ORIGINAL ARTICLE



Improved coral reefs optimization with adaptive β -hill climbing for feature selection

Shameem Ahmed¹ S · Kushal Kanti Ghosh¹ · Laura Garcia-Hernandez² · Ajith Abraham³ · Ram Sarkar¹

Received: 19 June 2020 / Accepted: 29 September 2020 © Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

For any classification problem, the dimension of the feature vector used for classification has great importance. This is because, in a high-dimensional feature vector, it is found that some are non-informative or even redundant as they do not contribute to the learning process of the classifier. Rather, they may be the reason for low classification accuracy and high training time of the learning model. To address this issue, researchers apply various feature selection (FS) methods as found in the literature. In recent years, meta-heuristic algorithms have been proven to be effective in solving FS problems. The Coral Reefs Optimizer (CRO) which is a cellular type evolutionary algorithms has good tuning between its exploration and exploitation ability. This has motivated us to present an improved version of CRO with the inclusion of adaptive β -hill climbing to increase the exploitation ability of CRO. The proposed method is assessed on 18 standard UCI-datasets by means of three distinct classifiers, KNN, Random Forest and Naive Bayes classifiers. It is also analyzed with 10 state-of-the-art meta-heuristics FS procedure, and the outputs show an excellent performance of the proposed FS method reaching better results than the previous methods considered here for comparison. The source code of this work is publicly available at https://github.com/ahmed-shameem/Projects.

Keywords Meta-heuristic · Feature selection · UCI · Coral reefs optimization · Adaptive β -hill climbing · Hybrid optimization

Laura Garcia-Hernandez ir1gahel@uco.es

Shameem Ahmed shameemahmed 20 apr 2000 @gmail.com

Kushal Kanti Ghosh kushalkanti 1999@gmail.com

Ajith Abraham ajith.abraham@ieee.org

Ram Sarkar rsarkar@ieee.org

- ¹ Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
- ² Area of Project Engineering, University of Cordoba, Córdoba, Spain
- ³ Machine Intelligence Research Labs, Auburn, WA, USA

1 Introduction

With the rapid advancement in the domain of computer and technology, the amount of data to be dealt with is increasing exponentially in almost every field that includes financial analysis, business management, social networks, medical studies, combinatorial chemistry, image processing and so on [5, 7, 8, 21, 29, 66], etc. But for retrieving useful information from high dimensional datasets, researchers face some challenges. Firstly, processing high dimensional datasets requires huge computation power. Secondly and most importantly, often these data-sets contain many redundant or irrelevant information or attributes, which cause overfitting problem of the learning algorithms [59]. As a result, learning algorithms may not perform as desired. In such cases, feature selection (FS) [34] plays an important role as a pre-processing step which is used to eliminate the possible non-informative and irrelevant features from the original datasets (here considered as feature

vector) [28]. In other words, FS is a process of choosing the most informative subset of features to be employed for model building in the categorization of the data under consideration. This also helps in decreasing the computational time requirement, and increasing the classification accuracy of that machine learning or data mining algorithm in consideration, and reduces storage space requirement [15, 18].

We can divide FS methods into two different categories based on evaluation criteria of features—filter and wrapper. Filter methods [32, 38] evaluate features based on some predefined criteria. Some examples are Symmetrical Uncertainty [40], Mutual Information [43], Chi-square [69], Laplacian Score [36], etc. Filter methods do not employ a learning algorithm. In contrast, wrapper methods [35] utilize learning algorithms (i.e., classifiers) to evaluate features and thereby selecting the optimal feature subset [49]. Due to the requirement of the learner in the latter method, the computational cost is greater than the filter method. But, generally, the wrapper methods provide better results than filter methods [31].

In recent times, meta-heuristic algorithms are quite common for solving optimization problems of different types due to their non-derivative nature, flexibility, and their ability to avoid local optima [51]. Two main aspects of a meta-heuristic algorithm are [16]: exploration and exploitation. Exploration guarantees the algorithm to achieve different favorable areas of the search space (i.e., global search) for a given problem, whereas exploitation guarantees the exploring of optimal individuals within the given area (i.e., local search). The fine-tuning between these components decide the goodness of an algorithm. It is tough to compensate these elements due to their stochastic properties.

Genetic algorithm (GA) [45] and Particle swarm optimization (PSO) [41] are the oldest and most famous metaheuristic algorithms. GA is based on the idea of the Darwinian theory of evolution. GA uses crossover and mutation operators to produce better solutions. PSO is a swarmbased algorithm that simulates the functioning of birds flying together in flocks searching for food. In [52], a local search is applied to improve the search capability of GA. In the work stated in [31], GA is employed for FS on Microarray datasets. In [46], binary version of PSO is proposed. In [14], binary PSO is used for FS on Sonar and Iris datasets. In [39], GA and PSO are hybridized and applied for FS on digital mammogram datasets. In the work reported in [27], hybrid of GA and PSO is used for FS on Indian Pines hyperspectral dataset. Simulated annealing (SA) [42] is proposed following the annealing procedure, i.e., heating and then controlled cooling to increase the strength of metals. Gravitational search algorithm (GSA) [54] utilizes Newton's law of gravitation between masses and their interaction to update the positions toward the optimal point. PSO and GSA have been hybridized in [44].

Social mimic optimizer (SMO) [13] is proposed by following the human nature of mimicking more successful humans, and applied for FS [30] on 18 standard UCI datasets, with a novel X-shaped transfer function. Following the mating procedure of barnacles, Barnacles mating optimization (BMO) algorithm [58] is proposed. Atom search optimization (ASO) [68] is proposed which is inspired by the movement of atom following Lennard– Jones (L–J) potential and applied for FS on 22 standard UCI datasets in the work reported in [62]. Butterfly optimization algorithm (BOA) [12] is proposed by mimicking the food searching and mating behavior of butterflies. In the work reported in [11], the authors have proposed binary version of BOA and applied for FS on 21 UCI datasets.

Based on the 'inspiration' point of view, classification of the meta-heuristic algorithms is depicted in Fig. 1.

Therefore, the presence of numerous meta-heuristic FS algorithms in the literature questions the need for another meta-heuristic FS method. However, No Free Lunch [65] theorem states that a single meta-heuristic algorithm cannot solve every optimization problem present. As FS is considered to be an optimization problem [49], hence, the research in this field is still ongoing, which is the inspiration of our present work where we have proposed a hybrid meta-heuristic FS method based on coral reefs optimization (CRO) [55]. CRO artificially replicates the natural behavior in the coral reefs. This way, each coral that made up the reef is considered as candidate solution to the optimization problem of our interest. In this ecosystem, each coral grows up and reproduces by means of sexual and asexual operations, battling with each other for being allocated in the coral reef. This battle for survival and reproduction characteristics have given researchers to come up with a robust meta-heuristic algorithm. In the past, CRO has been applied on many benchmark problems (both continuous and discrete), as well as in different areas and fields. These experiments produce promising results. Work reported in [17] utilizes CRO for environmental protection fields, emission reduction and cost reduction, etc. In the work reported in [56], the authors have applied CRO with the operators of Harmony search algorithm [26] and is used in short-term wind speed prediction for obtaining optimal meteorological variables which are used as input to a Extreme Learning Machine (ELM) network. The work reported in [57] proposes an improved version of CRO which is utilized for large scale continuous optimization. The work reported in [50] has used CRO for solving clustering problems. A modified version of CRO is proposed in [67] by hybridized with SA and applied for FS for high dimensional medical datasets. CRO has been deployed to solve unequal area facility layout problem in **Fig. 1** A general classification of meta-heuristic optimization algorithms based on their source of 'inspiration'



[22–25]. The CRO algorithm is a cellular type evolutionary scheme [4], which denotes that it can be implemented as a fully parallel program in a group of microprocessors. As CRO utilizes operators whose ideas are embraced from evolutionary computation and SA, hence it has a good trade-off between its exploration and exploitation capabilities. These facts have motivated us to explore its effectiveness in the FS domain by enhancing its exploitation ability.

The present work proposes a binary version of a hybrid of CRO, known as A β CRO, with another lately suggested meta-heuristic algorithm, *namely* Adaptive β -hill climbing (A β HC) [2]. A β HC is previously evaluated against 23 global optimization functions to prove its efficiency. It has also been used for FS along with Sailfish optimizer [28].

For hybridizing meta-heuristic algorithms, generally, two different approaches are followed [61]: high-level and low-level. In the high-level hybridization, the meta-heuristics are executed in sequence, one after another. In the low-level hybridization, a function or sub-procedure in a meta-heuristic is interchanged by another meta-heuristic or its sub-procedure. We have followed the high-level approach to hybridize CRO and $A\beta$ HC, following a pipe-line model, where the output of one meta-heuristic optimization algorithm is considered as the input of another optimization algorithm.

To the best of our knowledge, this is the first time CRO is hybridized with $A\beta$ HC algorithm for finding optimal solution for FS problem. In short, the main offering of this research are as follows:

- A new FS method, *namely* AβCRO is proposed by hybridizing CRO and recently proposed AβHC.
- 18 standard UCI datasets [19] are used to evaluate the proposed AβCRO by means of K-nearest Neighbors (KNN), Naive Bayes and Random Forest classifiers.

