 Ensemble multi-objective evolutionary algorithm for gene regulatory network reconstruction based on fuzzy cognitive maps

Jing Liu¹,², Yaxiong Chi¹, Zongdong Liu¹, Shan He²

¹School of Artificial Intelligence, Xiidian University, Xi’an, People’s Republic of China
²School of Computer Science, University of Birmingham, Birmingham, UK

E-mail: neouma@mail.xidian.edu.cn

Abstract: Many methods aim to use data, especially data about gene expression based on high throughput genomic methods, to identify complicated regulatory relationships between genes. The authors employ a simple but powerful tool, called fuzzy cognitive maps (FCMs), to accurately reconstruct gene regulatory networks (GRNs). Many automated methods have been carried out for training FCMs from data. These methods focus on simulating the observed time sequence data, but neglect the optimisation of network structure. In fact, the FCM learning problem is multi-objective which contains network structure information, thus, the authors propose a new algorithm combining ensemble strategy and multi-objective evolutionary algorithm (MOEA), called EMOEAFCM-GRN, to reconstruct GRNs based on FCMs. In EMOEAFCM-GRN, the MOEA first learns a series of networks with different structures by analysing historical data simultaneously, which is helpful in finding the target network with distinct optimal local information. Then, the networks which receive small simulation error on the training set are selected from the Pareto front and an efficient ensemble strategy is provided to combine these selected networks to the final network. The experiments on the DREAM4 challenge and synthetic FCMs illustrate that EMOEAFCM-GRN is efficient and able to reconstruct GRNs accurately.

1 Introduction

The so-called gene regulatory networks, or GRNs in short, refer to a collection of molecular regulators that interact with each other in the cell to govern the gene expression levels of mRNA and proteins. The regulator can be DNA, mRNA, protein and complexes of these [1].

These mRNA and proteins are extremely crucial to the reproduction and survival of organisms on earth. The GRNs contain many different genes and interactions, and one specific gene’s expression level is controlled by a complex network which is composed of many genes. Thus, the inference or reverse engineering of GRNs is essential for us to reveal the complicated networks of gene functions. Thus, it also refers to identify mutual interactions among distinct genes based on expression data [2]. Various models have been used to model GRNs. The simplest models are based on Boolean networks. These models are quite simple and benefit us a lot from offering high-level insights, which could help us understand the emerging properties and operational mechanism of GRNs much deeper [3]. In the reverse engineering, Boolean networks are used to infer both the underlying topology and the Boolean functions at the nodes from observed gene expression data. However, one limitation of the Boolean network models is that the inference algorithms need massive data to infer a densely connected Boolean network with general Boolean functions because of the high-dimensional parameters spaces [4].

Continuous networks, an extension of the Boolean networks, are also widely used to model GRNs. Nodes still represent genes and connections between them regulatory influences on gene expression. Genes in biological systems display a continuous range of activity levels and it has been argued that using a continuous representation captures several properties of GRNs not present in the Boolean model. Various methods based on continuous networks have been proposed to the inference of GRNs. For example, the (mutual information) MI-based methods, like ARACNE algorithm [5], calculate the value of MIs of all gene pairs. If the calculated value exceeds the given threshold, then one regulatory interaction is inferred [6].

Many probabilistic graphical models have been proposed by researchers to measure the high-order dependency among distinct gene expression patterns. The Bayesian network is one of the most popular methods used in the inference of GRNs. In the Bayesian network, directed acyclic graphs are used to indicate the conditional dependency among random variables [7].

Despite that plenty of proposed models using gene expression data to infer GRNs [8], these models only adopt scaled real values or Boolean values to indicate the levels of gene expression [9], which may result in the loss of information and thus the accuracy of the inferring will be reduced. Considering fuzzy reasoning ability, fuzzy cognitive maps (FCMs) are better tools compared with the aforementioned methods. FCMs use fuzzy values, which integrate the benefits of both real values and Boolean values, to figure out the real relationships in GRNs [9]. As a result, a good balance between real value and Boolean value representation can be achieved. Many researches [10] have validated that FCMs are an efficient and powerful tool when it comes to model complex regulatory network systems.

For the above considerations, within this context, we use FCMs to model GRNs and propose EMOEAFCM-GRN (for ‘ensemble multi-objective evolutionary algorithm’), a new GRN inference method.

In EMOEAFCM-GRN, a GRN is modelled as an FCM. The nodes in FCMs are the representation of genes in GRN and each node equals to one gene. As for the directed edges between the nodes, they represent causal relations. Then it could be considered as a problem about optimisation when learning the FCM based on data.

In recent years, many learning algorithms based on artificial intelligence such as genetic algorithm (GA) [11], particle swarm optimisation (PSO) [12], simulated annealing (SA) [13], have been
developed by scientists to learn FCMs. All these learning algorithms focus on modifying the interconnection weight matrix of FCM, and the aim is to minimise the value of Data Error which is a cost function that accounts for the approximation error on given training data. In the FCM learning problem, the training data are the time sequences which can be observed in real world. However, reducing the approximation error often leads to overfitting problem.

In fact, most FCMs learned by evolutionary-based algorithms are much denser than the target FCM, which means the model complexity is too high [14]. So an algorithm deals with the trade-off between approximation accuracy and model complexity is needed. That is to say, there are two objectives that need to be optimised in the learning process.

Traditional machine learning algorithms try to satisfy multiple objectives by combining the objectives into a scalar cost function. However, from the viewpoint of multi-objective optimisation, there is no single learning model that can satisfy different objectives at the same time.

In this sense, Pareto-based multi-objective optimisation is a powerful way to deal with the conflicting objectives in machine learning. Jin [15] presents a collection of most representative research work on multi-objective machine learning.

In our previous work [16], multi-objective evolutionary algorithm was proposed to reconstruct FCM (MOEA-FCM) from historical data, which first consider using network structure information as an objective in the learning algorithm.

In MOEA-FCM, FCM learning problems are modelled as multi-objective optimisation problems (MOPs) with two objectives. Evolutionary algorithms (EAs) are used for solving MOPs to create Pareto-optimal front consisting of many FCMs. The reason to use Data Error is for the minimisation of the discrepancy between the sequences generated by candidate FCM and the observed historical sequence. A low Data Error means a good performance of FCM simulation. Density is also employed to search for various FCMs with different network structures via decreasing density of links among FCMs.

