Signal, Image and Video Processing
https://doi.org/10.1007/s11760-022-02171-w

INVITED PAPER

An improved immune plasma algorithm with a regional pandemic restriction

Selcuk Aslan[1] · **Sercan Demirci**[2]

Received: 3 August 2021 / Revised: 7 December 2021 / Accepted: 3 February 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
The coronavirus (COVID-19) and its global effect have increased the interests of researchers from different disciplines to the medical methods such as immune or convalescent plasma treatment. Immune Plasma algorithm (IPA) that is the first meta-heuristic referencing the steps of the immune plasma treatment as the name implies has been proposed recently and its potential has been investigated. In this study, a pandemic management strategy based on limiting the free movements between regions was modeled and integrated into the workflow of the IPA and a new variant called regional IPA (rIPA) was introduced. For analyzing the contribution of the proposed method, twelve numerical benchmark problems were solved. Also, the performance of the rIPA was investigated by solving a new big data optimization problem that requires minimization of the measurement noise of electroencephalography signals. The results obtained by the rIPA were compared with the fourteen well-known and state-of-art meta-heuristics. Comparative studies showed that managing the relationship between the individuals of the population as in the proposed regional model significantly contributes to the capabilities and rIPA outperforms other meta-heuristics for most of the test cases.

Keywords Meta-heuristics · IPA · Regional restriction · Noise filtering

1 Introduction

The increasing difficulties of real world or engineering problems limit the solving capabilities of the well-known analytical methods [21,23]. In order to address the drawbacks stemmed from the computational and storage limitations of these methods, researchers increased the density of the their studies on a relatively new area called meta-heuristics [21,23]. A meta-heuristic algorithm provides a problem independent framework guiding a set of strategies and different types of optimization problems related with the fields ranging from image, video or signal processing [2] to pattern-motif-gene discovery [20], digital filter design [5] to unnamed vehicle planning [19] have been solved successfully.

✉ Selcuk Aslan
selcukaslan@erciyes.edu.tr

Sercan Demirci
sercan.demirci@omu.edu.tr

[1] Department of Aeronautical Engineering, Erciyes University, Kayseri, Turkey

[2] Department of Computer Engineering, Ondokuz Mayıs University, Samsun, Turkey

The meta-heuristics that model the brilliant behaviors of the animals such as ants, birds or other creatures are called swarm intelligence based algorithms. Particle Swarm Optimization (PSO) [6] referencing the bird flocking or fish schooling and Artificial Bee Colony (ABC) [2] referencing the foraging habits of the honey bees are listed in the most commonly used swarm intelligence based meta-heuristics. Yang introduced Firefly algorithm (FA) [13] after analyzing flashing and signaling capabilities of the fireflies and Bat algorithm (BA) [10] by investigating bats and their echolocation properties. In addition to these algorithms, self and cross pollinating flowers became the source of inspiration for Yang and Flower Pollination algorithm (FPA) was proposed [1]. Mirjalili introduced or contributed to the developments of the meta-heuristics such as Moth-Flame Optimization (MFO) algorithm [14] based on the flying characteristics of the moths, Salp Swarm algorithm (SSA) [16] based on swarming behaviors of the salps and Harris Hawks Optimizer (HHO) [11] based on chasing styles of a special type of hawks. Another group of meta-heuristics also called evolutionary algorithms tries to model biological mechanisms including natural selection, crossover and mutation. Genetic algorithm (GA) [12] introduced by Holland

Published online: 04 March 2022

brought a new perspective to the concept of meta-heuristics and promising capabilities of them. The idea lying behind the representation of the solutions, subtly designed selection, crossover and mutation operators still contributes to the new researches about the meta-heuristics. Differential Evolution (DE) [8] proposed by Storn and Price is another well balanced evolutionary algorithm. The final group of the meta-heuristics usually guides the prominent physical, chemical or mathematical laws, theorems or different phenomenas. Gravitational Search algorithm (GSA) proposed by Rashedi et al. used Newtonian law of gravity to manage the search operation [24]. Cuevas et al. considered the transition between solid, liquid and gas phases of a matter and State of Matter Search (SMS) was introduced [7]. Mirjalili showed that some trigonometric functions such as sine and cosine can be used to generate and fluctuate candidate solutions and then Sine-Cosine algorithm (SCA) was proposed [15] and helped understanding richness of the laws or phenomenas being referenced.

As in the other disciplines, the COVID-19 has changed the working routines and search motivations of the researchers from the computer and data sciences and various studies have been conducted. In one of these studies, a medical method that is known as the immune or convalescent plasma and also used commonly for the treatment of the COVID-19 infection was investigated and a new meta-heuristic called Immune Plasma algorithm (IP algorithm or IPA) was introduced [3]. Even though the IPA is capable of obtaining promising results for different types of problems, it still requires subtly adjustment of two control parameters for determining the number of donors and receivers. The performance of the IPA can be further improved by adapting some pandemic management operations such as controlling free movements of the citizens and applying internal or external border restriction. Moreover, using pandemic management operations can determine the number of donors and receivers without using special control parameters. In this study, some pandemic management operations were first modeled and ported to the IPA and then an improved IPA variant called the regional IPA (rIPA) was introduced.

