IntechOpen *Journals*

AI, Computer Science and Robotics Technology

RESEARCH PAPER

Addressing Optimisation Challenges for Datasets with Many Variables, Using Genetic Algorithms to Implement Feature Selection

Neil Gordon*, Chandrasekhar Kambhampati and Asmaa Alabed

Department of Computer Science and Technology, University of Hull, Cottingham Road, Hull, UK
*Corresponding author. E-mail: n.a.gordon@hull.ac.uk

Abstract

This article provides an optimisation method using a Genetic Algorithm approach to apply feature selection techniques for large data sets to improve accuracy. This is achieved through improved classification, a reduced number of features, and furthermore it aids in interpreting the model. A clinical dataset, based on heart failure, is used to illustrate the nature of the problem and to show the effectiveness of the techniques developed. Clinical datasets are sometimes characterised as having many variables. For instance, blood biochemistry data has more than 60 variables that have led to complexities in developing predictions of outcomes using machine-learning and other algorithms. Hence, techniques to make them more tractable are required. Genetic Algorithms can provide an efficient and low numerically complex method for effectively selecting features. In this paper, a way to estimate the number of required variables is presented, and a genetic algorithm is used in a "wrapper" form to select features for a case study of heart failure data. Additionally, different initial populations and termination conditions are used to arrive at a set of optimal features, and these are then compared with the features obtained using traditional methodologies. The paper provides a framework for estimating the number of variables and generations required for a suitable solution.

Keywords: feature selection, feature optimisation, genetic algorithms, human reasoning, wrapper selection

Citation
Neil Gordon, Chandrasekhar Kambhampati and Asmaa Alabed (2022), Addressing Optimisation Challenges for Datasets with Many Variables, Using Genetic Algorithms to Implement Feature Selection. *AI, Computer Science and Robotics Technology* 2022(0), 1–21.

DOI
https://doi.org/10.5772/acrt.01

Published
28 March 2022

1. Research background—introduction

The explosion of data capture in general, and specifically in the health industry, has created new challenges in terms of understanding and benefitting from the data [1]. This increase in data has led to an explosion in terms of the size and dimensional complexity of datasets, with respect to the number of distinct measurements or

features. When machine-learning and data mining algorithms are applied to high-dimensional data, the performance of pattern modelling and the effectiveness of classification is reduced [2], as the computational complexity increases in terms of both processing time and the storage space required (see [3, 4] and [5]).

Reducing dimensionality can be done by using one of two primary methods, namely feature selection and feature extraction. Feature extraction creates a new smaller set of features that projects the original high-dimensional features to a new feature subset with lower dimensionality. On the other hand, feature selection is the process of creating a smaller subset by removing irrelevant and redundant features. Hence, they are both considered as effective dimension reduction approaches, having the advantage of improving learning performance, increasing computational efficiency, decreasing memory storage, and building better generalisation models [6]. A distinction between the two is that, in feature selection, a subset of the original features is created, whilst for feature extraction, a set of novel features is produced from the original features [7].

Feature selection is based on selecting a subset of features from the original dataset where [8]:

(a) Reduction to the classification accuracy itself is minimised;
(b) The class distribution of the values for the selected features is optimised to be close to the class distribution of the original set of all the features.

There are three recognised approaches to feature selection, namely filter, wrapper and embedded approaches [9, 10]. Firstly, **filter selection methods** apply a statistical measure to assign a weight to each feature according to its degree of relevance. The advantages of filter methods are that they are fast, scalable and independent of a learning algorithm. The most distinguishing characteristic of filter methods is that a relevance index is calculated solely on a single feature without considering the values of other features [11]. Such an implementation implies that the filter assumes the orthogonality of features—which may not be true in practice—ignoring any conditional dependency (or independence). This is known to be one of the weaknesses of filters. Secondly, **wrapper methods evaluate features using a black box predictor** to evaluate a feature subset [3, 12]. With these methods, the possible number of subsets which can be created and evaluated could be 2^n, where n is the number of features and thus becomes an NP-hard problem. Finally, **the embedded approach** combines the previous two methods, where the feature selection is carried out alongside creating a model, with specific learning algorithms that perform feature selection during the process of training.

In a typical case, the feature selection problem can be solved optimally using an exhaustive search to evaluate all possible options. Typically, a heuristic method is

employed to generate subsets of features for evaluation procedures, where each subset is evaluated to determine the "goodness" of the selected features. The current work focuses on the application of Genetic Algorithms (GAs) to the feature selection problem, while also addressing some of the underlying fundamental problems of Genetics Algorithms, namely (i) when do we terminate the algorithms, (ii) how to determine the number of chromosomes to be used and (iii) how much of the search space has been explored. By integrating a GA within a feature selection problem, it is possible to develop a framework within which answers to these questions can be obtained.

This paper first casts the problem of feature selection as an optimisation problem, which automatically lends itself to the application of Genetic Algorithms (section 2). Furthermore, by casting exploration of the subset set space as a graph problem, it is feasible to derive specific properties for the GAs as applied to the feature selection problem. Finally, in section 3, through the use of a Heart Failure dataset [11, 13] the approach is applied and evaluated [14, 15]. Appendix A provides a list of all of the variables for the case considered below, and Appendix B, Table 4 provides a list of the main acronyms.

2. *Methodology: optimisation of feature selection*

This research was primarily based on an experimental approach, where feature selection was applied to a published data set to identify the most important factors. This was then compared with established practice by clinicians, to validate the computer based identification of the key factors.

Section 3 describes the theory and relevant background material on feature selection. It develops the theory and provides proofs of some of the relevant properties that enables estimates of the number of iterations required to achieve a solution. This is then applied to a large clinical data set to demonstrate the effectiveness of the techniques and approach.

2.1. Classical features selection

All feature selection algorithms typically consist of four steps (see figure 1), namely:

1. a subset generation step;
2. evaluation of the subset;
3. a step to check against the stopping criteria, and once this is met, finally
4. a result validation step [5, 16], and [17].

Classically, a heuristic method is employed to generate subsets of features for evaluation procedures, where each subset is evaluated to determine the "goodness" of the selected features. The validation is done by carrying out different tests and comparisons with the previous subset. If a new subset has worse performance,

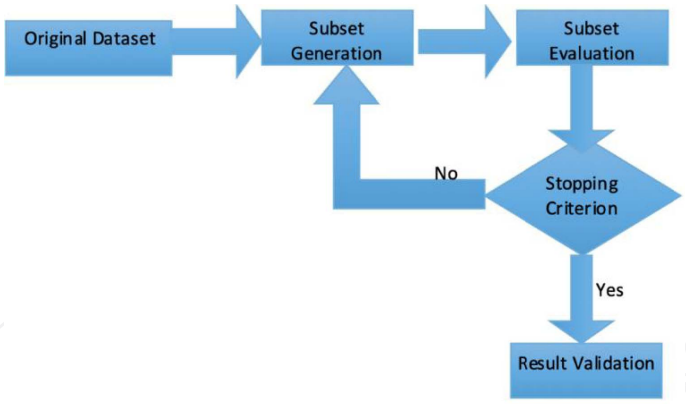

Figure 1. Four steps for feature selection process [8].

then the previous subset is retained. This process is repeated until a stopping criterion is reached, as shown in figure 1. A feature is a variable in a dataset, and as mentioned previously, if the number of variables is large, this can lead to problems that are intractable, i.e. cannot be solved in polynomial time. One approach that can aid in finding solutions is to reduce the number of features by selecting features, to drop those features that are deemed unnecessary, or that may actually impede the predictive performance of a model.