 The suggested FS method, AβCRO is analyzed against 10 state-of-the-art meta-heuristic and hybrid metaheuristic FS methods.

The remainder of this paper is structured as follows: Sect. 2 provides a brief overview of CRO and A β HC algorithms. Section 3 provides a detailed description of the proposed FS method. The experimentation performed to validate the proposed method is reported in Sect. 4. Additionally, this section display the achieved results obtained by our novel approach and discuss them comparing with other state-of-the-art solutions. In Sect. 5, the proposed A β CRO is compared *w.r.t.* the achieved classification accuracies and selected number of features, against 10 state-of-the-art meta-heuristic and hybrid meta-heuristic FS methods. At the end, Sect. 7 closes this article, debates its limitations and offers a promising future extension of this research.

2 Pre-requisites

2.1 Coral reefs optimization: an overview

CRO [55] is inspired by the fight for survival phenomenon of corals. This algorithm mainly has two stages, reef formation and coral reproduction.

In the reef formation stage, the algorithm is initialized by considering a model of reefs consisting of $\mathcal{X} \times \mathcal{Y}$ square grids. Each such square is meant to allocate a coral. A square is uniquely identified as its position (m, n), where 'm' denotes the row number and 'n' denotes the column number. We start by assigning some squares in the grid to be occupied by corals randomly. The others are left empty, for new corals to freely settle and grow in the later stages. There is a ratio between the occupied and unoccupied reefs, $\kappa_i \in (0,1)$, which is an important parameter for CRO. The next phase of reef formation is completed by different reproduction operators and involves the settling of new larvae in the reef. Each stage of CRO produces a coral reef larva, which is associated with a fitness value. The reef larvae with better fitness value survive longer, while the other gets perished eventually, to ensure better solutions in population.

CRO executes the stages of reproduction iteratively until the terminating condition is met. The next phase of the algorithm consists of four sub-phases: sexual reproduction (external and internal), larvae settling, asexual reproduction, and depredation in polyp stage.

- Sexual Reproduction
 - Broadcast Spawning (External Sexual Reproduction) In this sub-phase, a part of the population is selected randomly, let's say \mathcal{F}_{h} fraction of existing coral, out of which, corals are selected at random in pairs for sexual crossover. The coral larvae so produced are then released in water.
 - Brooding (Internal Sexual Reproduction) $(1 \mathcal{F}_h)$ fraction of the total population is selected at every step. The new offspring are produced by random mutation of the existing corals. These offspring are also released in water following the similar fashion as broadcast spawning.

- Larvae settling After the formation of offspring by broadcast spawning and brooding, they try to settle into the reefs. For this purpose, their strength is computed by a suitable fitness function. If any reef is already occupied beforehand then the new larva can only settle there if its fitness value is better than the already occupying coral. Each larva is given α number of chances to settle, after α unsuccessful attempts the larva is depredated. If the reef is empty, the new larva settles there irrespective of its fitness value.
- Asexual reproduction Corals perform asexual reproduction by budding or fragmentation. The population is sorted according to their fitness values, out of which certain fraction, \mathcal{F}_a is selected to duplicate itself in order to perform asexual reproduction.
- Depredation At each step, a small fraction of the coral population gets depredated by other corals based on their fitness value, thus liberating space for the new corals to settle there. This operator is employed with minimal probability, \mathcal{P}_d and only to a fraction \mathcal{F}_d of the worst fitness.

For the sake of our convenience, we can use $\mathcal{F}_d = \mathcal{F}_a$ or $\mathcal{F}_d + \mathcal{F}_d \leq 1$ as the relation between \mathcal{F}_d and \mathcal{F}_s . Algorithm 1 presents the pseudocode of CRO algorithm.

Algorithm 1 Pseudo-code of CRO algorithm

Input: Problem dependent information (the fitness function, $\mathcal{R}, \mathcal{R}_p, \mathcal{L}, \mathcal{X} \times \mathcal{Y}$) **Output**: The best coral 1: Initialize $\mathcal{R}, \mathcal{R}_p$ and parameter values $(\mathcal{F}_b, \mathcal{F}_a, \mathcal{P}_d, \mathcal{F}_d, \alpha, \kappa_i)$ 2: Calculate fitness value of each coral using Equation (1) 3: while Stopping criterion is not met do 4: Perform sexual crossover operator using broadcast spawning for Couples of broadcast spawning corals \mathcal{P}_1 and \mathcal{P}_2 do 5: 6: $\mathcal{P}_1 + \mathcal{P}_2 \longrightarrow \mathcal{O}_1 + \mathcal{O}_2$ 7: end for 8: Perform sexual crossover using brooding 9: for Each brooding coral \mathcal{P} do $\mathcal{P} \longrightarrow \mathcal{O}$ 10: end for 11: 12:Settle new larvae (as per the fitness values) 13:Perform asexual reproduction using budding for Each coral which would perform budding do 14:15: $\mathcal{P}_b \longrightarrow \mathcal{P}_b$ end for 16:17:Perform coral depredation 18:Compute the fitness value of each coral 19:Obtain the current optimal solution with best fitness

- 20: end while
- 21: Return the best coral

2.2 Adaptive β -hill climbing: an overview

A β HC [2] is an actually proposed meta-heuristic algorithm, an adaptive version of the β HC [1], which, in turn, is an enhanced version of the popular hill climbing (HC) algorithm. HC is a quite simple local search algorithm. But its main drawback is, more often than not it gets stuck in local optima, and consequently fails to reach global optima. β HC is proposed to overcome this problem. However, in β HC, parameter tuning is quite an issue and it requires exhaustive experiments for each of the problems in consideration. To avoid these exhaustive experiments to set the optimal values of parameters, an adaptive version, *namely* A β HC is proposed.

Given a particular solution $X = (x_1, x_2, ..., x_D)$, A β HC iteratively computes an improved solution $X'' = (x''_1, x''_2, ..., x''_D)$ by using its two operators: \mathcal{N} -operator (Neighborhood operator) and β -operator. The pseudo-code of this A β HC is given in Algorithm 2.

column number. The grids which are occupied by corals are set to 1, and the empty grids as 0. In Fig. 2a, every occupied square \mathcal{R}_{mn} in \mathcal{R} represents a coral larva of the coral population \mathcal{R}_{mn} . Figure 2b shows the initial population \mathcal{R}_p containing $\mathcal{X} \times \mathcal{Y} \times \kappa_i$ larvae. Each coral larva is represented as a binary vector of length \mathcal{L} . This vector contains a series of 0s and 1s representing the feature subset, where 1 means that feature is selected and 0 means it is not. For example, the \mathcal{R}_{11} in \mathcal{R} is represented as \mathcal{R}_{11} in \mathcal{R}_p . We have represented different corals in \mathcal{R} with different shades and colors. A binary individual in \mathcal{R}_p represents a potential solution, i.e., a feature subset. Each \mathcal{R}_{mn} in \mathcal{R}_p has several features denoted by $\mathcal{F} = (f_1, f_2, \dots, f_D)$, where \mathcal{D} is the total number of features whose values are set to be 1 if selected and 0 if not selected, which is accomplished randomly.

Algorithm 2 Pseudo-code of adaptive β -hill climbing algorithm
Input : $X = (x_1, x_2,, x_D)$
Output : $X = (x_1, x_2,, x_D)$ (improved)
1: Initialize $\beta_{min}, \beta_{max}, K, t_{max}$
2: for $t = 1, 2, \ldots, t_{max}$ do
3: $X' := X$
4: $C_t = \left(\frac{t}{t_{max}}\right)^{\frac{1}{K}}$
5: $\mathcal{N}_t = 1 - C_t$
6: $RandIndex \in (1, D)$
7: $X'_{RandIndex} := X'_{RandIndex} \pm \mathcal{N}_t$
8: $X'' := X'$
9: $\beta_t = \beta_{min} + t \times \frac{\beta_{max} - \beta_{min}}{t_{max}}$
10: for $i = 1,, D$ do
11: if $random(0,1) < \beta_t$ then
12: $X''[i] := random(LB_i, UB_i)$
13: end if
14: end for
15: if $fitness(X'') \leq fitness(X)$ then
16: $X := X''$
17: end if
18: end for

3 Proposed work

In this section, we describe the binary encoding version of CRO, for solving the problem of FS. Then we discuss the fitness function used in this work. Finally, the underlying steps of the proposed $A\beta$ CRO used for FS are described.

3.1 The encoding scheme

We denote the reef as \mathcal{R} , consisting of $\mathcal{X} \times \mathcal{Y}$ square grids. We denote each grid uniquely by denoting it as (m, n), where m represents the row number and n represents the

3.2 Fitness function

The main objective of any FS method to minimize the number of features and maximize the classification accuracy of a classification problem when this feature subset is used [53]. Here we apply A β CRO to find the best feature subset and calculate the accuracy of this subset with KNN, Naive Bayes and Random Forest classifiers. Let \mathcal{A} be the accuracy of the model calculated using a classifier, d be the number of selected features and \mathcal{D} be the total number of features present in the original dataset. Hence, $(1 - \mathcal{A})$ represents the classification error and $\frac{d}{\mathcal{D}}$ represents the

fraction of features selected from the original feature set. We define the fitness function as:

$$\downarrow \text{Fitness} = \omega \cdot (1 - \mathcal{A}) + (1 - \omega) \cdot \frac{d}{\mathcal{D}}$$
(1)

where $\omega \in [0, 1]$ denotes weightage given to the classification error.

