Theory of Approximation and Applications, 2020, Volume 14, No. 1: 109-117, DOR: 20.1001.1.25382217.2020.14.1.6.8

The combinatorial of Artificial Bee Colony algorithm and Bisection method for solving Eigenvalue Problem

P. Mansouri[a*], P. Zolfaghari[b]

[a] *Department of Computer Science and Mathematics, Arak Branch, Islamic Azad University, Arak-Iran*

[b] *Department of Computer Engineering, Tehran University, Tehran-Iran*

Article Info

Keywords

Bisection method

Root-Finding method

Eigenvalue Problem

Artificial Bee Colony algorithm.

Article History

Received:2019 August 25

Accepted:2020 January 10

Abstract

The aim of this paper is to find eigenvalues of square matrix A based on Artificial Bee Colony algorithm and Bisection method(BIABC). At first, we obtain initial interval $[a, b]$ that included all eigenvalues based on Gerschgorin's theorem, and then by using Artificial Bee Colony algorithm(ABC) at this interval to generate initial value for each eigenvalue. The bisection method improves them until the best values of eigenvalues with arbitrary accuracy will be achieved. This enables us to find eigenvalues with arbitrary accuracy without computing the derivative of the characteristic polynomial of the given matrix. We illustrate the proposed method with Some numerical examples.

1 Introduction

Many practical problems in engineering and physics lead to eigenvalue problems. Eigenvalues and eigenvectors play an important part in the applications of linear algebra. The naive method of finding the eigenvalues of a matrix involves finding the roots of the characteristic polynomial of the matrix. There exist several techniques for finding eigenvalues and eigenvectors of a matrix, some of which fall under the realm of iterative methods. These methods work by repeatedly refining approximations to the eigenvectors or eigenvalues, and can be terminated whenever the approximations reach to a suitable degree of accuracy. Iterative methods form the basis of much of modern day eigenvalue computation. In this paper, we introduce a new iterative method to find eigenvalues of square matrix $A_{n\times n},\ n \geq 1$ without computing derivative of characteristic polynomial $P(z) = 0$ of A. We assume that the characteristic polynomial $P(z) = 0$, has n real roots(we call $\lambda_1 \leq \lambda_2 \leq ...\lambda_n$, where $\lambda_i \neq \lambda_j,\ i, j = 1, ..., n$. In the proposed method first we determine $\lambda_1, \lambda_2, ...$ and finally λ_n. Artificial Bee Colony (ABC) is one of the most recently defined algorithms by Dervis Karaboga in 2005, motivated by the intelligent behavior of honey bees [1, 2]. It is as simple as Particle Swarm Optimization (PSO) and Differential Evolution (DE) algorithms, Genetic Algorithm (GA)[1], biogeography based optimization (BBO), and uses only common control parameters such as colony size and maximum cycle number. ABC as an optimization tool, provides a population-based search procedure in which individuals called foods positions are modified by the artificial bees with time and the bee's aim is to discover the places of food sources with high nectar amount and finally the one with the highest nectar.

* Corresponding Author's E-mail: p-mansouri@iau-arak.ac.ir

In ABC system, artificial bees fly around in a multidimensional search space and some (employed and onlooker bees) choose food sources depending on the experience of themselves and their nest mates, and adjust their positions. Once all onlookers have selected their food sources, each of them determines a new neighboring food source of its selected food source and computes it's nectar amount. Providing that this amount is higher than that of the previous one, and then the bee memorizes the new position and forgets the old one. The employed bee becomes a scout bee when the food source which is exhausted by the employed and onlooker bees is assigned as abandoned. In other words, if any solution cannot be improved further through a predetermined number of cycles which is called limit parameter, the food source is assigned as an abandoned source and employed bee of that source becomes a scout bee. Thus, ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process. Also, Karaboga and Basturk have compared the performance of the ABC algorithm with other works such as GA, DE and PSO methods on unconstrained problems[3]. Although ABC is a robust, easy and flexible algorithm. It, like other evolutionary algorithms, has some issues. For example, acceleration of convergence speed is one of the important goals of ABC. But, convergence speed of this method is typically slower than those of representative population-based algorithms[4]. P. Mansouri et all in 2011[6], introduced a novel method for solving Hard problems without compute derivative of functions.