A general feature selection problem, with n features and a fixed subset size of d, would require (nd) subsets to be examined to get the optimal result. Considering all subset sizes, it would generally require 2^n subsets to be evaluated, and it has been shown that no non-exhaustive sequential search can be guaranteed to produce an optimal subset [18]. However, by creating an ordered sequence of errors for each of the subsets, a branch-and-bound feature selection algorithm can be used to find an optimal subset more quickly than an exhaustive search [19]. The problem of feature selection can be stated as follows [20]:

Given a set F of features (variables or attributes) $\{f_1, f_2, f_3, \ldots f_m\}$ and a set C of classes; and where m is the number of features, let $S_d = \{(f^1, C^1), (f^2, C^2)(f^3, C^3) \ldots (f^n, C^n))\}$ be the dataset, where $F = \{f_1, f_2, f_3, \ldots f_m\}$ is the set of features, and $f_i \in R^n$, and $C_i \in Z_2$ are the set of features describing a possible class C. For simplicity, we assume two classes, but this can be extended to more classes. The problem learning classifier can be stated as follows: find a set of parameters θ such that

$$\sum_{i=1}^{n} D(C_i, \hat{C}_i) \tag{1}$$

where $\hat{C}_i = g(f_i, \theta)$, and g is the model predictor function, The cost function $D(C_i, \hat{C}_i)$ is an error squared function of the form $D(C_i, \hat{C}_i) = \frac{1}{2}(C_i - g(f_i, \theta))^2$; where $\theta \in R^n$ is the vector of parameters for the model.

Given the ease of data collection, a particular issue is that they often have far more variables than are strictly necessary to achieve a correct result. This is more acute in clinical datasets. The problem becomes one of reconstructing the data set with a reduced, optimal, number of features, $F_i \subseteq F$;

Definition 1. A feature set $F_i \subseteq F$, is said to be optimal if:

a. the relationships between F & C are maintained using F_i
b. $F_i \cap F^C = \varnothing$; where F^C is the complement of F_i.

Such a formulation lends itself to a combinatorial search for an optimal set of features (see [21–23]). An alternative approach is to consider the two problems of feature selection and classification together [24], through the use of scaling variables $\omega_i \in Z_2, i = 1, 2, \ldots, n$.. for the features, such that the feature selection problem becomes one of optimising an objective function over ω. Thus, a particular feature f_i is removed if $\omega_i = 0$. The overall problem becomes one of optimising for both θ *and* ω and can be written as

$$\sum_{i=1}^{n} \quad D(\omega_i F_i, \theta) \; s.t. \sum_{i=1}^{m} \quad \omega_{ij} \leq m, j = 1, 2, \ldots n \Bigg\} \tag{2}$$

Remark. The length of $\omega_i \in Z_2, i = 1, 2, \ldots, n$ is equal to the length of the number of features in the data set, where ω_i is equal to 1 if the feature is present, and 0 if the feature is to be ignored. Thus, the vector $\varpi = \{\omega_i\}$ is a binary bit-string, and as will be seen later is essentially the chromosome utilised in the genetic algorithm, where $\{\varpi_i\}, i = 1, 2, 3, \ldots, P$ is a set of chromosomes and P is the population size.

In order to investigate the properties of (2), we need the following proposition:

Proposition 1. Let

$$\Omega \triangleq \left\{ \omega_i : \sum_{i=1}^{m} \quad \omega_i - m < 0; \omega_i \geq 0 \right\} \tag{3}$$

Then for $\omega \in \Omega$;

$$D(\omega) \triangleq \arg\arg \left\{ \sum_{i=1}^{n} \quad D(\omega_i F_i, \theta); \theta \in R^k \right\} \text{ is a concave set.}$$

The problem $D\ (\omega)$; *s.t.* $\omega \in \Omega$ is equivalent to solving equation (2).

The proof is routine and follows the methodology that can be found in [21] & [24]. The approach taken by a feature search algorithm for selecting optimal feature sets is often an iterative one, where two problems are repeatedly solved, these are:

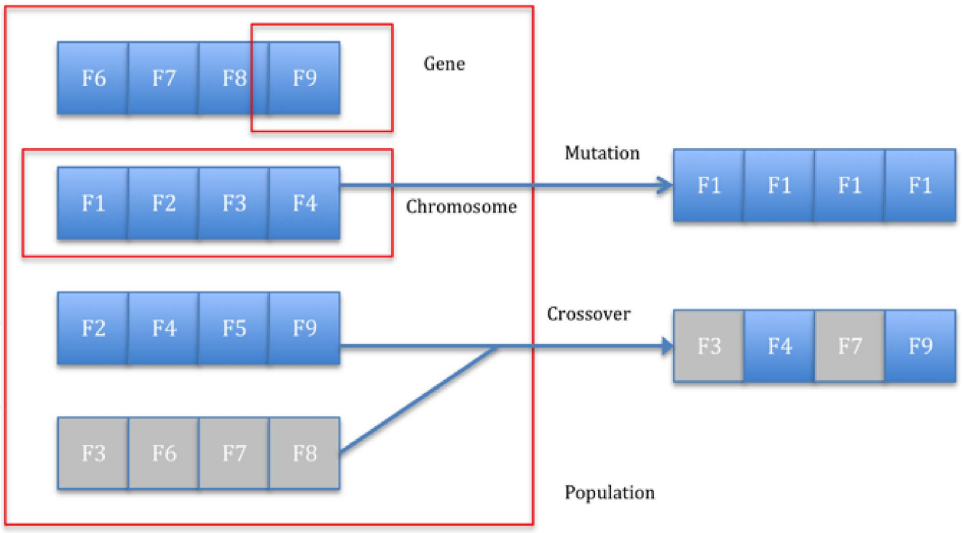

Figure 2. Genetic Algorithms operations of mutation and crossover for FSP2.

1. FSP1: for a given $\hat{\theta} \in R^k \min \left\{ \sum_{i=1}^{n} \quad D(\omega_i F_i, \hat{\theta}) \right\}$ (4)

2. FSP2: for a given $\hat{\omega} \in \Omega; \min \left\{ \sum_{i=1}^{n} \quad D(\hat{\omega}_i F_i, \theta) \right\}$ (5)

2.2. Feature selection using genetic algorithms (GAs)

A Genetic Algorithm ensures that over successive generations, the population "evolves" toward an optimal solution based on the principle of survival of the fittest (see [3, 9, 17]). GAs allow the combining of different solutions generation after generation to extract the best solution in the quickest time [25]. Generally, the search method used by Genetic Algorithms has lower complexity than the more formal approaches mentioned earlier [26].