However, there are two disadvantages in the previous work, (i) the Density only reflects the partial information of network structural features. For example, in Fig. 1, the following two FCMs (directed graphs) have the same density value, but the network structures are totally different, none of the connected edges are the same; (ii) need a pre-set parameter to guide how to select the final solution from Pareto-front. In our previous work [16], MOEA-FCM only selected one FCM from Pareto-optimal front which density is closest to 37%, the average density of FCMs in the real world [14]. However, 37% is a predefined value which contains a hypothesis that the density of network need to be reconstructed is really close to 37%. What if the real network is dense? So this pre-set value is not suitable for all cases. Moreover, this selection strategy leads to the abandon of other solutions in Pareto-optimal fronts, which means the final FCM may be losing some network information which is useful for learning the real FCMs.

In order to solve the shortcomings discussed above, in this work, we proposed a new objective Similarity which contains sufficient network structure information as the learning goal in the proposed algorithm, and this is one of the main contributions of this work.

Another contribution of this work is that we provide an effective ensemble strategy to obtain an FCM from a series of FCMs (Pareto-optimal front) learned by MOEA. As FCMs carry similarities with neural networks in many aspects, and the formation of neural networks ensemble has attracted much attention in the machine learning literatures. In [17], Abbass presents a theoretically sound approach for the formation of neural network ensembles.

Thus, we propose an ensemble multi-objective evolutionary algorithm (EMOEA) to model GRNs. The proposed algorithm is named as EMOEA\_FCM\_GRN. Just like the MOEA-FCM algorithm, in EMOEA\_FCM\_GRN, the FCM learning problems are also modelled as multi-objective. The Data Error is set as the first objective, and the new designed Similarity which is used to evaluate the difference of network structure between the candidate FCM and the other FCMs in the Pareto-optimal front is set as the second objective.

In this work, EMOEA\_FCM\_GRN has two steps. First, EAs are used for solving the above MOP. Then, an ensemble strategy is used to integrate the candidate FCMs with different optimal network information to the final FCMs without any prior knowledge.

Both the synthetic FCMs and a real-world application, namely, reconstruction of a biological GRN, benchmark DREAM4, validate the efficiency of EMOEA\_FCM\_GRN. The experimental results show that EMOEA\_FCM\_GRN is capable of learning FCMs and GRNs with 100 nodes, which means 10,000 weights need to be optimised. Also, it is validated by the systematic comparison that compared with other learning algorithms, the EMOEA\_FCM\_GRN has a much better performance.

The major contributions of this paper are summarised as follows:

(i) We propose a new FCM learning algorithm, named as EMOEA\_FCM\_GRN. In the new proposed algorithm, we designed a new optimisation objective based on structure similarity which first takes sufficient network structure information into consideration.

(ii) An efficient ensemble strategy is provided in EMOEA\_FCM\_GRN to combine the Pareto solutions to one solution which maintained at a stable good level in all performance measurements.

(iii) Apply the new proposed FCMs learning algorithm to reconstruct GRN, and leading accuracies are achieved on both synthetic FCMs and DREAM4 challenge project which illustrates the efficiency of the EMOEA\_FCM\_GRN.

The remaining part of this paper is organised in the following order. Section 2 gives related work on GRN reconstruction and FCM learning. In Section 3, the methods used to model GRNs based on FCMs are described. Details of EMOEA\_FCM\_GRN are given in Section 4. In Section 5, the results of various experiments are reported. Finally, the conclusion is made in Section 6.

2 Related work

Recently, systems biologists are sparing no efforts to use the gene expression data to infer GRNs [18]. Various methods [7, 19–21] have been proposed to uncover the mutual relationships between genes and proteins, which are the final products of genes. Since the framework of EMOEA\_FCM\_GRN is based on EAs, in this section, we first introduce existing EAs for inferring GRNs, and then give a brief overview of related work.

2.1 EAs for GRNs reconstruction

As one bionics optimisation algorithm, EAs are playing increasingly more important roles in indicating the relationship between genes from the whole genome levels [22].

Rispensber et al. [23] used the GA to simulate the related gene expression data by modelling multistate discrete networks.

CAAI Trans. Intell. Technol., 2019, Vol. 4, Iss. 1, pp. 24–36

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)
Eriksson and Olsson [24] used the genetic programming to model discrete GRNs. In this model, each gene takes a Boolean value. For the target gene, the regulatory networks are encoded as one tree and would be evolved to gain better structures.

Except qualitative models, EAs have been developed to infer continuous GRN also. Tomina et al. [25] used time series data to infer GRNs via using double-encode GA, which is quite classic. Yet, a very small network with only two genes was used in this work. Then, a method was proposed by Fomekong-Nanfac et al. in [26] which applied classic evolution strategy based on partial differential equations to optimise model parameters for a network with six genes. Obviously, the size of GRN is very small. According to the empirical comparison results in [27], for now on, classical EAs have not been successfully applied to the inference of large GRNs.

The most important reason is that the linear growth of the number of gene will result in a quadratic increase in the number of variables to infer [14]. One of the common strategies solving the above problem is using EAs to optimise variables for each individual gene [28]. The comparison study [27] validated that the algorithms using this distributed strategy outperform those which optimise the entire networks simultaneously.

2.2 FCMs learning

FCMs, proposed by Kosko in [29], are a kind of powerful tools for modelling complex systems. They have been used in a variety of applications such as time series analysis [30, 31], human reliability in maintenance [32], modelling of Parkinson’s disease [33] and so on.

Papageorgiou and his colleagues systematically reviewed learning algorithms [10]. According to this review, these learning algorithms could be classified into three distinct types based on the historical data and/or the experts’ knowledge. Namely, they are evolutionary-based algorithms, Hebbian-based algorithms and hybrid-learning algorithms. The last algorithm is based on the combination of the former two kinds of algorithms.

NHL, or non-linear Hebbian learning, is a representative method among the Hebbian-based learners [34]. This algorithm modifies the weight update formula and then extends the basic Hebbian rules non-linearly [10]. Then, Stach et al. [35] proposed a data-driven NHL (DDNHL) algorithm to learn FCMs. The DDNHL algorithm is a reinforced NHL version. It could raise the quality of learning by using output concepts and taking advantages of the historical data [10].

