A Constrained Evolutionary Computation Method for Detecting Controlling Regions of Cortical Networks

Yang Tang, Zidong Wang, Huijun Gao, Stephen Swift, and Jürgen Kurths

Abstract—Controlling regions in cortical networks, which serve as key nodes to control the dynamics of networks to a desired state, can be detected by minimizing the eigenratio R and the maximum imaginary part σ of an extended connection matrix. Until now, optimal selection of the set of controlling regions is still an open problem and this paper represents the first attempt to include two measures of controllability into one unified framework. The detection problem of controlling regions in cortical networks is converted into a constrained optimization problem (COP), where the objective function R is minimized and σ is regarded as a constraint. Then, the detection of controlling regions of a weighted and directed complex network (e.g., a cortical network of a cat), is thoroughly investigated. The controlling regions of cortical networks are successfully detected by means of an improved dynamic hybrid framework (IDyHF). Our experiments verify that the proposed IDyHF outperforms two recently developed evolutionary computation methods in constrained optimization field and some traditional methods in control theory as well as graph theory. Based on the IDyHF, the controlling regions are detected in a microscopic and macroscopic way. Our results unveil the dependence of controlling regions on the number of driver nodes l and the constraint r. The controlling regions are largely selected from the regions with a large in-degree and a small out-degree. When $r = +\infty$, there exists a concave shape of the mean degrees of the driver nodes, i.e., the regions with a large degree are of great importance to the control of the networks when l is small and the regions with a small degree are helpful to control the networks when l increases. When $r = 0$, the mean degrees of the driver nodes increase as a function of l. We find that controlling σ is becoming more important in controlling a cortical network with increasing l. The methods and results of detecting controlling regions in this paper would promote the coordination and information consensus of various kinds of real-world complex networks including transportation networks, genetic regulatory networks, and social networks, etc.

Index Terms—Cortical networks, synchronization, controllability, evolutionary computation, constrained optimization

1 Introduction

Complex networks can be used to model real-world systems and have attracted research interests from the fields of biology, social and technical society [1], [2], [3], [4], [5], [6]. Among others, synchronization of complex networks of coupled oscillators has been the subject of intensive research activity [7], [8]. In particular, synchronization of distributed brain activity has been found to play a vital role in neural information processing [9], [10], [11], [12]. The experimentally observed evidence has revealed that certain brain disorders, such as schizophrenia, epilepsy, autism, Alzheimer's disease, and Parkinson's disease, are relevant to abnormal neural synchronization [10].

In recent years, the analysis of the anatomical connectivity of the mammalian cortex, such as human, cat, and monkey brains, has shown that large-scale cortical networks reflect typical characteristics of small-world networks and hierarchy, e.g., high clustering and short path length [13], [14], [15], [16]. In addition, hub regions, which are thought to play pivotal roles in the coordination and transportation of information in neuronal networks [17], [18], [19], have been identified using graph theory such as degree, betweenness centrality (BC) and motif structure [17], [20].

In physical, social, and biological networks, the requirement of regulating the behavior of large ensembles of interacting units is a common feature [21], [22], [23]. Controllability of complex networks is an intuitive notion of control, capturing a capability to lead a network's state to a desired goal by suitable manipulation of a few input variables [21], [22], [24]. Based on the idea of feedback mechanism, some vertices in the networks serve as reference sites, leaders, or pacemakers [25], and regulate all the other vertices in the networks toward desired states. In [24], the pinning control of complex networks has been investigated

- Y. Tang is with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150080, China, the Institute of Physics, Humboldt University, Berlin 12489, Germany, and the Department of Transdisciplinary Concepts and Methods, Potsdam Institute for Climate Impact Research, Telegraphenberg A31, Potsdam 14415, Germany. E-mail: tangtany@gmail.com.
- Z. Wang and S. Swift are with the Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex UB8 3PH, United Kingdom. E-mail: {Zidong.Wang, Stephen.Swift}@brunel.ac.uk.
- H. Gao is with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, PO Box 3015, Yikuang Street 2#, Nangang District, Harbin 150080, Heilongjiang Province, China. E-mail: hjgao@hit.edu.cn.
- J. Kurths is with the Department of Transdisciplinary Concepts and Methods, Potsdam Institute for Climate Impact Research, Telegraphenberg A31, Potsdam 14473, Germany, the Institute of Physics, Humboldt University, Berlin, Germany, and the Institute for Complex Systems and Mathematical Biology, University of Aberdeen, Aberdeen AB24 3UE, United Kingdom. E-mail: Juergen.Kurths@pik-potsdam.de.

Manuscript received 26 Dec. 2011; revised 16 July 2012; accepted 1 Sept. 2012; published online 28 Sept. 2012.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2011-12-0342.
Digital Object Identifier no. 10.1109/TCBB.2012.124.

based on the degree information and eigenvalue analysis. Then, by defining the pinning controllability of a network in terms of the spectral properties of an extended network topology, the roles of coupling gains and the number of pinned nodes were illustrated in [21]. More recently, by utilizing control theory and assuming linear dynamics in each vertex, the results of "structure controllability" show that, in both model and real systems, the driver nodes tend to avoid the high-degree nodes [22], [23]. Unfortunately, it is still not clear how to identify the locations of driver nodes and represent the coupling between the desired state and the driver nodes, especially for a directed and weighted network. In addition, the existing results concentrated on how to minimize R (the ratio of the largest real part of the eigenvalues of the connected matrix and the second smallest real part of the eigenvalues of the connected matrix) while neglecting σ (the largest absolute value of the imaginary part of the eigenvalues of the connected matrix), which might induce unavoidable conservativeness. Moreover, how to embrace two measures of controllability R and σ into a framework to evaluate synchronizability and controllability of complex networks exactly is still open and remains a challenging problem [22], [26], [27], which is the first motivation of this paper.

As a special complex network, the cortical network of cat brain is a typical directed and weighted network. It is of great importance to investigate the controllability of cortical networks for the following two reasons: 1) it is widely thought that the ultimate proof of our understanding of social, natural, or technological systems is shown by our ability to control them efficiently [22], [23], [28]. As a typical weighted and directed natural network, the detection of controlling regions of cortical networks will be helpful to grasp how to control a natural system efficiently; 2) controllability is of special relevance with synchronization [21], which plays a significant role in the processing of high-level information in brain networks [15] and is useful for investigating their dynamical properties [20]. This paper is aimed to further understand the processing of high-level information in a neuronal network to avoid abnormal synchronization in typical neural diseases such as schizophrenia, epilepsy, autism, Alzheimer's disease, and Parkinson's disease, which is the second motivation of this paper [10].

Different from the above results of controllability of networks, in this paper, the controllability and the detection of controlling regions of cortical networks are transformed into a single objective optimization problem with a constraint, i.e., a *constrained* optimization problem (COP). However, the traditional easy enumeration method or degree-based methods in [21], [24], [29] cannot be simply applied to deal with the proposed problem. For handling the locations of driver nodes of cortical networks and designing their control gains under constraints, the evolutionary computation method emerges as a promising one.

Evolutionary algorithms (EAs) are population-based search approaches that take their inspiration from natural selection and survival of the fittest in biological society. Recently, EAs have been well utilized in studying multiobjective synchronization, controller design for synchronization of two coupled systems and inferring gene regulatory networks in [6], [30], [31]. The use of EAs for constrained optimization problems has been significantly increased in the past 10 years, leading to the development of constrained optimization evolutionary algorithms (COEAs) [32], [33], [34]. COEAs consist of two major parts: a search algorithm and a constraint-handling approach. The performance of COEAs depends largely on these two components. The goal of the search algorithm is to enhance the exploration and exploitation abilities of the population, while the constraint-handling approach is devoted to incorporate the constraints into the EAs. Recently, the constraint-handling approaches using multiobjective optimization concepts have drawn increasing attention due to their satisfactory efficiency [34], [35], [36]. Although the performance of the methods in [35], [36] has been verified to show promising performance by comparing with a number of well-known COEAs, there still remains much room to improve the search algorithm, thereby enhancing the entire search performance of the algorithm, which is the third motivation of this paper.

Motivated by the above discussion, in this paper, the two measures of controllability of cortical networks R and σ are embraced into a unified constrained optimization framework. Then, an improved dynamic hybrid framework (IDyHF) optimization approach is developed to handle the problem of identification of controlling regions in cortical networks. Through an appropriate encoding scheme, the selection of driver nodes and the design of their control gains are dealt with the proposed IDyHF. The effectiveness of the proposed method is analyzed and validated by several simulation examples. Some interesting findings about the relevance of the number of driver nodes, the controllability and constraints, are revealed by our proposed method. This paper will not only help us to understand the information processing of a typical biological network as a neural network of cat, but also be conducive to further understanding the controllability of a weighted and directed network. The main contributions of this paper are summarized as follows:

1. To the best of authors' knowledge, this paper is the first attempt to transform the controllability problem of a directed and weighted cortical network into a constrained optimization problem, in which minimizing R serves as the objective function and minimizing σ is viewed as a constraint. Therefore, the impacts of both R and σ on controllability are taken into account simultaneously, which has not been addressed in the references so far;
2. an improved dynamic hybrid framework is proposed, which performs better than two other recently developed COEAs and methods from control theory;
3. the controlling regions are identified by the IDyHF in a microscopic and macroscopic way;
4. some interesting findings regarding the controlling regions, the number of driver nodes, and the characterization of eigenvalues are illustrated.

The remainder of this paper is organized as follows: In Section 2, some preliminaries and problem formulation are briefly outlined. The encoding scheme of EAs and IDyHF is presented in Section 3. In Section 4, numerical examples and comparisons are provided and discussed. Conclusions are given in Section 5.