The individuals in the genetic space are called chromosomes. The chromosome is a collection of genes where a real value or a binary encoding can generally represent genes. The number of genes is the total number of features in the data set. Thus, the vector $\varpi = \{\omega_i\}$ is a chromosome, and ω_i is the ith gene. The collection of all chromosomes is called the "population", P. Typically, a set P of chromosomes, are generated randomly. The problem FSP2 is then solved iteratively through the use of two operators, namely Crossover and Mutation (see figure 2).

Every iteration or generation, a new set of chromosomes are evaluated through the use of a fitness function. This is typically the cost function used in FSP1. Given

the population, P, there are essentially P subsets of features being evaluated at every iteration. In this approach, the chromosomes with the least performance (or fitness) are also eliminated, thus not available for the cross over or mutation operations. In this manner, good subsets are "evolved" over time [22]. The commonly used methods for selection are Roulette-wheel selection, Boltzmann selection, Tournament selection, Rank selection, and Steady-state selection. The selected subset is ready for reproduction using crossover and mutation. The idea of a crossover operator is that some combination of the best features of the best set of chromosomes would yield a better subset. On the other hand, mutation takes one chromosome and randomly changes the value of a gene. Thus, adding or deleting features in a subset. This process is repeated until there is no improvement, or as is often the case, a set number of iterations or generations is reached. This is illustrated in figure 3.

The general structure of the algorithm is as follows

Step 0: Choose an initial population, *M*, of chromosomes of length *n*

Step 1: Evaluate the fitness of each chromosome

Step 2: Perform Selection

Step 3a: Perform Crossover

Step 3b: Perform Mutation

Step 4: Evaluate fitness,

if stopping criteria satisfied Stop; Else go to Step 2.

Steps 1 and 4 solve PSP2, steps 3a and 3b solve FSP1.

The solution to both FSP1 and FSP2, can be considered to be a set of Kuhn-Tucker points, i.e. points that satisfy the K_T conditions for FSP1 and FSP2, but never together at the same time [22].

For most optimisation techniques, there is often a requirement that the cost function is monotone. However, there is no guarantee that this is satisfied globally. However, if the cost function is not locally monotone somewhere, a solution to the problem will not exist. It is possible to estimate the number of generations required to arrive at an optimal set of features of the requirement of monotonicity is enforced in the GAs. Thus, if it were made explicit that from one generation to generation the best subset is an improvement on the previous best, the sequence of cost function values would be monotone. One of the consequences of the monotone requirement is that the initial population of chromosomes and the final set of chromosomes are disjointed.

This condition can be satisfied for the initial population only; for subsequent generations, this may be not possible, for the same operation on two different pairs

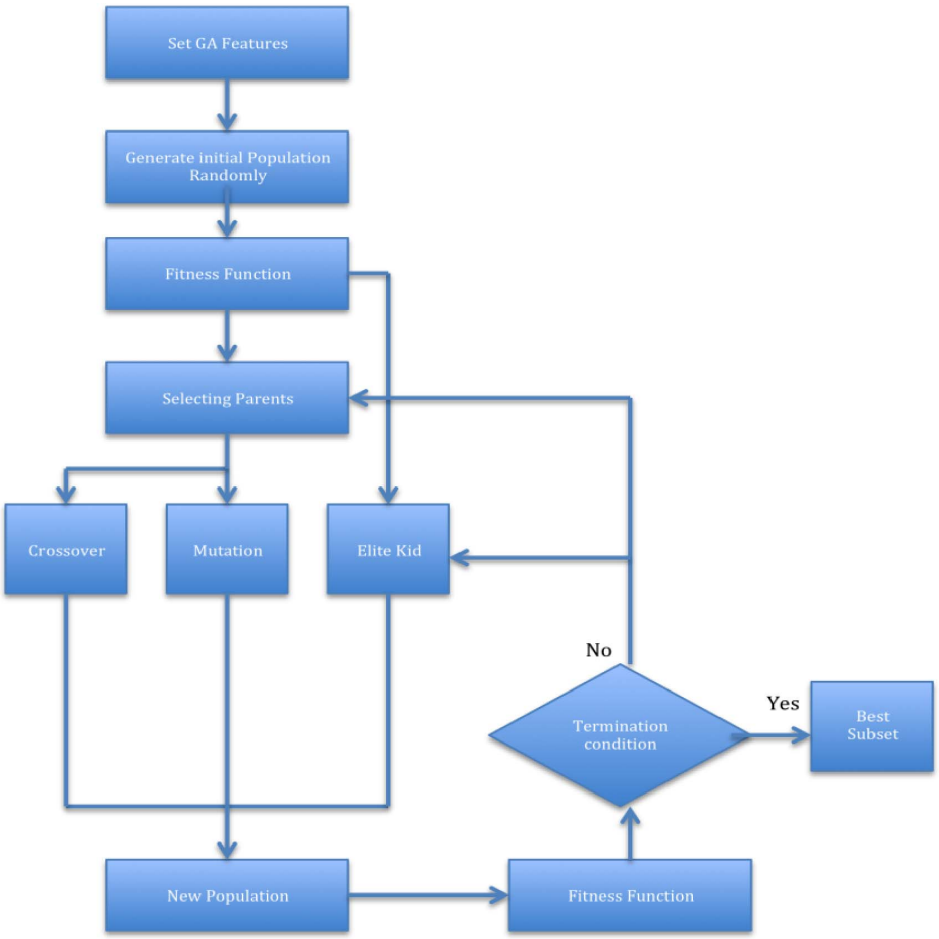

Figure 3. GAs as a feature selection [9].

of chromosomes may result in the same set of chromosomes as children. However, given this restriction, it is interesting to use this to gain an insight into the operation of GAs.

Definition 2. A generation g is an ordered set of M chromosomes, where M is the initial population of chromosomes. Thus if $g = \{f_1^g, f_2^g, \ldots f_M^g\}$, then the fitness function J has the property

$$J(f_1^g) < J(f_2^g) < \cdots < J(f_M^g) \tag{6}$$

Definition 3. The fitness function J (.) is said to be ***genetically monotone*** if for the i'th generation

$$J(f_1^{i-1}) < J(f_1^i) \tag{7}$$

The second definition is interesting and required for further analysis. It essentially focuses on the first or leading chromosome of any generation, since for

each generation, the chromosomes are ordered (definition 2). This definition also suggests that with every successive generation, the fitness of the leading chromosome of the i'th generation is better than the previous generation.

Definition 4. Two generations g^i and g^{i+1} are said to be generationally disjoint if $g^i \cap g^{i+1} = \varnothing$.