The subsequent sections of the paper are organized as follows: Basic workflow and used mathematical models by the standard IPA are given in Sect. 2. Implementation details of the border restriction mechanism and details of the rIPA are presented in Sect. 3. Results of the experiments and comparative studies are detailed in Sects. 4 and 5. Finally, conclusion and some assumptions about the future works are summarized in Sect. 6.

2 Immune plasma algorithm

When solving a D-dimensional problem with the IPA for which the size of the population is PS, the jth parameter of the x_k individual is initialized as in the Eq. (1) [4]. In Eq. (1), x_{kj} is used on behalf of the jth parameter of the x_k. x_j^h and x_j^l correspond to the upper and lower bounds of the jth parameter, respectively. Finally, the $rand(0, 1)$ represents a number generated randomly between 0 and 1.

$$x_{kj} = x_j^l + rand(0, 1)(x_j^h - x_j^l) \tag{1}$$

For modeling how an infection is transferred from an individual to another, IPA uses the Eq. (2) in which x_{kj} and x_{mj} are matched with the randomly determined jth parameters of the x_k and x_m individuals [4]. Also, the x_{kj}^{inf} is used to represent the jth parameter of the infected x_k showed by x_k^{inf}. While the k index is chosen from the set of $\{1, 2, \ldots, PS\}$ sequentially, the m index is chosen randomly from the same set but it must be different than the selected k. When the x_k is infected by the x_m, IPA computes the immune response of the x_k^{inf} by using the objective function f that will be minimized for the considered problem. If the immune response of the infectious x_k found as the $f(x_k^{inf})$ is less than the immune response of the x_k before the infection found as the $f(x_k)$, it is assumed that the x_k is capable of handling infection without requiring extra treatments and x_{kj} is changed with the x_{kj}^{inf} in order to support the immune memory for the subsequent encountering with the same or similar infection. Otherwise, x_k is not changed [4].

$$x_{kj}^{inf} = x_{kj} + rand(-1, 1)(x_{kj} - x_{mj}) \tag{2}$$

IPA continues its operations by determining how many receivers and donors will be selected. In order to decide that how many receivers and donors will be selected, IPA introduces two control parameters called NoR and NoD [4]. While the NoR is abbreviation for the number of receivers and the worst NoR individuals (or individual) are chosen as the receivers (or receiver), the NoD is abbreviation for the number of donors and the best NoD individuals (or individual) are chosen as the possible donors (or donor) [4]. When the k indexed receiver showed by x_k^{rcv} is decided to be treated with the plasma from the randomly assigned x_m^{dnr} donor individual, the IPA uses the Eq. (3) where j ranges from 1 to D. In Eq. (3), x_k^{rcv-p} shows the plasma transferred x_k^{rcv} receiver. x_{kj}^{rcv-p} corresponds to the jth parameter of the plasma transferred x_k^{rcv} receiver and the value of the j is set sequentially to the elements of $\{1, 2, \ldots, D\}$ [3]. If the newly calculated immune response showed as the $f(x_k^{rcv-p})$ for the x_k^{rcv} after the first plasma dose is better than the response showed as the $f(x_m^{dnr})$ for the x_m^{dnr}, the x_k^{rcv} is updated with the x_k^{rcv-p}

and second does of plasma from x_m^{dnr} is prepared for the next transfer. Otherwise, all of the parameters belonging to the x_k^{rcv} are changed with the corresponding parameters of the x_m^{dnr} in order to ensure that one dose of plasma is transferred and the treatment is completed for x_k^{rcv}. When the IPA decides to continue plasma treatment for the x_k^{rcv} with the second dose, the Eq. (3) is used again. However, for making a decision about the transfer of the third or more dose of plasma, a comparison between $f(x_k^{rcv-p})$ and $f(x_k^{rcv})$ is carried out [4].

$$x_{kj}^{rcv-p} = x_{kj}^{rcv} + rand(-1, 1)(x_{kj}^{rcv} - x_{mj}^{dnr}) \tag{3}$$

The immune memory controlling how quickly antibodies of a plasma donor will be synthesized can change with time or incidence of infection. The IPA manages this type of change on the immune memory of a donor with a mechanism that utilizes from the ratio between t_{cr} and t_{max} [3]. While the t_{cr} shows the current evaluation number, t_{max} corresponds to the maximum evaluation number. If t_{cr}/t_{max} is equal or less than the random number generated between 0 and 1, the x_m^{dnr} or its immune memory is re-organized by using Eq. (1). Otherwise, the x_m^{dnr} donor is modified slightly with the Eq. (4) where j ranges from 1 to D [3].