2 Problem Formulation

In this section, some representative references regarding cat graphs [13], complex networks [8], evolutionary computation methods [32], and controllability [21] are provided here for a clear presentation. The details of cat graphs are provided in [19], [37].

2.1 Notations and Cat Graph

Throughout this paper, $l \in [1, N]$ stands for the number of driver nodes of a network, where N is the network size. $\delta_{\mathcal{M}}(\cdot)$ denotes the characteristic function of the set $\mathcal{M}$, i.e., $\delta_{\mathcal{M}}(i) = 1$ if $i \in \mathcal{M}$; otherwise, $\delta_{\mathcal{M}}(i) = 0$. Define a graph by $\mathcal{G} = [\mathcal{V}, \mathcal{E}]$, where $\mathcal{V} = \{1, \ldots, N\}$ represents the vertex set and $\mathcal{E} = \{e(i,j)\}$ is the edge set. The graph $\mathcal{G}$ is assumed to be directed, weighted and simple (without self-loops and multiple edges). Let the weighted and directed matrix $G = [g_{ij}]_{i,j=1}^{N}$ be the adjacency matrix of cat graph $\mathcal{G}$, which is defined as follows: for any pair $i \neq j, g_{ij} < 0$ if $e(i,j) \in \mathcal{E}$; otherwise, $g_{ij} = 0$. $g_{ii} = -\sum_{j=1, j\neq i}^{N} g_{ij} (i = 1, 2, \ldots, N)$. One can convert the adjacency matrix G into the Laplacian matrix L by neglecting the weights over the networks. For any pair $i \neq j, l_{ij} = -1$ if $e(i,j) \in \mathcal{E}$; otherwise, $l_{ij} = 0$. $l_{ii} = -\sum_{j=1, j\neq i}^{N} l_{ij}$, $(i = 1, 2, \ldots, N)$. The output-degree $k_{out}(i) = -\sum_{j=1, i\neq j}^{N} l_{ij}$ of a node i is the number of efferent connections that it projects to other nodes, and its input-degree $k_{in}(i) = -\sum_{j=1, i\neq j}^{N} l_{ji}$, is the number of the afferent connections it receives.

Hereafter, the cat corticocortical network, which shows the anatomical connectivity, is employed as an example for a complex network. The cerebral cortex of a cat can be partitioned into 53 cortical regions ($N = 53$), connected by about 830 fibers of different densities into a weighted and directed complex network as shown in Fig. 1. This network has been found to exhibit typical small-world properties, i.e., short average path length and high clustering coefficient, indicating an optimal coordination for effective interarea communication and for achieving high functional complexity [17]. The cat cortical network also exhibits a hierarchically clustered organization [37]. Recent studies have shown that there exist four topological clusters that widely agree with four functional cortical communities: visual cortex (16 areas), auditory (seven areas), somato-motor (16 areas), and fronto-limbic (14 areas). Here, the topological clusters are also referred to communities or modules. The connection matrix G of cortical networks is both weighted and directed, which is more general than an unweighted or undirected connection matrix studied in most of the existing well-studied works regarding controllability of networks [24], [29], [38].

2.2 Controllability of Cortical Networks

In the following, let a reference evolution/state (desired state) be as follows:

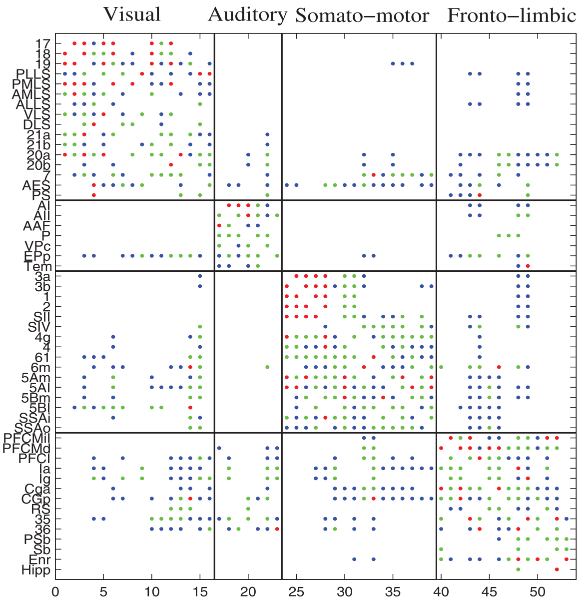


Fig. 1. The brain cortical network of the cat. The weighted adjacency matrix is shown (red: three dense, green: two intermediate, blue: one sparse). The matrix shows the partition of the network into four main modules (communities) of modally related areas: visual, auditory, somatosensory-motor, and fronto-limbic.

$$\frac{ds(t)}{dt} = f(s(t)).$$

The above differential equation is general enough to represent extensive real-world complex systems such as social networks, biological systems, and other natural systems [23].

The diffusively coupled array of identical systems with a feedback controller is written as

$$\begin{aligned} \frac{dx_i(t)}{dt} = f(x_i, t) - c\sum_{j=1}^{N} g_{ij} h(x_j(t)) \\ - c\delta_{\mathcal{M}}(i)\kappa_i(h(s(t)) - h(x_i(t))), \\ i = 1, \ldots, N, \end{aligned} \tag{1}$$

where $x_i(t) = [x_{i1}(t), x_{i2}(t), \ldots, x_{in}(t)]^T \in \mathbb{R}^n \quad (i = 1, 2 \ldots, N)$ is the state vector of the ith node/region/area and $f(x_i, t) = [f_1(x_i, t), \ldots, f_n(x_i, t)]^T$ is a continuous vector function. n denotes the dimensional size of each node. c is the global coupling gain of the networks. In the coupling term, the node is connected through a generic output function $h(x_i(t))$. The coupling matrix G describes the information about the cortical networks topology, as described in Fig. 1. Let $\mu_p = \mu_p^r + j\mu_p^i (j = \sqrt{-1}), (p = 1, 2, \ldots, N)$, be the set of eigenvalues of G and assume that they are ordered in such a way that $\mu_1^r \leq \mu_2^r \leq \cdots \leq \mu_N^r$. κ_i are the control gains or coupling strengths between the vertex and the desired state. It is clear that $1 \leq \sum_{i=1}^{N} \delta_{\mathcal{M}}(i) \leq N$. The goal of pinning control is to guide a cortical network (1) toward the desired reference state $s(t)$, i.e., $x_1(t) = x_2(t) = \cdots = x_N(t) = s(t)$.

In order to seek to assess the controllability of network (1), we consider an extended network of $N+1$ dynamical systems y_i, where $y_i = x_i$ for $i = 1, 2, \ldots, N$ and $y_{N+1} = s$. Then, (1) can be rewritten as follows [21]:

$$\frac{dy_i(t)}{dt} = f(y_i, t) - c\sum_{j=1}^{N+1} \mathcal{W}_{ij} h(y_j(t)), \quad i = 1, \ldots, N+1, \tag{2}$$

where $\mathcal{W} = [\mathcal{W}_{ij}] \in \mathbb{R}^{(N+1)\times(N+1)}$ in the form of

$$\mathcal{W} = \begin{pmatrix} \mathcal{A}_1 & g_{12} & \dots & g_{1N} & -\delta_{\mathcal{M}}(1)\kappa_1 \\ g_{21} & \mathcal{A}_2 & \dots & g_{2N} & -\delta_{\mathcal{M}}(2)\kappa_2 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ g_{N1} & g_{N2} & \dots & \mathcal{A}_N & -\delta_{\mathcal{M}}(N)\kappa_N \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}, \tag{3}$$

in which $\mathcal{A}_i = g_{ii} + \delta_{\mathcal{M}}(i)\kappa_i$. Let $\lambda_p = \lambda_p^r + j\lambda_p^i$ be the pth eigenvalue of $\mathcal{W}$ and assume that λ_p is sorted as $\lambda_1^r \leq \lambda_2^r \leq \cdots \leq \lambda_{N+1}^r$, where $\lambda_1^r = 0$.

Similar to the analysis method of checking synchronizability of networks [21], the controllability can be evaluated in terms of

$$R = \frac{\lambda_{N+1}^r}{\lambda_2^r},$$

and

$$\sigma = \max_p \{\lambda_p^i\}.$$

The smaller R and σ are, the easier the network is controllable [21], [26]. Liu et al. [22] consider the controllability of state transfer in the network, while this paper and Sorrentino et al. [21] investigate the controllability of synchronization. In [22], if a network is controllable, one state can be driven by a proper control input to any state. However, in [21] and in this paper, the smaller R and σ, the easier the network can be controlled to a synchronization manifold. Besides, there are two main differences between [22] and this paper: 1) The model considered here is nonlinear, which is very typical in real-life but the model in [22] is limited to a linear one; 2) The adopted methods are also different. In [22], the controllability in control theory is used whereas we use evolutionary methods and control approaches to handle the controllability problem.

2.3 Problem Transformation into a Constrained Optimization Problem

In previous works, σ is usually neglected, since σ is very small in most of the coupling graph and thus has only minor impacts on synchronizability [27] and controllability of the network. However, this assumption might induce unavoidable conservativeness and cannot reflect the actual synchronizability and controllability of networks. In addition, in some special networks, e.g., normalized Laplacian matrix, the value of σ is comparable to that of R. In this paper, we transform the controllability of cortical networks into a constrained optimization problem, in which optimizing R is regarded as an objective and minimizing σ is viewed as a constraint, since usually R plays a more important role in measuring controllability of networks than σ. In the following, we first give some preliminaries of the constrained optimization problem and then transform the controllability of cortical networks into a COP.