Our objective is to minimize the value of fitness function, as we try to reduce the classification error as well as the number of selected features. For each feature subset, we calculate the fitness value and decide which is better depending on whose fitness value is lower. In our case, we emphasize on selecting the minimum number of features along with reducing the error of our model and hence $\omega =$ 0.9 is used.

3.3 A β CRO: the proposed method

In the proposed method, a coral larva exemplify a solution, which is settled by competing with other corals for growth space. Here, $A\beta$ HC is used to escape local optima. The proposed method, $A\beta$ CRO is established by combining the benefits of global search algorithm CRO and local search of $A\beta$ HS to find the optimal feature subset. The steps used in the proposed method are given as follows:

- *Initialization* In this stage, we initialize the reef \mathcal{R} and the coral population \mathcal{R}_p randomly. For FS, each entry is initialized with binary value, i.e., 0 or 1. κ_i is set to 0.6. This stage tells us which squares are empty and which features are selected or rejected.
- Broadcast spawning We implement the broadcast spawning using the two-point crossover [60]. We select two parents \mathcal{P}_1 and \mathcal{P}_2 randomly for external sexual reproduction to produce two offspring \mathcal{O}_1 and \mathcal{O}_2 . Illustration of this step is provided in Fig. 3.
- *Brooding* To perform brooding, we select a parent \mathcal{P} randomly to produce an offspring \mathcal{O} through random mutation. The idea is to select some positions of the parent and invert them to produce the new offspring. The method is illustrated in Fig. 4.

1	1	0	0	1	
1	0	1	1	0	
0	1	0	1	1	
1	0	1	1	0	
0	1	1	0	1	
1	1	0	1	0	
(a) $\operatorname{REEF}(\mathcal{R})$					

- *Larvae settling* To settle on the reefs, a larva gets α number of chances. After α number of unsuccessful attempts, we eliminate it from the population considering it as not fit enough to survive. If any position in the reef is already filled by another larva, to settle a new larva there, we compare their fitness values. The fitter one settles there and the other one gets perished. If the position is empty, we allow the larva to settle there without considering its fitness value.
- Budding or fragmentation Budding is a process where the parent \mathcal{P}_b produces buds from its body to produce new offspring, which are genetically similar to their parent. In fragmentation, the parent's body gets divided into pieces to produce genetically similar offspring. We achieve this by selecting a certain ratio of the coral population having the best fitness values and then replicate them.
- Depredation In every step, we eliminate a certain portion of the coral population based on the fitness value to make room for new and better offspring. For our case, we use \mathcal{F}_d as 0.1.
- *Fitness evaluation* To evaluate and compare the effectiveness of a solution, we determine the fitness value using Eq. (1). This value gives us the insight to decide which feature subset to be considered as the optimal one.
- Update solutions Every potential solution is updated using $A\beta$ HC. $A\beta$ HC tries to find a neighbor with better fitness value than the current solution has (Fig. 5).

4 Experimentation and discussion

We have used three classifiers, KNN [6], Naive Bayes and Random Forest, to calculate the classification accuracies with the feature subset selected by the proposed FS model. As per the recommendations provided in [20, 48, 49], we divided the datasets into two parts: training and testing. 80% of the dataset is employed in order to train the

R ₁₁	1	1	0	1	0	 0	1	0	1
R ₁₂	1	1	1	1	0	 1	1	0	0
R15	0	1	1	1	0	 0	1	0	1
R ₂₁	1	0	0	1	1	 0	0	1	1
•••	0	1	1	0	0	 1	1	1	1
R64	1	0	0	1	1	 0	1	0	1

(b) Coral Population (\mathcal{R}_p)

Fig. 2 Representation of a reef and a population considered in CRO based FS method

classification model, and the rest 20% for testing purpose. The suggested FS model is implemented by means of Python3 [63] and graphs are plotted employing Matplotlib [37].

4.1 Dataset definition

In order to inspect the performances of CRO and A β CRO, 18 standard UCI datasets [19] are used. The datasets are chosen from varied domains [9, 10].

As the datasets used here are divergent in terms of the number of features and instances, so it helps us to understand the firmness of the suggested FS method. We have summarized the datasets below. For quick reference, we can have a look at Table 1.

4.2 Software

The conducted experiments are executed on a system with 8th Gen, i5 processor having 4 GB RAM. Each dataset are run 15 times and the best result is taken for further experiments. The execution time in seconds for each dataset using CRO and A β CRO for a single run is provided in Table 2.

4.3 Tuning parameters

It is to be noted that for a multi-agent evolutionary algorithm, two parameters are very important which are population size or search agents and the maximum number of iterations applied to execute the algorithm. Population size reflects how a single agent gets the chance to learn from the other agents present in the population. On the other hand, the maximum number of iterations is used to terminate the evolutionary algorithm. In order to determine proper values for these parameters, experiments have been carried out by varying one parameter *w.r.t.* the other.

Figure 6 displays the impact of the population size on the classification accuracy of a classifier by means of the suggested FS method. Figure 7 shows the value of the fitness function achieved in each iteration of the algorithm. Table 3 shows the value of the parameters that are used in the proposed FS method. Considering the constraint of computational time, we proceed further keeping in mind the parameter values given in Table 3.

4.4 Complexity analysis

The complexity analysis of any meta-heuristic algorithm mainly depends on the searching time. The initialization of parameters and the fitness evaluation also affect the time complexity. In our case, the initialization process takes O(n) time, where 'n' is the number of corals considered. The time complexity of fitness evaluation depends upon the classifier used, so in general, we can say fitness evaluation takes O(fit) time. The main operations which are involved for the searching procedure include broadcast spawning, brooding, larvae settling, budding and A β HC. Broadcast spawning takes $O(d^2)$ and brooding takes O(d) time, where 'd' is the dimension of the problem. The larvae settling stage takes $O(n^2)$ time and budding has time the complexity of O(n). A β HC has complexity of O(mI * d), where 'ml' is the number of iterations considered for A β HC. If we consider 'maxIter' as maximum number of iteration for the whole search process, then the total time complexity of A β CRO is given by Eq. 2.





Fig. 5 Flowchart of the proposed $A\beta$ CRO

 $\overline{\textcircled{D}}$ Springer

 Table 1
 Elucidation of the datasets used here to evaluate the proposed

 FS
 method

Sl. no.	Dataset	#Attributes	#Samples	Dataset domain
1	Breastcancer	9	699	Biology
2	Tic-tac-toe	9	958	Game
3	WineEW	13	178	Chemistry
4	HeartEW	13	270	Biology
5	Exactly	13	1000	Biology
6	Exactly2	13	1000	Biology
7	M-of-n	13	1000	Biology
8	Zoo	16	101	Artificial
9	Vote	16	300	Politics
10	CongressEW	16	435	Politics
11	Lymphography	18	148	Biology
12	SpectEW	22	267	Biology
13	BreastEW	30	569	Biology
14	IonosphereEW	34	351	Electromagnetic
15	KrvskpEW	36	3196	Game
16	WaveformEW	40	5000	Physics
17	SonarEW	60	208	Biology
18	PenglungEW	325	73	Biology

Table 2 Execution time (in s) for a single run of 18 standard UCI datasets using CRO and A β CRO

Sl. no.	Dataset	CRO	ΑβCRO
1	Breastcancer	3.134	29.499
2	Tic-tac-toe	4.517	47.774
3	WineEW	2.186	22.239
4	HeartEW	2.224	20.48
5	Exactly	5.34	57.743
6	Exactly2	4.405	44.094
7	M-of-n	5.383	59.011
8	Zoo	1.782	20.479
9	Vote	2.063	19.888
10	CongressEW	2.449	22.575
11	Lymphography	2.064	20.088
12	SpectEW	2.391	24.898
13	BreastEW	3.414	33.938
14	IonosphereEW	3.163	30.136
15	KrvskpEW	67.988	525.761
16	WaveformEW	235.347	2043.582
17	SonarEW	3.177	30.985
18	PenglungEW	2.864	54.751

4.5 Result analysis

This section provides the findings of the suggested FS method called A β CRO, when experimented on the datasets mentioned in Sect. 4.1.

Tables 4, 5 and 6 provide the outcomes reached by the proposed $A\beta$ CRO algorithm when evaluated by KNN, Naive Bayes and Random Forest classifiers respectively. Compared to the basic CRO algorithm, the obtained results show the effectiveness of $A\beta$ CRO in finding a better solution. It can be extracted that the suggested $A\beta$ CRO algorithm works the best over UCI datasets with KNN classifier. Also, KNN classifier is widely employed in the references for FS on UCI datasets [20, 48, 49]. Hence, for further experiment and discussion, we have utilized KNN classifier with K = 5.