$$x_{mj}^{dnr} = x_{mj}^{dnr} + rand(-1, 1)x_{mj}^{dnr} \tag{4}$$

3 Regional immune plasma algorithm

An infection for which a well known treatment method does not exist as in the initial stage of the ongoing COVID-19 pandemic can spread easily among the population members and lead to dramatic changes on our daily routines. After detecting the spread of an infection, governments must take measures to control the infection and protect the health of their citizens. One of the most common and effective precautions is applying external and internal border restrictions. While external border restrictions try to control infection by limiting travels between countries, internal border restrictions help to decrease active cases within a country by limiting the free movements of citizens between states, cities or regions. Moreover, if a specialized treatment method is applied and the distribution of the infection is suppressed in a region, other regions are informed about the used medical method or methods and the distribution of the infection is also controlled for these regions. The Fig. 1 illustrates how the transitions between the regions named A, B, M and N are restricted with the start of the infection and each region tries to control infection by using their local capabilities. Each region shares the information about the successful treatment methods with other regions and guides them for controlling the infection.

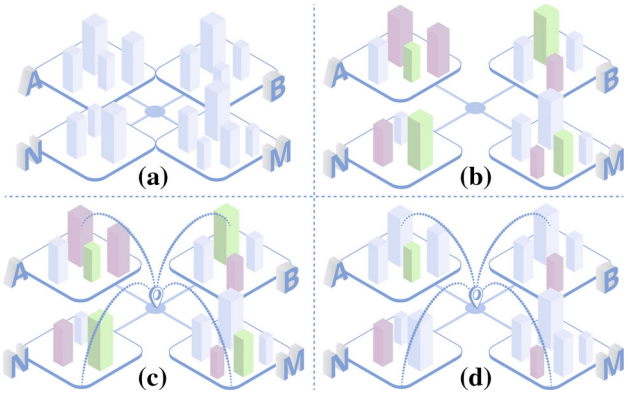

Fig. 1 Border restrictions between A, B, M and N regions

The positive effect of the border restrictions on the management of a pandemic can be modeled appropriately and integrated into the workflow of the IPA for further improving the solving capabilities of it. Assume that PS individuals of the IPA are related with a state for which the external border restrictions are maintained. In addition to this, mentioned state is tried to be divided randomly into RG different regions (or cities) each containing at least two or more individuals and sum of the individuals in all of the RG regions is equal to the PS. In the Algorithm 1, it is described how the sizes of the RG different regions are determined randomly and initialization of the other control parameters is conducted.

Algorithm 1 Sizes of the RG different regions

Assign initial values to PS, D and RG
Set x_{best} as the best individual of the population
Select t_{max} and set t_{cr} to PS
$rgSizes[1 \ldots RG] \leftarrow 2$
for $k \leftarrow 1 \ldots RG$ **do**
  $sumOfrgSizes = sum(rgSizes[1 \ldots RG])$
  if $k == RG$ **then**
    $rgSizes[k] = rgSizes[k] + (PS - sumOfrgSizes)$
  else
    $const = \lfloor rand(0, 1) \times (PS - sumOfrgSizes) \rfloor$
    $rgSizes[k] = rgSizes[k] + const$
  end if
end for

The IPA that uses the internal border restrictions described above is named as the regional IPA (rIPA). In the rIPA, the whole population is distributed between RG different regions by considering that each region contains at least two individuals. After generating isolated regions and determining number of individuals for them, rIPA utilizes from the fundamental operations of the IPA about the distribution of infection between individuals of each region. However, it should be considered that the interactions between individuals must be limited by considering the number of individuals in a region and individuals in a region. The Algorithm 2 below describes how the infection distributes between individuals

of a region and interactions between them are specialized for a region.

Algorithm 2 Distributing infection in rIPA

for $i \leftarrow 1 \ldots RG$ **do**
  if $i == 1$ **then**
    $lBound \leftarrow 1$ and $uBound \leftarrow regions[i]$
  else
    $sumOfrgSizes = sum(rgSizes[1 \ldots (i-1)])$
    $lBound = sumOfrgSizes + 1$
    $uBound = sumOfrgSizes + rgSizes[i]$
  end if
  for $k \leftarrow lBound \ldots uBound$ **do**
    if $t_{cr} < t_{max}$ **then**
      $t_{cr} \leftarrow t_{cr} + 1$
      select x_m randomly between $lBound$ and $uBound$
      $x_k^{inf} \leftarrow$ infect x_k with x_m using Eq. (2)
      if $f(x_k^{inf}) < f(x_k)$ **then**
        Update x_k with x_k^{inf}
        if $f(x_k) < f(x_{best})$ **then**
          Update x_{best} with x_k
        end if
      end if
    end if
  end for
end for

When the regions complete the distribution of infection, each region determines the worst individual as a receiver and the best individual as the donor candidate. From the RG donor candidates, rIPA selects the best one as the main donor and RG receivers are treated with the main donor as described by the Eq. (3). Even though the infection still continues to spread, each region gets a chance of evaluating their own status, determining, isolating and treating critical individuals more quickly. For accessing the details of the plasma transfer operations of the rIPA, the Algorithm 3 should be investigated. As stated earlier in the standard implementation of the IPA, the immune memory of the individuals who are also helped treatment operations of the critical patients as plasma donor can change with time. For the rIPA, there is only one donor and its immune memory is updated by guiding the workflow of the IPA.