This definition is required to keep in line with the requirements of analysis based on graph searches.

Proposition 2. For a given fixed M, where M is the number of chromosomes, and a genetically monotone fitness function $J(.)$, if the genetic operators are such that every generation is generationally disjoint, the maximum number of generations, N_g, needed to determine the fittest chromosome is then given by

$$N_g = \frac{2(n-1)}{M} \tag{8}$$

The proof follows from the definitions. The conditions (assumptions) here are often difficult to satisfy. Getting an ordered set for a generation is already present in the algorithm, and so is the presence of a genetically monotone function in the form of a fitness function. However, the genetically disjoint requirement is difficult to satisfy, and there is no guarantee it will be fulfilled at all during a given run of the algorithm. Thus, often the terminating criteria of a genetic algorithm are the number of generations. In most random search algorithms, the function $J(.)$ is guaranteed to be monotone.

Definition 5. Two generations g^i and g^{i+1} are said to be partially generationally disjoint if $g^i \cap g^{i+1} \neq \varnothing$ and $|g^i \cap g^{i+1}| < M$.

Proposition 3. If the sequence of generation sets $g^1, g^2, g^3, \ldots, g^N$, where N is the Nth generations are all partially genetically disjoint, and given a monotone fitness function, then as $N \to \aleph/M$ the top feature subset is given by f_1^g is that that $f_1 \to \hat{f}_1$ where $\hat{f}_1$ is the optimal feature subset.

The proposition is both intuitive and follows the fact that every generation set g^i is ordered and that the fitness function is genetically monotone thus $J(f_1^{i-1}) \leq J(f_1^i)$ for all i. Essentially this proposition is suggesting that in the limit the maximum number of generations leads to an exhaustive search through all subsets of features. In practice there are feature sets, which can be discounted, these the sets with no features, all the features and one feature; therefore, the number of subsets to be evaluated is $2^n - (n + 2)$. Simultaneously, if M is sufficiently large, the number of generations is reduced significantly. However, this is a technological limitation

rather than an algorithmic limitation. In the limiting case as $M \to n$ what is obtained is nothing more than an exhaustive search.

Louis and Rawlins [27] have shown that the crossover operator has no role to play in the convergence of the algorithms, as the average Hamming distance of the population in every generation is maintained. Their analysis indicated that the selection criteria are on which dictates the convergence rate of the genetic algorithm. Their analysis shows that the traditional crossover operation, like 1-point, two-point l-point, uniform crossover, does not change the average Hamming distance of the chromosomes within a generation, and thus, the difference of the distance from generation to generation is maintained. They suggested that selection criteria are the main parameter for determining convergence. However, Greenhalgh and Marshall [27], showed that irrespective of the coding (binary or otherwise), the key controller of convergence (or the ability of the genetic algorithm to have visited all the possible points available) is given by the mutation probability. They indicated, following on from Aytug *et al.* (1996) [28] that a mutation rate of $\mu \leq \frac{K-1}{K}$; *where K* is the *cardinality* of *coding* rate $\mu \leq 0.5$, allows for convergence in the worst case (it is essentially an upper bound). However, most implementations of genetic algorithms allow for a mutation rate far smaller than this. This then leads to the following.

Proposition 4. For a given $0 \leq \gamma < 1$, and $\mu \leq 0.5$ (where γ is the *crossover parameter*) the genetic algorithm generates feature subsets which are generationally disjoint. i.e.

$$g^i \cap g^{i+1} \neq \emptyset \quad \text{and} \quad |g^i \cap g^{i+1}| < M. \tag{9}$$

The proof is intuitive. Even though crossover may lead to all the subsets of a particular generation being identical, the presence of mutation changes this. However, with both present, it is always the case [27]. These two conditions imply that every generation has a set of chromosomes which have not been evaluated in the previous generation, that the crossover operations do not generate the same solutions within a generation, and we would need at most $\frac{p}{2} + n - 1$ generations to arrive at a solution. Within the definitions of the GAs, it is not possible to guarantee both of these can be satisfied. If they are, it is feasible to determine the number of generations required to arrive at a solution given the initial population size.

2.3. Complexity of the GAs

In order to assess the complexity of the Genetic Algorithms, we need to consider the following:

(a) The initial population size, and
(b) The number of generations required to satisfy the performance criteria.

GAs can often reach a state of stagnation, a possible reason for this that the above two points act as constraints, and it becomes feasible that the same population is generated at every generation. In order to get a deeper insight into the process, consider the following:

Since n-features means that each chromosome consists of n-bits, the total number of possible chromosomes is 2^n. However, two of these can be discarded—these are when all the genes are either 0 or 1. If we consider these 2^n chromosomes as the vertices of an n-dimensional hypercube, an exhaustive search by the Genetic Algorithm will cover all the possible vertices, given by

$$\aleph = 2^n - 2^2 \left(\frac{60!}{2(58!)} + \frac{60!}{59!} \right). \tag{10}$$

However, the problem of the search can be reduced in complexity if searching vertices can be interpreted as a search through a graph [26].

3. Selecting features for a clinical dataset

The dataset under consideration is a real-life heart failure dataset [29]. In this dataset, there are 60 features for 1944 patient records (See Table 3 in Appendix A). The classification here is simply "dead" or "alive". The data sets were imputed by different methods such as Concept Most Common Imputation (CMCI) and Support Vector Machine (SVM). The different datasets obtained were tested in order to select a good set for feature selection [29]. The selection of a dataset was based on accuracy, sensitivity and specificity. The dataset where the imputation was based on SVM was thus selected. The experiments were designed using Weka (version 3.8.1-199-2016), and validation was done using a 10-fold validation. It can be seen from Alabed *et al.* (2020) [30] that this dataset, with all 60 features present gave the best results for both the Random Forest and Bayes Net classifiers, the other two not far behind.

The GAs was implemented as described earlier and can be seen in [31]. The parameters for the GAs are shown in Table 1.

3.1. Population size

In most evolutionary methods, and Genetic Algorithms in particular, the population size—i.e. the number of chromosomes—is of particular interest. This often influences the quality of the solution and computing time and is also dependent on the available memory [32–34]. There have been a number of studies on appropriate population sizes, see [35, 36], with many suggesting that a "small" population size would result in poor solutions [37–39], while a "large" population size would result in greater complexity in finding a solution [40, 41]. Indeed, this is apparent if the problem is viewed as a search through the 2^n vertices of the

Table 1. GAs parameters.

GAs Parameter	Value
Number of features	60
Population size	50, 75, 100
Genome length	60
Population type	Bit strings
Fitness function	kNN-based classification error
Number of generations	130
Crossover	Arithmetic crossover
Mutation	Uniform mutation
Selection scheme	Roulette wheel
Elite count	2

n-dimensional hypercube. However, none have so far suggested an appropriate size, or at least an effective estimate.