Observing Table 4, we can see that $A\beta$ CRO produces accuracy > 90% in 15 datasets (83.33%). Amongst these, for 10 datasets (55.55%), namely BreastEW, WineEW, Exactly, M-of-n, Zoo, Vote, CongressEW, Lymphography, SonarEW and PenglungEW, it produces 100% accuracy. Out of 18 datasets, it has achieved the highest classification accuracy in case of all the datasets. Analyzing these results with the obtained results by the basic CRO algorithm, we can see that CRO produces equivalent results in 13 datasets. Although in case of Tic-tac-toe, HeartEW, BreastEW, KrvskpEW, and WaveformEW, A β CRO outperforms CRO in terms of achieved classification accuracy.

Talking about the number of selected features, in 16 datasets (88.89%) A β CRO selects the least number of features. Of these, in 10 datasets, namely Breastcancer, WineEW, Exactly, Exactly2, M-of-n, Zoo, Vote, CongressEW, BreastEW and Lymphography, CRO selects the same number of features. In the case of HeartEW and WaveformEW, CRO outperforms A β CRO. Now, considering the achieved classification accuracy and number of selected features, we can say that A β CRO is able to perform better than CRO while evaluated with KNN classifier.

From Table 5 we can say that A β CRO produces classification accuracy > 90% in 15 datasets (83.33%) while evaluated by Naive Bayes classifier. It achieves 100% accuracy in 8 datasets (44.44%). Whereas, it achieves the highest accuracy in 17 datasets (94.44%). In the case of Tic-tac-toe, WineEW, HeartEW, Exactly2, Zoo and Vote, it produces an equivalent result as the CRO. Only in the case of SpectEW, CRO outperforms A β CRO. So overall, we can say in terms of classification accuracy A β CRO produces better results than CRO.

Considering the number of selected features, $A\beta CRO$ selects the least number of features in 14 datasets



Fig. 6 Graphs showing the impact of the population size on classification accuracy of the classifiers when tested over 18 UCI datasets using CRO and $A\beta$ CRO

Deringer



Fig. 7 Graphs showing the convergence of solution using best fitness value obtained in each iteration for 18 UCI datasets using CRO and $A\beta$ CRO

(77.77%). It selects the same number of features as selected by CRO in the case of HeartEW, Exactly2, Vote,

CongressEW and SpectEW. CRO outperforms $A\beta$ CRO in BreastEW, Exactly, KrvskpEW and M-of-n. So,

Table 3Value of theparameters used in the proposedFSmethod

Parameters	Values
\mathcal{R}	5
Iterations	30
${\cal F}_b$	0.6
\mathcal{F}_{a}	0.4
${\cal F}_d$	0.1
κ _i	0.6
α	3
\mathcal{P}_d	0.1

considering the achieved classification accuracy along with number of selected features, we can say that A β CRO has an upperhand while evaluated by Naive Bayes classifier.

If we look at Table 6, we can see that $A\beta$ CRO produces classification accuracy > 90% in 16 datasets (88.89%) while evaluated using Random Forest classifier. It produces 100% accuracy in 6 datasets (33.33%). It also achieves the highest accuracy in 16 datasets (88.89%), out of which it provides equivalent result in 7 datasets with CRO and 3 datasets with Random Forest classifier. CRO is unable to beat $A\beta$ CRO in any case in terms of achieved classification accuracy but Random Forest classifier outperforms $A\beta$ CRO in cases of Tic-tac-toe and WaveformEW.

Looking at the number of selected features, we can say that A β CRO selects least number of features in 12 datasets (66.67%). It selects same number of features as CRO in 4

datasets. In the case of Tic-tac-toe, HeartEW, BreastEW, SpectEW, KrvskpEW and SonarEW, CRO outperforms A β CRO. Now, considering the produced classification accuracy and number of selected features, we can say for sure that A β CRO outperforms CRO while being evaluated by Random Forest classifier.

As we can witness the superiority of A β CRO over CRO from the above discussion and Tables 4, 5 and 6, we can state that A β CRO outperforms CRO in every aspect while evaluated using KNN, Naive Bayes and Random Forest classifiers. Generalizing this, we can say that A β CRO has the capability to find a better solution than CRO and hence producing better results in terms of achieved classification accuracy and the number of selected features.

From Tables 4 and 5 if we compare the results obtained by the proposed method when evaluated by KNN and Naive Bayes classifiers only in terms of achieved classification accuracy, we can say that the proposed method produces better results while evaluated using KNN than using Naive Bayes in 8 datasets (44.44%). In 10 datasets (55.55%) they produce equivalent results. So, while evaluated on Naive Bayes, it cannot beat $A\beta$ CRO in any case while evaluated using KNN classifier.

Considering the number of selected features, $A\beta$ CRO produces better results while evaluated by KNN than Naive Bayes in 4 datasets only. They select the same number of features in 6 datasets. In the rest 8 datasets, $A\beta$ CRO selects fewer features while evaluated using Naive Bayes. Looking at the number of selected features, it may seem that

Dataset	Original		CRO		ΑβCRO	
	Accuracy	Features	Accuracy	Features	Accuracy	Features
Breastcancer	96	9	99.28	3	99.28	3
Tic-tac-toe	81.1	9	83.854	6	84.375	5
WineEW	66.67	13	100	2	100	2
HeartEW	68.15	13	90.74	4	94.44	7
Exactly	72.3	13	100	6	100	6
Exactly2	73.3	13	76	1	76	1
M-of-n	87.4	13	100	6	100	6
Zoo	87	16	100	5	100	5
Vote	92.33	16	100	1	100	1
CongressEW	92.18	16	100	3	100	3
Lymphography	81.33	18	100	6	100	6
SpectEW	82.22	22	94.44	7	94.44	5
BreastEW	92.63	30	97.36	4	100	4
IonosphereEW	83.43	34	98.57	11	98.57	3
KrvskpEW	96.1	36	98.435	14	98.748	10
WaveformEW	81.44	40	85.8	16	87.8	17
SonarEW	80.95	60	100	21	100	11
PenglungEW	81.33	325	100	108	100	77

Table 4 Performance of CRO and $A\beta$ CRO in terms of classification accuracy and number of selected features using KNN-classifier (highest classification accuracies and lowest no. of selected features are highlighted) **Table 5** Performance of CRO and $A\beta$ CRO in terms of classification accuracy and number of selected features using Naive Bayes classifier (highest classification accuracies and lowest no. of selected features are highlighted)

Dataset	Original		CRO		$A\beta CRO$	
	Accuracy	Features	Accuracy	Features	Accuracy	Features
Breastcancer	89.28	9	97.87	3	98.57	2
Tic-tac-toe	75.52	9	72.92	6	72.92	5
WineEW	100	13	100	3	100	2
HeartEW	94.44	13	96.296	4	96.296	4
Exactly	69.5	13	96	1	100	6
Exactly2	76	13	76	1	76	1
M-of-n	96.5	13	98.5	6	100	7
Zoo	100	16	100	5	100	3
Vote	98.33	16	100	3	100	3
CongressEW	98.85	16	98.85	1	100	1
Lymphography	86.67	18	90	5	100	4
SpectEW	72.22	22	94.44	4	92.59	4
BreastEW	96.49	30	97.36	2	100	4
IonosphereEW	95.71	34	92.88	8	98.57	5
KrvskpEW	65.88	36	95.31	9	97.18	12
WaveformEW	82.2	40	82.5	15	85.8	10
SonarEW	80.95	60	80.95	23	97.61	11
PenglungEW	60	325	73.33	120	93.33	50

Table 6Performance of CROand $A\beta$ CRO in terms ofclassification accuracy andnumber of selected featuresusing Random Forest classifier(highest classificationaccuracies and lowest no. ofselected features arehighlighted)

Dataset	Original		CRO		ΑβCRO	
	Accuracy	Features	Accuracy	Features	Accuracy	Features
Breastcancer	97.8	9	97.86	3	97.86	2
Tic-tac-toe	95.8	9	85.94	6	93.23	8
WineEW	100	13	100	4	100	3
HeartEW	81.5	13	88.89	3	94.44	8
Exactly	78.5	13	100	6	100	6
Exactly2	74	13	76	1	76	1
M-of-n	100	13	100	6	100	6
Zoo	100	16	100	4	100	3
Vote	95	16	98.33	3	98.33	1
CongressEW	97.7	16	98	1	98.85	1
Lymphography	90	18	93.33	8	96.67	4
SpectEW	88.9	22	90.74	5	96.3	7
BreastEW	98.2	30	95.61	2	100	4
IonosphereEW	91.4	34	97.14	8	98.57	5
KrvskpEW	99.5	36	98.12	8	99.53	17
WaveformEW	85.8	40	83	15	85.3	14
SonarEW	90.7	60	92.86	13	95.24	14
PenglungEW	86.7	325	93.33	140	100	103

A β CRO performs better while evaluated using Naive Bayes, but considering the achieved classification accuracy, it shows just the opposite. As classification accuracy has utmost importance for us, we can say that A β CRO produces better results while evaluated using KNN classifier. Observing Tables 4 and 6 we can say that A β CRO produces better results in 6 datasets (33.33%) while evaluated using KNN than using Random Forest. They produce equivalent results in 9 datasets (50%). Although when A β CRO is evaluated using Random Forest, it produces

Table 7 Comparison of CRO and $A\beta$ CRO based on Precision, Recall and fscore evaluated on 18 standard UCI datasets