In the section 2-1 of this work, we define the linear algebra background required to understand the concepts that are discussed. In Section 2-2 and 2-3, we review Artificial Bee colony algorithm and Bisection Method. In Section 3 we introduce the proposed method and discuss about its details. Finally, in Section 4, we provide some concluding remarks and mention some of the additional algorithm refinements that are used in practice. For the purposes of this paper, we restrict our attention to real-valued, square matrices with a full set of real eigenvalues.

2 Preliminaries

2.1 The Eigenvalue Problem

The eigenvalues and eigenvectors are properties of the equations that simulate the behavior of a real structure.

2.1.1 Definition1.

Let $A \in R^{n\times n}$. A nonzero vector $x \in R^n$ is called an eigenvector of A with corresponding eigenvalue $\lambda \in C$ if $Ax = \lambda x$. Note that eigenvectors of a matrix are precisely the vectors in R^n, whose direction is preserved when multiplied with the matrix. Although eigenvalues may be not be real in general, we will focus on matrices whose eigenvalues are all real numbers. It is often necessary to compute the eigenvalues of a matrix. The most immediate method for doing so involves finding the roots of characteristic polynomials.

2.1.2 Definition2.

The characteristic polynomial of square matrix A, denoted$P_A(x)$ for $x \in R$, is the degree n polynomial defined by

$$P_A(x) = det(xI - A),$$

is straightforward to see that the roots of the characteristic polynomial of a matrix is exactly the eigenvalues of the matrix since the matrix $\lambda I - A$ is singular precisely when λ is an eigenvalue of A. It follows that computation of eigenvalues can be reduced to finding the roots of polynomials. Unfortunately, solving polynomials is generally a difficult problem as there is no closed formula for solving polynomial equations of degree 5 or higher. The only

way to proceed is to employ numerical techniques to solve these equations. We have just seen that eigenvalues may be found by solving polynomial equations. The converse is also true. Given characteristic polynomial,

$$f(z) = z^n + a_{n-1}z^{n-1} + ... + a_1 z + a_0,$$

Its zeroes are eigenvalues of matrix A. If $\lambda_1 \leq \lambda 2 \leq ... \leq \lambda_k$ are distinct roots of characteristic polynomial, then $f(z)$ can be represented in the form:

$$f(z) = (-1)^n (z - \lambda_1)^{\alpha_1} (z - \lambda_2)^{\alpha_2} ... (z - \lambda_k)^{\alpha_k}$$

The integer $\alpha_i,\ i = 1, 2, ..., k,\ k \leq n$ is called the multiplicity of the eigenvalue λ_i more precisely, its algebraic multiplicity.

2.1.3 [6] Gerschgorin's Theorem.

Let A be the matrix $A = (a_{ij})$, and let $x = (x_1, x_2, ..., x_n)'$ be an eigenvector of A corresponding to the eigenvalue λ. For some i we have $|x_i| \geq |x_j|$ for $|j \neq i$, and since x is an eigenvector, $|x_i| > 0$. Now

$$Ax = \lambda x,$$

or

$$(A - I\lambda)x) = 0,$$

which represents n simultaneous equations for the x_j. In particular, the coefficients x_j satisfy the i^{th} equation

$$(\lambda - a_{ii})x_i - \sum_{j \neq i} a_{ij} x_j = 0,$$

or

$$(\lambda - a_{ii})x_i = \sum_{j \neq i} a_{ij} x_j.$$

Therefore

$$|(\lambda - a_{ii})x_i| = |\sum_{j \neq i} a_{ij} x_j|.$$

Then

$$|\lambda - a_{ii}||x_i| \leq \sum_{j \neq i} |a_{ij}||x_j|,$$

$$\Rightarrow |\lambda - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|.$$