However, Hebbian-based methods need prior knowledge from experts, such as input and output concept and weighted interconnection, to build the initial network [10]. In order to solve the above problem, Stach et al. [11] developed a new algorithm called the real coded genetic algorithm (RCGA) to directly use observed data to learn FCMs. It is the objective of RCGA to figure out the best FCM, which could simulate the observed data best.

In addition to GA, other EAs, like PSO [36], ACO [37] and so on, are also used to learn FCMs. All these learning algorithms are based on using the observed available historical data and are modulated to fit the network which use optimisation operators to mimic historical data.

Since the densities of FCMs learned by EAs are denser than the real value, Stach et al. in [14] puts forward a new kind of RCGA, named the spare RCGA or SRCGA, to guide the learning towards finding maps of one predefined density by using density estimate parameters.

Besides the above two mentioned learning types, some hybrid approaches based on evolutionary learning algorithms and Hebbian learning methods were proposed to learn the FCM.

Papageorgiou et al. presented the couple of NHL and DE algorithms based on the efficacy of NHL rule and the global search capabilities of EAs [38]. Zhu and Zhang [39] combined NHL and RCGA to propose one new method to learn FCM, and applied this method in the selection of partners.

3 FCM for GRN reconstruction

The FCM is a simple and efficient tool to describe network regulated system. In this work, the FCM is used to model GRNs; that is, a GRN can be described as a signed directed graph with the number of $N$ nodes. Each of the nodes equals one gene, and gene $i$ directs to another gene $j$ to form an edge that means the gene $j$ is regulated by gene $i$.

From the perspective of graph theory, the FCM refers to one signed digraph, structured as $N$ concept nodes and the digraph can be denoted as an $N \times N$ weight matrix $W$ which is used to describe the casual relationships of FCMs

\[
W = \begin{bmatrix}
    w_{11} & \cdots & w_{1N} \\
    \vdots & \ddots & \vdots \\
    w_{N1} & \cdots & w_{NN}
\end{bmatrix}
\]

where $w_{ij}$ represents the influence of node $i$ to node $j$, $i, j = 1, 2, \ldots, N$. The value of $w_{ij}$ is in the range of $[-1, 1]$, where $-1$ means the strongest negative impact, 0 means no impact, and by that analogy, $+1$ means the strongest positive impact. Fig. 2a shows a simple FCM with five nodes, and the interconnection weight matrix is shown in Fig. 2b. For instance, $w_{23} = 0.3$ means node 2 has a positive excitatory influence to node 1; $w_{24} = 0$ denotes there is no influence from node 2 to node 4; $w_{13} = -0.7$ means a negative excitatory pointing from node 1 to node 3 with a strength value of 0.7.

As for the goal of GRN reconstruction, it aims at inferring the complicated networks based on the expression data of gene, which are the measurements of the gene in various time points or conditions. We concentrate on time series data in the paper.

In what follows, we define the time series measurements for each gene as a vector with $T$ measurements. Similarly, in FCMs, each node has a series of state values varying by the time point. For example, the state values for node $i$ at different time point can be denoted as a vector $C_i$:

\[
C_i = [C_i(0), C_i(1), \ldots, C_i(T)]
\]

where $C_i(t)$ is the state value of node $i$ at the $t$th time point and it is in the range of $[0, 1]$, $t = 1, 2, \ldots, T$. $N$ is the number of nodes in FCMs and $T$ is the length of the time series data. The state value of a node at the $(t + 1)$th time point is determined by the strength of the relationship and the state value of related nodes at the $t$th time point. Namely, the dynamics of FCMs is determined by

\[
C_i(t + 1) = g \left( \sum_{j \neq i} w_{ij} C_j(t) \right), \quad \forall i \in \{1, 2, \ldots, N\}
\]

where $C_i(t + 1)$ is the state value of node $i$ at the $(t + 1)$th time point, and $g(\bullet)$ function is used as a transfer function to bound the expression level to the range $[0, 1]$. Transfer functions are commonly used in the analysis of control systems and they have

\[\text{Fig. 2. FCM with five nodes}\]

\[\begin{align*}
    a & \quad \text{Topological structure} \\
    b & \quad \text{Interconnection weight matrix}
\end{align*}\]
various formula forms. From the study in [40], in general, sigmoid function performs the best. Therefore, in this work, the sigmoid function is used as well

\[ g(x) = \frac{1}{1 + e^{-\lambda x}} \]  

(4)

where \( \lambda \) is a real value and it is usually determined by choosing distinct value for lambda and the best one is confirmed according to their performances [37]. In this paper, we choose 5 as the value of \( \lambda \) because a lot of researchers use this value in FCM learning [41].

### 4 EMOEA_FCM-GRN

In this section, we develop a novel network learning algorithm that combines multi-objective evolutionary algorithm (MOEA) and ensemble strategy. The proposed algorithm is labelled as EMOEA_FCM-GRN.

The learning procedure is composed of two steps: MOEA step and ensemble step. (i) First, the problem of FCM learning is modelled as an MOP with two objectives. A candidate FCM is a solution to an MOP, as Coello mentioned that there are no other practicable solutions because all the known solutions could not increase criterion without affecting other criterion [42]. In the work, we use an MOEA for learning the above MOP in consideration of the most welcomed optimisation method to solve MOP is the EA. From the MOEA step, we can obtain a series of Pareto optimal front. (ii) Then, we use ensemble strategy to integrate the trade-off solutions to one solution which combines various networks information from Pareto optimal solutions.

#### 4.1 Two objective functions

The FCM learning problem aims at reconstructing the actual network structure. The task of reconstructing GRNs is turned into a MOP because the weight and bias of each node are considered as the decision variable. The FCM learning problem aims at reconstructing the actual network structure. The task of reconstructing GRNs is turned into a MOP because the weight and bias of each node are considered as the decision variable.