Sl. no.	Dataset	CRO			AβCRO	ΑβCRO		
		Precision	Recall	fscore	Precision	Recall	fscore	
1	Breastcancer	0.948	0.957	0.953	1	1	1	
2	Tic-tac-toe	0.776	0.778	0.778	0.864	0.8	0.8204	
3	WineEW	0.969	0.976	0.971	0.976	0.976	0.976	
4	HeartEW	0.87	0.867	0.868	0.9	0.879	0.884	
5	Exactly	0.345	0.5	0.408	1	1	1	
6	Exactly2	0.38	0.5	0.432	0.38	0.5	0.432	
7	M-of-n	1	1	1	1	1	1	
8	Zoo	0.452	0.571	0.495	1	1	1	
9	Vote	0.909	0.916	0.913	0.9647	0.9647	0.9647	
10	CongressEW	0.899	0.908	0.903	0.961	0.966	0.964	
11	Lymphography	0.556	0.569	0.56	0.6	0.62	0.6	
12	SpectEW	0.398	0.5	0.443	0.651	0.702	0.662	
13	BreastEW	0.962	0.962	0.962	0.9741	0.97	0.97	
14	IonosphereEW	0.941	0.88	0.9	0.95	0.9	0.9	
15	KrvskpEW	0.9518	0.949	0.9496	0.981	0.982	0.981	
16	WaveformEW	0.812	0.813	0.812	0.825	0.8365	0.838	
17	SonarEW	0.833	0.829	0.831	0.87	0.9	0.8542	
18	PenglungEW	0.752	0.738	0.726	0.82	0.84	0.8	

Best achived values are highlighted in bold

Table 8 Pairwise *p* values obtained by Wilcoxon rank-sum test considering entire feature set and selected features by A β CRO using different classifiers ($p \le 0.05$ are highlighted)

	KNN	$A\beta CRO + KNN$	Random forest	$A\beta CRO + Random forest$	Naive Bayes	$A\beta CRO + Naive Bayes$
KNN	_	0.000	0.000	0.000	0.248	0.004
CRO+KNN	0.000	-	0.019	0.754	0.000	0.006
Random_forest	0.000	0.019	_	0.008	0.010	0.642
CRO+Random_forest	0.000	0.754	0.008	-	0.001	0.033
Naive Bayes	0.248	0.000	0.010	0.001	_	0.001
CRO+ Naive Bayes	0.004	0.006	0.642	0.033	0.001	_

better results than while it is evaluated using KNN in 3 datasets (16.67%).

Talking about the number of selected features, the proposed method selects fewer features while evaluated using KNN than using Random Forest in 9 datasets. Although, in 4 datasets A β CRO selects less number of features while evaluated using Random Forest. Now, considering the achieved classification accuracy as well as the number of selected features, we can say that A β CRO produces better results while evaluated using KNN classifier.

From the above discussions, it is quite clear that $A\beta$ CRO produces better results while evaluated using KNN classifier. So, for our comparison purpose, we consider the results achieved by our algorithm when evaluated using KNN only. Also, the other state-of-the-art methods were evaluated using KNN, so the comparison is done in the common platform.

Table 7 shows the results obtained by CRO and A β CRO evaluated on 18 standard UCI datasets using precision, recall and f1 score as evaluation metrics. A β CRO achieves 100% score in Breastcancer, Exactly, M-of-n and Zoo while evaluated using precision, recall and f1 score. It is pretty evident from Table 7 that A β CRO performs much better than CRO in every dataset. In case of WineEW, the recall score is same, for Exactly2 and M-of-n all the three scores, namely precision, recall and f1 score are equivalent. In the rest cases, A β CRO outperforms CRO in all these three metrics. From these observations, we may conclude that A β CRO is able to find better solutions than CRO in almost every case.

For determining the statistical significance of the proposed A β CRO, Wilcoxon rank-sum test [64] has been carried out. It is a non-parametric statistical test and it is done to verify whether the results produced by an

🖄 Springer

Algorithm	Parameters
BGA	popSize = 8
	maxIter = 70
	Mutation and Crossover rate $= 0.8$
BPSO	popSize = 8
	maxIter = 70
	inertia factor $= 0.1$
	individual-best acceleration factor $= 0.1$
BALO	popSize = 8
	maxIter = 200
BGSA	popSize = 8
	maxIter = 20
	G = 1
	$\alpha = 20$
WOASAT-2	popSize = 10
	maxIter =100
	a in WOA = [2 0]
	selection pressure $= 0.5$
BGWOPSO	popSize = 10
	maxIter = 100
	c1 = c2 = c3 = 0.5
	$w = 0.5 + \operatorname{rand}()/2$
	$l \in [1, 0]$
WOA-CM	popSize = 10
	maxIter = 100
	a in WOA = [2 0]

 Table 9 Setting of parameter values for state-of-the-art FS methods

 used here for comparison

algorithm are statistically distinct from the results produced by other algorithms. Here, the null hypothesis is that the two sets of results produced by two algorithms are drawn from the same distribution. So any difference in the produced results comes only from sampling error. If the two set of results belong to different (statistically) distributions, then the generated p value (by Wilcoxon rank-sum test) from the test statistics will be <0.05 (level of significance), as we have done the test at 0.05% significance level, which in turn rejects of the null hypothesis (Table 8).

5 Comparison

To prove the supremacy of the proposed FS procedure, comparison with 10 state-of-the-art methods have been made. These state-of-the-art methods include four very popular meta-heuristic algorithms, namely ALO, GA, GSA and PSO. The rest six methods comprise of hybrid metaheuristic FS methods, namely BGWOPSO, adaptive switching grey-whale optimizer (ASGW), serial greywhale optimizer (HSGW), random switching grey-whale optimizer (RSGW), WOA-CM and WOASAT-2. BGWOPSO [3] is the result of hybridization between GWO and PSO. ASGW, HSGW and RSGW are three different FS methods developed by hybridizing GWO and WOA [48] in different fashion. WOA-CM [47] elevates its performance by applying both crossover and mutation operators. WOASAT-2 [49] is a hybrid of WOA and SA methods. The values of the control parameters of these FS methods are described in Table 9.

Table 10 displays the fulfillment of A β CRO in terms of achieved classification accuracy. From this table it is understood that A β CRO achieves the highest classification accuracy on 16 datasets (88.9%) which is quite impressive. In the case of Exactly2, it holds the *eighth* position along with BGWOPSO. For Tic-tac-toe, it stands at the *third* position.

Comparing A β CRO with other methods, we can observe that ASGW beats A β CRO in Tic-tac-toe and Exactly2, and produces equivalent result in WineEW, M-of-n, Zoo, BreastEW and PenglungEW. But, A β CRO outperforms ASGW in the rest 11 datasets (61.11%). A β CRO beats BALO in 17 datasets (94.44%). Only in the case of Eaxctly2, BALO outperforms A β CRO. A β CRO outperforms BGA in 15 datasets (83.33%), produces equivalent classification accuracy in Exactly and M-of-n, but loses in Exactly2 with a narrow margin. BGSA outperforms A β CRO only in the case of Exactly2, but in rest 17 datasets (94.44%), it fails to produce better results. BGWOPSO performs on equal terms in WineEW, Exactly, Exactly2, M-of-n and Zoo, but it unable to outperform $A\beta$ CRO in other cases. A β CRO beats BGWOPSO in 13 datasets (72.22%). We can say the similar about BPSO while comparing it with A β CRO. BPSO outperforms A β CRO in Exactly2 only and produces similar results in the case of Exactly and M-of-n. In the rest 15 datasets (83.33%), A β CRO has the upperhand. HSGW is able to produce the best result in Exactly2, beating A β CRO with significant margin. In the case of WineEW, Exactly, M-of-n and Zoo, it produces an equivalent result with A β CRO. But, the suggested method beats HSGW in 13 datasets (72.22%). RSGW outperforms A β CRO in the case of Tic-tac-toe and Exactly2, and produces similar result in WineEW, M-of-n, Zoo and PenglungEW. A β CRO beats RSGW in the rest 12 datasets (66.67%). WOA-CM is unable to produce a better result than $A\beta$ CRO in any other datasets and performs equivalently in the case of Exactly only. A β CRO beats WOA-CM in 17 datasets (94.44%). WOASAT-2 is unable to beat the proposed method in any dataset. It produces a similar result in the case of Exactly and M-of-n. A β CRO beats WOASAT-2 in the rest 16 datasets (88.89%).