In general, the constrained optimization problem is formulated as follows: find the decision variables $x = (x_1, \ldots, x_D) \in \mathbb{R}^D$ to minimize the objective function

$$\min \quad F_j(x), x \in \Omega \subseteq S,$$

where Ω is the feasible region and S is the decision space defined by the parametric constraints $L_i \leq x_i \leq U_i, i = 1, 2, \ldots, D$. x should satisfy m constraints including u inequality constraints

$$Q_j(x) \leq 0, j = 1, 2, \ldots, u,$$

and $\iota = m - u$ equality constraints

$$H_j(x) = 0, j = u+1, 2, \ldots, m.$$

In general, the degree of constraint violation of a vector x on the jth constraint is defined as

$$M_j(x) = \begin{cases} \max\{0, Q_j(x)\}, & 1 \leq j \leq u, \\ \max\{0, |H_j(x)|\}, & u+1 \leq j \leq m. \end{cases} \tag{4}$$

Then, $\Sigma(x) = \sum_{j=1}^m M_j(x)$ reflects the degree of constraint violation of the vector x.

Therefore, based on the above analysis, we consider three cases in this work:

1. The first case is formulated as follows:

$$\min R = \frac{\lambda_{N+1}^r}{\lambda_2^r}, \quad \text{subject to} : Q_1(x) \leq 0, \tag{5}$$

where $Q_1(x) = \sigma - r, r = +\infty$. The first problem means that the problem considered here is unconstrained and one should only minimize R as small as possible. Hence, there is no need to use COEAs to solve the first case. EAs for a single objective are strong enough to tackle this problem.

2. The second case can be written as

$$\min R = \frac{\lambda_{N+1}^r}{\lambda_2^r}, \quad \text{subject to} : Q_1(x) \leq 0, \tag{6}$$

where $Q_1(x) = \sigma - r, r \in (0, +\infty)$. Here, we just use a representative number $r = 0.1$, which can distinguish the case II from the case I and the case III. The second case is to minimize R as well as make the inequality constraint feasible.

3. The third case is formulated in the following way:

$$\min R = \frac{\lambda_{N+1}^r}{\lambda_2^r}, \quad \text{subject to: } H_1(x) = 0, \tag{7}$$

where $H_1(x) = \sigma - r, r = 0$. The third case indicates that one should minimize R and suppress $\sigma = 0$. In this way, the effect of σ can be neglected completely.

2.4 The Approaches for Determining the Locations of Driver Nodes

In this section, the following strategies for detecting the locations of driver nodes or controlling regions are taken into account:

1. **Degree-based scheme**. The driver nodes are selected according to out-degree information k_{out} in an ascending or a descending way. The two schemes

are called the ascending and the descending degree-based scheme, respectively.

2. **Betweenness centrality-based scheme**. Similar with the degree-based scheme, a descending and an ascending BC-based scheme are used here.
3. **Closeness-based scheme**. Two types of closeness-based strategies, i.e., the descending and the ascending closeness-based strategy are employed.
4. **Node importance-based scheme**. The node importance for undirected networks has been proposed in [39]. It is worthwhile to point out that few works are devoted to the node importance for directed networks. We consider two measures of node importance for the cortical network, which are widely used in characterizing synchronizability of a directed network. The first one is to minimize $\frac{\mu_N^r}{\mu_2^r}$ of G upon sequential removal of nodes, which is called $U=\frac{\mu_N^r}{\mu_2^r}$**-based scheme**. The other one is to minimize $S=\max_p\{\mu_p^i\}$ of G upon sequential removal of nodes, which is called **S-based scheme**.
5. **Evolutionary algorithm-based scheme**. Constrained optimization evolutionary algorithms are used to select driver nodes and design their control gains.

Similar with the work in [21], in the degree-based, the BC-based, the closeness-based, and the node importance-based schemes, the control gains in all the nodes are assumed to be identical and are gradually tuned.

3 Improved Dynamic Hybrid Framework and Its Encoding Scheme

3.1 IDyHF

In [35], a dynamic hybrid framework (DyHF) was proposed which includes a global search and a local search. In the global and local search models, differential evolution (DE) acts as search algorithm, and Pareto dominance used in multiobjective optimization works as a constraint-handling technique to compare the individuals in the population. The DyHF transforms a constrained optimization problem into a biobjective optimization problem $\overrightarrow{F}(x)=(F(x),\Sigma(x))$ by treating the degree of constraint violation $\Sigma(x)$ as an additional objective. Consequently, the original objective function $F(x)$ and the degree of constraint violation $\Sigma(x)$ should be taken into consideration simultaneously when comparing the individuals in the population. A scientific set of experiments and solid comparisons have validated the effectiveness and performance of the DyHF, which is very promising to deal with real-world problems [35].

As discussed in the Introduction, COEAs include a search algorithm and a constraint-handling technique. In this paper, we retain the constraint-handling technique of the DyHF due to its efficiency and make some modifications on the part of the search algorithm. The modifications concentrated on the part of global search in the DyHF. The global search scheme in the DyHF utilizes a simple mutation and a crossover scheme from the traditional DE. However, the scheme adopted in [35] is too simple and thus it is not adaptive to fit the search circumstances. Therefore, we adopt a recently developed self-adaptive differential evolution (jDE) to generate offspring to enhance the global search scheme, which can efficiently adjust the control parameters in differential evolution and thus adapt to the search situations [40]. The experiments have shown that the jDE is easily implemented, has the ability of a good convergence speed, and a good search accuracy.

Since the main algorithm structure of the DyHF, the local search scheme, and the constraint handling technique are retained in the IDyHF, we do not repeat the details here. The global search model proposed in this paper focuses on exploring more promising regions and refining the overall performance of the population, which adopts the following framework:

- **Step 1**. Each target vector $x_i(i=1,2,\dots,NP)$ in the population P is utilized to generate a trial vector u_i through the mutation and crossover operations of the jDE. The control parameters of jDE is updated according to the self-adaptive scheme [40].
- **Step 2**. Compute the two objectives, i.e., $F(x)$ and the $\Sigma(x)$, for the trial vector u_i.
- **Step 3**. If u_i dominates x_i, the trial vector u_i will replace the target vector x_i, else no replacement occurs.

By using the trial vector to remove the inferior target vector, the update of the population P is achieved through Pareto dominance. It is clear that our modification of the DyHF is simple and easily implemented; however, the experimental results in the next section show the performance is encouraging and promising. Another point that should be taken into account is that IDyHF is based on DyHF and therefore its complexity is similar to DyHF, which can be referred to [35]. All evolutionary methods with an elitism method such as IDyHF can be ensured to find the global optimum with probability 1 if the number of generation is infinity. For more details concerning the DyHF and the jDE, we refer the readers to [35], [40], and the appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2012.124.

3.2 The Encoding Scheme

In this section, an appropriate encoding scheme is used. It consists of two components with the equal dimension size l: the first one is an integer search space to denote the locations of driver nodes; the second one is a continuous search space to represent the control gains of driver nodes.

In order to illustrate our encoding scheme, an example for the encoding scheme is given and explained. For the sake of simplicity, $l=2$ nodes are selected as driver nodes and thus the dimension size D of the population is $D=2\times l$. The parameter constraints are adopted as $L_i=0, U_i=N$. Let an individual be $x=(x_1,x_2,x_3,x_4)=(51.1,8.7,21.9,4.2)$. Since the node index is an integer, the round operators are performed in the first part of each individual and thus $x=(51,9,21.9,4.2)$. The encoding denotes that the regions (nodes) $i=51$ and $i=9$ are chosen as driver nodes and injected with feedback controllers, i.e., $\delta_{\mathcal{M}}(51)=1$ and $\delta_{\mathcal{M}}(9)=1$. In addition, the second part of encoding means that the control gains of regions 51 and 9 are 21.9 and 4.2, respectively, which indicates that $\kappa_{51}=21.9$ and $\kappa_9=4.2$. From the above discussion, the encoding scheme is very simple and easily implemented. The evolutionary computation algorithms adopted in this paper can use this encoding scheme to locate the position of driver nodes and determine their control gains.

The advantages of using COEAs to study the controllability of cortical networks and identify the controlling regions are obvious and discussed in the following. The search range of each dimension is assumed to be the same and written as $\triangle x = (U_i - L_i)$. In order to find the driver nodes from $N = 53$ as a function of l, one has to locate l driver nodes out of N vertices. Then, there are C_N^l distinct combinations. For instance, there exist 9.0×10^{14} combinations when $l = 25$. It is impossible to adopt a Brute-force method to select the driver nodes. In addition, even if the locations of driver nodes are known a priori, the problem is then reduced into an l-dimensional continuous optimization problem. In [21], [41], by simply assuming the control gains of each driver node to be identical, the control gains are tuned gradually with a step size ϵ, which will give rise to unavoidable conservativeness.

4 Main Results

In this section, several examples are presented to confirm the performance of the IDyHF in comparison with two COEAs and methods from control theory and graph theory. The controlling regions are identified in a microscopic, and macroscopic way, respectively.

To show the advantages of the IDyHF is suitable to treat the controllability and identification of controlling regions of cortical networks, the IDyHF is compared with two evolutionary computation approaches the CMODE [36], and the DyHF [35]. Note that the CMODE and the DyHF have been recently developed and shown their advantages over some well-known COEAs [35], [36]. In addition, the IDyHF is also compared with other methods in the field of control theory and graph theory, which are listed in Section 2.4.