Thus the eigenvalue λ lies in one of the circles

$$|t - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|.$$

These circles are known as Gerschgorin's Circles, and this result is known as Gerschgorin's theorem. There are n circles corresponding to $i = 1, 2, ..., n$. Suppose that $B(r), 0 \leq r \leq 1$ is the $n \times n$ matrix given by

$$b_{ii} = a_{ii}$$

$$b_{ij} = ra_{ij}, i \neq j$$

Then the eigenvalues of $B(r)$ lie in the circles

$$|(t - a_{ii})| \leq r \sum_{j \neq i} |a_{ij}|.$$

In particular, when $r = 0$, the eigenvalues of B are the diagonal entries a_{ij}, and there is precisely one eigenvalue in each circle. As r increases to 1, the eigenvalues vary continuously, so that where the circles are distinct, there is one eigenvalue in the circle. Where the circles overlap, there are the appropriate number of eigenvalues in the combined region.

2.2 Artificial Bee Colony Algorithm(ABC)

Artificial Bee Colony algorithm (ABC) is an algorithm based on the intelligent foraging behavior of honey bee swarm,purposed by Karaboga in 2005 [2]. In ABC model, the colony consists of three groups of bees: employed bees, onlookers and scouts. In the ABC algorithm, the number of employed bees is equal to the number of food sources which is also equal to the number of onlooker bees. There is only one employed bee for each food source whose first position is randomly generated. At each iteration of the algorithm, each employed bee determines a new neighboring food source of its currently associated food source and computes the nectar amount of this new food source by equation:

$$v_{ij} = z_{ij} + \Theta_{ij}(z_{ij} - z_{kj}), \tag{2.1}$$

where Θ_{ij} is a random number between $[0, 1]$. If the nectar amount of this new food source is higher than that of its currently associated food source, then this employed bee moves to this new food source, otherwise it continues with the old one. After all employed bees complete the search process, they share the information about their food sources with onlooker bees. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount by equation:

$$p_i = \frac{fit_i}{\sum_1^n fit_k}, \tag{2.2}$$

where fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i and n is the number of food sources which is equal to the number of employed bees. This method, known as roulette wheel selection method, provides better candidates to have a greater chance of being selected. Once all onlookers have selected their food sources, each of them determines a new neighboring food source of its selected food source and computes it's nectar amount. Providing that this amount is higher than that of the previous one, and then the bee memorizes the new position and forgets the old one. The employed bee becomes a scout bee when the food source which is exhausted by the employed and onlooker bees is assigned as abandoned. In other words, if any solution cannot be improved further through a predetermined number of cycles which is called limit parameter, the food source is assigned as an abandoned source and employed bee of that source becomes a scout

bee. In that position, scout generates randomly a new solution by equation :

$$z_i^j = z_{min}^j + rand(0,1)(z_{max}^j - z_{min}^j), \tag{2.3}$$

where j is determined randomly which is different from i and assume that z_i is the abandoned source and $j \in 1, 2, ..., D$ where D is the solution vector, the scout discovers a new food source which will be replaced with z_i. The employed bee whose food source has been abandoned becomes a scout and starts to search for finding a new food source. Onlookers watch the dances of employed bees and choose food sources depending on dances.

2.2.1 The main steps of the Artificial Bee Colony Algorithm

Based on the above explanation of initializing the algorithm population, employed bee phase,probabilistic selection scheme, onlooker bee phase and scout bee phase, the pseudo-code of the ABC algorithm is given bellow:

2.2.2 Algorithm 1 (ABC's algorithm)

01. Initialize population with random solutions.
02. Evaluate fitness of the population.
03. **While** (stopping criterion not met) Forming new population.
04. Select sites for neighborhood search.
05. Recruit bees for selected sites (more bees for best sites) and evaluate fitnesses.
06. Select the fittest bee from each patch.
07. Assign remaining bees to search randomly and evaluate their fitnesses.
08. **End While**.