#### 4.1.1 The objective function of the MOEA

The objective function of the MOEA is defined as follows:

Minimise \[ f_1(W) = \text{Data Error}(W) \]  

(5)

where \( N \) refers to the number of nodes, \( T \) means length of time sequence, \( C_n(t) \) and \( \hat{C}_n(t) \) are the state value of concept node \( n \) at the time \( t \) in the observed time series data and the generated time sequence, respectively. Since the initial state value of \( C_n(t) \) is the same as \( \hat{C}_n(0) \), Data Error only considers the difference of state values of the last \( T-1 \) time points between observed time sequence and the generated time sequence.

As EMOEA_FCM-GRN is an evolutionary-based (or can be called population-based) algorithm, selecting the individuals which fit better is the basic aim. However, if Data Error is used as the only objective, the entire population may converge towards a single network structure.

However, the second step of EMOEA_FCM-GRN, ensemble method, tends to yield better results when there is a significant diversity among the models [43], therefore, EMOEA_FCM-GRN needs to seek to promote diversity among the candidate FCMs we combine; that is, we need to make sure the individuals in the population keep certain distance between each other.

Thus, we design the second objective as Similarity, which is used to evaluate the diversity of an individual in the population. More specifically, Similarity is employed to assess the structure distance of candidate FCM and all other FCMs in the population.

Since Similarity only consider the difference between network structures, the candidate FCM needs to transfer into binary one according to the viewpoint in [37], which is if the absolute value of weight between two nodes is smaller than 0.05, the relation between these two nodes can be ignored in practical application, the weight value is set to 0; otherwise, the weight value is set to 1. Here, we use \( W_{01} \) to represent the binary form of interconnection matrix \( W \).

Then, we define Euclidean distance-based metric as \( \Omega \) for individual. In EMOEA_FCM-GRN, each individual is a candidate interconnection matrix. In an individual, candidate interconnection matrix is pulled into 1D vector, in the following equation, we use \( \Omega(W_{01}) \) to represent the vector form of an interconnection matrix \( W \).

\[ \Omega(W_{01}) = \sum_{j=1}^{N_{pop}} \|W_{01} - W_{01}^j\| \]  

(6)

where \( N_{pop} \) is the population size, \( \|W_{01} - W_{01}^j\| = \sum_{n=1}^{N\times N} |w_n - w_n^j| \), where \( w_n^j \) is the \( n \)th element of \( W_{01} \) and \( N \times N \) is the length of \( W_{01} \). \( \Omega(W_{01}) \) represents the sum of the distances of the solution \( W_{01} \) from all other solutions \( (W_{01}^j, j = 1, 2, ..., N_{pop}) \) of the population. Then, the two objectives of each individual in the population can be defined as follows:

Minimise \[ f_1(W) = \text{Data Error}(W) \]  

(7)

Minimise \[ f_2(W) = \text{Similarity}(W_{01}) = -\frac{\Omega(W_{01})}{N_{pop} - 1} \]  

(8)

where \( f_1 \) implies the ability of a solution of fitting the observed historical data. \( f_2 \) is the opposite value of the average distance ((\( \Omega(W_{01})/N_{pop} - 1 \)) of a solution from all other solutions in the population of size \( N_{pop} \). A solution with low \( f_2 \) implies that it is far from the other ones in the population, conversely, a solution with high \( f_2 \) means there are some other solutions with similar network structure near it.

#### 4.2 MOEA step

The task of reconstructing GRNs is turned into a MOP because the two objective functions obtained above contradict with each other. Therefore, we need to use a multi-objective optimisation algorithm to get a set of solutions on the Pareto front, which corresponds to a set of networks.

In the past few years, many studies have been devoted to apply EAs to MOPs [43–45], where NSGA-II [45] showed an excellent performance among the EAs. Thus, in the MOEA step of EMOEA_FCM-GRN, we used NSGA-II as the framework to optimise the above objectives. In the following subsection, we introduce the representation and suitable evolutionary operators used in MOEA step.

##### 4.2.1 Individual representation: In EMOEA_FCM-GRN

Each individual is a candidate weight matrix which has been transformed to a 1D vector with \( N \times N \) elements by linking each row. Thus, each individual can be represented as \( W = [w_{11}, w_{12}, ..., w_{1N}, w_{21}, w_{22}, ..., w_{2N}, ..., w_{N1}, w_{N2}, ..., w_{NN}] \).

##### 4.2.2 Non-dominated sorting: Non-dominated sorting is the key operator in NSGA-II. The non-dominated sort [44, 45] can be described as follows.

Each individual \( p \) in the population \( P \) has two entities. \( S_p \) is a set containing all the individuals that are dominated by \( p \), and \( n_p \) is the number of individuals that dominate \( p \). First, initialise \( S_p = \Phi \) and \( n_p = 0 \); Then, check each individual \( q \) in \( P \), if \( p \) is dominated by \( q \), then \( q \) is a non-dominated solution, add \( q \) to the first front \( F_1 \). Now, for each individual \( p \) in front \( F_1 \), we visit each member \( q \) in \( S_p \), which
is the set of individuals dominated by \( p \), and reduce its \( n_p \) count by one. In doing so, if \( n_p = 0 \), we put it in the set \( Q \) which is the set for storing individuals for front \( F_2 \). When all individuals in \( F_1 \) have been checked, we declare the individuals in set \( Q \) as members of the second \( F_2 \), i.e. \( F_2 = Q \).

This process continues till all fronts \( (F_1, F_2, \ldots, F_n) \) are identified [45]. Front \( F_1 \) is completely non-dominated set, the individuals in front \( F_2 \) are dominated by individuals in \( F_1 \) only and the front goes so on.

### 4.2.3 Crowding distance

When all fronts are identified based on the above non-dominated sort, the crowding distance value is calculated for each individual at the same front. Comparing the crowding distance between two individuals in different front is meaningless [46]. The crowding distance is used to measure the distance of an individual from its neighbours in the objective space. Individuals with large crowding distance own a better diversity (or survival probability) than other individuals in the same front. More details about crowding distance can be found in [45, 46].

### 4.2.4 Generation of new individuals and evolutionary operators

Parent individuals are selected from the sorted population using the binary tournament selection based on the front level \( (F_i) \) is the best level) and the crowding distance (compare the front level first, then compare the crowding distance if and only if they are at the same level).

Then, evolutionary operators are performed on the parent population to create an offspring population \( Q \) of size \( N \). Many different crossover operators [47] can be employed in EMOEA\(_{FCM-GRN}\). In this work, we consider a discrete crossover because it effectively expands the search scope of network structure and carries a low computational cost. Also, random mutation [48] is applied.