Algorithm 3 Applying plasma transfer in rIPA

$doseControl[1 \ldots RG] \leftarrow$ set each element to 1
$treatmentControl[1 \ldots RG] \leftarrow$ set each element to 1
$x^{dnr} \leftarrow$ get the best individual as donor from RG regions
$rIndexes[1 \ldots RG] \leftarrow$ get indexes of receivers from RG regions
for $i \leftarrow 1 \ldots RG$ **do**
  $k \leftarrow rIndexes[i]$ and $x_k^{rcv} \leftarrow$ get the receiver from ith region
  while $treatmentControl[i] == 1$ **do**
    if $t_{cr} < t_{max}$ **then**
      $t_{cr} \leftarrow t_{cr} + 1$
      $x_k^{rcv-p} \leftarrow$ treatment to x_k^{rcv} with x^{dnr} using Eq. (3)
      if $doseControl[i] == 1$ **then**
        if $f(x_k^{rcv-p}) < f(x^{dnr})$ **then**
          Update x_k^{rcv} with x_k^{rcv-p}
          $doseControl[i] \leftarrow doseControl[i] + 1$
        else
          Update x_k^{rcv} with x^{dnr}
          Set $treatmentControl[i]$ to 0
        end if
      else
        if $f(x_k^{rcv-p}) < f(x_k^{rcv})$ **then**
          Update x_k^{rcv} with x_k^{rcv-p}
        else
          Set $treatmentControl[i]$ to 0
        end if
      end if
      if $f(x_k^{rcv}) < f(x_{best})$ **then**
        Update x_{best} with x_k^{rcv}
      end if
    end if
  end while
end for

4 Comparison of rIPA with other meta-heuristics on classical problems

The performance investigation of the rIPA was started by solving the classical benchmark problems ranging from f_1 to f_{12} [3]. When solving these classical benchmark problems, number of parameters was set to 100 and population size of the rIPA was taken equal to 30 [3]. While the maximum evaluation number was 30,000, the RG parameter was set to 2, 3, 4, 5, 6, 7 and 8 for understanding the change trends of the final solutions with the different number of regions. For each combination of benchmark function and RG parameter, rIPA was tested 30 times with random seeds and the best solutions and corresponding objective function values obtained were recorded. For the f_6, f_9, f_{10} and f_{11} functions, the RG parameter can be chosen equal or higher than 5. While the appropriate RG parameter value is 8 for the f_2, f_3, f_4 and f_7 functions, the appropriate RG parameter is 7, 4 and 1 for the f_1, f_8 and f_{12} functions, respectively. Only for the f_5 function, the values assigned to the RG does not generate dramatic changes on the qualities of the final solutions of the rIPA. However, it should be stated that the value being assigned to the RG for f_5 function can be less than 6.

The validation of the results found by the rIPA requires a comparison with the other meta-heuristics. For this purpose, the results of the rIPA were compared with the results of the different meta-heuristics including IPA [3], MFO [14], PSO [6,14], GSA [14,24], BA [13,14], FPA [1,14], SMS [7,14], FA [10,14] and GA [12,14]. In order to guarantee that all of these algorithms obtain their final solutions under the same conditions, population sizes were set to 30 and maximum evaluation number was taken equal to 30, 000 [14]. The RG parameter of rIPA was 7 for the f_1 and 4 for the f_8. Also, it was set to 8 for the remaining functions. When the mean best objective function values and standard deviations calculated after 30 independent runs given in the Table 1 are analyzed, it is understood that rIPA outperforms its competitors for ten of twelve benchmark functions. Only for the f_8 and f_{12} functions, rIPA lags slightly behind some of the tested algorithms.