In first step, the algorithm starts with the scout bees (n) being placed randomly in the search space. In step 2, the fitnesses of the sites visited by the scout bees are evaluated. In step 4, bees that have the highest fitnesses are chosen as "selected bees" and sites visited by them are chosen for neighborhood search. Then, in steps 5 and 6, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the best sites. The bees can be chosen directly according to the fitnesses associated with the sites they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Searches in the neighborhood of the best sites which represent more promising solutions are made more detailed by recruiting more bees to follow them than the other selected bees. Together with scouting, this differential recruitment is a key operation of the bees algorithm. However, in step 6, for each patch only the bee with the highest fitness will be selected to form the next bee population. In nature, there is no such a restriction. This restriction is introduced here to reduce the number of points to be explored. In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts to its new population, those that were the fittest representatives from a patch and those that have been sent out randomly.

2.3 Bisection Method

The Bisection Method is a numerical method for estimating the roots of the real-valued problem $f(x) = 0$. It is one of the simplest and most reliable, but it is not the fastest method. Assumption that $f(x)$ is continuous.

2.3.1 Algorithm for the Bisection Method

Given the continuous function $f(x)$. Find points a and b such that $a < b$ and $f(a).f(b) < 0$. Take the interval $[a, b]$ and find its midpoint x_1.

$$\Delta x_i = |x_i - x_{i-1}| = (0.5)^i(b - a)$$

and the new midpoint is

$$x_i = (a_{i-1} + b_{i-1})/2$$

3 The Proposed Method

We present a novel iterative method to find eigenvalues of square matrix $A = (a_{ij})$ by using numerical iterative methods and ABC algorithm. First we obtained initial search interval with Gerschgorin's theorem. Let $A \in R^{n\times n}$ and the characteristic polynomial of square matrix A, denoted $P_A(x)$ for $x \in R$, is a polynomial of degree n, defined by $P_A(x) = det(xI - A)$. Assumption that $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_k$ are distinct roots. By using Gerschgorin's theorem, we determined n circles that each eigenvalue is belongs to one of them. Each eigenvalue λ lies in one of the circles

$$|t - a_{ii}| \leq \sum_{j\neq i} |a_{ij}|.$$

The interval $[a, b]$ that included all eigenvalues is obtained as follows

$$a = \min_i l_i,\ l_i\ is\ lower\ bound\ of\ i^{th}\ circle,\ i = 1, 2, ..., n$$

and,

$$b = \max_i u_i,\ u_i\ is\ upper\ bound\ of\ i^{th}\ circle,\ i = 1, 2, ..., n$$

So the best lower and upper bounds has been achieved. We are defining initial search domain as follows,

$$I_0 = [a_0, b_0] = [a, b],$$

The ABC algorithm with respect to initial search domain $I = [a_0, b_0]$, randomly generate the initial sequence values $z_k, k = 1, 2, ..., colonysize/2$. If there exist $j, 1 \leq j \leq colonysize/2$ that $P(z_j) = \min_k P(z_k), 1 \leq k \leq colonysize/2$ and that satisfies to condition $P(a_0)P(z_1) < 0$, then $\lambda_1 \in [a_0, z_1]$, else return to ABC algorithm and continues randomly generate the initial sequence values $z_k, k = 1, 2, ..., colonysize/2$ again and again, until value of z_1 that satisfies to up condition is obtained. Hence, after that by using Bisection method after some iteration, we obtained λ_1 and interval $[a_0, \beta_1] \subset [a_0, z_1]$ that included λ_1 with arbitrary accuracy. Any way, after finding λ_1, with omit small size interval including λ_1, we update I_0 to I_1 as follows,

$$I_1 = [a_1, b_1] = I_0 - [a_0, \beta_1],$$

It is clear that $I_1 \subset I_0$. Since the new interval width is smaller and with same process we obtain λ_2, after that λ_3

and so on. In this work, eigenvalues of square matrix A determined sequentially with arbitrary accuracy, without compute derivative of characteristic polynomial of A. Also at each iteration i for find $\lambda_i,\ i = 1, 2, ..., n,\ I_i$ width is smaller than I_{i-1} at iteration $i - 1$. Thus convergence speed of algorithm is fast and complexity of computations will be decrease.

