

# SCIENTIFIC REPORTS



OPEN

## A new method to reduce the number of time delays in a network

Alexandre Wagemakers<sup>1</sup> & Miguel A. F. Sanjuán<sup>1,2,3</sup>

**Time delays may cause dramatic changes to the dynamics of interacting oscillators. Coupled networks of interacting dynamical systems can have unexpected behaviours when the signal between the vertices are time delayed. It has been shown for a very general class of systems that the time delays can be rearranged as long as the total time delay over the constitutive loops of the network is conserved. This fact allows to reduce the number of time delays of the problem without loss of information. There is a theoretical lower bound for this number that can be numerically improved if the time delays are commensurable. Here we propose a formulation of the problem and a numerical method to even further reduce the number of time delays in a network.**

Transmission delays are intrinsic to any process that exchanges information. While in many applications these time delays are small enough to be neglected, in other cases they have a critical influence on the dynamics. Examples of connected dynamical systems appear frequently in physics, engineering and natural sciences<sup>1–3</sup>. A problem of interest in the dynamical systems community is the synchronization of coupled oscillators. Some progress has been made to understand the synchronization of oscillators when an identical time delay is present on every connection of a coupled system<sup>4,5</sup>. Otherwise, the problem of synchronization with nonidentical time delays spreaded across a network is still difficult to address, yet there are succesfull attempts of analysis using a mean field approach of the dynamical system<sup>6,7</sup>.

In an effort to simplify the study of such networks, a new method called componentwise time-shift transformation<sup>8</sup> has been developed in order to transform the time delays of the network. This transformation allows to change the time delays on the network following some precise rules without affecting the dynamics of the system<sup>8,9</sup>. The purpose of the transformation is to set  $n - 1$  time delays to zero, being  $n$  the number of vertices of the network. A brief summary is described in the next section.

Here we take on this idea and propose a new formulation of this transformation that allows to use common optimization algorithms to reduce the number of time delays on a network. Our results show that on networks with different topologies, the number of time delays that can be reduced to zero is larger than  $n - 1$ . We claim that the number  $n_z$  of zero time delays can be larger than the lower bound  $n_z = n - 1$  in the case of a set of commensurable time delays. Moreover, within our framework we can devise other optimization strategies to find a suitable configuration of time delays given a specific need.

The technique described in ref. 9 hinges on the observation that we can change the time delays in a network with a single cycle of length  $n$  without altering the dynamics as long as the sum of the time delays around the cycle is conserved. The authors extended the reasoning over arbitrary networks by establishing the constitutive constraints between the time delays in a network. In this article, we reformulate the fundamental property of conservation of the time delays over a loop using algebraic graph theory. The problem of finding minimal time delays on the network is next transformed into a linear optimization problem. We show that the simplex optimization algorithm<sup>10</sup> provides a solution for the transformed time delays where at least  $n - 1$  time delays are set to zero.

### Componentwise time-shift transformation

We consider a graph  $G$  with a collection of  $l$  oriented edges  $e_i$  and  $n$  vertices  $v_i$ . At each vertex we have a very general dynamical system in the form of a system of  $n$  delay differential equations

<sup>1</sup>Nonlinear Dynamics, Chaos and Complex Systems Group, Departamento de Física, Universidad Rey Juan Carlos, Móstoles, Madrid, Tulipán s/n, 28933, Spain. <sup>2</sup>Department of Applied Informatics, Kaunas University of Technology, Studentu 50-415, Kaunas, LT-51368, Lithuania. <sup>3</sup>Institute for Physical Science and Technology, University of Maryland, College Park, Maryland, 20742, USA. Correspondence and requests for materials should be addressed to A.W. (email: [alexandre.wagemakers@urjc.es](mailto:alexandre.wagemakers@urjc.es))

with  $i = 1, \dots, n$  and  $S_i$  is the set of indices  $k$  such that the edges  $e_k$  connect the vertex  $j$  to the vertex  $i$ . We assume a discrete time delay  $\tau_k$  on the edge  $e_k$ .