### 4.2.5 Selection

Suppose we have the parent population \( G \) of size \( N \), and offspring population \( Q \) of size \( N \) is created by evolutionary operators. Thus, a combined population \( R = G + Q \) is formed of size \( 2N \). All individuals in \( R \) are sorted by the non-dominated sorting approach, and each individual is assigned a front level \( (F_1, F_2, \ldots, F_n) \). Then, the algorithm adds individuals from good fronts \( (F_i) \) is the best front and so on) to the new population until the size exceed \( N \) [45]. Then individuals in the last accepted front are selected based on their crowding distance in the descending order until the size reaches \( N \) [45].

The above several operators are repeated until some stop criteria have been reached. In the end of MOEA step, a set of non-dominated (trade-off) solutions (Pareto optimal solutions) are obtained. Pareto optimal solutions provide the best FCM under their respective network structure.

### 4.3 Ensemble step

Ensemble step is a learning paradigm that uses a set of learned networks to form a single super-network.

In contrast to our previous work (MOEA-FCM) which tries to select only one network from the Pareto optimal front, EMOEA\(_{FCM-GRN}\) constructs a set of learned networks and combines them to use. Each of Pareto optimal solutions is either better in terms of Data Error, or better in terms of Similarity. Krogh and Vedelsby [49] claim that for an ensemble to be effective, the integrated elements in ensemble method should be as more accurate as possible. Unfortunately, the solutions with low similarity value and high data error in Pareto optimal front own very unique network structures, but these network structures do not perform well in simulating the observed historical data. That is, these solutions do not provide useful network information. Also, combination of these solutions reduces the accuracy of the final reconstructed network as demonstrated through experiments. So, it is necessary to develop a method to select useful solutions from the Pareto optimal front.

#### Algorithm 1: EMOEA\(_{FCM-GRN}\)

1. Initialize population \( P \) with \( N \) individuals, where each element in the individuals is randomly generated from the range \([1,1]\).
2. \( F \) = fast non-dominated sort \( (P) \).
3. Binary tournament selection and evolutionary operators are employed on parent population to generate offspring population \( Q \) with size \( N \).
4. Combine \( P \) and \( Q \) to form the new population \( P \), with size \( 2N \).
5. \( F \) = fast non-dominated sort \( (R) \).
6. Select individuals from fronts \( F \) in the order of \( \{F_1, F_2, F_3, \ldots, F_n\} \) to \( P \), until the size exceed \( N \).
7. Suppose the last accept front is \( F_i \), crowding-distance-assignment \( (F_t) \) and sort \( F_i \) in descending order according to the crowding distance.
8. \( P_{t+1} = P_{t+1} \cup F_{t+1}(1: \text{N-size}(P_{t+1})) \).
9. \( t = t + 1 \).
10. If \( t \geq \text{maximum number of generations} \), then output Pareto optimal front of \( P_t \).
11. Sort Pareto optimal solutions in descending order according to the Data Error, and select the top \( 30\% \) solutions \( W_1, W_2, \ldots, W_k \).
12. Final network = Combine \( (W_1, W_2, \ldots, W_k) \).

In order to solve this problem, we sort the Pareto optimal solutions in descending order according to the Data Error, and average the top \( 30\% \) networks to integrate. For example, if there are \( N \) networks selected from the Pareto optimal front, the final network is calculated by the following:

\[
W = \frac{1}{N} \sum_{i=1}^{N} W_i
\]

where \( W_i \) is the \( i \)th network selected from the Pareto optimal front, and \( N \) is the total number. Fig. 3 shows the learning procedure of ensemble step, and Algorithm 1 summarises the whole step-wise description of EMOEA\(_{FCM-GRN}\).

In the ensemble step, some better solutions are selected from all solutions first. Then, these selected solutions are combined to form a final network according to (9). So the diversity of solutions is taken into account as well.

### 5 Experiments

In order to assess EMOEA\(_{FCM-GRN}\), both synthetic FCM and benchmark DREAM4 [50] are used. The experimental results are compared with our previous work, MOEA-FCM and other mainstream methods, DD-NHL [35], RCGA [11], sparse RCGA [14] and ACO\(_{AD}\) (an improved version over ACO) [51], where DD-NHL is a representative method based on Hebbian method and the other methods are based on EAs.

DD-NHL, RCGA, sparse-RCGA were used for FCM learning. ACO\(_{AD}\) was used to reconstruct GRN and MOEA-FCM was used to learn both FCM and GRN. The results used in the comparison are cited from existing literature. ACO\(_{AD}\), MOEA-FCM and EMOEA\(_{FCM-GRN}\).
EMOEA\textsubscript{FCM-GRN} were running under the same experimental environment for DREAM4 datasets. The effectiveness of MOEA-FCM shows that the parameter settings are reasonable. Thus, the parameters used in EMOEA\textsubscript{FCM-GRN}, namely the crossover probability, mutation probability and number of objective function evaluations, are set as the same as those used in MOEA-FCM, which are 0.9, 0.5 and 3 \times 10^6, respectively.

In addition to Data Error, Out-of-Sample-Error, Model Error and SS Mean are also used to evaluate the quality of the obtained FCM network. Out-of-Sample-Error is used to test the generalisation capability of the obtained FCM in dealing with over-fitting to the input historical data. First, ten randomly chosen initial state vectors that are not used to learn the obtained FCM are assigned to both the real FCM and the obtained FCM. Then, the error can be calculated for the ten simulated sequences generated by the real FCM and the obtained FCM

\[
\text{Out-of-Sample-Error}(W) = \frac{1}{R \cdot (T - 1) \cdot N} \sum_{r=1}^{R} \sum_{t=1}^{T-1} \sum_{j=1}^{N} |C_{rj}(t) - \hat{C}_{rj}(t)|
\]

where $R$ refers to the number of different initial state vectors, $T$ is the length of time sequence and $N$ is the number of nodes in the network. Model Error is used to compare the interconnection matrix of the obtained FCM network and the original FCM network

\[
\text{Model Error} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} |w_{ij} - \hat{w}_{ij}|
\]