Hereafter, COEAs will be repeated 10 times independently for eliminating random discrepancy and terminated when COEAs algorithms attain $f_{e,\max}$. The parameter setting of the CMODE and the DyHF is according to [35], [36]. The parameter setting of the IDyHF follows [35], [40]. The maximum fitness evaluation $f_{e,\max}$ is set to $f_{e,\max} = \eta * D$ and $D = 2 * l$ is the dimension size. $\eta = 12{,}500$ is a predefined constant, which is popular in evolutionary computation area.

4.1 Comparisons of the IDyHF with the CMODE and the DyHF

In the following, the reliability of evolutionary computation methods is shown in terms of the mean value M and the best value B of 10 runs. In COEAs, the best solution among each running time is recorded. Since 10 times are used in this paper, there exist 10 solutions for each algorithm under different l. Using the recorded solutions, we calculate the best value B and the mean value M of 10 runs under different l. Both B and M of the solutions are of great significance for measuring the reliability of the algorithm; hence, we introduce as a combined measure

$$Q = \sqrt{B \times M}. \tag{8}$$

TABLE 1
Search Result Comparisons among Three Algorithms for Different l of Driver Nodes in Cortical Networks with a Network Size $N = 53$ and $r = 0.1$, See Fig. 1

		CMODE		DyHF		IDyHF	
		R	Σ	R	Σ	R	Σ
$l = 6$	M	36.1718	0	30.3335	0	**29.017**	0
	B	30.9949	0	28.5985	0	**28.4976**	0
	Q	33.4834	0	29.4533	0	**28.7561**	0
$l = 12$	M	22.1119	0	17.924	0	**16.0513**	0
	B	21.3298	0	15.7408	0	**15.5056**	0
	Q	21.7174	0	16.797	0	**15.7761**	0
$l = 18$	M	18.122	0	14.1315	0	**11.1256**	0
	B	15.9598	0	10.6539	0	**10.6253**	0
	Q	17.0066	0	12.2701	0	**10.8726**	0
$l = 24$	M	14.2728	0	11.0648	0	**8.706**	0
	B	12.923	0	7.9463	0	**7.3016**	0
	Q	13.5812	0	9.3768	0	**7.973**	0
$l = 30$	M	11.9477	0	8.4731	0	**7.0268**	0
	B	10.8937	0	6.9599	0	**6.2412**	0
	Q	11.4085	0	7.6793	0	**6.6224**	0
$l = 36$	M	10.8115	0	6.6889	0	**5.2418**	0
	B	9.9942	0	5.7471	0	**4.7158**	0
	Q	10.3948	0	6.2002	0	**4.9719**	0
$l = 42$	M	10.5946	0	8.1058	0	**4.7643**	0
	B	9.2208	0	4.7948	0	**3.9905**	0
	Q	9.8838	0	6.2343	0	**4.3603**	0
$l = 48$	M	8.8032	0	6.1074	0	**3.0804**	0
	B	8.6091	0	4.0594	0	**2.7845**	0
	Q	8.7056	0	4.9792	0	**2.9287**	0

The calculation of Q is given in (8). The best results among the three algorithms are shown in Bold fonts.

It is obvious that Q should be made as small as possible. B, M, and Q of the solutions are listed in Table 1. In addition, the degree of violation Σ is also presented. The number of driver nodes is increased from 6 to 48 with a step size 6. The best results among the three algorithms are presented in Bold fonts. From Table 1, it can be observed that Σ can achieve zero in three algorithms with different l, which indicates that all the three algorithms can find feasible solutions. Since all the results are feasible, we only focus on the objective value R. It can be easily seen that the IDyHF performs best among the three algorithms, especially the dimension size D is large. The DyHF performs a little better than the CMODE and worse than the IDyHF. The IDyHF is the most powerful algorithm among these algorithms, since it is equipped with the jDE. The mechanism will help the IDyHF to explore the search space with good accuracy and high convergence speed. In the following, we use the IDyHF to carry out all the following simulations.

4.2 Comparison of the IDyHF with the Schemes in Section 2.4

In this section, the IDyHF is compared with other schemes in Section 2.4. The control gains of these methods are considered to be identical in each dimension and tuned by a step size 0.1 gradually, except the IDyHF. Note that the methods 1-4 in Section 2.4 only focus on minimizing R and σ is neglected, except the IDyHF. The best solutions in 10 runs of the IDyHF under different l are used to yield the following results. In the following, we only consider the third case in Section 2.3, i.e., $r = 0$. $r = 0.1$ will produce similar results. The comparison of different r will be presented in Section 4.5.

From Fig. 2a, the IDyHF performs better than the other methods in terms of R in most of cases, and the IDyHF

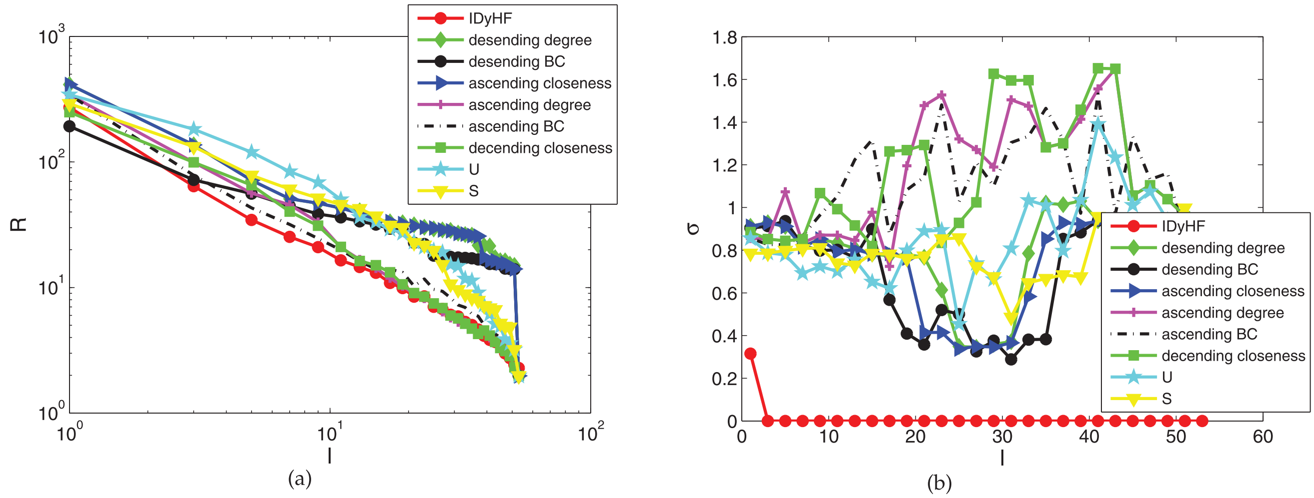


Fig. 2. Optimizing R with different schemes and $r = 0$ as a function of l. (a) Comparison of R with different schemes as a function of l. (b) Comparison of σ with different schemes as a function of l.

works worse than a few methods in terms of R in some cases, since all the other methods do not take the index of σ into account. It indicates that although a few methods are better than the IDyHF on R in some situations, they are not successful in having feasible solutions as a function of l, as seen from Fig. 2b. In order to satisfy the constraint as much as possible, the IDyHF has to encounter the situation that it loses the accuracy of R. From Fig. 2a, at the very beginning, the descending BC-based method performs best. When l increases, the descending degree-based strategy, the descending BC-based strategy, and the ascending closeness-based strategy are getting worse. In converse, the ascending degree-based strategy, the ascending BC-based strategy, and the descending closeness-based strategy are becoming better. Although U and S can characterize the synchronizability of networks suitably, they cannot be simply applied to study the controllability of networks, since the controllability problem considered here is a combinatorial optimization problem. And the optimal combinations of driver nodes is largely dependent on the number of driver nodes l.

From Fig. 2b, it can be found that the IDyHF always performs best than the other methods in terms of σ and when $l > 2$, σ always attains zero using the IDyHF. However, other methods cannot work well in terms of σ, since they are only concentrated on minimizing R without considering a constraint. From Figs. 2a and 2b, one can also find that the values of σ are smaller than R, especially when l is small. This is consistent with the findings in [26], [27], in which $\max_p\{\mu_p^i\}$ can be neglected in studying synchronizability of complex networks in most of the cases. It is also worthwhile to point out that, when $l \to N$, the effect of σ cannot be neglected, since R is getting smaller and the value of R is comparable to the value of σ. Therefore, σ should be taken into account for measuring the controllability of networks, especially when l is large.

4.3 Identification of Controlling Regions Using the IDyHF—A Microscopic Way

In Figs. 3a, 3b, and 3c, the mean values of the degree, BC and closeness information of driver nodes by various methods as a function of l are depicted, when $r = 0$. It can be found that the mean values of driver nodes selected by the IDyHF are intermediate, belonging to the range of mean values of the ascending and the descending degree-based schemes, which are less than the mean value of network degree. As l increases, the mean values of driver nodes selected by the IDyHF are getting larger and then converge to the mean value of network degree. Moreover, the standard deviation of the degree of the driver nodes selected by the IDyHF is relatively stable as a function of l. For a clearer presentation of which nodes should be selected as driver nodes, details of the degree and BC of driver nodes as a function of l are provided in Fig. 4. From Fig. 4, when l is small, no regions with a large degree or BC are chosen as driver nodes. With increasing l, the driver nodes are picked from the regions with a small degree and a large degree at the same time. This tendency is much clearer as l increases, as seen from Figs. 3a, 3b, 3c, and 4. In the latter stage, the driver nodes are distributed evenly in the networks. In summary, one should pick more nodes with a large degree as l increases, while the nodes with a small degree should also be chosen.