The previous system in Eq. (1) can be transformed with a redefinition of the time delays  $\tau_k$  without changing the dynamical properties of the system. We set

$$\frac{dy_i}{dt} = f_i(y_i, y_j(t - \tilde{\tau}_k)_{k \in S_i}), \quad (2)$$

with the following change of variables

$$y_i(t) = x_i(t - \eta_i) \quad (3)$$

$$\tilde{\tau}_k = \tau_k + \eta_{s(k)} - \eta_{t(k)}, \quad (4)$$

being  $\eta_i$  constants and  $s(k)$  is the source vertex of the edge  $k$  and  $t(k)$  the target vertex of the same edge. The authors in ref. 9 noticed that the *algebraic sum* of the time delay around any cycle of the network is constant for every choice of the time-shifts  $\eta_i$ . The term *algebraic sum* means here that, given an oriented cycle in the graph, the time delay associated to the edges on the cycle with the same orientation should be summed up and the time delays on edges with opposite direction subtracted.

Now the problem is to find the time-shifts  $\eta_i$  associated to each vertex for a desired configuration of time delays  $\tilde{\tau}_k$ .

### Graph characteristics

The topology of the graph can be described in terms of algebraic structures associated to the topology<sup>11</sup>. We first give some definitions to set the context of the work. We define  $G(V, E, A)$  as a directed and connected graph, where  $V$  is a set of  $n$  vertices and  $E$  a set of  $l$  directed edges. Before going into the details, we need to number the edges from 1 to  $l$  and we note  $\tau_k$  as the time delay of the edge  $e_k$ .

To represent the connectivity, we define the incidence matrix  $A \in \mathbb{Z}^{n \times l}$  that relates the vertices to the edges. The elements  $a_{jk}$  of the matrix  $A$  are expressed in the following way:  $a_{jk} = 1$  if the edge  $e_k$  points towards the vertex  $j$  and  $a_{jk} = -1$  if the edge points outwards. All other entries are zero. All the information about the connections of the network is contained in this matrix. It is also possible to develop the method for multiple edges connecting two vertices. We restrain here the case to a maximum of two edges to represent a bidirectional connection.

If the graph is connected, or weakly connected, we can define an acyclic subgraph called spanning tree that connects all the vertices and have exactly  $n - 1$  edges. This structure is important for the decomposition of the graph  $G$  into elementary cycles. Given a spanning tree  $T$  and an edge  $e$  not in  $T$ , there is a unique cycle in  $G$  containing only edges of  $T$  and  $e$ . As a consequence, we can decompose the network into  $c = l - (n - 1)$  independent cycles. This decomposition can be expressed as a matrix  $B \in \mathbb{Z}^{(l-(n-1)) \times l}$  that expresses the cycle space associated to the tree  $T$ . First, we set the orientation of the cycle as the direction of the edge *not* in  $T$ . Being  $b_{jk}$  an element of  $B$ , we set  $b_{jk} = 1$  if the edge  $k$  is in the cycle  $j$  with the same direction, and  $b_{jk} = -1$  if the orientations are opposite. All other numbers are zero. This matrix  $B$  is of special interest for our study since the sum of time delays around each cycle is given by a simple matrix multiplication

$$B\tau = \sigma, \quad (5)$$

where  $\tau = (\tau_1 \dots \tau_l)^T$  is the column vector of the time delay  $k$  associated to the edge  $e_k$  and  $v^T$  denotes the transpose of the vector  $v$ . The vector  $\sigma$  is what matters for the dynamics of the coupled system of delay differential equations. The time delays can be shuffled and changed into a new vector  $\tilde{\tau}$ , but the vector  $\sigma$  should be constant, so that

$$B\tau = B\tilde{\tau}. \quad (6)$$

This is the key property of the graph that we need to explore the space of possible solutions of  $\tilde{\tau}$ .

The last necessary step to obtain the full characterization of the system is to derive the time-shifts  $\eta_i$  in Eq. (3) that can lead us back to the time series of the original configuration of Eq. (1). These time-shifts  $\eta_j$  associated to the vertex  $j$  in the network is computed from a recursive relation on the spanning tree  $T^9$ ,

$$\tilde{\tau}_k - \tau_k = \eta_{s(k)} - \eta_{t(k)}, \quad (7)$$

with  $s(k)$  the source vertex of the edge  $k$  and  $t(k)$  the target vertex of the same edge. Since the time-shifts are defined up to a constant, we can choose the value  $\eta_1 = 0$  as a reference for all the other vertices. The rest of the time-shifts can be obtained from the incidence matrix  $A$  restricted to the edges of the spanning tree. The columns of the incidence matrix contain exactly the values  $s(k)$  and  $t(k)$  for any edge  $k$ . If we partition the edges into two subsets of edges in and out of the spanning tree, we can rearrange the incidence matrix in two blocks:

$$A = [A_{in} | A_{out}]. \quad (8)$$

The first matrix  $A_{in} \in \mathbb{Z}^{n \times (n-1)}$  contains the information on the edges in the spanning tree. We also split the time delays in two similar sets  $\tau_{in}$  and  $\tau_{out}$  and we define a column vector with the time-shifts  $\eta = (\eta_1 \dots \eta_n)^T$ . From the recursive relation given in Eq. (7), we can infer that

$$\tilde{\tau}_{in} - \tau_{in} = A_{in}^T \eta. \quad (9)$$

However, we are looking for the time-shifts  $\eta$  as a function of the time delays of the tree  $T$ . Notice that the matrix  $A_{in}$  has a rank  $n - 1$  and that the vector  $\eta$  has  $n - 1$  unknowns since  $\eta_1 = 0$ . Consequently, we can construct a full rank square matrix  $A_r^T$  by removing the first column of  $A_{in}^T$ . We define the vector  $\eta^- = (\eta_2 \dots \eta_n)^T$  and transform the last equation into:

$$\tilde{\tau}_{in} - \tau_{in} = A_r^T \eta^-. \quad (10)$$

Now we have a linear system with a single solution:

$$\eta^- = (A_r^T)^{-1}(\tilde{\tau}_{in} - \tau_{in}). \quad (11)$$

The matrices  $A$  and  $B$  in Eqs (6) and (8) are straightforward to derive and have a strong dependence to each other<sup>12</sup>. Notice also that the matrices  $A_{in}$ ,  $A_{out}$  and  $B$  depend on the initial chosen spanning tree. We can demonstrate that this choice does not affect the space of possible solutions that can be reached with Eq. (6). Any spanning tree can give us a valid basis to reconfigure the time delays in the network.

### Optimization of network time delays

The main problem is stated in Eq. (6) where all the possible vectors  $\tilde{\tau}$  are contained. We have to restrain however the problem to positive time delays  $\tau_k$  to avoid complications with negative time delays. This consists of finding a vector  $\tilde{\tau}$  that will minimize the sum of the time delays over the network. This problem takes naturally the form of a standard linear program, that is,

$$\begin{aligned} & \text{Minimize: } \sum \tilde{\tau}_k \\ & \text{Constrained to: } B\tilde{\tau} = \sigma \\ & \quad \tilde{\tau}_k \geq 0 \end{aligned} \quad (12)$$

This standard linear program can be solved with conventional techniques such as the simplex optimization algorithm<sup>10</sup>. We show that the simplex method reduces the time delays of the network with at least  $n - 1$  zero time delays.

**Proposition 1.** *Using the simplex algorithm, we can guarantee that there is a feasible solution  $\tilde{\tau}$  to the problem in Eq. (12) such that at least  $n - 1$  time delays in the vector  $\tilde{\tau}$  are set to zero.*

*Proof.* In the simplex algorithm, there is a first search for a basic feasible solution to the problem in a  $l$ -dimensional space. For such a solution, the columns of the matrix  $B$  are rearranged into  $[D|Z]$  where  $D$  is an invertible  $c \times c$  matrix and  $Z$  is a  $c \times (n - 1)$  matrix. The vector  $\tilde{\tau} = (\tau_D \tau_Z)$  solution to the equation  $B\tilde{\tau} = \sigma$  can be decomposed into  $\tau_D = D^{-1}\sigma$  and  $\tau_Z = 0$  a vector with all zeros. Being  $n - 1$  the size of the vector  $\tau_Z$ , we have a valid reduction of the network with  $n - 1$  time delays set to zero. The other part  $\tau_D$  contains only positive time delays.  $\square$

The existence of one basic feasible solution gives us a valid reduction, however the algorithm looks further for an optimal solution minimizing the sum of the time delays. The solver will find a solution with  $n_z \geq n - 1$  and a total sum of the time delays below or equal than the initial sum of the time delays  $\sum \tau_k$ . There are plenty of efficient implementations of the simplex algorithm to solve linear programs<sup>10</sup>, and we can obtain the reduction of the network in a polynomial time.

We now have all the ingredients to construct a optimized network. All we need is any spanning tree  $T$ , the incidence matrix  $A$  and a fundamental loop matrix  $B$  of the graph  $G$ .