Table 1 Comparison between rIPA and other meta-heuristics for classical problems

Fn.		rIPA	IPA	MFO	PSO	GSA	BA	FPA	SMS	FA	GA
f_1	Mn.	**5.878e–103**	7.467e−27	1.170e−04	1.321e+00	6.082e+02	2.079e+04	2.036e+02	1.200e+02	7.480e+03	2.188e+04
	Sd.	3.219e−102	1.933e–26	1.500e−04	1.153e+00	4.646e+02	5.892e+03	7.839e+01	0.000e+00	8.948e+02	2.879e+03
f_2	Mn.	**4.200e–71**	9.236e−26	6.390e−04	7.715e+00	2.275e+01	8.978e+01	1.116e+01	2.053e−02	3.932e+01	5.651e+01
	Sd.	2.300e−70	2.226e−25	8.770e−04	4.132e+00	3.365e+00	4.195e+01	2.919e+00	4.718e−03	2.465e+00	5.660e+00
f_3	Mn.	**3.172e–39**	4.504e−14	6.967e+02	7.363e+02	1.357e+05	6.248e+04	2.375e+02	3.782e+04	1.735e+04	3.701e+04
	Sd.	1.737e−38	2.454e−13	1.885e+02	3.617e+02	4.865e+04	2.976e+04	1.366e+02	0.000e+00	1.740e+03	5.572e+03
f_4	Mn.	**7.073e–03**	2.346e+01	7.068e+01	1.297e+01	7.878e+01	4.974e+01	1.257e+01	6.917e+01	3.395e+01	5.914e+01
	Sd.	3.874e−02	1.644e+01	5.275e+00	2.634e+00	2.814e+00	1.014e+01	2.290e+00	3.876e+00	1.869e+00	4.648e+00
f_5	Mn.	**9.891e+01**	1.102e+02	1.391e+02	7.736e+04	7.410e+02	1.995e+06	1.097e+04	6.382e+06	3.795e+06	3.132e+07
	Sd.	8.320e−02	4.307e+01	1.202e+02	5.115e+04	7.812e+02	1.252e+06	1.205e+04	7.299e+05	7.590e+05	5.264e+06
f_6	Mn.	**0.000e+00**	**0.000e+00**	1.130e−04	2.866e+02	3.081e+03	1.705e+04	1.753e+02	4.143e+04	7.828e+03	2.096e+04
	Sd.	0.000e+00	0.000e+00	9.870e−05	1.070e+02	8.986e+02	4.917e+03	6.345e+01	3.295e+03	9.752e+02	3.868e+03
f_7	Mn.	**2.926e–04**	7.075e−03	9.115e−02	1.037e+00	1.129e−01	6.045e+00	1.359e−01	4.952e−02	1.906e+00	1.337e+01
	Sd.	2.819e−04	9.401e−03	4.642e−02	3.103e−01	3.760e−02	3.045e+00	6.121e−02	2.401e−02	4.600e−01	3.081e+00
f_8	Mn.	−1.282e+04	**–2.512e+04**	−8.496e+03	−3.571e+03	−2.352e+03	6.553e+04	-8.086e+03	−3.942e+03	−3.662e+03	−6.331e+03
	Sd.	3.911e+03	9.553e+02	7.258e+02	4.308e+02	3.821e+02	0.000e+00	1.553e+02	4.041e+02	2.141e+02	3.325e+02
f_9	Mn.	**0.000e+00**	**0.000e+00**	8.460e+01	1.243e+02	3.100e+01	9.621e+01	9.269e+01	1.528e+02	2.149e+02	2.368e+02
	Sd.	0.000e+00	0.000e+00	1.616e+01	1.425e+01	1.366e+01	1.958e+01	1.422e+01	1.855e+01	1.721e+01	1.903e+01
f_{10}	Mn.	**8.881e–16**	6.960e−14	1.260e+00	9.167e+00	3.741e+00	1.594e+01	6.844e+00	1.913e+01	1.456e+01	1.784e+01
	Sd.	2.005e−31	1.758e−13	7.295e−01	1.569e+00	1.712e−01	7.749e−01	1.250e+00	2.385e−01	4.675e−01	5.311e−01
f_{11}	Mn.	**0.000e+00**	**0.000e+00**	1.908e−02	1.241e+01	4.868e−01	2.202e+02	2.716e+00	4.205e+02	6.965e+01	1.799e+02
	Sd.	0.000e+00	0.000e+00	2.173e−02	4.165e+00	4.978e−02	5.470e+01	7.277e−01	2.525e+01	1.211e+01	3.244e+01
f_{12}	Mn.	1.178e+00	8.157e+03	8.940e−01	1.387e+01	**4.634e–01**	2.893e+07	4.105e+00	8.742e+06	3.684e+05	3.413e+07
	Sd.	6.705e−02	4.467e+04	8.812e−01	5.853e+00	1.376e−01	2.178e+06	1.043e+00	1.405e+06	1.721e+05	1.893e+06

Bold values are represents the better results

5 Comparison of rIPA with other meta-heuristics on signal decomposition

The big data concept that differs from a conventional database or dataset has changed the current status of the data-dependent real world or complex optimization problems into another stage [9]. One of these data-dependent real world optimization problems has been introduced by Abbass and Goh recently [9]. They measured electroencephalography (EEG) signals and used them for generating different problem instances. While the problem instances named D4 and D4N are related with four time series each containing 256 samples, D12 and D12N instances are related with twelve time series each containing 256 samples [9]. EEG signals can be employed for determining brain or non-brain activities, mental disorders, abnormalities and should be filtered before further processing [17,18]. The problem introduced by Abbass and Goh mainly depends on splitting a matrix formed EEG signal into noise and noise-free compartments [9]. Given that X is an $N \times M$ dimensional matrix where N shows the number of time series and M shows the length of time series or number of samples and used on behalf of the transformed problem instance with the A matrix of size $N \times N$. Also, S is another matrix of size $N \times M$ and it is multiplied with the A transformation matrix for obtaining X [9]. If the S matrix is appropriately splitted into S_1 and S_2 matrices where S_1 represents the noise-free part of the original S matrix and S_2 represents the noise part of the original S matrix, the X matrix is calculated simply with the A, S_1 and S_2 matrices by using Eq. (5).