In the following, the dependence of R, λ_2^r, and λ_{N+1}^r on l is investigated. From Fig. 5, it is observed that $R(l) \propto l^{-\gamma}$, which is useful to predict R when knowing l a priori. In addition, one can see that in order to minimize R with a small l, λ_2^r should be enlarged as much as possible, while λ_{N+1}^r should be suppressed near a constant value. With a large l, both λ_{N+1}^r and λ_2^r grow exponentially and the growth of the amplitude of λ_2^r is larger than that of λ_{N+1}^r. The finding indicates that λ_2^r plays a more important role to enhance controllability than λ_{N+1}^r. Fig. 5 also displays that the shape of R is largely dependent on λ_2^r. When $l \to N$, $\lambda_2^r \approx \lambda_{N+1}^r$, which gives rise to $R \approx 1$. To summarize, in order to enhance controllability, one should put more efforts on enlarging λ_2^r than reducing λ_{N+1}^r.

4.4 Identification of Controlling Regions Using the IDyHF—A Macroscopic Way

By means of the IDyHF, the identification of controlling regions of the cortical networks with different r as a function of l is illustrated. Denote

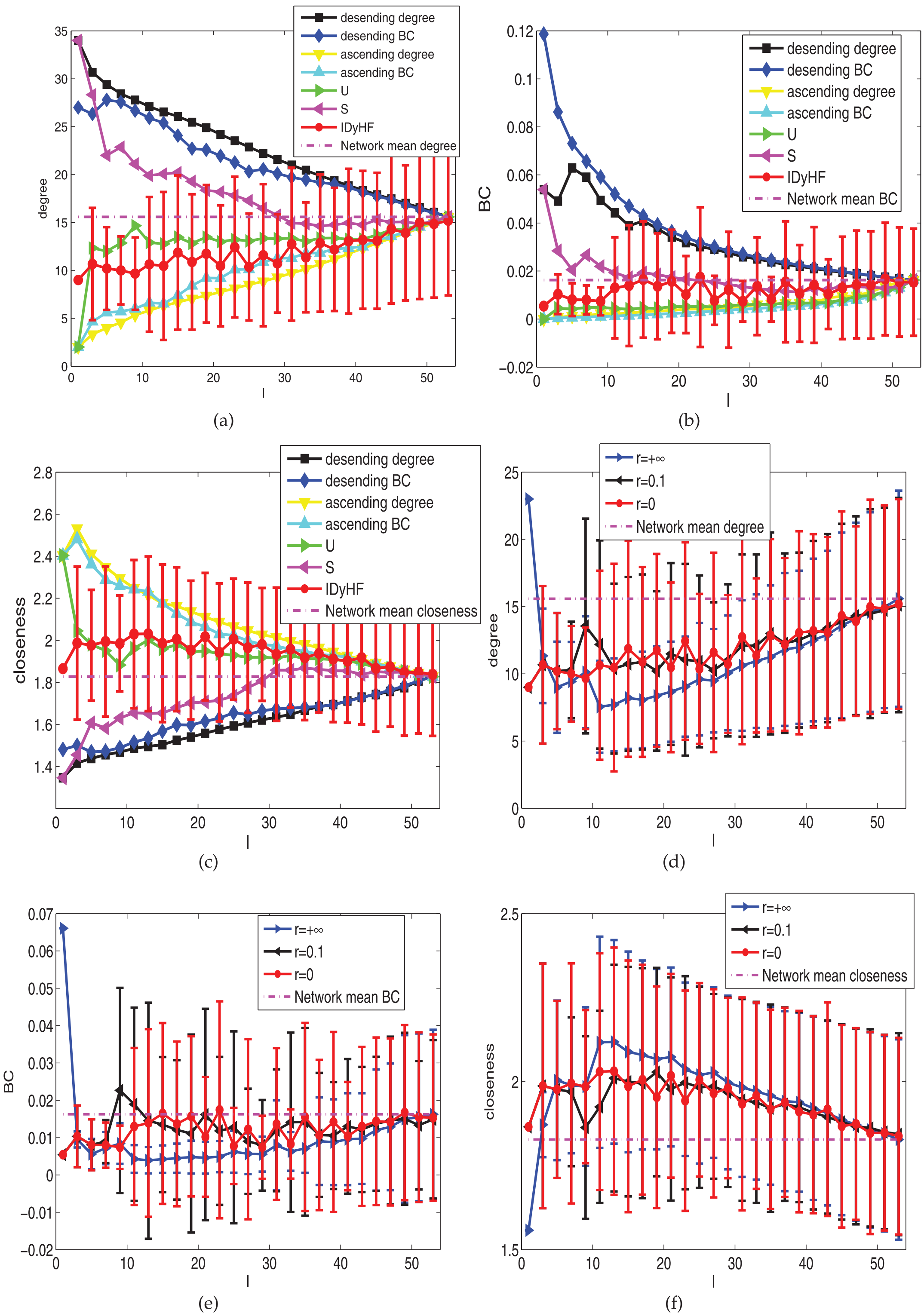


Fig. 3. The degree, BC, and closeness information of driver nodes with different schemes as a function of l. (a) The mean values of degree information of driver nodes with $r = 0$ as a function of l. (b) The mean values of BC information of driver nodes with $r = 0$ as a function of l. (c) The mean values of closeness information of driver nodes with $r = 0$ as a function of l. (d) The mean values of degree of driver nodes under different r as a function of l. (e) The mean values of BC of driver nodes under different r as a function of l. (f) The mean values of closeness of driver nodes under different r as a function of l.

$$\xi_i = \sum_{l=1}^{N} \delta_{\mathcal{M}}(i), \tag{9}$$

which qualifies the times of selection as driver nodes of each node in cortical networks. Apparently, the regions with a large ξ_i play a vital role in controlling cortical networks. After the control of a cortical network with an increase of l (step size 1), we sort ξ_i under $r = +\infty, r = +0.1$, and $r = 0$. As mentioned in Section 2.3, when $r = +\infty$, the constrained optimization problem is reduced into a single

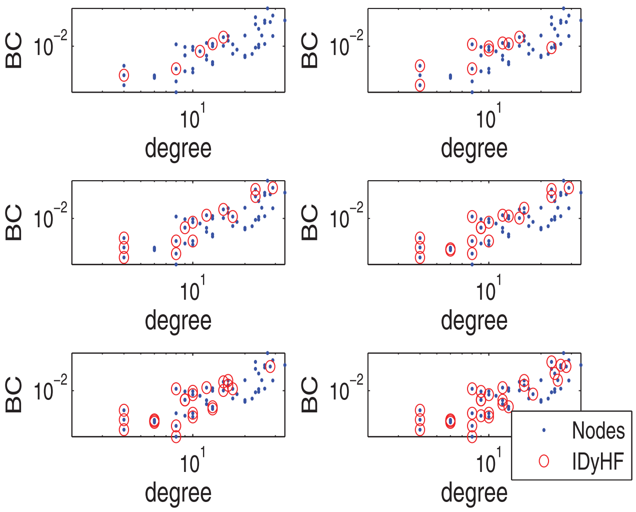


Fig. 4. Driver nodes in terms of degree and BC as a function of $l(l = 5, 10, 15, 20, 25, 30)$, when $r = 0$.

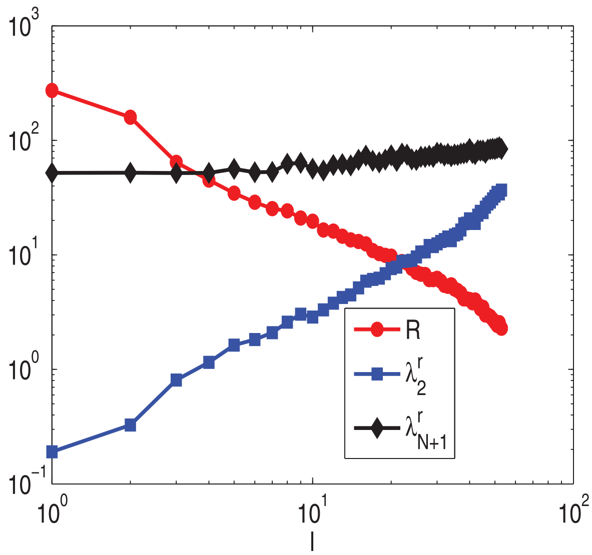


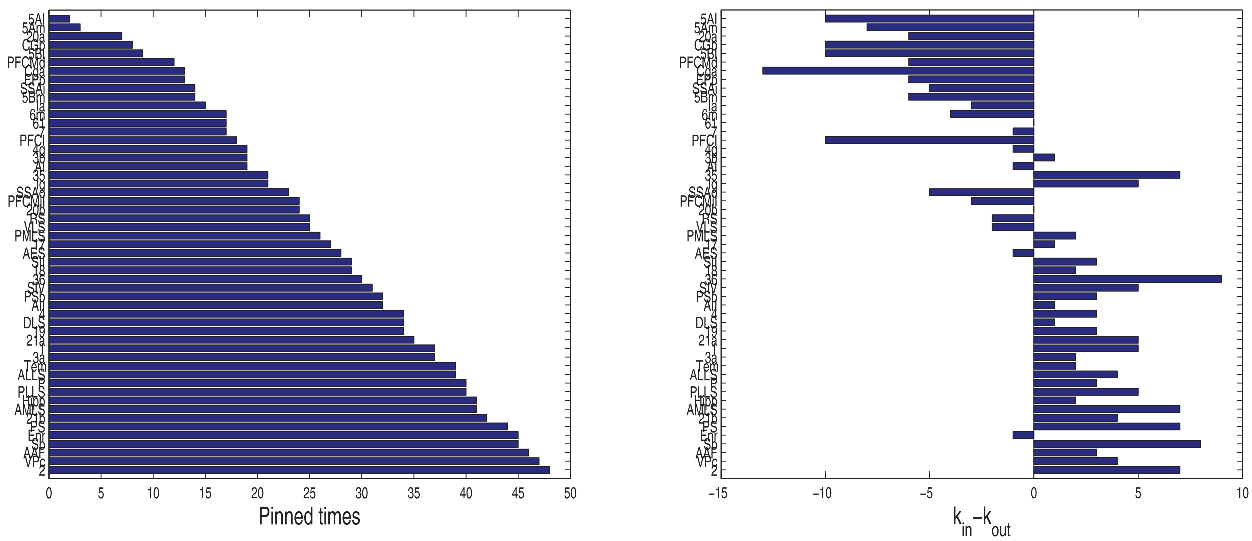
Fig. 5. R, λ_2^r, and λ_{N+1}^r as a function of l, when $r = 0$.