Table 11 shows the number of selected features by the proposed method and 10 state-of-the-art methods. As we

Dataset	AβCRO	ASGW	BALO	BGA	BGSA	BGWOPSO	BPSO	HSGW	RSGW	WOA-CM	WOASAT-2
Breastcancer	0.9928	0.985	0.974	0.9743	0.9686	0.98	0.9629	0.986	0.971	0.968	0.97
Tic-tac-toe	0.84375	0.865	0.783	0.7996	0.7766	0.81	0.7996	0.828	0.859	0.785	0.79
WineEW	1	1	0.972	0.9888	0.9775	1	0.9775	1	1	0.959	0.99
HeartEW	0.9444	0.831	0.838	0.8741	0.8296	0.85	0.837	0.923	0.848	0.807	0.85
Exactly	1	0.999	0.965	1	0.994	1	1	1	0.997	1	1
Exactly2	0.76	0.777	0.762	0.77	0.77	0.76	0.768	0.815	0.779	0.742	0.75
M-of-n	1	1	0.967	1	0.994	1	1	1	1	0.991	1
Zoo	1	1	0.98	0.902	0.9804	1	0.9608	1	1	0.98	0.97
Vote	1	0.984	0.972	0.9733	0.96	0.97	0.96	0.983	0.996	0.939	0.97
CongressEW	1	0.994	0.981	0.9679	0.9633	0.98	0.9633	0.975	0.961	0.956	0.98
Lymphography	1	0.884	0.917	0.8378	0.8649	0.92	0.8919	0.934	0.893	0.852	0.89
SpectEW	0.944	0.87	0.899	0.8955	0.8433	0.88	0.8881	0.862	0.815	0.866	0.88
BreastEW	1	1	0.974	0.9754	0.9544	0.97	0.9719	0.981	0.982	0.971	0.98
IonosphereEW	0.9857	0.972	0.904	0.9489	0.9432	0.95	0.9489	0.944	0.978	0.926	0.96
KrvskpEW	0.98748	0.971	0.973	0.985	0.9549	0.97	0.9731	0.973	0.972	0.972	0.98
WaveformEW	0.878	0.746	0.797	0.7836	0.9549	0.8	0.756	0.748	0.757	0.753	0.76
SonarEW	1	0.948	0.845	0.9904	0.9135	0.96	0.9423	0.964	0.979	0.919	0.97
PenglungEW	1	1	0.827	0.9189	0.8333	0.96	0.9189	0.942	1	0.792	0.94
Average rank	1.44	3.77	5.61	4.33	7.27	4.16	5.67	3.55	3.72	7.38	4.55
Assigned rank	1	4	8	6	10	5	9	2	3	11	7

 Table 10 Comparison of classification accuracy obtained by proposed FS method with some state-of-the-art FS methods for 18 UCI datasets (highest classification accuracies are highlighted)

 Table 11 Comparison of proposed FS method with some state-of-the-art FS methods for 18 UCI datasets in terms of number of feature selected by the methods (lowest no. of selected features are highlighted)

Dataset	AβCRO	ASGW	BALO	BGA	BGSA	BGWOPSO	BPSO	HSGW	RSGW	WOA-CM	WOASAT-2
Breastcancer	3	4.867	4.7	4	4	4.4	4	5	5.933	4.302	4.2
Tic-tac-toe	5	7	5	5	4	5.2	6	7	7	6.903	6
WineEW	2	5.933	5.4	4	4	6	5	4.533	5.867	6.799	6.4
HeartEW	7	6.367	8.6	5	3	5.8	3	8.767	6.133	6.995	5.4
Exactly	6	6.867	5.75	6	4	6	6	6.7	7.1	6.045	6
Exactly2	1	7.933	1.5	1	1	1.6	1	9.033	9.2	5.252	2.8
M-of-n	6	6.867	6	6	5	6	6	6.8	7.1	6.006	6
Zoo	5	7.6	5.7	4	6	6.8	5	5.33	5.3	6	5.6
Vote	1	8.967	6.6	5	4	3.4	3	7.567	8.8	7.408	5.2
CongressEW	3	8.833	6.65	2	4	4.4	3	8.867	9.7	6.448	6.4
Lymphography	6	11.2	7.35	5	6	9.2	5	10.567	10.567	8.208	7.2
SpectEW	8	10.167	7.65	5	5	8.4	6	10.233	13.3	8.025	9.4
BreastEW	4	15.833	13.85	8	10	13.6	9	16.667	17.5	15.81	11.6
IonosphereEW	3	17.3	11.75	7	9	13	7	18.167	20.5	14.416	12.8
KrvskpEW	10	24.5	16.15	11	14	15.8	12	24.8	24.8	18.54	18.4
WaveformEW	17	25.833	20.5	15	14	14.2	15	26.933	27.533	25.4	20.6
SonarEW	11	35.3	26.6	19	24	31.2	22	34.3	36.433	35.64	26.4
PenglungEW	77	170.3	133.1	84	140	130.8	130	165.333	181.2	128.05	127.4
Average rank	2.11	7.78	5	2	2.83	4.83	2.5	7.5	8.33	6.27	4.72
Assigned rank	2	10	7	1	4	6	3	9	11	8	5

2 Springer

Table 12 Comparison of the proposed method with the top - 3 FS methods in terms of precision, recall and f-score for 18 UCI datasets

Dataset	HSGW			RSGW			ASGW			ΑβCRO		
	Precision	Recall	fscore									
Breastcancer	1	1	1	0.9552	0.9678	0.9609	0.98	0.9891	0.9843	1	1	1
Tic-tac-toe	0.6864	0.645	0.6515	0.8995	0.8204	0.8433	0.9139	0.806	0.8325	0.864	0.8	0.8204
WineEW	0.9697	0.9762	0.9718	0.9744	0.9762	0.9743	0.9697	0.9762	0.9718	0.9762	0.9762	0.9762
HeartEW	0.5907	0.5917	0.5903	0.739	0.7417	0.7393	0.85	0.85	0.85	0.9	0.879	0.884
Exactly	0.8143	0.8086	0.8113	0.7779	0.7779	0.7779	0.9006	0.8734	0.8852	1	1	1
Exactly2	0.5505	0.5192	0.5007	0.6241	0.5718	0.5766	0.5609	0.534	0.5291	0.38	0.5	0.432
M-of-n	0.9379	0.9329	0.9353	0.8527	0.8443	0.8481	0.9922	0.9865	0.9892	1	1	1
Zoo	1	1	1	0.9841	0.9286	0.944	1	1	1	1	1	1
Vote	0.951	0.943	0.9467	0.9259	0.9459	0.9314	0.9647	0.9647	0.9647	0.9647	0.9647	0.9647
CongressEW	0.9512	0.9512	0.9512	0.9662	0.9604	0.9632	0.9714	0.9815	0.9759	0.961	0.966	0.964
Lymphography	0.4234	0.4409	0.432	0.3946	0.4006	0.3944	0.4039	0.4069	0.401	0.6	0.62	0.6
SpectEW	0.7778	0.7378	0.7545	0.7473	0.6924	0.7126	0.7976	0.8171	0.8066	0.651	0.704	0.7333
BreastEW	0.9539	0.9335	0.9422	0.9672	0.9573	0.9619	0.936	0.9315	0.9337	0.9741	0.97	0.97
IonosphereEW	0.9018	0.78	0.8045	0.9327	0.86	0.8825	0.9245	0.84	0.8639	0.95	0.9	0.9
KrvskpEW	0.9646	0.9634	0.9639	0.9765	0.9764	0.9765	0.9704	0.97	0.9702	0.981	0.982	0.981
WaveformEW	0.811	0.8111	0.811	0.8202	0.8195	0.8187	0.8367	0.8361	0.836	0.825	0.8365	0.838
SonarEW	0.8413	0.8249	0.8286	0.8612	0.8513	0.8542	0.8333	0.8295	0.8309	0.87	0.9	0.8542
PenglungEW	0.881	0.9048	0.8762	0.7929	0.7857	0.7732	0.7571	0.8095	0.7651	0.82	0.84	0.8

Table 13 p values generated by the Wilcoxon rank-sum test for the classification accuracies generated by CRO and A β CRO and 10 state-of-theart meta-heuristic FS methods considered in this work

	ASGW	BALO	BGA	BGSA	BGWOPSO	BPSO	HSGW	RSGW	WOA-CM	WOASAT-2
CRO + KNN	0.041	0.000	0.001	0.000	0.001	0.000	0.022	0.022	0.000	0.001
$A\beta CRO + KNN$	0.037	0.013	0.02	0.000	0.022	0.000	0.022	0.019	0.03	0.041

can observe, in the case of 9 datasets (50%), A β CRO selects the least number of features.

For BreastCancer and WineEW, $A\beta$ CRO holds the first position. In the case of Vote, BreastEW, IonosphereEW, KrvskpEW, SonarEW and PenglungEW, it selects the least number of features. In Tic-tac-toe, it holds the *second* position along with BGA and BALO after BGSA. For HeartEW, $A\beta$ CRO stands at the *ninth* position. For

 Table
 14 Description
 of
 the
 microarray
 datasets
 used
 for

 experimentation

Sl. no.	Dataset	#Instances	#Features	#Classes
1	DLBCL	77	7070	2
2	Leukemia	72	5147	2
3	SRBCT	83	2308	4

Exactly, it stands at the *third position* along with BGA, BPSO, WOASAT-2 and BGWOPSO. In the case of Eaxctly2, it selects only one feature hence selecting the same number of features as BGA, BPSO and BGSA do. It again holds the *second* position in M-of-n after BGSA. In the case of Zoo, it stands at *second* position with BPSO after BGA. In the case of CongressEW, it is slightly behind BGA and produces an equivalent result with BPSO. In the case of Lymphography, it holds the *second* position after BGA and BPSO. BGA, BPSO, BALO and BGSA outperform A β CRO in the case of SpectEW. It stands at *fifth* position in the case of WaveformEW.