### Numerical experiments

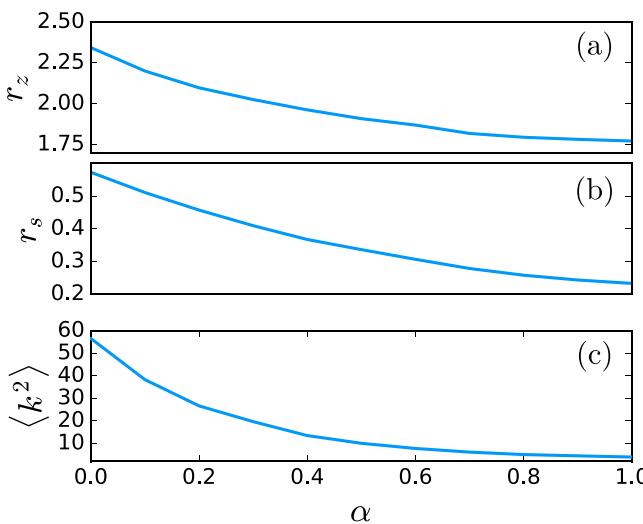
The results of the optimization method may vary with the topology of the network and the statistical distribution of the time delays. Initially, we will focus on the effects of the topology of the network on the optimization output by setting identical time delays on every edge. To quantify the results of the algorithm, we have selected two representative parameters: the ratio  $r_z = n_z/(n - 1)$  that represents the ratio of the number of zero time delays over the lower bound  $n - 1$ , and the ratio

$$r_s = 1 - \frac{\sum_{k=1}^l \tilde{\tau}_k}{\sum_{k=1}^l \tau_k}, \quad (13)$$

which measures the reduction of the sum of the time delays after the optimization process. A number  $r_s = 0$  means that the algorithm was unable to find a lower sum of time delays, while  $r_s = 1$  happens when all the time delays have been reduced to zero.

To understand the role of the topology we have centered our attention on two markers: the density  $\rho$  of the graph and the second moment of the degree distribution  $\langle k^2 \rangle$  as a measure of degree heterogeneity<sup>13,14</sup>. The density of the graph is the ratio between the actual and the maximum number of edges possible for a given number of vertices. In average, it is equal to  $\rho = \langle k \rangle / (n - 1)$  where  $\langle k \rangle$  is the mean vertex degree.

The influence of the degree heterogeneity of the network can be studied by interpolating a network between an Erdős-Rényi random network and a scale-free network<sup>15</sup>. The shape of the degree distribution can be changed



**Figure 1.** Influence of the second moment of the vertex degree distribution  $\langle k^2 \rangle$  on the optimization. The variance  $\langle k^2 \rangle$  is modulated with a parameter  $\alpha$ . The plots represent in (a) the ratio  $r_z = n_z/(n-1)$ , in (b) the ratio  $r_s$  defined in Eq. (13), and in (c) the variance of the vertex degree  $\langle k^2 \rangle$ . The correlation between the variation of  $\langle k^2 \rangle$  and the two ratios  $r_z$  and  $r_s$  is clear. The simulations have been averaged over 30 networks of  $n = 400$  vertices with average degree  $\langle k \rangle = 4$  for each value of  $\alpha$ .