$$X = A \times S = A \times (S_1 + S_2) = A \times S_1 + A \times S_2 \tag{5}$$

Although the relationship between X, A, S, S_1 and S_2 matrices is represented by utilizing from relatively simple equations, a straightforward method for dividing the S matrix into noise-free and noise compartments or in other words obtaining S_1 and S_2 matrices from the S does not exist. However, Abbass and Goh stated that the Pearson Correlation Coefficients abbreviated as C and calculated by using Eq. (6) in which $var(X)$ and $var(A \times S_1)$ correspond to the variance matrices and $covar(X, A \times S_1)$ corresponds to the covariance matrix can be guided [9]. For a proper division, the diagonal elements of the C should be maximized and remaining elements of it should by minimized [9]. By considering the minimization and maximization purposes about the C matrix, an objective function showed with f_1 can be defined as given in Eq. (7).

$$C = \frac{covar(X, A \times S_1)}{var(X) \times var(A \times S_1)} \tag{6}$$

$$f_1(C) = \frac{1}{(N^2 - N)} \sum_{i \neq j} (C_{ij})^2 + \frac{1}{N} \sum_{i} (1 - C_{ii})^2 \tag{7}$$

Another important situation that should be considered for a proper division is the similarity between the S_1 and S matrices. Because of the S_1 matrix represents the original S matrix, S_1 should be determined relatively close to the S. In order to decide that how the obtained S_1 matrix is similar to the S, an objective function showed with f_2 can be defined as given in Eq. (8) [9]. If it is assumed that both f_1 and f_2 objectives are weighted equally and their sum is tried to be minimized, a challenging optimization problem is described.

$$f_2(S, S1) = \frac{1}{N \times M} \sum_{ij} (S_{ij} - S1_{ij})^2 \tag{8}$$

When the noise-free parts of the D4, D4N, D12 and D12N instances were tried to be determined with the rIPA, the population size and number of maximum evaluations were set to 100 and 10, 000, respectively [3]. The lower and upper bounds of the elements belonging to the S_1 were -8 and $+8$ [3]. As in the previous experimental scenarios, seven different values including 2, 3, 4, 5, 6, 7 and 8 were assigned to the RG parameter. For each combination of the problem instance and value of the RG parameter, rIPA was tested 30 times with random seeds and best solutions of runs were recorded. From the recorded results, it can be roughly generalized that the qualities of the solutions obtained by the rIPA for the D4 and D12 instances are getting better with the RG increasing from 2 to 6. Moreover, same generalization can be extended for the D4N and D12N instances. The qualities of the solutions obtained by the rIPA for D4N and D12N instances are getting better with the RG increasing from 2 to 7.

In order to evaluate the status of the rIPA within other meta-heuristics for the mentioned real world problem, a comparative study has been carried out. The results obtained by the rIPA for which the RG is 6 for D4 and D12 instances and 7 for D4N and D12N instances were compared with the results obtained by the IPA, GA [3,12], PSO [3,22], DE [3,8], ABC [2,3], GSA [3,24], MFO [3,14], SCA [3,15], SSA [3,16] and HHO [11] under the same conditions as seen in the Table 2. In the experiments, the NoR and NoD parameters of the IPA were set to 1. When the results given in the Table 2 are controlled, it is seen that rIPA outperforms selected competitors for all of the four test instances. The difficulty of the problem increasing with the number of time series leads to trapping initial local minimums for some of the tested meta-heuristics including PSO, DE, ABC and MFO. However, rIPA has different implicit or explicit mechanisms that manage converging to the optimum solutions and escaping local minimums. The division of the population in the rIPA allows the interactions between the individuals of the same region. This type of mechanism increases the interaction probability between the qualified individual and other members of same region and contributes to the convergence

Table 2 Comparison between rIPA and other meta-heuristics for D4, D4N, D12 and D12N instances