Comparison between A β CRO and top 3 state-of-the-art methods (based on achieved classification accuracy) using precision, recall and f1 score is also performed. Table 12 illustrates the comparison results. We can observe that A β CRO outperforms the other 3 methods in almost every dataset in terms of precision, recall and f1 score. From this

Terrormanee companion of the proposed method with some emisting methods on intercurrary datasets										
Dataset	GA	MA	PSO	ALO	GSA	LAHC-SSD	ECWSA	ΑβCRO		
DLBCL	97 (88)	96.4 (105)	96.2 (90)	95.1 (92)	95.7 (59)	96 (60)	96 (57)	97.6 (52)		
Leukemia	95.8 (85)	97 (65)	97 (50)	97.2 (70)	97.6 (80)	94 (61)	95.2 (50)	98.4 (47)		
SRBCT	100 (78)	100 (50)	99.1 (49)	98.7 (45)	100 (62)	99.2 (54)	100 (40)	100 (37)		

Table 15 Performance comparison of the proposed method with some existing methods on Microarray datasets

table we can say that $A\beta$ CRO is proving its superiority over the other state-of-the-art methods.

Here, Wilcoxon test has been employed in order to demonstrate that the achieved results by the suggested algorithm is statistically significant from the reached results by the 10 state-of-the-art methods considered (with KNN classifier). The *p* values achieved for pairwise comparison of the CRO and A β CRO methods with state-of-the-art algorithms are provided in Table 13.

From Tables 10 and 11 and the above discussions, we can safely say that $A\beta$ CRO produces the best results considering the achieved classification accuracy and the number of selected features as the deciding factors.

6 Additional testing on microarray data

It is evident from the previous discussion, $A\beta$ CRO is an effective algorithm for FS. In this section, additional experiments are performed to check how $A\beta$ CRO can scale to high dimensional datasets which are challenging binary optimization problems.

Microarray datasets [31] are high dimensional in nature and FS becomes difficult due to the existence of extremely large search space. So, these datasets are very effective in testing the robustness of any FS model. For this experimentation, three publicly available Microarray datasets have been considered details of which is presented in Table 14.

The classification accuracies obtained over the Microarray datasets [31] have further been compared with some classical as well as recently proposed meta-heuristics such as: Genetic Algorithm (GA), Memetic Algorithm (MA), Particle Swarm Optimization (PSO), Ant Lion Optimizer (ALO), Gravitational Search Algorithm (GSA), Social Ski-Driver Algorithm with Late Acceptance Hill Climbing (SSD-LAHC), Embedded Chaotic Whale Survival Algorithm (ECWSA). The results of these algorithms are reported in [16, 33]. Table 15 contains the classification accuracy (in %) obtained by these algorithms over the utilized Microarray datasets. The number of features used to achieve the accuracy has been provided in parenthesis corresponding to the accuracy.

7 Conclusions

Feature selection is a relevant and a fundamental pre-processing phase in the domains of data mining and machine learning. In the last two decades, many meta-heuristic algorithms have been proposed to select an optimal number of features from various high dimensional datasets. These methods have proved their effectiveness in this domain by showing promising results. CRO algorithm has an efficient exploration capability and it has also good trade-off between its exploitation and exploration capabilities. In present work, we have proposed an improved version of CRO by enhancing its searching capability with the help of a recently proposed local search algorithm, A β HC. The proposed method is named as A β CRO. To establish the effectiveness and superiority of A β CRO, it is evaluated on 18 standard UCI datasets and compared with 10 state-ofthe-art FS methods. The achieved results demonstrate that A β CRO is a good choice for FS problem. To prove the robustness of this algorithm, it is applied on microarray data; where the results have shown the effectiveness of the proposed algorithm.

There are several parameters which are used in this proposed algorithm. To exploit the perfect value for each and every parameter is a time-consuming process. To find out the best-suited combination of these parameters, it requires exhaustive experiments. On top of that, there can be some problems where $A\beta$ CRO may fail to find the optimal solution, as it may be unable to find the global optimum. Although this is in accordance with *No Free Lunch* theorem [65], it can still be treated as a weakness of the proposed method.

As future work, it would be intriguing to hybridize CRO with any other meta-heuristic method on low level. We can also apply this algorithm in other challenging research problems such as facial emotion recognition, handwriting recognition, human activity recognition, etc.

Acknowledgements This research has been partially funded by Junta de Andalucía, under the Research Project UCO-FEDER 18 REF. 1265277 MD A1. We would like to thank the CMATER research laboratory of the Computer Science and Engineering Department, Jadavpur University, India for providing us the infrastructural support.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding this work.

References

- 1. Al-Betar MA (2016) β -hill climbing: an exploratory local search. Neural Comput Appl 28(S1):153–168. https://doi.org/10.1007/ s00521-016-2328-2
- 2. Al-Betar MA, Aljarah I, Awadallah MA, Faris H, Mirjalili S (2019) Adaptive β -hill climbing for optimization. Soft Comput 23(24):13489–13512. https://doi.org/10.1007/s00500-019-03887-7
- Al-Tashi Q, Kadir SJA, Rais HM, Mirjalili S, Alhussian H (2019) Binary optimization using hybrid grey wolf optimization for feature selection. IEEE Access 7:39496–39508. https://doi.org/ 10.1109/access.2019.2906757
- Alba E, Dorronsoro B (2008) Introduction to cellular genetic algorithms. Operations Research/Computer Science Interfaces Series. Springer, Berlin, pp 3–20. https://doi.org/10.1007/978-0-387-77610-1_1
- Almomani A, Alweshah M, Khalayleh SA, Al-Refai M, Qashi R (2019) Metaheuristic algorithms-based feature selection approach for intrusion detection. In: Machine learning for computer and cyber security. CRC Press, Boca Raton, pp 184–208, https://doi. org/10.1201/9780429504044-8
- Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175. https://doi.org/ 10.2307/2685209
- Alweshah M, Alzubi OA, Alzubi JA, Alaqeel S (2016) Solving attribute reduction problem using wrapper genetic programming. Int J Comput Sci Netw Secur (IJCSNS) 16(5):77
- 8. Alweshah M, Al-Daradkeh A, Al-Betar MA, Almomani A, Oqeili S (2019) β -hill climbing algorithm with probabilistic neural network for classification problems. J Ambient Intell Human Comput 11(8):3405–3416. https://doi.org/10.1007/s12652-019-01543-4
- Alweshah M, Alkhalaileh S, Albashish D, Mafarja M, Bsoul Q, Dorgham O (2020a) A hybrid mine blast algorithm for feature selection problems. Soft Comput. https://doi.org/10.1007/s00500-020-05164-4
- Alweshah M, Khalaileh SA, Gupta BB, Almomani A, Hammouri AI, Al-Betar MA (2020b) The monarch butterfly optimization algorithm for solving feature selection problems. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05210-0
- Arora S, Anand P (2019) Binary butterfly optimization approaches for feature selection. Expert Syst Appl 116:147–160. https:// doi.org/10.1016/j.eswa.2018.08.051
- Arora S, Singh S (2018) Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 23(3):715–734. https://doi.org/10.1007/s00500-018-3102-4
- Balochian S, Baloochian H (2019) Social mimic optimization algorithm and engineering applications. Expert Syst Appl 134:178–191. https://doi.org/10.1016/j.eswa.2019.05.035
- Chakraborty B (2008) Feature subset selection by particle swarm optimization with fuzzy fitness function. In: 2008 3rd international conference on intelligent system and knowledge engineering. IEEE. https://doi.org/10.1109/iske.2008.4731082
- Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28. https://doi.org/10. 1016/j.compeleceng.2013.11.024
- Chatterjee B, Bhattacharyya T, Ghosh KK, Singh PK, Geem ZW, Sarkar R (2020) Late acceptance hill climbing based social ski

driver algorithm for feature selection. IEEE Access 8:75393–75408. https://doi.org/10.1109/ACCESS.2020.2988157