3.2. Selection of chromosomes for genetic operations

The chromosomes are evaluated using a fitness function based on Oluleye's fitness function [31]. Here the function is minimising the error while at the same time reducing the number of features (see equation (2)). The fitness function which evaluates the chromosomes is based on the kNN-function [31]. The chromosomes are ranked based on the kNN based fitness. In this case, chromosomes with the lowest fitness have a better chance of surviving.

A roulette wheel selection approach was used for selecting the chromosomes for the crossover operations where each individual is assigned as a "slice" of the wheel in proportion to the fitness value of the individual. Therefore, the fitter an individual is, the larger the slice of the wheel. The wheel is simulated by a normalisation of fitness values of the population of individuals as discussed in FSP2. The selection process is shown below

Step 1 Determine the sector angles for each chromosome

Step 2 Generate a random number $r \in [0,2\pi]$

Step 3 Select the i'th chromosome

Step 4 And repeat.

It should be noted that the chromosomes are a bit string, clearly for such representation we could use a single point or a n-point crossover. Of course, a single point crossover would be limiting if the number of variables is large. However, an n-point crossover strategy would be more suitable for a large set of variables [42–44].

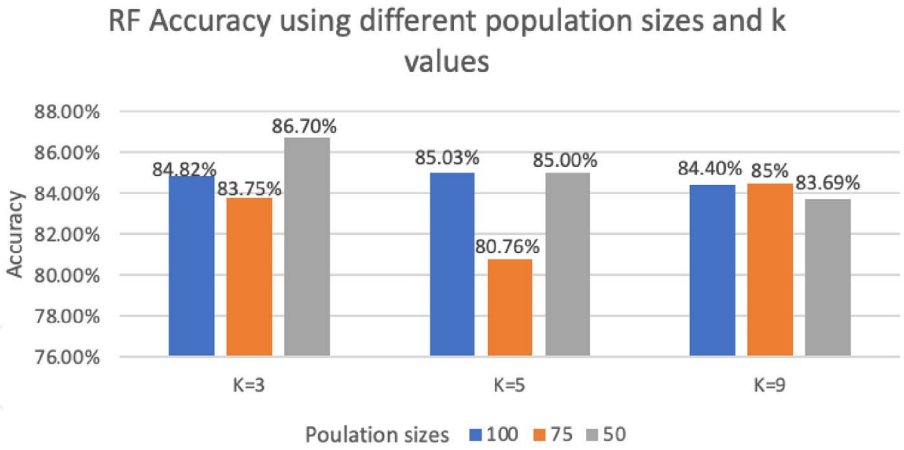

Figure 4. Performance results of GAs for different population sizes and generations (RF classifier- using 27 features).

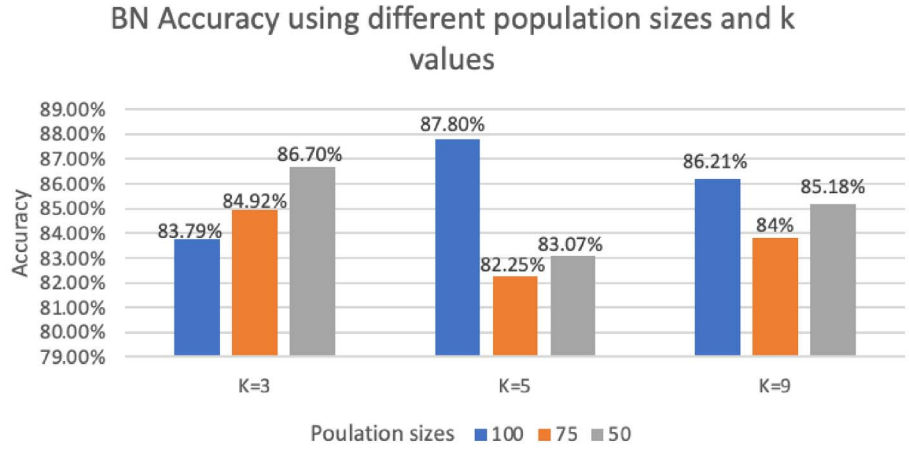

Figure 5. Performance results of GAs for different population sizes and generations (BN classifier- using 27 features).

4. Research overview—results

This paper focuses on two aspects of feature selection, (a) the total number of subsets (population size) to be evaluated at each iteration, and (b) the number of features in each subset.

In order to assess the appropriate population size, different population sizes were tested. The results are presented in figures 4 and 5. The best results were obtained where the population is 100 and $k = 5$ as shown in figure 5, using 27 features from the 60. As K is increased, the accuracy changes as shown in figures 4 and 5.

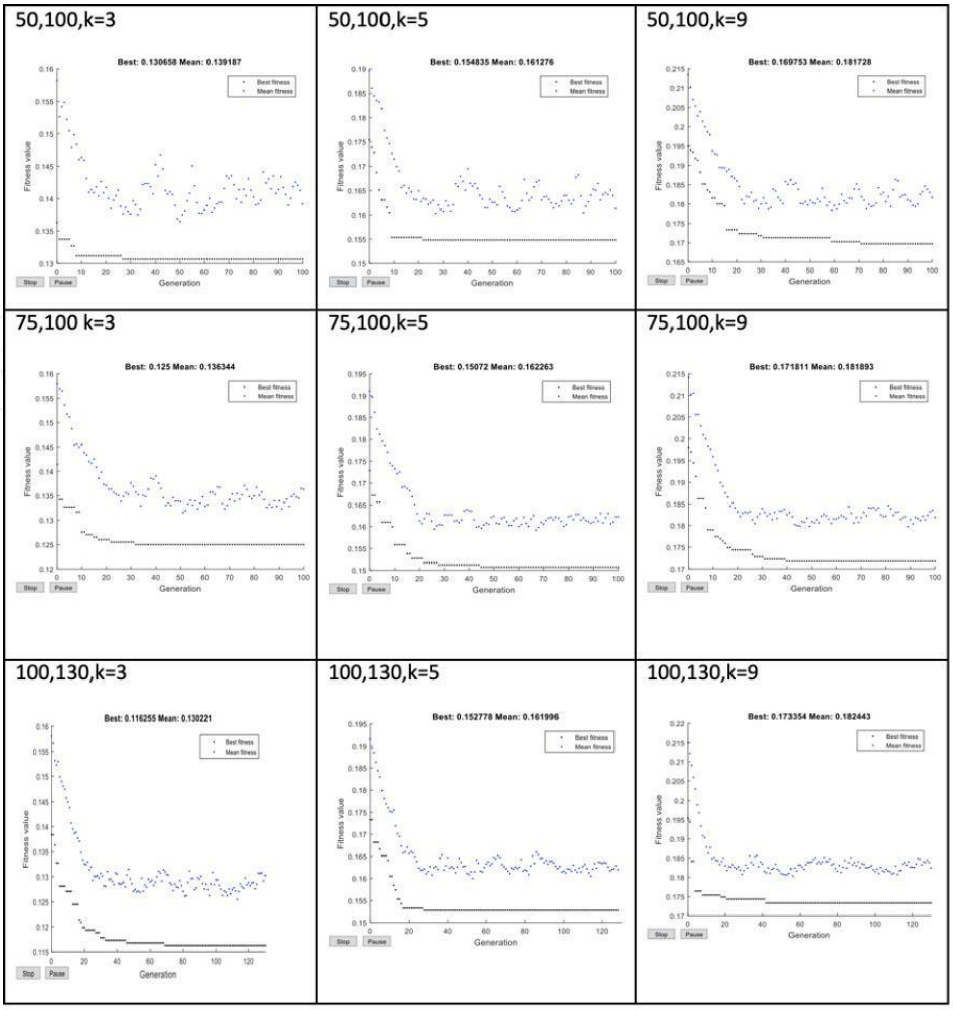

Figure 6. Outcomes from the models with a variety of parameters. The lower set of data in each figure shows the Best Fit.