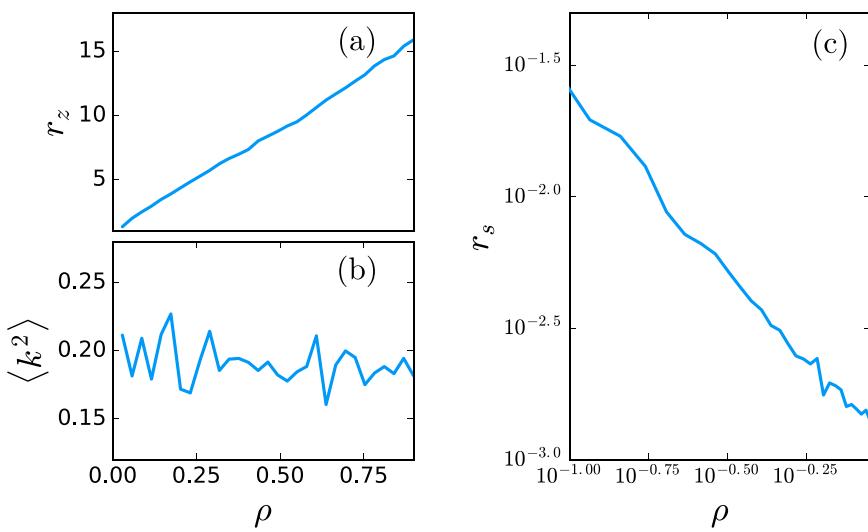
continuously between the two limiting cases with a single parameter  $\alpha$ . For  $\alpha = 0$  we obtain a scale-free network with an exponent 2.8 of the power law degree distribution. When  $\alpha = 1$  we get an Erdős-Rényi random network with a probability  $p = \langle k \rangle / n$  to have an edge between two vertices. The second moment of the degree distribution  $\langle k^2 \rangle$  diverges for the scale-free network in the thermodynamic limit  $n \rightarrow \infty$ . For the Erdős-Rényi random network the degree distribution is binomial and the variance tends to  $np(1-p)$ . So we can expect that  $\langle k^2 \rangle$  decreases when  $\alpha$  takes values from 0 to 1. The results in Fig. 1 show a clear correlation between the ratios  $r_z$ ,  $r_s$  and the variance  $\langle k^2 \rangle$  of the vertex degree. For this simulation, the network is sparse with a constant low density  $\rho = 0.01$ . The average path length is almost constant for all the simulation and, as we will explain later, we have ruled out the influence of the clustering coefficient. It seems that the heterogeneity in the degree of the vertex has an important role in the possible outcome of the reduction process, although the exact relationship between the three measures  $\langle k^2 \rangle$ ,  $r_z$  and  $r_s$  is elusive.

The density  $\rho$  of the graph has also a significant influence on the measures  $r_z$  and  $r_s$ . To illustrate this assertion we will build a network with an adjustable density. Nevertheless it would be good to eliminate the influence of the variance  $\langle k^2 \rangle$ . Noticing that the Watts-Strogatz random graph model with a rewiring probability  $p$  and mean degree  $\langle k \rangle$  has a variance degree roughly equal to  $\langle k^2 \rangle \simeq \langle k \rangle p(1-p) \simeq \langle k \rangle p$  for a small  $p$ , we can construct a network with an arbitrary density and almost a constant degree variance  $\langle k^2 \rangle$ . We start with a regular regular ring network where  $n$  vertices are coupled to the  $k$  nearest neighbors. In this case the network is symmetric and all the vertices have the same degree. We break this symmetry by rewiring each edge with a probability  $p = \varepsilon/\langle k \rangle$  leading to a variance of the vertex degree approximately  $\langle k^2 \rangle \simeq \langle k \rangle p = \varepsilon$ . This is very similar to the small-world model construction but we focus on tuning the density while keeping the variance of the vertex degree constant  $\langle k^2 \rangle$  and very low. We can also assure with this construction that the graph is weakly connected. The results summarized in Fig. 2 clearly uncover the dependence between the density and the ratios  $r_z$ . On the one hand,  $r_z$  is directly proportional to  $\rho$  and on the other hand  $r_s$  and  $\rho$  seem to follow a power law functional as the log-log plot in Fig. 2(c) suggests. This dependence means that  $\rho$ , and consequently the mean degree  $\langle k \rangle$ , has relevant influence on the optimization.

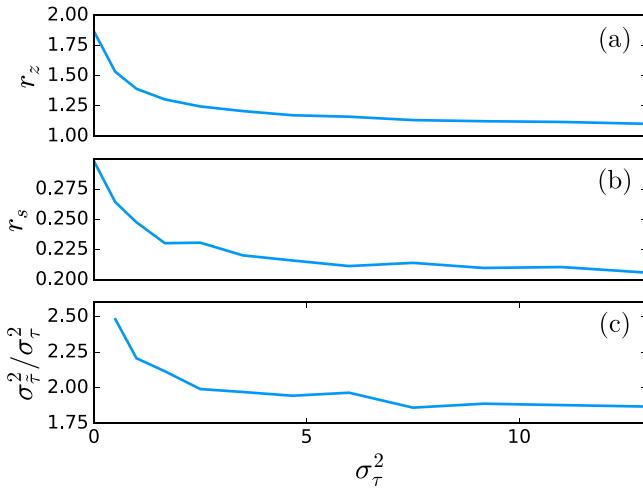
Other factors such as the clustering coefficient or the average path length do not seem to have a significant effect on the ratios  $r_z$  and  $r_s$ . The clustering coefficient of a scale-free network has been tuned using a technique that adds triangles in the network without changing the degree distribution of the network<sup>16</sup>. While the clustering coefficient of the network evolves from 0 to 0.3, the ratios  $r_z$  and  $r_s$  remain almost unchanged. This is a counter-intuitive result since it would have been reasonable to think that the presence of more triangles in the network would have brought an enhancement of the results.