objective problem without constraints. Therefore, there is no need to use COEAs to handle the case of $r = +\infty$. Here, we use a recently developed evolutionary algorithm, i.e., adaptive deferential evolution (JADE) [42], for solving this single objective optimization problem. When $r = 0.1$ and $r = 0$, the COPs are dealt with the IDyHF. The results are shown in Fig. 6a in an ascending way and Table 2 in a descending way. It can be checked from Table 2 that there exists some minor differences for the pinned times of each node in the three cases. It can be found that the regions with a large ξ_i can be viewed as controlling regions and are widely spread in four communities. Therefore, the cortical networks can be controlled with high efficiency. Table 2 infers that the regions such as VPc and 2 are very important to control the cortical networks. Meanwhile, the regions such as 5AI can be neglected to serve as driver nodes. From the controlling regions in all three cases, it can be found that the controlling nodes are different from the usual hubs detected by degree, BC, and motif-based methods in [17], in which the hubs are usually selected from the nodes with a large degree.

In order to illustrate which potential rules govern the selection of controlling nodes, the degree differences $\triangle k = k_{in} - k_{out}$ of each region in the cortical network are shown in Fig. 6b. From Fig. 6b, it can be observed that the controlling regions are selected from the nodes with a large k_{in} and a small k_{out}.

4.5 Comparisons of the IDyHF with Different r

In this section, enhancing controllability of cortical networks under different constraints r is investigated using EAs. The comparisons of R and σ are presented in Figs. 7a and 7b. From Fig. 7a, JADE performs better than the IDyHF, since JADE neglects the effect of σ. However, the differences between the lines are close to each other, which also verifies the performance of the IDyHF. Although the IDyHF



(a) (b)

Fig. 6. ξ_i and $\triangle k$ is sorted in ascending way according to ξ_i when $r = 0$. (a) ξ_i of each node in cortical networks of cat. (b) $\triangle k$ of each node in cortical networks of cat.

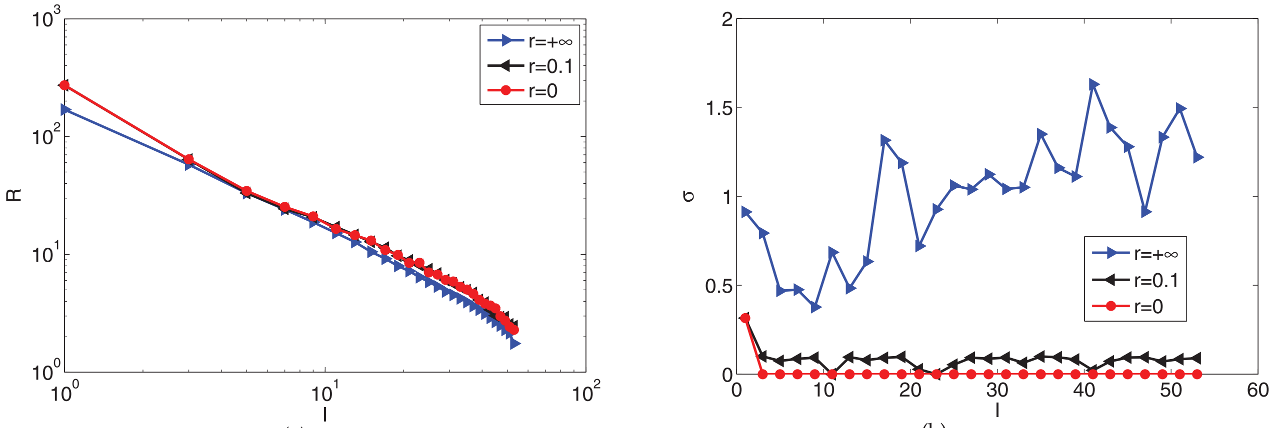
TABLE 2
Controlling Times and Its Community of Each Node Belonging to when Optimizing R under Different r

$r=+\infty$			$r=0.1$			$r=0$		
Node Name	ξ	Community	Node Name	ξ	Community	Node Name	ξ	Community
VPc	50	Auditory	VPc	51	Auditory	2	48	Somato-motor
2	50	Somato-motor	2	48	Somato-motor	VPc	47	Auditory
AMLS	48	Visual	AMLS	45	Visual	AAF	46	Auditory
21b	48	Visual	21b	45	Visual	Sb	45	Frontolimbic
PS	48	Visual	Sb	45	Frontolimbic	Enr	45	Frontolimbic
21a	47	Visual	AAF	43	Auditory	PS	44	Visual
ALLS	45	Visual	PS	42	Visual	21b	42	Visual
Sb	45	Frontolimbic	ALLS	41	Visual	AMLS	41	Visual
Hipp	45	Frontolimbic	P	41	Auditory	Hipp	41	Frontolimbic
AAF	44	Auditory	Enr	41	Frontolimbic	PLLS	40	Visual
Tem	42	Auditory	Hipp	41	Frontolimbic	P	40	Auditory
P	40	Auditory	21a	38	Visual	ALLS	39	Visual
SIV	40	Somato-motor	PLLS	37	Visual	Tem	39	Auditory
3a	39	Somato-motor	Tem	37	Auditory	3a	37	Somato-motor
1	39	Somato-motor	1	37	Somato-motor	1	37	Somato-motor
DLS	38	Visual	DLS	36	Visual	21a	35	Visual
SII	38	Somato-motor	SIV	36	Somato-motor	19	34	Visual
PSb	38	Frontolimbic	4	36	Somato-motor	DLS	34	Visual
4	36	Somato-motor	3a	35	Somato-motor	4	34	Somato-motor
PLLS	33	Visual	36	35	Frontolimbic	AII	32	Auditory
AII	33	Auditory	PSb	34	Frontolimbic	PSb	32	Frontolimbic
RS	32	Frontolimbic	19	32	Visual	SIV	31	Somato-motor
PMLS	30	Visual	SII	29	Somato-motor	36	30	Frontolimbic
20b	30	Visual	RS	28	Frontolimbic	18	29	Visual
VLS	29	Visual	17	26	Visual	SII	29	Somato-motor
PFCMiI	29	Frontolimbic	AII	26	Auditory	AES	28	Visual
Enr	29	Frontolimbic	PFCMiI	26	Frontolimbic	17	27	Visual
19	27	Visual	20b	25	Visual	PMLS	26	Visual
3b	27	Somato-motor	18	24	Visual	VLS	25	Visual
17	25	Visual	VLS	24	Visual	RS	25	Frontolimbic
SSAo	25	Somato-motor	Ig	24	Frontolimbic	20b	24	Visual
18	22	Visual	PMLS	23	Visual	PFCMiI	24	Frontolimbic
4g	21	Somato-motor	AES	23	Visual	SSAo	23	Somato-motor
AI	20	Auditory	SSAo	22	Somato-motor	Ig	21	Frontolimbic
PFCI	20	Frontolimbic	3b	21	Somato-motor	35	21	Frontolimbic
36	20	Frontolimbic	4g	21	Somato-motor	AI	19	Auditory
61	19	Somato-motor	AI	20	Auditory	3b	19	Somato-motor
7	18	Visual	PFCI	20	Frontolimbic	4g	19	Somato-motor
SSAi	16	Somato-motor	35	18	Frontolimbic	PFCI	18	Frontolimbic
5Bm	14	Somato-motor	7	16	Visual	7	17	Visual
Ig	14	Frontolimbic	EPp	16	Auditory	61	17	Somato-motor
6m	12	Somato-motor	61	16	Somato-motor	6m	17	Somato-motor
PFCMd	11	Frontolimbic	5Bm	16	Somato-motor	Ia	15	Frontolimbic
AES	10	Visual	SSAi	16	Somato-motor	5Bm	14	Somato-motor
Ia	9	Frontolimbic	Cga	13	Frontolimbic	SSAi	14	Somato-motor
EPp	8	Auditory	6m	12	Somato-motor	EPp	13	Auditory
5BI	7	Somato-motor	20a	11	Visual	Cga	13	Frontolimbic
Cga	6	Frontolimbic	Ia	10	Frontolimbic	PFCMd	12	Frontolimbic
35	5	Frontolimbic	5BI	9	Somato-motor	5BI	9	Somato-motor
5Am	4	Somato-motor	CGp	7	Frontolimbic	CGp	8	Frontolimbic
20a	3	Visual	5Am	6	Somato-motor	20a	7	Visual
CGp	2	Frontolimbic	PFCMd	6	Frontolimbic	5Am	3	Somato-motor
5AI	1	Somato-motor	5AI	1	Somato-motor	5AI	2	Somato-motor

ξ can be seen from (9).

performs worse than JADE in terms of R, the IDyHF works much better than JADE in terms of σ, as seen from Fig. 7b. With increasing l, σ obtained by JADE gradually turns larger, which implies that only minimizing R will give rise to an increase of σ, when l increases. In addition, the line of R when $r = 0.1$ is very close to that of R when $r = 0$. Meanwhile, it is also clear to see that there are differences between the line of σ when $r = 0.1$ and the line of σ when $r = 0$. It is also worth mentioning that, when l is very small, e.g., $l = 1$, the IDyHF cannot find feasible solutions and, therefore, it produces solutions with the least degree of constraint violation in the population. Once l exceeds a threshold, the IDyHF is able to find feasible solutions and, therefore, the feasible solutions with the least fitness will be regarded as the best solution and presented. In summary, the IDyHF can enhance the controllability of cortical networks when l increases, while keeping the solutions in a feasible space as much as possible.