In order to investigate if larger population sizes improve the performance dramatically, the population was increased in steps to 400, 600, and 800. Figure 6 shows the accuracy for different generations, and the optimal accuracy is 86.3% which is less than 87.7% that was achieved using a population of 100. The results also indicate that an increase in population size does not change the results significantly to warrant the increase in complexity to achieve these results.

Another test was carried out with 3 population sizes of 50, 75, 100 (see figure 4). And running the Genetic Algorithms for a different number of generations, and different values of k (the fitness function parameter). A small value of k means that the noise will have a higher influence on the algorithm, while larger k not only reduces the effect of noise, but also increases computational and numerical

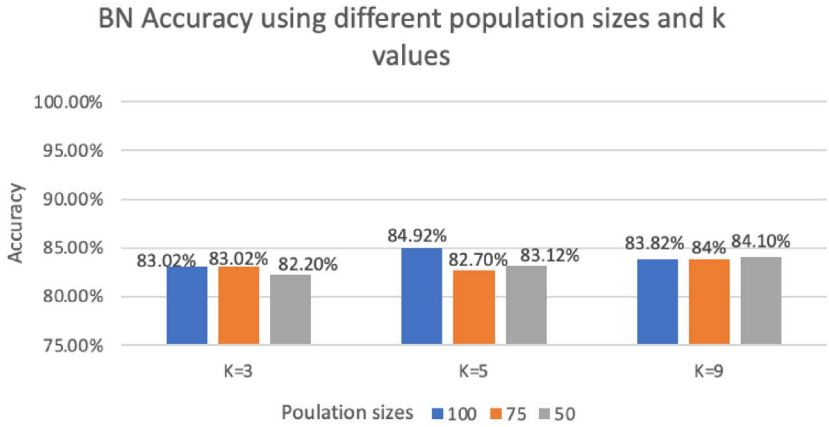

Figure 7. Performance GAs for different population sizes and generations (RF classifier- using 14 features).

complexity. In all these tests, it can be seen that the mean value of the fitness function was best with a k = 5.

At the same time, it can be noticed that the value of the best-fit chromosome in each generation oscillates after the initial improvements. This clearly indicates the algorithm has arrived at a set of features which are good, but it starts to oscillate, and an increase in the number of generations does not change either the value of fitness with the best chromosome, nor does the mean value change. Indeed, the number of generations was increased to 130 and the results are similar. The results shown in figure 6 show that the performance has not significantly improved but confirmed the earlier analysis of the possible number of generations needed and in [15]. Thus, for this dataset approximately 50 generations are needed for convergence of the best-fit chromosome.

This is further illustrated in figures 7 and 8. Here the performance of the selected features in classification is shown for the tests runs shown in figure 6. This further confirms, what was shown in figures 4 and 5.

The question then remains one of determining whether an optimal solution has been obtained. Figure 6, illustrates two aspects of the feature selection problem. The first is that independent of the number of generations or population size the optimal value of the fitness function is reached. However, what is the chromosome which is the best is an answer which is difficult to arrive at independently. The oscillatory behaviour of the fitness of the best chromosome in all cases shows that there is no one chromosome which is the best and that it is possible there is more than one chromosome which is good. This is where, an expert in the application has to be consulted in order to pick an appropriate sub-set of features for the application.

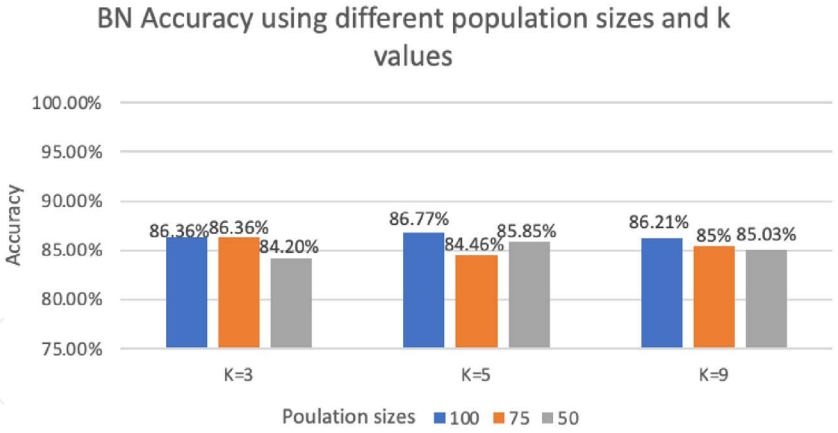

Figure 8. Performance GAs for different population sizes and generations (BN classifier- using 14 features).

It has been suggested [43, 44] that the performance can be further improved by running a second level GAs. Thus, using the 27 features, selected by the genetic algorithm, the number of selected features were reduced from 27 to 14 features as shown in figure 7; however, the performance of GAs has slightly reduced by 1% where the accuracy was 87.8% to 86.77%. Figure 7 shows the results obtained for these tests.

In order to assess this, further tests were carried out. Two further feature selection algorithms were used. The Symmetrical Uncertainty [43, 44] and Correlation-Based-Selector (CFS) [45]. These results were compared to those obtained by Al-Khaldy [46]. Al Khaldy investigated several feature selection methods, including wrapper and filters methods, and further used a representative set of classification methods for evaluating the features selected. These methods enabled the identification of a core set of features, from the same dataset. Table 2 represents the common features between this work and Al Khaldy [46] work. It could be noticed that there is a common feature between both of them.

It is said that there is no so-called "best feature selection method" [47] since the performance of feature selection relies on the performance of the learning method. The number of features selected by the GAs was 27 features using GAs and the accuracy was 87.8%. Of these features, three variables are the ones used by clinicians in diagnosing heart failure [46], namely Urea, Uric acid and Creatinine.

5. Conclusions

Feature selection algorithms based on analytical methods are numerically complex. On the other hand, Genetic algorithms, are less complex and can be used to arrive at

Table 2. Common Features between GAs, CFS, Systematical uncertainty and the Al Khaldy [46] study.

