Tsang, Guided local search, Technical Report CSM-247, Department of Computer Science, University of Essex, Colchester, UK, 1995.

[35] T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures, Journal of Global Optimization 6 (1995) 109–133.

[36] H. Lourenco, O. Martin, T. Stützle, Iterated local search, Handbook of Metaheuristics, vol. 57 of International Series in Operation research & Management Science, McGraw-Hill, 2002, pp. 321–353.

[37] M.A. Faust, R.A. Gulledge, Identifying harmful marine dinoflagellates Contributions from the U.S. National Herbarium, vol. 42, Smithson Institutions, 2002, pp. 1–144.

[38] S. Willcox, N.A. Moltschaniwskyj, C. Crawford, Asexual reproduction in scyphistomae of *Aurelia* sp.: effects of temperature and salinity in an experimental study, Journal of Experimental Marine Biology and ecology 353 (1) (2007) 107–114.

[39] C. Zilberberg, A.M. Solé-Cava, M. Klautau, The extent of asexual reproduction in sponges of the genus Chondrilla (Demospongiae: Chondrosida) from the Caribbean and the Brazilian coasts, Journal of Experimental Marine Biology and Ecology 336 (2) (2006) 211–220.

[40] K. Holmstrom, H.J. Jensen, Who runs fastest in an adaptive landscape: sexual versus asexual reproduction, Physica A: Statistical and Theoretical Physics 337 (1–2) (2004) 185–195.

[41] V.G. Martinez, G.J. Menger III, M.J. Zoran, Regeneration and asexual reproduction share common molecular changes: upregualtion of a neural glycoepitope during morphallaxis in Lumbriculus, Mechanical Development 122 (5) (2005) 721–732.

[42] W. Song, Morphogenesis of *Cyrtohymena tetracirrata* (Ciliophora, Hypotrichia, Oxytrichidae) during binary fission, European Journal of Protistology 40 (3) (2004) 245–254.

[43] K. Lee, P. Singh, W.C. Chung, J. Ash, T.S. Kim, L. Hang, S. Park, Light regulation of asexual development in the rice blast fungus, Magnaporthe oryzae, Fungal Genetics and Biology 43 (10) (2006) 694–706.

[44] T. Yanagi, T. Yachi, N. Okuda, Photoperiodic reaction of sexual and asexual reproduction in *Fragaria chiloensis* L. CHI-24-1 plants grown at various temperatures, Scientia Horticulturae 110 (2) (2006) 187–191.

[45] F. Prall, C. Ostwald, High-degree tumor budding and podia-formation in sporadic colorectal carcinomas with K-ras gene mutations, Human Pathology 38 (2007) 1696–1702.

[46] A.V. Ramayya, J.H. Hamilton, J.K. Hwang, C.J. Beyer, G.M. Ter Akopian, A.V. Daniel, J.O. Rasmussen, S.-C. Wu, R. Donangelo, J. Kormicki, X.Q. Zhang, A. Rodin, A. Formichev, J. Kliman, L. Krupa, Y.T. Oganessian, G. Chubaryan, D. Seweryniak, R.V.F. Janssens, W.C. Ma, et al., Binary and ternary fission studies with 252Cf, Progress in Particle and Nuclear Physics 46 (1) (2001) 221–229.

[47] S. Stertz, M. Reichelt, M. Spiegel, T. Kuri, L. Martínez-Sobrido, A. García-Sastre, F. Weber, G. Kochs, The intracellular sites of early replication and budding of SARS-coronavirus, Virology 361 (2) (2007) 304–315.

[48] R.F. Green, D.L.G. Noakes, Is a little bit of sex as good as a lot? Journal of Theoretical Biology 174 (1) (1995) 87–96.

[49] M.T. Ghiselin, The evolution of sex: a history of competing points of view, in: R.E. Michod, B.R. Levin (Eds.), The evolution of sex: an examination of current ideas, Sinauer Associates Inc., Sunderland, 1988, pp. 7–23.

[50] G.C. Williams, Sex and Evolution, Monographs in Population Biology, vol. 8, Princeton University Press, 1975.

[51] J. Maynard Smith, The Evolution of Sex, Cambridge Univ. Press, Cambridge, UK, 1978.

[52] D.A. Levins, Pest pressure and recombination systems in plants, American Naturalist 109 (1975) 437–451.

[53] R.R. Glesener, D. Tilman, Sexuality and the components of environmental: clues from geographic parthenogenesis in terrestrial animals, American Naturalist 112 (1978) 659–673.

[54] H.L. Sanders, Marine benthic diversity: a comparative study, American Naturalist 102 (1968) 243–282.

[55] L. Francis, Contrast between solitary and colonial lifestyle in the sea anemone anthopleura eleyantissima, American Zoology 19 (1979) 669.

[56] K.P. Sebens, Asexual reproduction in an anthopleura elegantissina (Anthozoa: Actiniaria): seasonality and spatial content of clones, Ecology 63 (1982) 434–444.

[57] P.C. Schroeder, C.O. Hermans, Reproduction of marine invertebrates, Annelida: Polychaeta. In: A.C. Giese, J.S. Pearse (Eds.), Annelids and Echiurans, vol. 3. Academic Press, New York, 1975, pp. 1–213.

[58] A. Farasat, A.M. Amani, M.B. Menhaj, Minimum-delay link determination in networked control systems using asexual reproduction optimization, IEEE Symposium Series on Computational Intelligence, 2009.

[59] M. Clerc, J. Kennedy, The particle swarm explosion, stability, and convergence in a multidimensional complex space, IEEE Transaction on Evolutionary Computation 6 (2002) 58–73.

[60] J. Ludvig, J. Hesser, R. Manner, Tackling the representation problem by stochastic averaging, in: Proceedings of the 7th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Francisco, 1997, pp. 196–203.

[61] D. Gross, C.M. Harris, Fundamental of Queuing, 2th ed., Wiley, New York, 1984.

[62] J.A. White, J.W. Schimidt, G.K. Bennett, Analysis of Queuing System, Academic Press, New York, 1975.

[63] Corne D., Dorigo M. Golver F., "*New Ideas in Optimization*", McGraw-Hill International (UK) Limited, 1999.

[64] Yang Zhenyu, Ke Tang, Xin Yao, "Differential Evolution for High-Dimensional Function Optimization", IEEE Congress on Evolutionary Computation (2007) 3523–3530.

[65] R.M. Schuster, Ecology reproductive biology and dispersal of Hepaticae in the tropics, Journal of the Hattori Botanical Laboratory 76 (1988) 127–145.