The topology of the network has certainly a strong effect on the outcome of the optimization. However, the distribution of the time delays  $\tau_k$  is also critical as shown in Fig. 3(a,b) where the measures  $r_z$  and  $r_s$  are represented as a function of the variance of the time delay distribution. The time delays have been chosen randomly in the discrete interval  $[1; N_{\max}]$  in order to control the variance  $\sigma_\tau^2$  of the distribution. As this variance increases, the algorithm has more difficulty to find a solution with  $r_z \geq 1$ . The ratio  $r_s$  also tends to diminish but it remains above 0. In Fig. 3(c) we can see that the ratio between the variance of the time delays  $\sigma_\tau^2$  after and  $\sigma_\tau^2$  before the optimization is larger than 1. In general the variance of the time delay distribution will increase after optimization but it seems that the ratio  $\sigma_\tau^2 / \sigma_\tau^2$  is bounded.

When the time delays are distributed following a continuous real valued distribution, it is almost impossible to find a solution with  $n_z > n - 1$ . The simplex method finds only the basic feasible solution  $n_z = n - 1$ , which is the minimum number of zero time delays achievable<sup>9</sup>. If there are special relations between time delays, for example if they are all identical, it might be possible to reach a better solution. In the case of incommensurable real-valued



**Figure 2.** Influence of the density  $\rho$  of the graph. The figure shows the effect of the density  $\rho = \langle k \rangle / (n - 1)$  of a random graph on the optimized delays results. The panels represent in (a) the ratio  $r_z = n_z / (n - 1)$ , in (b) the variance of the vertex degree  $\langle k^2 \rangle$ , and (c) the ratio  $r_s$ . The ratio  $r_z$  is directly proportional to  $\rho$  and  $r_s$  is inversely proportional to  $\rho$ . The simulations have been averaged over 40 realizations of a network with 70 vertices for each value of  $\rho$ .

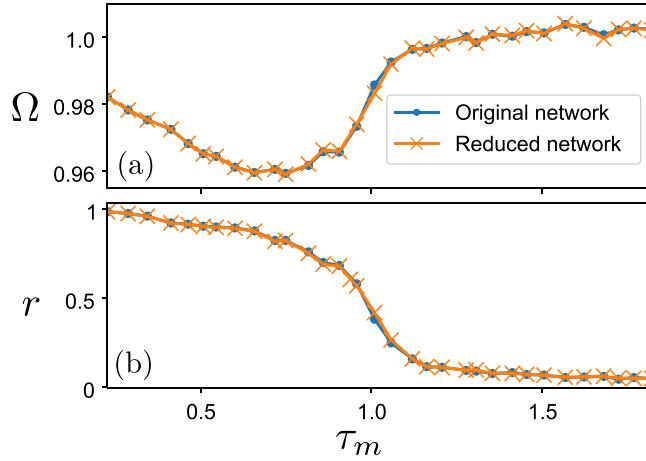


**Figure 3.** Influence of the variance  $\sigma_\tau^2$  of delay distribution. This plot shows the importance of the distribution of the time delay on the results of the algorithm. As the width of the distribution increases, the performance of the optimization get worse. The panels represent in (a) the ratio  $r_z = n_z / (n - 1)$ , in (b) the ratio  $r_s$ , and (c) the ratio between the variance of the time delays  $\sigma_\tau^2$  after and  $\sigma_\tau^2$  before the optimization. The simulations have averaged over 60 realizations of a network with  $n = 400$  vertices with mean degree  $\langle k \rangle = 4$  for each value of  $\sigma_\tau^2$ .

time delays, such relations vanish. However, the simplex algorithm is still capable of finding a lower total sum of time delays, which may be of interest.