where $\hat{w}_{ij}$ is the estimated weight of the edge from node $i$ to $j$. As identifying plausible causal relation among genes plays an important role in the reconstruction of GRNs, the learning algorithm is required to predict the existence of an edge between two nodes in an FCM. More specifically, the obtained FCM and the real FCM model are transformed into binary networks according to a predefined threshold based on the rule [37]. That is, the absolute weights that are larger than the threshold are transformed to 1 (negative edges); otherwise 0 (positive edges) [37]. In this paper, we choose 0.05 as the value of threshold which is the same value used in [41]. Two measures, Sensitivity and Specificity, are employed to measure the performance, which are defined as follows:

\[
\text{Sensitivity} = \frac{N_{TP}}{N_{TP} + N_{FN}}
\]

\[
\text{Specificity} = \frac{N_{TN}}{N_{TN} + N_{FP}}
\]

where $TP$, $FN$, $TN$ and $FP$ are defined in Table 1. $T$ (True) means the correctly identified edges, $F$ (False) means the edge is identified as the opposite character. $P$ (Positive) means positive edge and $N$ (Negative) means negative edge which defined above (12). For example, $N_{TP}$ is the number of correctly identified positive edges.

In addition, SS Mean, the harmonic mean of Sensitivity and Specificity, is also popularly used to evaluate the ability of algorithm for predicting the existence of links. It is calculated as (14)

\[
\text{SS Mean} = \frac{2 \times \text{Sensitivity} \times \text{Specificity}}{\text{Sensitivity} + \text{Specificity}}
\]

The value of SS Mean is 100% means a perfect reconstruction.

### Table 1  Meaning of TP, FN, TN and FP

| Target networks | 0 | 1 |
|-----------------|---|---|
| Obtained networks | 0 | TP | FP |
| | 1 | FN | TN |

5.1 Experiments on synthetic data

In order to compare other methods fairly, the input data in all the comparison methods are generated by the same method in [37], which include two steps.

First, random real numbers which should be within the interval $[-1, 1]$ are assigned to the weights of the interconnection matrix. However, according to the viewpoint in [37], which is if the absolute value of weight between two nodes is smaller than 0.05, the relation between these nodes can be ignored in practical application, we check each weight whether its absolute value is smaller than 0.05. If so, the weight will be set to 0. Second, the state value of response time sequences at the initial time point are assigned by random value ranging from 0 to 1. Then, the subsequent sequences can be generated according to (3) based on the FCM model and initial state values.

In the following experiments, the number of nodes in the generated FCM varies from 5 to 100, and for each size, the edge density is set of 20 and 40%, and the length of response sequence is 10. The Pareto-optimal fronts obtained by EMOEA\textsubscript{FCM-GRN} are reported in Figs. 4–8.

![Fig. 4 Pareto-optimal fronts obtained by EMOEA\textsubscript{FCM-GRN} for synthetic FCMs with five nodes](image)

- Target density = 20%
- Target density = 40%

![Fig. 5 Pareto-optimal fronts obtained by EMOEA\textsubscript{FCM-GRN} for synthetic FCMs with ten nodes](image)

- Target density = 20%
- Target density = 40%

![Fig. 6 Pareto-optimal fronts obtained by EMOEA\textsubscript{FCM-GRN} for synthetic FCMs with 20 nodes](image)

- Target density = 20%
- Target density = 40%
As can be seen, EMOEA\textsubscript{FCM}-GRN is able to find the Pareto-optimal fronts which show the conflict between Data Error and Similarity. EMOEA\textsubscript{FCM}-GRN provides a series of candidate FCM models with different network structure, and most of these models have relative small Data Error. However, it is always difficult to decide which networks in the Pareto-set to include or exclude from the ensemble. We conduct the following experiment to discuss how many solutions included is the best result.

First, we sorted these FCM networks in descending order according to the value of related Data Error; Then, we ensemble the first $P$% solutions, where the value of $P$ ranges from 10 to 100 in the step of 10; Last step, we calculated the Data Error, Model Error and SS Mean for each ensemble solution. The results are shown from Figs. 9–11.

As can be seen, the more solutions included in the ensemble, the bigger the Data Error. However, the Data Error cannot reflect the quality of the model. In terms of Model Error, no matter how large the scale of the network, the best performance happens when the proportion of the included solutions is from 20 to 40%. Only one case is an exception when the FCM consists of five nodes and the target density is 0.4. The similar situation happens in the SS Mean terms, best ensemble proportion lies in the range of [20%, 40%].

If only a little fraction of the solutions are included in the ensemble, the Data Error of ensemble result is the very low, but the number of network is too few to provide sufficient model information. On the other hand, although the solutions with low similarity own very unique network structures, the Data Error of these networks are too large to provide useful network information. Combination of these networks would reduce the accuracy of the final ensemble solution. We need to provide a suitable ensemble proportion to integrate the advantages of both.

According to the above experiments, we selected the top 30% networks from the Pareto-optimal fronts and averaged them to the final FCM. In Figs. 4–8, the red points represent Pareto solutions obtained by MOEA, and the ‘star’ points represent the good solution selected from the Pareto-optimal fronts according to the above rule.

As can be seen from the results, although the Similarity of these good solutions are very large which means the network structures of these solutions are very different, the Data Error is very low. A low value of Data Error reflects the candidate FCM has some useful information on network features in the real FCM.

Figs. 12–26 show the performance measures of candidate FCMs selected from Pareto-optimal fronts and the experimental results of
the final FCM (red dotted line) after ensemble method. As can be seen, whatever the size of FCMs, the candidate FCM which performs the best in one performance measure terms is not always the best in the other performance measure terms. As can be seen from the Pareto-optimal front from Figs. 4–8, there is no FCM could be the best in terms of both objectives. Although the ensemble FCM did not show the best results in all the solutions, it still provides a stable and relative good result.
In order to further discuss the choice of the ensemble strategy, we choose three types of solutions to compare, which is the best network based on Data Error from Pareto-set, simple averaged network and the weighted average network. The simple averaged network is calculated by (9), and the weighted average network is calculated by following:

$$\overline{W} = \sum_{i=1}^{N} x_i \cdot W_i$$

(15)

where $W_i$ is the $i$th network for ensemble, $N$ is the number of integrated networks and $x_i$ is the weight for $i$th network $x_i = \frac{\text{Fitness}_i}{\sum_{i=1}^{N} \text{Fitness}_i}$

(16)

where Fitness$_i$ is the first fitness value of the $i$th network, and the first fitness value corresponds to the first optimization objective, Data Error. That is to say, the network with smaller data error will get a larger weight value.