In the following, we also show the differences between the driver nodes under different r. It is clear to see that the mean values of degree, BC, and closeness under $r = 0.1$ and $r = 0$ are close to each other, as seen from Figs. 3d, 3e, and 3f. However, there are apparent distinctions between the mean values of degree, BC and closeness of $r = 0$ and those of $r = +\infty$. In the case of $r = +\infty$ and when l is small, one should choose the regions with a large degree,



(a) (b)

Fig. 7. Optimizing R by the IDyHF with different r as a function of l. (a) Comparison of R with different r as a function of l. (b) Comparison of σ with different r as a function of l.

and then abruptly change to the regions with a small degree with increasing l. There exists a clear transition from the regions with a large degree to the regions with a small degree when $r = +\infty$, as seen from Fig. 3d. It is quite different from the case of $r = 0$ and $r = 0.1$, in which one should always pick the regions with a small degree below the mean degree of the cortical network. With increasing l, the tendency of mean degrees of driver nodes increase accordingly. In addition, it can be observed that one should select the regions with a smaller degree in the case of $r = +\infty$ than the cases of $r = 0.1$ and $r = 0$, when l exceeds a threshold. This reveals that in order to only minimize R with $r = +\infty$, it is necessary to choose the regions with a large degree at first, then select the regions with rather small degrees. However, when minimizing R under a constraint r, it is efficient to pick the regions with a small degree as well as the regions with a large degree σ, as seen from Figs. 3d and 4. That is, the existence of constraint r will result in a different control rule.

5 Conclusion

How to control a complex system is deemed to be crucial in understanding natural systems. In this paper, the problem of detection of controlling regions of a realistic cortical network is first transformed into a constrained optimization problem, which combines the two measures R and σ into one unified framework. Then, in order to deal with this constrained optimization problem, an improved dynamic hybrid framework is developed based on the adaptive differential evolution and the DyHF. The IDyHF does not only minimize R, but also seeks to find feasible solutions to satisfy the constraint of σ. Our experiments have verified clear advantages of the IDyHF over the DyHF and the CMODE. In addition, the comparisons of some well-known methods, such as degree, BC, closeness, and node importance-based methods are presented, which illustrates the effectiveness of the IDyHF. By using the IDyHF, the controlling regions are detected in a microscopic and macroscopic way. The dependence of R, λ_2^r, and λ_{N+1}^r on l is illustrated. More importantly, the relevance of selection of driver nodes to the number of driver nodes l and constraint of r is discussed. The impacts of r on the controllability and selecting driver nodes are also discussed, as seen in Figs. 3d, 3e, 3f, 7a, and 7b. We have found that the controlling regions depend on the number of driver nodes, especially the driver regions with a large in-degree and a small out-degree should be considered to be driver nodes with priority. It is found that controlling σ is becoming more important in controlling a cortical network with increasing l. Additionally, with increasing of l, the regions with a large degree is becoming more important. All the findings imply that the controlling regions are different from the usual hubs in [17], [23], in which the hubs in networks are usually selected from the regions with a large degree [17], [19].

At the end, it is valuable to provide some future discussions. First, one can develop more promising evolutionary computation methods to handle the controllability of cortical networks. Second, COEAs can be applied to other realistic natural complex systems to unveil the potential controlling rules. Third, it is of great importance to develop more intelligent techniques such as genetic algorithms or particle swarm optimization (PSO) [43], [44] for detection of efficient controlling regions. This achievement would require further explorations in neuroscience, in dynamical complex systems, in EAs as well as in control technology society. Fourth, the limitation of this paper is to linearize (2) and how to present global controllability is a future research topic [45], [46]. Finally, it is also interesting to extend our results to study the observability and controllability of cortical networks or multiagent systems [47].

Acknowledgments

This research is supported by 973 Project (2009CB320600), the National Natural Science Foundation of China (60825303, 61021002, 61203235), and the Key Laboratory of Integrated Automation for the Process Industry (Northeastern University), SUMO (EU), IRTG 1740 (DFG), grant of the German Research Foundation (DFG) in the Research Group FOR 868 Computational Modeling of Behavioral, Cognitive, and Neural Dynamics and the Alexander von Humboldt Foundation of Germany.

REFERENCES

[1] A.L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science,* vol. 286, pp. 509-512, 1999.

[2] M. Piraveenan, M. Prokopenko, and A. Zomaya, "Assortative Mixing in Directed Biological Networks," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 9, no. 1, pp. 66-78, Jan./Feb. 2012.

[3] Z. Wang, X. Liu, Y. Liu, J. Liang, and V. Vinciotti, "An Extended Kalman Filtering Approach to Modeling Nonlinear Dynamic Gene Regulatory Networks via Short Gene Expression Time Series," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 6, no. 3, pp. 410-419, July-Sept. 2009.

[4] T. Perkins and J. Hallett, "A Trade-Off between Sample Complexity and Computational Complexity in Learning Boolean Networks from Time-Series Data," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 7, no. 1, pp. 118-125, Jan.-Mar. 2010.

[5] S. Mitra, R. Das, and Y. Hayashi, "Genetic Networks and Soft Computing," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 8, no. 1, pp. 94-107, Jan./Feb. 2011.

[6] N. Noman and H. Iba, "Inferring Gene Regulatory Networks Using Differential Evolution with Local Search Heuristics," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 4, no. 4, pp. 634-647, Oct.-Dec. 2007.

[7] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences.* Cambridge Univ. Press, 2001.

[8] A. Arenas, A. Guilera, J. Kurths, Y. Moreno, and C. Zhou, "Synchronization in Complex Networks," *Physics Reports,* vol. 469, pp. 93-153, 2008.

[9] A. Engel, P. Fries, and W. Singer, "Rapid Feature Selective Neuronal Synchronization through Correlated Latency Shifting," *Nature Neuroscience,* vol. 2, pp. 704-716, 2001.

[10] P. Uhlhaas and W. Singer, "Neural Synchrony in Brain Disorders: Relevance for Cognitive Dysfunctions and Pathophysiology," *Neuron,* vol. 52, pp. 155-168, 2006.

[11] S. Schinkel, G. Zamora, O. Dimigen, W. Sommer, and J. Kurths, "Functional Network Analysis Reveals Differences in the Semantic Priming Task," *J. Neuroscience Methods,* vol. 197, pp. 333-339, 2011.

[12] N. Kopell and B. Ermentrout, "Chemical and Electrical Synapses Perform Complementary Roles in the Synchronization of Interneuronal Networks," *Proc. Nat'l Academy of Sciences USA,* vol. 101, pp. 15482-15487, 2004.

[13] E. Bullmore and O. Sporns, "Complex Brain Networks: Graph Theoretical Analysis of Structural and Functional Systems," *Nature Rev. Neuroscience,* vol. 10, pp. 186-198, 2009.

[14] M. Rubinov and O. Sporns, "Complex Network Measures of Brain Connectivity: Uses and Interpretations," *NeuroImage,* vol. 52, pp. 1059-1069, 2010.

[15] C. Zhou, L. Zemanová, G. Zamora, C. Hilgetag, and J. Kurths, "Hierarchical Organization Unveiled by Functional Connectivity in Complex Brain Networks," *Physical Rev. Letters,* vol. 97, p. 238103, 2006.

[16] M. Muller, C. Hilgetag, and M. Hutt, "Organization of Excitable Dynamics in Hierarchical Biological Networks," *PLoS Computational Biology,* vol. 4, p. e1000190, 2008.

[17] O. Sporns, C. Honey, and R. Kotter, "Identification and Classification of Hubs in Brain Networks," *Plos One,* vol. 10, p. e1049, 2007.

[18] R. Morgan and I. Soltesz, "Nonrandom Connectivity of the Epileptic Dentate Gyrus Predicts a Major Role for Neuronal Hubs in Seizures," *Proc. Nat'l Academy of Sciences USA,* vol. 105, pp. 6179-6184, 2008.

[19] G. Zamora, C. Zhou, and J. Kurths, "Cortical Hubs form a Module for Multisensory Integration on Top of the Hierarchy of Cortical Networks," *Frontiers in Neuroinformatics,* vol. 4, pp. 1-12, 2010.

[20] J. Gomez, G. Zamora, Y. Moreno, and A. Arenas, "From Modular to Centralized Organization of Synchronization in Functional Areas of the Cat Cerebral Cortex," *Plos one,* vol. 5, p. e12313, 2010.

[21] F. Sorrentino, M. Bernardo, F. Garofalo, and G. Chen, "Controllability of Complex Networks via Pinning," *Physical Rev. E,* vol. 75, p. 046103, 2007.