Common feature between GAs, CFS, systematical uncertainty	**Al Khaldy**
Urea (mmol/L)	4
Uric Acid (mmol/L)	4
MCV (fL)	5
Iron (umol/L)	6
Ferritin (ug/L)	4
CRP (mg/L)	3
CT-proET1	7
LVEDD (HgtIndexed)	6
E	3
Height (Exam)(m)	2
FVC (L)	6

a solution to complex optimisation problems. Casting the feature selection problem as an optimisation problem, and using a Genetic Algorithm for its solution provides us with a solution which is useful. However, the key questions of population sizes, the number of generations etc. remain to be answered. This paper, provides partial answers to these questions. Through the analysis of a complex dataset, it has illustrated the rules of thumb. What is interesting, is that the required number of generations does not change much given the different population sizes. What changes is the chromosome with the best fitness, this is natural for most feature selection problems. It shows that although it is possible to select a good sub-set, this subset is not unique and that there will always be a small variation in them. The compromise here is based around (a) expert advice, (b) the extra computational effort and (c) the marginal improvement in performance. The trade-offs are often then dependent on the nature of the application, if the margins are very fine, then a small marginal improvement in performance is well worth it.

Data availability

The data and code used in the work is available on request from the authors.

Conflict of interests/funding details

This work was assisted with support from the institutional research support fund. There is no conflict of interest in the work itself or the analysis.

Appendix A

Table 3. All 60 features of the heart failure dataset [29].

Feature Number	Feature Name (measurement)	Feature Number	Feature Name
1	Age	31	MR-proADM Mid regional pro-adrenomedullin
2	Sodium (mmol/L)	32	CT-proET1 C-terminal proendothelin-1
3	Potassium (mmol/L)	33	CT-proAVP Copeptin
4	Chloride (mmol/L)	34	PCT procalcitonin
5	Bicarbonate (mmol/L)	35	Rate (ECG) (bpm)
6	Urea (mmol/L)	36	QRS Width (msec) QRS complex width
7	Creatinine (umol/L)	37	QT QT Interval
8	Calcium (mmol/L)	38	LVEDD(cm) left ventricular end-diastolic diameter
9	Adj Calcium (mmol/L)	39	LVEDD (HgtIndexed)
10	Phosphate (mmol/L)	40	BSA (m^2) Body Surface Area
11	Bilirubin (umol/L)	41	Left Atrium (cm)
12	Alkaline Phophatase (iu/L)	42	LeftAtrium (BSAIndexed)
13	ALT (iu/L) Alanine transaminase	43	Left Atrium (HgtIndexed)
14	Total Protein (g/L)	44	Aortic Velocity (m/s)
15	Albumin (g/L)	45	E Examination
16	Uric Acid (mmol/L)	46	Height (Exam) (m)
17	Glucose (mmol/L)	47	Weight (Exam) (kg)
18	Cholesterol (mmol/L)	48	BMI Body Mass Index
19	Triglycerides (mmol/L)	49	Pulse (Exam) (bpm)
20	Haemoglobin (g/dL)	50	Systolic BP (mmHg)
21	White Cell Count (10^9/L)	51	Diastolic BP (mmHg)
22	Platelets (10^9/L)	52	Pulse BP (mmHg)
23	MCV (fL) mean corpuscular volume	53	Pulse BP (mmHg)
24	Hct (fraction) hematocrit	54	FEV1 (l) Forced expiratory volume
25	Iron (umol/L)	55	FEV1 Predicted (l)
26	Vitamin B12 (ng/L)	56	FEV1
27	Ferritin (ug/L)	57	FVC (l) Forced vital capacity
28	CRP (mg/L) C-Reactive Protein	58	FVCPredicted (l)
29	TSH (mU/L) Thyroid-Stimulating Hormone	59	FVC
30	MR-proANP Mid-regional pro atrial natriuretic peptide	60	PEFR (l) peak expiratory flow rate

Appendix B

Table 4. Acronyms and abbreviations.

Acronym	Meaning
CFS	Correlation-based-selector
D	Cost function
F	Set of features
FSP	Feature search problem
GA	Genetic algorithms
J	Fitness function
NP	Non-deterministic polynomial-time

References

1 **Katoch S.**, **Chauhan S. S.**, **Kumar V.** A review on genetic algorithm: past, present, and future. ***Multimed. Tools Appl.***, 2021; **80**: 8091–8126. https://doi.org/10.1007/s11042-020-10139-6.

2 **Moslehi F.**, **Haeri A.** An evolutionary computation-based approach for feature selection. ***J. Ambient Intell. Human Comput.***, 2020; **11**: 3757–3769. https://doi.org/10.1007/s12652-019-01570-1.

3 **Chandrashekar G.**, **Sahin F.** A survey on feature selection methods. ***Comput. Electr. Eng.***, 2014; **40**(1): 16–28.

4 **Panthong R.**, **Srivihok A.** Wrapper feature subset selection for dimension reduction based on ensemble learning algorithm. ***Procedia Comput. Sci.***, 2015; **72**: 162–169.

5 **Kumar V.**, **Minz S.** Feature selection: a literature review. ***Smart Comput. Rev.***, 2014; **4**(3): 211–229.

6 **Cheng L. J.**, **Wang K.**, **Morstatter S.**, **Trevino F.**, **Tang, J. R. P.**, **Liu H.** Feature selection: a data perspective. ***ACM Comput. Surv.***, 2017; **50**(6): 1–45.

7 **Xue B.**, **Zhang M.**, **Browne W. N.**, **Yao X.** A survey on evolutionary computation approaches to feature selection. ***IEEE Trans. Evolut. Comput.***, 2016; **20**(4): 606–626.

8 **Dash M.**, **Liu H.** Feature selection methods for classifications. ***Intell. Data Anal.***, 1997; **1**(4): 131–156.

9 **Cai J. et al.** Feature selection in machine learning: a new perspective. ***Neurocomputing***, 2018; **300**: 70–79.

10 **Shikhpourand R. et al.** A survey on semi-supervised feature selection methods. ***Pattern Recognit.***, 2017; **64**: 141–158.

11 **Anbarasi M. et al.** Enhanced prediction of heart disease with feature subset selection using genetic algorithm. ***Int. J. Eng. Sci. Technol.***, 2010; **2**(10): 5370–5376.

12 **Kohavi R.**, **John G. H.** The wrapper approach. In: Liu H., Motoda H. (eds), Feature Extraction, Construction and Selection. The Springer International Series in Engineering and Computer Sciencevol. 453, Boston, MA: Springer, 1998; p. 33.

13 **Akhil J.**, **Deekshatulu B.**, **Chandra P.** Classification of heart disease using K-nearest neighbour and genetic algorithm. ***Procedia Technol.***, 2013; **10**: 85–94.

14 **Tiwari R.**, **Singh M. P.** Correlation-based attribute selection using genetic algorithm. ***Int. J. Comput. Appl.***, 2010; **4**(8): 28–34.