Figs. 27 and 28 show that in terms of Model Error and SS Mean, simple average network and weighted average network did not show a big difference. Both of them perform better than the best network (based on Data Error) which has a worse Model Error and SS Mean.
Stach et al. [14] also gave a similar conclusion that the FCMs with minimum Data Error always performs not well in terms of density which consists of partial model information.

Figs. 29–32, respectively, show the comparison results on FCMs with 5–40 nodes in terms of Data Error, Out-of-Sample-Error, Model Error and SS Mean. The results are averaged over ten independent runs. As can be seen, in terms of Data Error, EMOEA\textsubscript{FCM}-GRN can reach 0 for FCMs with five nodes, no matter the density is 20 or 40%. Since the Data Error of DD-NHL is more than 0.1 which is about several to more than ten times of that of the other algorithms, we did not show the Data Error results of DD-NHL in Fig. 29 for ease of comparison between the other algorithms.
For FCMs with 10, 20 and 40 nodes, the Data Error of EMOEA\textsubscript{FCM-GRN} is lower than the other methods except one case (node = 40, density = 20%), which means EMOEA\textsubscript{FCM-GRN} has better performance in simulating the historical data. In terms of Model Error, EMOEA\textsubscript{FCM-GRN} performs not as well as MOEA-FCM and sparse-RCGA for FCM with 5 nodes, but for FCM with 10, 20 and 40 nodes, EMOEA\textsubscript{FCM-GRN} always performs the best in all cases. Similarly, EMOEA\textsubscript{FCM-GRN} outperforms the other algorithms in 7 out of all the 8 cases in Out-of-Sample-Error terms. In addition, in terms of SS Mean, EMOEA\textsubscript{FCM-GRN} always outperforms RCGA and DD-NHL, and performs better than MOEA-FCM in 5 out of 8 cases.

We further validate the performance of EMOEA\textsubscript{FCM-GRN} on large-scale synthetic data with 100 nodes. Since the experimental results of sparse-RCGA, RCGA and DD-NHL are obtained from the existing literatures which are only used for learning FCM with small-scale size, i.e. not more than 40 nodes, we only compare the results with MOEA-FCM in Table 2.

The results are averaged for ten dependent runs. As can be seen, EMOEA\textsubscript{FCM-GRN} outperforms MOEA-FCM in terms of Data Error and Out-of-Sample-Error.

In addition, in Model Error and SS Mean terms, EMOEA\textsubscript{FCM-GRN} outperforms MOEA-FCM in all cases except two, which means EMOEA\textsubscript{FCM-GRN} improves the ability of reconstruction of large-scale network structure greatly.

As the FCMs in real world are always sparse, unlike our previous work, in EMOEA\textsubscript{FCM-GRN}, the density is not set as a direct objective to minimise. In the following experiments, we calculate the average density of obtained FCM with various numbers of nodes and shown in Fig. 33.

As can be seen, no matter what the size of FCMs is, the densities of obtained FCMs are always fluctuating around their densities. The experimental results show that EMOEA\textsubscript{FCM-GRN} can still learn FCMs with sparse structure.

5.2 Experiments on DREAM4 in silico network challenge

DREAM4 in silico network challenge [50] provides biologically plausible simulated gene expression datasets in order to evaluate various reverse engineering methods in an unbiased manner. The goal of the challenge is to reverse engineer gene regulation networks from simulated time-series data. The sub-challenges differ in the size of the network (10 genes and 100 genes). Each sub-challenge consists of five networks.

For each network, there are five time series under different perturbations generated based on stochastic differential equation models with both internal and measurement noise. Every time series contains 21 time points, but only the last 11 time points reflect the response of the gene network after the wild type network is restored. In this experiment, we use the last 11 time points and test the performance of EMOEA\textsubscript{FCM-GRN} in terms of Data Error and SS Mean.

Figs. 34 and 35 show the Pareto-optimal fronts obtained by EMOEA\textsubscript{FCM-GRN} for sub-challenges with 10 genes and 100 genes, respectively. The comparison results are shown in Figs. 36 and 37.

In terms of Data Error, most of the results for GRNs with 10 genes are smaller than 0.1, and results for GRNs with 100 genes are smaller than 0.2. EMOEA\textsubscript{FCM-GRN} performs the best for GRNs with 10 genes. For GRNs with 100 genes, the Data Error of EMOEA\textsubscript{FCM-GRN} and MOEA-FCM achieved are much lower than ACO\textsubscript{RD} in all cases, and EMOEA\textsubscript{FCM-GRN} outperforms MOEA-FCM in 3 cases out of 5 cases.

![Fig. 33](image-url) Average density of obtained FCMs with 5, 10, 20, 40 and 100 nodes

| #Genes | Edge density, % | Data Error | Out-of-Sample-Error | Model Error | SS Mean |
|--------|----------------|------------|---------------------|-------------|---------|
| EMOEA\textsubscript{FCM-GRN} | 20 | 0.084 ± 0.025 | 0.109 ± 0.064 | 0.390 ± 0.176 | 0.49 |
| MOEA-FCM | 20 | 0.135 ± 0.082 | 0.193 ± 0.107 | 0.373 ± 0.120 | 0.51 |
| EMOEA\textsubscript{FCM-GRN} | 40 | 0.105 ± 0.064 | 0.209 ± 0.087 | 0.293 ± 0.143 | 0.42 |
| MOEA-FCM | 40 | 0.200 ± 0.108 | 0.240 ± 0.126 | 0.398 ± 0.184 | 0.40 |

Fig. 34 Pareto-optimal fronts obtained by EMOEA\textsubscript{FCM-GRN} for first sub-challenge of DREAM4 in silico network with ten genes

a Network 1  
b Network 2  
c Network 3  
d Network 4  
e Network 5

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)
In terms of SS Mean, EMOEA\textsubscript{FCM}-GRN outperforms MOEA-FCM for GRNs with 5 genes, and it also performs better than ACO\textsubscript{RD} in 3 cases out of 5 cases. Although EMOEA\textsubscript{FCM}-GRN performs not as well as ACO\textsubscript{RD} in 2 cases, for large scale GRNs with 100 genes, EMOEA\textsubscript{FCM}-GRN outperforms ACO\textsubscript{RD} in all cases and is still better than MOEA-FCM in all cases except the experiment on Network 5. The results show that EMOEA\textsubscript{FCM}-GRN improves the ability of reconstructing GRNs with high accuracy.