Ins.		rIPA	IPA	GA	PSO	DE	ABC	GSA	MFO	SCA	SSA	HHO
D4	Mn.	**1.669e+00**	1.814e+00	2.130e+00	7.878e+00	1.465e+01	2.004e+01	1.759e+00	2.046e+01	6.074e+00	2.777e+00	1.743e+00
	Bs.	1.567e+00	1.617e+00	1.893e+00	7.243e+00	7.423e+00	1.898e+01	1.638e+00	1.954e+01	3.041e+00	2.602e+00	1.597e+00
	Sd.	6.097e−02	8.091e−02	1.555e−01	2.710e−01	6.328e+00	4.280e−01	7.934e−02	3.197e−01	1.427e+00	1.175e−01	5.011e−02
D4N	Mn.	**1.671e+00**	1.828e+00	2.149e+00	7.896e+00	1.688e+01	2.010e+01	1.758e+00	2.036e+01	6.438e+00	2.825e+00	1.755+00
	Bs.	1.544e+00	1.673e+00	1.871e+00	7.552e+00	5.342e+00	1.960e+01	1.566e+00	1.985e+01	3.531e+00	2.670e+00	1.642e+00
	Sd.	6.161e−02	7.678e−02	1.564e−01	2.252e−01	7.048e+00	3.269e−01	8.609e−02	3.510e−01	1.036e+00	8.552e−02	4.297e−02
D12	Mn.	**1.810e+00**	1.926e+00	2.801e+00	1.063e+01	2.239e+01	2.195e+01	2.196e+00	2.202e+01	6.915e+00	3.099e+00	1.856e+00
	Bs.	1.701e+00	1.863e+00	2.545e+00	1.002e+01	2.189e+01	2.159e+01	2.058e+00	2.181e+01	5.808e+00	2.966e+00	1.798e+00
	Sd.	4.350e−02	4.248e−02	1.507e−01	1.925e−01	2.015e−01	1.968e−01	5.620e−02	1.247e−01	4.699e−01	7.307e−02	3.129e−02
D12N	Mn.	**1.805e+00**	1.914e+00	2.776e+00	1.061e+01	2.243e+01	2.195e+01	2.210e+00	2.197e+01	6.805e+00	3.121e+00	1.860e+00
	Bs.	1.691e+00	1.766e+00	2.384e+00	1.027e+01	2.202e+01	2.143e+01	2.115e+00	2.139e+01	2.313e+00	2.954e+00	1.810e+00
	Sd.	4.623e−02	5.506e−02	1.508e−01	1.569e−01	1.702e−01	2.240e−01	4.337e−02	2.282e−01	1.044e+00	7.736e−02	3.367e−02

Bold values are represents the better results

Table 3 Statistical comparison between rIPA and other meta-heuristics for D4, D4N, D12 and D12N instances

rIPA vs.	D4			D4N			D12			D12N		
	Z-val	*p*-val.	Sign.	*Z*-val.	*p*-val.	Sign.	*Z*-val.	*p*-val.	Sign.	*Z*-val.	*p*-val.	Sign.
IPA	–4.5765	0.00001	**rIPA**	–4.6587	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.6382	0.00001	**rIPA**
GA	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**
PSO	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**
DE	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**
ABC	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**
GSA	–3.5069	0.00044	**rIPA**	–3.6303	0.00028	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**
MFO	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**
SCA	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**
SSA	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**	–4.7821	0.00001	**rIPA**
HHO	–2.9721	0.00298	**rIPA**	–3.7743	0.00016	**rIPA**	–3.6920	0.00022	**rIPA**	–3.8771	0.00010	**rIPA**

Bold values are represents the statistically significant algorithms

speed of the algorithm. Moreover, if the members of a region get stuck of a local minimum because of the density of the interaction, the newly proposed approach can help to these trapped individuals for escaping by applying a plasma treatment with a donor of different region.

The comparative studies between rIPA and other meta-heuristics for the noise minimization problem are ended with the Wilcoxon signed rank test. The significance level (p) of the test was set to 0.05 and its results were summarized in the Table 3. When the results given in the Table 3 are controlled, it is decided that the results obtained by the rIPA are enough to generate statistical difference in favor of the same algorithm for all of the four problem instances. While the p values calculated between rIPA and its nearest competitor are equal to 0.00298 and 0.00016 for the D4 and D4N instances, they are equal to 0.00022 and 0.00010 for the D12 and D12N instances. Small p values for the D12 and D12N instances compared to the D4 and D4N instances also indicate that solving capabilities of the rIPA become more dominant compared to the tested meta-heuristics with the increasing number of time series.

6 Conclusion

The COVID-19 has changed the working routines of the researchers from different disciplines and focused their interests on the diagnosis approaches, medical techniques or governmental measures taken to control and manage the pandemic. In this study, border restriction based pandemic control precaution was modeled and and then a new version of the Immune Plasma algorithm (IPA) named regional Immune Plasma algorithm (rIPA) was introduced. In order to analyze the capabilities of the rIPA, a set of numerical and engineering problems was solved by assigning different values to the algorithm-specific control parameters. The results obtained by the rIPA were also compared with the various meta-heuristics.

The comparative studies helped to state that the region based restriction model significantly improves the performance and rIPA outperforms its competitors for most of the tested problems. rIPA manages donor and receiver selection operations without requiring extra control parameters used by the standard IPA. However, it should be noticed that distributing the individuals of the population into different regions as in the rIPA can limit the overall solution diversity and decrease the exploration capability. In future, different pandemic management approaches by referencing the positive contribution of the introduced restriction approach on the performance of the algorithm and qualities of the solutions can be modeled and used with the IPA when solving various numerical or discrete optimization problems.