While real valued time delays are more general, integer valued time delays are very relevant when it comes to the numerical integration of differential equations. For the numerical algorithms involving finite and constant step size, the values of the time delays, that may have been issued from a continuous distribution, have to be discretized and rounded to the closest integer multiple of the time step. The set of continuous time delays is transformed into a new set of commensurable time delays that will give much better results from the point of view of the optimization.

The previous examples focus on the properties of the networks and delay distribution and do not involve any specific dynamical system. We present an application where a network of Kuramoto phase oscillators is coupled with time delays<sup>4</sup>. The phase oscillator model is a very simple abstraction of the essential properties of limit cycle oscillators. We can use this model to test our optimization method on a complex network of simple dynamical systems. The setup consists of a unidirectional Erdős-Rényi random network with average degree  $d$ , where the vertices represent Kuramoto oscillators with an identical intrinsic frequency  $\omega$ . The edges of the network represent a time delayed interaction chosen randomly according to a statistical distribution. The coupled delay differential equation can be written as



**Figure 4.** Average network network frequency  $\Omega$  and order parameter  $r$  of a coupled network of Kuramoto phase oscillators coupled with time delays. The curves, that are superposed in (a), represent the average network frequency for the original (dot markers) and reduced network (cross markers). For each dot the average network frequency has been computed and averaged for several initial histories of the network to avoid numerical artifacts caused by the integration method. Both the original and reduced network lead to the same asymptotic frequency. In (b), the mean value of the order parameter has been represented for the two sets of simulations. Here again both curves agree on the same synchronization value for the two kind of network. Parameters are:  $\omega = 1$ ,  $K = 0.1$ ,  $\tau_k \in [\tau_m; \tau_m + 0.5]$ ,  $n = 50$ ,  $d = 4$ .

$$\frac{d\theta_i}{dt} = \omega + \frac{K}{d} \sum_{k \in S_i} \sin(\theta_j(t - \tau_k) - \theta_i), \quad (14)$$

where  $S_i$  is the set of edges going from vertex  $j$  to the vertex  $i$  and  $K$  is the coupling strength. We distribute the time delays  $\tau_k$  following a uniform distribution in the continuous interval  $[\tau_m, 0.5 + \tau_m]$ . Notice however that since we integrate the equation numerically, we have to discretize this interval, as said earlier, due to the finite time step size of the algorithm. In order to test the validity of the reduction in a dynamical system, we use the average frequency of the network since this measurement is independent of the initial history of the delay differential equation<sup>17</sup>.

We let evolve the network in time and we compute the average frequency  $\Omega_i$  of each oscillator over a finite interval of time  $T$

$$\Omega_i = \frac{1}{T} \int_0^T \dot{\theta}_i dt. \quad (15)$$

Then we compute the average network frequency  $\Omega$  in this manner

$$\Omega = \frac{1}{n} \sum_i \Omega_i. \quad (16)$$

This last frequency is independent of the chosen initial conditions and should be the same for both the original network and the reduced network given by Eq. (12). In Fig. 4(a), we show an example where a network of  $n = 50$  oscillators has been simulated with a realization of the random time delays. The average frequency of the original and reduced network are consistent in both simulations showing that the asymptotic behavior is the same.

Another quantity of interest in the study of coupled oscillators is the synchronization order parameter

$$r(t) = \frac{1}{n} \left| \sum_{j=0}^n e^{i\theta_j(t)} \right|. \quad (17)$$

This parameter can be averaged over time to characterize the state of the network with a single number

$$r = \frac{1}{T} \int_0^T r(t) dt. \quad (18)$$

We cannot compare directly the order parameters of the original and reduced network since the time series are related through the change of variable in Eq. (3). Being  $\tilde{\theta}_j$  the variables of the reduced system, we can compute the order parameter introducing the time-shifts  $\eta_j$  in the Eq. (19)

$$\tilde{r}(t) = \frac{1}{n} \left| \sum_{j=0}^n e^{i\tilde{\theta}_j(t)} e^{-i\eta_j} \right|. \quad (19)$$

Figure 4(b) represents the average order parameter  $r$  and  $\tilde{r}$  for the original and reduced network for the same parameters as the previous example. Both results overlap almost exactly meaning that the dynamics in the reduced system is conserved.

The simulations have been performed with the programming language Julia<sup>18</sup> using LightGraphs, JuMP and Coin-or Linear Programming (Clp) packages.