6 Conclusions

In this paper, EMOEA\textsubscript{FCM}-GRN is proposed to learn GRN based on FCM model. First, EMOEA\textsubscript{FCM}-GRN models the learning problem of FCMs as multi-objective. Then it selects a good part of solutions from Pareto-optimal front based on simulation degree of the observed input data, and integrates these solutions into a complete FCM. Both synthetic data and DREAM4 challenge validated that the superiority of the proposed algorithm outperforms existing methods in terms of various performance measures. Future work will focus on reconstructing larger scale GRNs by developing a distributed system.

7 References

[1] Bolouri, H.: ‘Computational modeling of gene regulatory networks – a primer’ (Imperial College Press, London, 2008)
[2] Bansal, M., Belcastro, V., Ambesi-Impiombato, A., et al.: ‘How to infer gene networks from expression profiles’, Mol. Syst. Biol., 2007, pp. 3–78
[3] Kauffman, S.A.: ‘The origins of order: self organization and selection in evolution’ (Oxford University Press, New York, USA, 1993)
[4] Flick, V.: ‘Identifying gene regulatory networks from gene expression data’, in Aluru, S. (ED.): ‘Handbook of computational molecular biology’ (CRC Press I Llc, New York, USA, 2005), vol. 27, p. 1–29
[5] Margolin, A.A., Wang, K., Lim, W.K., et al.: ‘Widows of crowds for robust gene network inference’, Nat. Protoce., 2006, 1, (2), pp. 662–671
[6] Mittel, A.J., Kohane, I.S.: ‘Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements’, Pac. Symp. Biocomput., 2000, 5, pp. 418–429
[7] Friedman, N., Linial, M., Nachman, I., et al.: ‘Using Bayesian networks to analyze expression data’, J. Comput. Biol., 2000, 7, (3–4), pp. 601–620
[8] Marbach, D., Costello, J.C., Kuffner, R., et al.: ‘Widows of crowds for robust gene network inference’, Nat. Methods, 2012, 9, (8), pp. 796–804
[9] Chen, Y., Mazzack, L.J., Alt, A.M., et al.: ‘Inferring causal networks using fuzzy cognitive maps and evolutionary algorithms with application to gene regulatory network reconstruction’, Appl. Soft Comput., 2015, 37, pp. 667–679
[10] Papageorgiou, E.I.: ‘Learning algorithms for fuzzy cognitive maps – a review study’, IEEE Trans. Syst. Man Cybern. C, 2012, 42, (2), pp. 150–163
[11] Stach, W., Kurgan, L., Pedrycz, W., et al.: ‘Genetic learning of fuzzy cognitive maps’, Fuzzy Sets Syst., 2005, 153, pp. 371–401
[12] Papageorgiou, E.I., Panopoulos, K.E., Stylios, C.S., et al.: ‘Fuzzy cognitive maps learning using particle swarm optimization’, J. Intell. Inf. Syst., 2005, 25, pp. 95–121
[13] Ghazanfari, M., Alizadeh, S., Fathian, M., et al.: ‘Learning of fuzzy cognitive maps using a density estimate and genetic algorithm in learning FCM’, Appl. Math. Comput., 2007, 192, pp. 56–68
[14] Stach, W., Pedrycz, W., Kurgan, L.A.: ‘Learning of fuzzy cognitive maps using density estimate’, IEEE Trans. Syst. Man Cybern., B, 2012, 42, (3), pp. 900–912
[15] Jin, Y.: ‘Multi-objective machine learning’, vol. 16 (Springer-Verlag, Berlin, Heidelberg, 2006)
[16] Chi, Y., Liu, J.: ‘Learning of fuzzy cognitive maps with varying densities using a multi-objective evolutionary algorithm’, IEEE Trans. Fuzzy Syst., 2016, 24, (1), pp. 71–81
[17] Abbas, H.A.: ‘Pareto neuro-ensemble’. Australasian Joint Conf. on Artificial Intelligence (Springer-Verlag Berlin, Heidelberg, 2003), pp. 554–566
[18] Przytycka, T.M., Singh, M., Slonim, D.K.: ‘Toward the dynamic interactive: it’s about time’, Brief. Bioinform., 2010, 11, (1), pp. 15–29
[19] Kauffman, S., Peterson, C., Samuelsson, B., et al.: ‘Random Boolean network models and the yeast transcriptional network’, Proc. Natl. Acad. Sci., 2003, 100, (25), pp. 14796–14799
[20] Kim, S.Y., Imoto, S., Miyano, S.: ‘Inferring gene networks from time series microarray data using dynamic Bayesian networks’, Bioinformatics, 2003, 19, (3), pp. 228–235
[21] Penfold, C.A., Wild, D.L.: ‘How to infer gene networks from expression profiles’, revised’, Interface Focus, 2011, 1, pp. 857–870
[22] Stibò, A., Ruskin, H.J., Crane, M.: ‘Stages of gene regulatory network inference: the evolutionary algorithm role’ (INTECH Open Access Publisher, 2011)
[23] Repsüber, D., Liljenström, H., Anderson, S.G.E.: ‘Reverse engineering of regulatory networks: simulation studies on a genetic algorithm approach for ranking hypotheses’, Biosystems, 2002, 66, (1), pp. 31–41
[24] Eriksson, R., Olsson, B.: ‘Adapting genetic regulatory models by genetic programming’, Biosystems, 2004, 76, (1), pp. 217–227
[25] Tominaga, D., Okamoto, M., Maki, Y., et al.: ‘Nonlinear numerical optimization technique based on a genetic algorithm for inverse problems: towards the inference of genetic networks’, German Conf. on Bioinformatics, Hannover, Germany, 1999, vol. 99, pp. 101–111
This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/).