15 **Alander J. T.** On optimal population size of genetic algorithms. In: CompEuro 1992 Proceedings Computer Systems and Software Engineering, The Hague, Netherlands. 1992; pp. 65–70.

16 **Liu H.**, **Yu L.** Toward integrating feature selection algorithms for classification and clustering. ***IEEE Trans. Knowl. Data Eng.***, 2005; **17**(4): 491–502.

17 **Jain A. K.**, **Duin R. P. W.**, **Mao J.** Statistical pattern recognition: a review. ***IEEE Trans. Pattern Anal. Mach. Intell.***, 2000; **22**(1): 4–37.

18 **Cover T. M.**, **Van Campenhout J. M.** On the possible orderings in the measurement selection problem. ***IEEE Trans. Syst. Man Cybern.***, 1977; **7**: 657–661.

19 **Narendra P. M.**, **Fukunaga K.** A branch and bound algorithm for feature subset selection. ***IEEE Trans. Comput.***, 1977; **26**: 917–922.

20 **Jain A. K.**, **Zongker D. E.** Feature selection: evaluation, application, and small sample performance. ***IEEE Trans. Pattern Anal. Mach. Intell.***, 1997; **19**(2): 153–158.

21 **Selim S. Z.**, **Ismail M. A.** K-means-type algorithms: a generalised convergence theorem and characterisation of local optimality. ***IEEE Trans. Pattern Anal. Mach. Intell.***, 1984; **PAMI-6**(1): 81–87.

22 **Siedlecki W.**, **Sklansky J.** A note on genetic algorithms for large-scale feature selection. ***Pattern Recog. Lett.***, 1989; **10**(5): 335–347.

23 **Sushil L. J.**, **Gregory J. E.** Predicting Convergence Time for Genetic Algorithms. 1993; pp. 141–161.

24 **Bradley P. S.**, **Mangasarian O. L.** Feature selection via concave minimisation and support vector machines. In: Shavlik J. (ed.), Learning Proceedings of the Fifteenth International Conference (ICML' '98). San Francisco, CA: Morgan Kaufmann, 1998; pp. 82–90.

25 **Leardi R.**, **Boggia R.**, **Terrile M.** Genetic algorithms as a strategy for feature selection. ***J. Chemometr.***, 1992; **6**(5): 267–281.

26 **Louis S. J.**, **Rawlins G. J.** Predicting convergence time for genetic algorithms. ***Found. Genet. Algorithms***, 1993; **2**: 141–161.

27 **Greenhalgh D.**, **Marshall S.** Convergence criteria for genetic algorithms. ***SIAM J. Comput.***, 2000; **30**: 269–282.

28 **Aytug H.**, **Bhattacharrya S.**, **Koehler G. J.** A Markov chain analysis of genetic algorithms with power of 2 cardinality alphabets. ***Eur. J. Oper. Res.***, 1997; **96**: 195–201.

29 **Khaldy M.**, **Kambhampati C.** Performance analysis of various missing value imputation methods on heart failure dataset. In: SAI Intelligent Systems Conference, London, UK. 2016.

30 **Alabed A.**, **Kambhampati C.**, **Gordon N.** Genetic algorithms as a feature selection tool in heart failure disease. In: Advances in Intelligent Systems and Computing. vol. 1229 AISC, 2020; pp. 531–543, https://doi.org/10.1007/978-3-030-52246-9_38.

31 **Oluleye B.**, **Armstrong L.**, **Leng J.**, **Dieeven D.** Zernike moments and genetic algorithm: tutorial and application. ***Br. J. Math. Comput. Sci.***, 2014; **4**(15): 2217–2236.

32 **Alander J. T.** On optimal population size of genetic algorithms. In: CompEuro 1992 Proceedings Computer Systems and Software Engineering, The Hague, Netherlands. 1992; pp. 65–70.

33 **Diaz-Gomez P. A.**, **Hougen D. F.** Initial population for genetic algorithms: a metric approacs. In: Proceedings of the 2007 International Conference on Genetic and Evolutionary Methods, GEM, Nevada, USA. 2007; pp. 55–63.

34 **Piszcz A.**, **Soule T.** Genetic programming: optimal population sizes for varying complexity problems. In: Proceedings of the Genetic and Evolutionary Computation Conference, Seattle, Washington, USA. 2006; pp. 953–954.

35 **Reeves C. R.** Using genetic algorithms with small populations. In: International Conference on Genetic Algorithms. vol. 5, San Mateo, CA: Kaufmann, 1993; pp. 90–92.

36 **Roeva O.** Improvement of genetic algorithm performance for identification of cultivation process models. In: Proceeding of the 9th WSEAS International Conference of Evolutionary Computing. Sofia, Bulgaria: World Scientific and Engineering Academy and Society WSEAS, 2008; pp. 34–39.

37 **Koumousis V. K.**, **Katsaras C. P.** A sawtooth genetic algorithm combining the effects of variable population size and reinitialisation to enhance performance. ***IEEE Trans. Evol. Comput.***, 2006; **10**(1): 19–28.

38 **Pelikan M.**, **Goldberg D. E.**, **Cantu-Paz E.** Bayesian optimisation algorithm, population sizing, and time to convergence. In: Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computing. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000; pp. 275–282.

39 **Lobo F. G.**, **Goldberg D. E.** The parameter-less genetic algorithm in practice. ***Inform. Comput. Sci.***, 2004; **167**(1–4): 217–232.

40 **Lobo F. G.**, **Lima C. F.** A review of adaptive population sizing schemes in genetic algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference. 2005; pp. 228–234.

41 **Raymer M. L.**, **Punch W. F.**, **Goodman E. D.**, **Khun L. A.**, **Jain A. K.** Dimensionality reduction using genetic algorithm. ***IEEE Trans. Evol. Comput.***, 2000; **4**(2): 164–171.

42 **Yang X.** Nature Inspired Optimisation Algorithms. London, UK: Elsevier, 2014ISBN: 978-0-12-416743-8.

43 **Lengler J.** General dichotomy of evolutionary algorithms on monotone functions. ***IEEE Trans. Evol. Comput.***, 2020; **24**(6): 995–1009.

44 **Quinlan J. R.** Induction of decision trees. ***Machine Learning***, 1986; **1**: 81–106.

45 **Hall M. A.** "Correlation-based feature subset selection for machine learning", Ph.D. thesis, Department of Computer Science, The University of Waikato, Hamilton, New Zealand, 1999.

46 **Al Khaldy M.** "Autoencoder for clinical data analysis and classification: data imputation, dimensional reduction, and pattern recognition", PhD thesis, Engineering and Computing Department, University of Hull, Hull, 2017.

47 **Bolón-Canedo V.**, **Sánchez-Maroño N.**, **Alonso-Betanzos A.** A review of feature selection methods on synthetic data. ***Knowl. Inf. Syst.***, 2013; **34**(3): 483–519.