[22] Y. Liu, J. Slotine, and A. Barabasi, "Controllability of Complex Networks," *Nature,* vol. 473, pp. 167-173, 2011.

[23] M. Egerstedt, "Degrees of Control," *Nature,* vol. 473, pp. 158-159, 2011.

[24] X. Wang and G. Chen, "Synchronization in Scale-Free Dynamical Networks: Robustness and Fragility," *IEEE Trans. Circuits Systems I: Regular Papers,* vol. 49, no. 1, pp. 54-62, Jan. 2002.

[25] W. Ren and R. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control, Communications and Control Engineering Series.* Springer-Verlag, 2008.

[26] D.U. Hwang, M. Chavez, A. Amann, and S. Boccaletti, "Synchronization in Complex Networks with Age Ordering," *Physical Rev. Letter,* vol. 94, p. 138701, 2005.

[27] S. Son, B. Kim, H. Hong, and H. Jeong, "Dynamics and Directionality in Complex Networks," *Physics Rev. Letters,* vol. 103, p. 228702, 2009.

[28] X. Qian and E. Dougherty, "Intervention in Gene Regulatory Networks via Phenotypically Constrained Control Policies Based on Long-Run Behavior," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 9, no. 1, pp. 123-136, Jan./Feb. 2012.

[29] W. Yu, G. Chen, and J. Lu, "On Pinning Synchronization of Complex Dynamical Networks," *Automatica,* vol. 45, pp. 429-435, 2009.

[30] Y. Tang, Z.D. Wang, and J. Fang, "Controller Design for Synchronization of an Array of Delayed Neural Networks Using a Controllable Probabilistic Pso," *Information Sciences,* vol. 181, pp. 4715-4732, 2011.

[31] Y. Tang, Z.D. Wang, W.K. Wong, J. Kurths, and J. Fang, "Multiobjective Synchronization of Coupled Systems," *Chaos,* vol. 21, pp. 025114-1-025114-12, 2011.

[32] Z. Michalewicz and M. Schoenauer, "Evolutionary Algorithm for Constrained Parameter Optimization Problems," *Evolutionary Computation,* vol. 4, pp. 1-32, 1996.

[33] C. Coello, "Theoretical and Numerical Constraint-Handling Techniques Used with Evolutionary Algorithms: A Survey of the State of the Art," *Computer Methods in Applied Mechanics and Eng.,* vol. 191, pp. 1245-287, 2002.

[34] Z. Cai and Y. Wang, "A Multiobjective Optimization-Based Evolutionary Algorithm for Constrained Optimization," *IEEE Trans. Evolutionary Computation,* vol. 10, no. 6, pp. 658-75, Dec. 2006.

[35] Y. Wang and Z. Cai, "A Dynamic Hybrid Framework for Constrained Evolutionary Optimization," *IEEE Trans. Systems, Man, and Cybernetics - Part B: Cybernetics,* vol. 42, no. 1, pp. 203-217, Feb. 2012.

[36] Y. Wang and Z. Cai, "Combining Multiobjective Optimization with Differential Evolution to Solve Constrained Optimization Problems," *IEEE Trans. Evolutionary Computation,* vol. 16, no. 1, pp. 117-134, Feb. 2012.

[37] C. Hilgetag, G. Burns, M. O'eill, J. Scannell, and M. Young, "Anatomical Connectivity Defines the Organization of Clusters of Cortical Areas in the Macaque Monkey and the Cat," *Philosophical Trans. R Soc. London B,* vol. 355, pp. 91-110, 2000.

[38] J. Zhou, J. Lu, and J. Lu, "Pinning Adaptive Synchronization of a General Complex Dynamical Network," *Automatica,* vol. 44, pp. 996-1003, 2008.

[39] J. Restrepo, E. Ott, and B. Hunt, "Characterizing the Dynamical Importance of Network Nodes and Links," *Physical Rev. Letters,* vol. 97, p. 094102, 2006.

[40] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Trans. Evolutionary Computation,* vol. 10, no. 6, pp. 646-657, Dec. 2006.

[41] F. Sorrentino, "Effects of the Network Structural Properties on Its Controllability," *Chaos,* vol. 17, p. 033101, 2007.

[42] J.Q. Zhang and A.C. Sanderson, "Jade: Adaptive Differential Evolution with Optional External Archive," *IEEE Trans. Evolutionary Computation,* vol. 13, no. 5, pp. 945-958, Oct. 2009.

[43] A. Esmin and G. Lambert-Torres, "Application of Particle Swarm Optimization to Optimal Power Systems," *Int'l J. Innovative Computing Information and Control,* vol. 8, pp. 1705-1717, 2012.

[44] J. Pahasa and I. Ngamroo, "Pso Based Kernel Principal Component Analysis and Multi-Class Support Vector Machine for Power Quality Problem Classification," *Int'l J. Innovative Computing Information and Control,* vol. 8, pp. 1523-1540, 2012.

[45] J.Q. Lu, D.W.C. Ho, and L.G. Wu, "Exponential Stabilization in Switched Stochastic Dynamical Networks," *Nonlinearity,* vol. 22, pp. 889-911, 2009.

[46] J.Q. Lu, D.W.C. Ho, and J.D. Cao, "A Unified Synchronization Criterion for Impulsive Dynamical Networks," *Automatica,* vol. 46, pp. 1215-1221, 2010.
[47] Y. Tang, H. Gao, W. Zou, and J. Kurths, "Distributed Synchronization in Networks of Agent Systems with Nonlinearities and Random Switchings," *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics,* accepted in 2012, doi: 10.1109/TSMCB.2012.2207718.

Yang Tang (M'11) received the BS and PhD degrees in electrical engineering from Donghua University, Shanghai, China, in 2006 and 2011, respectively. From December 2008 to December 2010, he was a research associate in The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China. He has been an Alexander von Humboldt research fellow at Humboldt University, Berlin, Germany, since 2011. He held a visiting fellow position at Brunel University, United Kingdom from May 2012 to June 2012. He has published more than 30 refereed international journals. His main research interests are synchronization/consensus, networked control system, evolutionary computation, bioinformatics and their applications. He is a very active reviewer for many international journals. He is a member of the IEEE.

Zidong Wang (SM'03) received the BSc degree in mathematics in 1986 from Suzhou University, China, the MSc degree in applied mathematics in 1990, and the PhD degree in electrical and computer engineering in 1994, both from Nanjing University of Science and Technology, Nanjing, China. He is now a professor of dynamical systems and computing at Brunel University, United Kingdom. His research interests include dynamical systems, signal processing, bioinformatics, control theory and applications. He has published more than 200 papers in refereed international journals. He is currently serving as an associate editor for 12 international journals including *IEEE Transactions on Automatic Control, IEEE Transactions on Neural Networks*, *IEEE Transactions on Signal Processing*, *IEEE Transactions on Systems, Man, and Cybernetics - Part C*, and *IEEE Transactions on Control Systems Technology*. He is a senior member of the IEEE.

Huijun Gao (SM'09) received the PhD degree in control science and engineering from the Harbin Institute of Technology, China, in 2005. He was a research associate with the Department of Mechanical Engineering, The University of Hong Kong, from November 2003 to August 2004. From October 2005 to October 2007, he carried out his postdoctoral research with the Department of Electrical and Computer Engineering, University of Alberta, Canada. Since November 2004, he has been with the Harbin Institute of Technology, where he is currently a professor and the director of the Research Institute of Intelligent Control and Systems. His research interests include network-based control, robust control/filter theory, time-delay systems and their engineering applications. He is an associate editor for *Automatica*, *IEEE Transactions on Industrial Electronics*, *IEEE Transactions on Systems Man and Cybernetics Part B: Cybernetics*, *IEEE Transactions on Fuzzy Systems*, *IEEE Transactions on Circuits and Systems - I*, *IEEE Transactions on Control Systems Technology*, etc. He is a senior member of the IEEE.

Stephen Swift received the BSc degree in mathematics and computing from the University of Kent, Canterbury, United Kingdom, the MSc degree in artificial intelligence from Cranfield University, United Kingdom, and the PhD degree in intelligent data analysis from Birkbeck College, University of London, United Kingdom. He is currently a research lecturer with the School of Information Systems, Computing, and Mathematics at Brunel University, London. He has also spent four years in industry as a web designer, programmer, and technical architect, and has four years postdoctoral research experience. His research interests include multivariate time series analysis, heuristic search, data clustering, and evolutionary computation. He has applied his research to a number of real-world areas including software engineering, bioinformatics, and health care.

Jürgen Kurths studied mathematics at the University of Rostock and received the PhD degree in 1983 from the GDR Academy of Sciences. He was a full professor in the University of Potsdam from 1994 to 2008 and has been a professor of nonlinear dynamics at the Humboldt University, Berlin, and the chair of the research domain Transdisciplinary Concepts of the Potsdam Institute for Climate Impact Research since 2008 and a sixth century chair of Aberdeen University (UK) since 2009. He is a fellow of the American Physical Society. He received the Alexander von Humboldt research award from CSIR (India) in 2005 and a Honory Doctorate in 2008 from the Lobachevsky University Nizhny Novgorod and one in 2012 from the State University Saratov. He has become a member of the Academia Europaea in 2010 and of the Macedonian Academy of Sciences and Arts in 2012. His main research interests are synchronization, complex networks, time series analysis and their applications. He has published more than 480 papers which are cited more than 18,000 times (H-factor: 56). He is an editor for *PLoS ONE, Philosophical Transaction of The Royal Society A, CHAOS*, etc.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.