3.1 Algorithm 2(The proposed Method's Algorithm)

Assumption that characteristic polynomial of square matrix $A_{n\times n}$ given as follows,

$$P(z) = z^n + a_{n-1}z^{n-1} + ... + a_1 z + a_0,$$

and $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ are distinct roots of characteristic polynomial and $\lambda_i \in [a_0, b_0],\ i = 1, 2, ..., n$. The main steps of algorithm is as follows,

01. Determine initial interval $[a_0, b_0]$ that included eigenvalues with Gerschgorin's theorem.
02. $i = 1$
03. **While** $(i \leq n)$
04. $cycle = 1$,
05. While($cycle \leq maxcycles$)
06. Determined initial value z_i at the interval $[a_{i-1}, b_{i-1}]$ based on ABC algorithm ,
07. If $P(a_{i-1})P(z_i) < 0$
08. Find the best approximate value of $\lambda_i \in [a_{i-1}, \beta_i] \subseteq I_{i-1}$ by Bisection method.
09. $cycle = maxcycle$,
09. Else
06. $cycle = cycle + 1$,
06. End while,
07. Update domain of search as follows

$$I_i = [a_i, b_i] = I_{i-1} - [a_{i-1}, \beta_i],$$

08. $i = i + 1$,
09. **End while.**

4 Numerical Examples

In this section, some examples are propose to illustrate the method.

4.1 Example1.

Consider the square matrix $A_{4\times 4}$

$$A = \begin{pmatrix} 2 & 1 & 11 & 6 \\ -1 & -2 & 4 & -7 \\ 4 & 3 & -3 & 3 \\ 3 & 0 & 5 & 1 \end{pmatrix}$$

Eigenvalue problem is as follows:

$$P(\lambda) = det(A - \lambda I) = det \begin{pmatrix} 2-\lambda & 1 & 11 & 6 \\ -1 & -2-\lambda & 4 & -7 \\ 4 & 3 & -3-\lambda & 3 \\ 3 & 0 & 5 & 1-\lambda \end{pmatrix}$$

or

$$P(\lambda) = \lambda^4 + 2\lambda^3 - 95.00042\lambda^2 - 180.0009\lambda + 511.0002 = 0$$

where $\lambda \in [-16, 20]$(initial domain search obtained with Gerschgorin's theorem). Four roots of equation $P(\lambda) = 0$ has shown as follows,

$$\lambda_1 = -9.437410268258554136562504 6956,$$
$$\lambda_2 = -3.606190522073592382694375 90932,$$
$$\lambda_3 = 1.587911852072726362979423 66391,$$
$$\lambda_4 = 9.455688938259420156277456 94103$$

It is clear that $\lambda_1 \in [-16, 20]$. After 44 iteration $\lambda_1 = -9.43739$, $P(\lambda_1) = -5.09922e-006$ achieved. With change initial interval I_0 to I_1 based on algorithm 4-1,$I_1 = [-9.43739, 20]$, it is clear that$\lambda_2, \lambda_3, \lambda_4 \in I_1$. In next step, λ_2 determined with same process in interval $I_1 = [-9.43739, 20]$. After 55 iteration, $\lambda_2 = -3.60608$, $P(\lambda_2) = 4.42179e-0003$ determined. Then,$\lambda_3 \in [-2, 20]$ determined with same process, after 20 iteration, $\lambda_3 = 1.58792, P(\lambda_3) = -9.11291e-005$. Finally, after45 iteration, $\lambda_4 = 9.4556653, P(\lambda_4) = -7.738e-004$. Summarize the mentioned process of algorithm $4-1$ has been simplified as follows

Table1: Result of solve eigenvalue problem of Examle1

λ	$P(\lambda)$	No-Iteration
-9.437390	-5.09922e-006	44
-3.606080	4.42179e-0003	52
1.5879200	-9.11291e-005	20
9.4556653	-7.738 e-004	18