References

1. Abdel-Basset, M., Mohamed, R., Saber, S., Askar, S., Abouhawwash, M.: Modified flower pollination algorithm for global optimization. Mathematics **9**(14), 1661 (2021). https://doi.org/10.3390/math9141661
2. Akay, B., Karaboga, D.: A survey on the applications of artificial bee colony in signal, image, and video processing. SIViP **9**(4), 967–990 (2015). https://doi.org/10.1007/s11760-015-0758-4
3. Aslan, S., Demirci, S.: Immune plasma algorithm: a novel meta-heuristic for optimization problems. IEEE Access **8**, 220227–220245 (2020). https://doi.org/10.1109/ACCESS.2020.3043174
4. Aslan, S., Demirci, S.: Performance investigation of parallel immune plasma algorithm. In: 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1–5 (2021). https://doi.org/10.1109/INISTA52262.2021.9548547
5. Chandra, A., Chattopadhyay, S.: A novel approach for coefficient quantization of low-pass finite impulse response filter using differential evolution algorithm. SIViP **8**(7), 1307–1321 (2014). https://doi.org/10.1007/s11760-012-0359-4
6. Couceiro, M.S., Rocha, R.P., Ferreira, N.F., Machado, J.T.: Introducing the fractional-order Darwinian pso. SIViP **6**(3), 343–350 (2012). https://doi.org/10.1007/s11760-012-0316-2
7. Cuevas, E., Diaz, P., Camarena, O.: A states of matter search-based scheme to solve the problem of power allocation in plug-in electric cars. In: Metaheuristic Computation: A Performance Perspective, pp. 161–176. Springer (2021). https://doi.org/10.1007/978-3-030-58100-8-6
8. Deng, W., Shang, S., Cai, X., Zhao, H., Song, Y., Xu, J.: An improved differential evolution algorithm and its application in optimization problem. Soft. Comput. **25**(7), 5277–5298 (2021). https://doi.org/10.1007/s00500-020-05527-x
9. Goh, S.K., Tan, K.C., Al-Mamun, A., Abbass, H.A.: Evolutionary big optimization (bigopt) of signals. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 3332–3339 (2015). https://doi.org/10.1109/CEC.2015.7257307
10. Gupta, D., Agrawal, U., Arora, J., Khanna, A.: Bat-inspired algorithm for feature selection and white blood cell classification. In: Nature-Inspired Computation and Swarm Intelligence, pp. 179–197. Elsevier (2020). https://doi.org/10.1016/B978-0-12-819714-1.00022-1
11. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: algorithm and applications. Futur. Gener. Comput. Syst. **97**, 849–872 (2019). https://doi.org/10.1016/j.future.2019.02.028
12. Katoch, S., Chauhan, S.S., Kumar, V.: A review on genetic algorithm: past, present, and future. Multimedia Tools Appl. **80**(5), 8091–8126 (2021). https://doi.org/10.1007/s11042-020-10139-6
13. Kumar, V., Kumar, D.: A systematic review on firefly algorithm: past, present, and future. Arch. Comput. Methods Eng. **28**(4), 3269–3291 (2021). https://doi.org/10.1007/s11831-020-09498-y
14. Mirjalili, S.: Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowl. Based Syst. **89**, 228–249 (2015). https://doi.org/10.1016/j.knosys.2015.07.006
15. Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. Knowl. Based Syst. **96**, 120–133 (2016). https://doi.org/10.1016/j.knosys.2015.12.022
16. Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. **114**, 163–191 (2017). https://doi.org/10.1016/j.advengsoft.2017.07.002
17. Mokatren, L.S., Ansari, R., Cetin, A.E., Leow, A.D., Ajilore, O., Klumpp, H., Vural, F.T.: Eeg classification based on image configuration in social anxiety disorder. In: 2019 9th International

IEEE/EMBS Conference on Neural Engineering (NER), pp. 577–580 (2019). https://doi.org/10.1109/NER.2019.8717152
18. Mokatren, L.S., Ansari, R., Cetin, A.E., Leow, A.D., Ajilore, O.A., Klumpp, H., Yarman Vural, F.T.: Eeg classification by factoring in sensor spatial configuration. IEEE Access **9**, 19053–19065 (2021). https://doi.org/10.1109/ACCESS.2021.3054670
19. Pattnaik, S., Mishra, D., Panda, S.: A comparative study of meta-heuristics for local path planning of a mobile robot. Eng. Optim. (2021). https://doi.org/10.1080/0305215X.2020.1858074
20. Sherly, L.A., Jaya, T.: Improved firefly algorithm-based optimized convolution neural network for scene character recognition. Signal Image Video Process. (2021). https://doi.org/10.1007/s1760-020-0180-4
21. Urfalioglu, O., Cetin, A.E., Kuruoglu, E.E.: Levy walk evolution for global optimization. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, pp. 537–538 (2008). https://doi.org/10.1145/1389095.1389200
22. Wang, F., Zhang, H., Zhou, A.: A particle swarm optimization algorithm for mixed-variable optimization problems. Swarm and Evolutionary Computation 60, 100808 (2021). https://doi.org/10.1016/j.swevo.2020.100808
23. Yang, X.S.: Nature-inspired optimization algorithms: challenges and open problems. J. Comput. Sci. (2020). https://doi.org/10.1016/j.jocs.2020.101104
24. Zandevakili, H., Rashedi, E., Mahani, A.: Gravitational search algorithm with both attractive and repulsive forces. Soft. Comput. **23**(3), 783–825 (2019). https://doi.org/10.1007/s00500-017-2785-2

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.