## Conclusions

Reorganizing the time delays in a network does not seem to be an easy task at first sight. But once the basic mechanisms of time delay conservation are understood, it is possible to change the time delays and at the same time to conserve the dynamical properties of the network. Our formulation along with the componentwise time-shift transformation technique opens a way to reduce even further the time delay space. When the problem is stated in the form of a linear program, the simplex algorithm provides a higher number of zero time delays than the theoretical lower bound  $n_z$ , that corresponds to the dimension of the cycle space of the network. It also finds the solution with the lowest sum of time delays, which can represent a reduction up to 60% of the initial sum of the time delays.

The numerical integration of coupled dynamical systems with the presence of different time delays among the network usually involves a high computational and storage cost. The memory usage can be reduced up to 30% with the optimization of the delay of the network. Another possible application is to modify the fitness function of the optimization algorithm such that the time delays fit a desired distribution more suitable to the problem at glance.

## References

1. Soriano, M. C., García-Ojalvo, J., Mirasso, C. R. & Fischer, I. Complex photonics: Dynamics and applications of delay-coupled semiconductors lasers. *Rev. Mod. Phys.* **85**, 421 (2013).
2. Liang, X., Tang, M., Dhamala, M. & Liu, Z. Phase synchronization of inhibitory bursting neurons induced by distributed time delays in chemical coupling. *Phys. Rev. E* **80**, 066202 (2009).
3. Perez, T. et al. Effect of the topology and delayed interactions in neuronal networks synchronization. *PLoS One* **6**, e19900 (2011).
4. Yeung, M. S. & Strogatz, S. H. Time delay in the kuramoto model of coupled oscillators. *Phys. Rev. Lett.* **82**, 648 (1999).
5. Earl, M. G. & Strogatz, S. H. Synchronization in oscillator networks with delayed coupling: A stability criterion. *Phys. Rev. E* **67**, 036204 (2003).
6. Lee, W. S., Ott, E. & Antonsen, T. M. Large coupled oscillator systems with heterogeneous interaction delays. *Phys. Rev. Lett.* **103**, 044101 (2009).
7. Petkoski, S. et al. Heterogeneity of time delays determines synchronization of coupled oscillators. *Phys. Rev. E* **94**, 012209 (2016).
8. Lücke, L., Pade, J. P., Knauer, K. & Yanchuk, S. Reduction of interaction delays in networks. *EPL* **103**, 10006 (2013).
9. Lücke, L., Pade, J. & Knauer, K. Classification of Coupled Dynamical Systems with Multiple Delays: Finding the Minimal Number of Delays. *SIAM Journal on Applied Dynamical Systems* **14**, 286–304 (2015).
10. Bazaraa, M. S., Jarvis, J. J. & Sherali, H. D. *Linear programming and network flows* (John Wiley & Sons, 2011).
11. Biggs, N. *Algebraic Graph Theory* (Cambridge university press, 1993).
12. Deo, N. *Graph Theory with Applications to Engineering and Computer Science* (Courier Dover Publications, 2016).
13. Small, M., Hou, L. & Zhang, L. Random complex networks. *National Science Review* **1**, 357 (2014).
14. Newman, M. *Networks: an introduction* (Oxford university press, 2010).
15. Gómez-Gardeñes, J. & Moreno, Y. From scale-free to Erdős-Rényi networks. *Phys. Rev. E* **73**, 056124 (2006).
16. Holme, P. & Kim, B. J. Growing scale-free networks with tunable clustering. *Phys. Rev. E* **65**, 026107 (2002).
17. Nordenfelt, A., Wagelmakers, A. & Sanjuán, M. A. F. Frequency dispersion in the time-delayed kuramoto model. *Phys. Rev. E* **89**, 032905 (2014).
18. Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM Review*, **59**, 65–98 (2017).

## Acknowledgements

This work was supported by the Spanish Ministry of Economy and Competitiveness under Project No. FIS2013-40653-P and by the Spanish State Research Agency (AEI) and the European Regional Development Fund (FEDER) under Project No. FIS2016-76883-P. M.A.F.S. acknowledges the jointly sponsored financial support by the Fulbright Program and the Spanish Ministry of Education (Program No. FMECD-ST-2016).

## Author Contributions

A.W. and M.A.F.S. devised the research. A.W. performed the numerical simulations. A.W. and M.A.F.S. analyzed the results and wrote the paper.

## Additional Information

**Competing Interests:** The authors declare that they have no competing interests.

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



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.