4.1.1 Example2.

For the following matrix determine the eigenvalues.

$$A = \begin{pmatrix} 0 & 5 & 7 \\ -2 & 7 & 7 \\ -1 & 1 & 4 \end{pmatrix}$$

The eigenvalue problem is as follows:

$$P(\lambda) = det(A - \lambda I) = det \begin{pmatrix} -\lambda & 5 & 7 \\ -2 & 7-\lambda & 7 \\ -1 & 1 & 4-\lambda \end{pmatrix}$$

or

$$P(\lambda) = \lambda^3 - 11\lambda^2 + 38\lambda - 40 = (\lambda - 2)(\lambda - 4)(\lambda - 5) \tag{4.1}$$

Assume $\lambda \in I_0 = [-12, 16]$(initial domain search obtained with Gerschgorin's theorem). Three roots of equation $P(\lambda) = 0$ has been shown as follows,

$$\lambda_1 = 2,$$
$$\lambda_2 = 4,$$
$$\lambda_3 = 5$$

It is clear that $\lambda_1 \in I_0 = [-12, 16]$, after 22 iteration $\lambda_1 = 2.000000,\ P(\lambda_1) = 3.82322e - 005$.
Update initial interval I_0 to I_1 as follows:
$I_1 = [a_1, b_1] = [-12, 16] - [-12, 2.00000] = (0, 14]$,
In next step ,λ_2 determined with same process .
After 46 iteration, $\lambda_2 = 4,\ P(\lambda_2) = -8.27136e - 005$ determined. We can show $I_2 = [a_2, b_2] = [a_1, b_1] - [a_1, 4.00000] = (0, 10]$,
After 18 iteration $\lambda_3 = 5.00000,\ P(\lambda_3) = -2.24904e - 005$.
Summarize the mentioned process of algorithm $4 - 1$ has been simplified as follows

Table 2 : Result of solving eigenvalue problem $\lambda^3 - 11\lambda^2 + 38\lambda - 40 = 0$ by algorithm $4 - 1$

λ	$P(\lambda)$	No- Iteration
2.00000	$3.82322e - 005$	**22**
4.0000	$-8.27136e - 005$	**46**
5.00000	$-2.24904e - 005$	**18**

Accuracy of our method is very good and the important point is, without computing derivative of $P(\lambda)$, we determined eigenvalues from minimum eigenvalue to maximum eigenvalue one by one. During this process, the domain of search in each iteration gets smaller than the previous one.

5 Conclusion

In this work, we have proposed a new iterative method to solve eigenvalue problem by combining ABC algorithm and numerical iterative Bisection method. For a given square matrix $A_{n\times n},\ n \in N,\ n >> 2$, it is very important that without computing the derivative of characteristic polynomial:

$$P(\lambda) = det(A - \lambda I) = 0$$

We directly search the domain that includes all eigenvalues obtained with Gerschgorin's theorem and sequentially determine eigenvalues with the proposed algorithm. Also the convergence speed is remarkably fast.
It is possible to developed this method to solve complex eigenvalue problems easily.

References

[1] J.H.Holland," Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor, MI (1975)

[2] D. Karaboga," An Idea Based On Honey Bee Swarm for Numerical Optimization," Technical Report-TR06,Erciyes University, Engineering Faculty, Computer Engineering Department 2005.

[3] D. Karaboga, B. Basturk, "On the performance of Artificial Bee Colony (ABC) algorithm," Applied Soft Computing, vol. 8, pp. 687–697, January 2008.

[4] R. Storn , K. Price,"Differential evolution–Asimple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization23, pp.689–94 ,2010.

[5] B. Nobel, J. W. Daniel, " Applied Linear Algebra,"3rd edition,Printice-Hall International Editions, 1988.

[6] P. Mansouri, B. Asady, N. Gupta," A Novel Iteration Method for solve Hard Problems(Nonlinear Equations) with Artificial Bee Colony algorithm," Paris-4-11 International Conference on Intelligent Systems,2011.