- Cui Y, Geng Z, Zhu Q, Han Y (2017) Review: multi-objective optimization methods and application in energy saving. Energy 125:681–704. https://doi.org/10.1016/j.energy.2017.02.174
- Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1(3):131–156. https://doi.org/10.3233/IDA-1997-1302
- 19. Dua D, Graff C (2019) Uci machine learning repository
- Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381. https://doi.org/10.1016/j.neucom.2015.06.083
- Faris H, Heidari AA, Al-Zoubi AM, Mafarja M, Aljarah I, Eshtay M, Mirjalili S (2020) Time-varying hierarchical chains of salps with random weight networks for feature selection. Expert Syst Appl 140:112898. https://doi.org/10.1016/j.eswa.2019.112898
- 22. Garcia-Hernandez L, Salas-Morera L, Garcia-Hernandez J, Salcedo-Sanz S, de Oliveira JV (2019) Applying the coral reefs optimization algorithm for solving unequal area facility layout problems. Expert Syst Appl 138:112819. https://doi.org/10.1016/ j.eswa.2019.07.036
- 23. Garcia-Hernandez L, Garcia-Hernandez J, Salas-Morera L, Carmona-Muñoz C, Alghamdi N, de Oliveira JV, Salcedo-Sanz S (2020a) Addressing unequal area facility layout problems with the coral reef optimization algorithm with substrate layers. Eng Appl Artif Intell 93:103697. https://doi.org/10.1016/j.engappai. 2020.103697
- 24. Garcia-Hernandez L, Salas-Morera L, Carmona-Muñoz C, Abraham A, Salcedo-Sanz S (2020b) A novel multi-objective interactive coral reefs optimization algorithm for the unequal area facility layout problem. Swarm Evol Comput 55:100688. https:// doi.org/10.1016/j.swevo.2020.100688
- 25. Garcia-Hernandez L, Salas-Morera L, Carmona-Muñoz C, Garcia-Hernandez J, Salcedo-Sanz S (2020c) A novel island model based on coral reefs optimization algorithm for solving the unequal area facility layout problem. Eng Appl Artif Intell 89:103445. https://doi.org/10.1016/j.engappai.2019.103445
- Geem ZW, Kim JH, Loganathan G (2001) A new heuristic optimization algorithm: Harmony search. Simulation 76(2):60–68. https://doi.org/10.1177/003754970107600201
- Ghamisi P, Benediktsson JA (2015) Feature selection based on hybridization of genetic algorithm and particle swarm optimization. IEEE Geosci Remote Sens Lett 12(2):309–313. https://doi. org/10.1109/lgrs.2014.2337320
- 28. Ghosh KK, Ahmed S, Singh PK, Geem ZW, Sarkar R (2020a) Improved binary sailfish optimizer based on adaptive β -hill climbing for feature selection. IEEE Access 8:83548–83560. https://doi.org/10.1109/access.2020.2991543
- Ghosh KK, Ghosh S, Sen S, Sarkar R, Maulik U (2020b) A twostage approach towards protein secondary structure classification. Med Biol Eng Comput 58(8):1723–1737. https://doi.org/10.1007/ s11517-020-02194-w
- 30. Ghosh KK, Singh PK, Hong J, Geem ZW, Sarkar R (2020) Binary social MIMIC optimization algorithm with x-shaped transfer function for feature selection. IEEE Access 8:97890–97906. https://doi.org/10.1109/ACCESS.2020.2996611
- 31. Ghosh M, Adhikary S, Ghosh KK, Sardar A, Begum S, Sarkar R (2018) Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods. Med Biol Eng Comput 57(1):159–176. https://doi.org/10.1007/s11517-018-1874-4
- 32. Guha R, Ghosh KK, Bhowmik S, Sarkar R (2020) Mutually informed correlation coefficient (MICC)—a new filter based feature selection method. In: 2020 IEEE Calcutta conference (CALCON), pp 54–58. https://doi.org/10.1109/CALCON49167. 2020.9106516

- Guha R, Ghosh M, Mutsuddi S, Sarkar R, Mirjalili S (2020) Embedded chaotic whale survival algorithm for filter-wrapper feature selection. Soft Comput. https://doi.org/10.1007/s00500-020-05183-1
- 34. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(null):1157–1182
- 35. Hall MA, Smith LA (1999) Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In: FLAIRS conference 1999, pp 235–239
- He X, Cai D, Niyogi P (2006) Laplacian score for feature selection. In: Advances in neural information processing systems, pp 507–514
- Hunter JD (2007) Matplotlib: a 2D graphics environment. Comput Sci Eng 9(3):90–95. https://doi.org/10.1109/mcse.2007. 55
- Jain A, Zongker D (1997) Feature selection: evaluation, application, and small sample performance. IEEE Trans Pattern Anal Mach Intell 19(2):153–158. https://doi.org/10.1109/34.574797
- 39. Jona J, Nagaveni N (2012) A hybrid swarm optimization approach for feature set reduction in digital mammograms. WSEAS Trans Inf Sci Appl 9(11):340–349
- Kannan SS, Ramaraj N (2010) A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm. Knowl-Based Syst 23(6):580–585. https://doi.org/10. 1016/j.knosys.2010.03.016
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95—international conference on neural networks, vol 4, pp 1942–1948. https://doi.org/10.1109/ICNN. 1995.488968
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680. https://doi.org/ 10.1126/science.220.4598.671
- Koller D, Sahami M (1996) Toward optimal feature selection. Technical Report 1996–1977, Stanford InfoLab, http://ilpubs. stanford.edu:8090/208/, previous number = SIDL-WP-1996-0032
- 44. Lai X, Zhang M, (2009) An efficient ensemble of GA and PSO for real function optimization. In: 2009 2nd IEEE international conference on computer science and information technology. IEEE. https://doi.org/10.1109/iccsit.2009.5234780
- Leardi R (1994) Application of a genetic algorithm to feature selection under full validation conditions and to outlier detection. J Chemom 8(1):65–79. https://doi.org/10.1002/cem.1180080107
- Lee S, Soak S, Oh S, Pedrycz W, Jeon M (2008) Modified binary particle swarm optimization. Prog Nat Sci 18(9):1161–1166. https://doi.org/10.1016/j.pnsc.2008.03.018
- Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453. https:// doi.org/10.1016/j.asoc.2017.11.006
- Mafarja M, Qasem A, Heidari AA, Aljarah I, Faris H, Mirjalili S (2019) Efficient hybrid nature-inspired binary optimizers for feature selection. Cogn Comput 12(1):150–175. https://doi.org/ 10.1007/s12559-019-09668-6
- Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 260:302–312. https://doi.org/10.1016/j.neucom. 2017.04.053
- Medeiros IG, Xavier JC, Canuto AMP, (2015) Applying the coral reefs optimization algorithm to clustering problems. In: 2015 International joint conference on neural networks (IJCNN). IEEE. https://doi.org/10.1109/ijcnn.2015.7280845
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft. 2013.12.007
- Oh IS, Lee JS, Moon BR (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26(11):1424–1437. https://doi.org/10.1109/tpami.2004.105

- Pudil P, Novovičová J, Kittler J (1994) Floating search methods in feature selection. Pattern Recogn Lett 15(11):1119–1125. https://doi.org/10.1016/0167-8655(94)90127-9
- 54. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
- 55. Salcedo-Sanz S, Ser JD, Landa-Torres I, Gil-López S, Portilla-Figueras JA (2014) The coral reefs optimization algorithm: a novel metaheuristic for efficiently solving optimization problems. Sci World J. https://doi.org/10.1155/2014/739768
- 56. Salcedo-Sanz S, Pastor-Sanchez A, Ser JD, Prieto L, Geem Z (2015) A coral reefs optimization algorithm with harmony search operators for accurate wind speed prediction. Renew Energy 75:93–101. https://doi.org/10.1016/j.renene.2014.09.027
- 57. Salcedo-Sanz S, Camacho-Gomez C, Molina D, Herrera F (2016) A coral reefs optimization algorithm with substrate layers and local search for large scale global optimization. In: 2016 IEEE Congress on evolutionary computation (CEC). IEEE. https://doi. org/10.1109/cec.2016.7744242
- Sulaiman MH, Mustaffa Z, Saari MM, Daniyal H (2020) Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems. Eng Appl Artif Intell 87:103330. https://doi.org/10.1016/j.engappai.2019.103330
- Sun X, Liu Y, Li J, Zhu J, Chen H, Liu X (2012) Feature evaluation and selection with cooperative game theory. Pattern Recogn 45(8):2992–3002. https://doi.org/10.1016/j.patcog.2012. 02.001
- Syswerda G (1993) Simulated crossover in genetic algorithms. In: Foundations of genetic algorithms. Elsevier, pp 239–255. https:// doi.org/10.1016/b978-0-08-094832-4.50021-0
- Talbi EG (2009) Metaheuristics: from design to implementation, vol 74. Wiley, New York
- Too J, Abdullah AR (2020) Binary atom search optimisation approaches for feature selection. Connect Sci. https://doi.org/10. 1080/09540091.2020.1741515
- 63. Van Rossum G, Drake FL (2011) The Python language reference manual. Network Theory Ltd
- Wilcoxon F (1992) Individual comparisons by ranking methods. Springer Series in Statistics. Springer, New York, pp 196–202. https://doi.org/10.1007/978-1-4612-4380-9_16
- Wolpert D, Macready W (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82. https://doi. org/10.1109/4235.585893
- Xiong L, Chen RS, Zhou X, Jing C (2019) Multi-feature fusion and selection method for an improved particle swarm optimization. J Ambient Intell Human Comput. https://doi.org/10.1007/ s12652-019-01624-4
- 67. Yan C, Ma J, Luo H, Patel A (2019) Hybrid binary coral reefs optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical datasets. Chemom Intell Lab Syst 184:102–111. https://doi.org/10.1016/j.chemolab. 2018.11.010
- Zhao W, Wang L, Zhang Z (2019) Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. Knowl-Based Syst 163:283–304. https://doi.org/10. 1016/j.knosys.2018.08.030
- Zheng Z, Wu X, Srihari R (2004) Feature selection for text categorization on imbalanced data. ACM SIGKDD Explor Newsl 6(1):80–89. https://doi.org/10.1145/1007730.1007741

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

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for smallscale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

- 1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
- 2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
- 3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
- 4. use bots or other automated methods to access the content or redirect messages
- 5. override any security feature or exclusionary protocol; or
- 6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com