Proceedings Paper:
Clark, R., Punzo, G. orcid.org/0000-0003-4246-9045 and Macdonald, M. (2016) Consensus speed optimisation with finite leadership perturbation in k-nearest neighbour networks. In: 2016 IEEE 55th Conference on Decision and Control (CDC). 55th IEEE Conference on Decision and Control, 12-14 Dec 2016, Las Vegas, NV, USA. IEEE (Institute of Electrical and Electronics Engineers) , pp. 879-884. ISBN 9781509018383

https://doi.org/10.1109/CDC.2016.7798378

Consensus Speed Optimisation with Finite Leadership Perturbation in k-Nearest Neighbour Networks

Ruaridh Clark, Giuliano Punzo and Malcolm Macdonald

***Abstract*— Near-optimal convergence speeds are found for perturbed networked systems, with N interacting agents that conform to k-nearest neighbour (k-NNR) connection rules, by allocating a finite leadership resource amongst selected nodes. These nodes continue averaging their state with that of their neighbours while being provided with the resources to drive the network to a new state. Such systems are represented by a directed graph Laplacian with two newly presented semi-analytical approaches used to maximise the consensus speed. The two methods developed typically produce near-optimal results and are highly efficient when compared with conventional numerical optimisation, where the asymptotic computational complexity is $\mathcal{O}(n^3)$ and $\mathcal{O}(n^4)$ respectively. The upper limit for the convergence speed of a perturbed k-NNR network is identified as the largest element of the first left eigenvector (FLE) of a graph's adjacency matrix. The first semi-analytical method exploits this knowledge by distributing leadership resources amongst the most prominent nodes highlighted by this FLE. The second method relies on the FLEs of manipulated versions of the adjacency matrix to expose different communities of influential nodes. These are shown to correspond with the communities found by the Leicht-Newman detection algorithm, with this method enabling optimal leadership selection even in low outdegree (< 12 connections) graphs, where the first semi-analytical method is less effective.**

I. INTRODUCTION

The focus of this paper is on developing the capabilities to effectively control a mobile engineered swarm. Primarily by considering how to identify optimal leadership for achieving fast consensus, which provides a mechanism for influencing the entire network without requiring control of all nodes. But this work could be extended to leadership selection on a range of networked systems, including leader election for distributed computing, targeted release of genetically altered mosquitoes or effective use of campaigning resources.

For a distributed system, attempting to reach consensus, optimising the speed of convergence is desirable as it enables the system to respond faster to inputs while also facilitating cohesion. There has been notable work on how topology can influence consensus with [1] finding that random rewiring and the creation of small-world networks can dramatically increase consensus speed.

In [1] unperturbed consensus dynamics are considered whereas in this paper perturbation-driven consensus is the focus, where *leader* nodes are required to drive the network from one state to another. The selection of effective leaders is therefore key and the problem has been considered by [2], [3] for achieving controllability and by [4], [5], [6] for robustness by minimising the system error in the presence of noise. There are also contributions to the problem of achieving fast convergence through leadership. In [7] fast consensus is pursued with leaders that are given external input but only considers a single leader node, which will be shown to often but not always be an optimal strategy. The approach taken in the current paper is the same as [8] where a limited leadership resource is applied to nodes across the network. Leader selection is required due to the limited resource constraint, without a limit all nodes could be fully controlled making consensus trivial. Limited communication is a viable constraint when considering the operation of such a system where point-to-point communication may provide greater security but prevent communication to all nodes at once.

In [8] the leadership perturbation is defined as the network Laplacian's first left eigenvector (FLE), or Perron vector, with a similar metric also vital in this paper where the FLE of the adjacency matrix shall form the basis of the leadership selection algorithms. The FLE of a graph's adjacency matrix is associated with the largest eigenvalue in magnitude and details each node's relative ability to serve as the origin of many arbitrarily long walks on the network [9]. However the perturbations, defined by the algorithms described herein, provide a significant improvement upon the convergence speeds achieved using the unmodified FLE.

To the best of the author's knowledge there are no other comparable methods for the limited perturbation case and therefore much of this work requires comparison with conventional numerical optimisers. Near-optimal is the phrase used to describe the results of the algorithms developed herein, highlighting that the results produced are on a par with those of the numerical optimiser. The combinatorial nature of the optimal perturbation problem results in common numerical approaches often finding local minima. This in turn makes identification of an optimal solution only possible for a limited set of networks where the optimality can be analytically proven. Numerical approaches also scale poorly when tackling combinatorial problems, with leader selection being no exception. It shall be seen that the computational requirements of purely numerical approaches become prohibitive as the network size grows.

When considering an engineered swarm there are many sources of inspiration in nature. For example starling flocks, which maintain a k-nearest neighbour (k-NNR) network structure, with each starling usually observing its seven nearest neighbours ($k \approx 7$). These topological rules are found to provide the starling swarm with robustness benefits

R. A. Clark, G. Punzo and M. Macdonald are with the Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow G1 1XQ, UK {ruaridh.clark, malcolm.macdonald.101}@strath.ac.uk

[10], therefore this work shall take the k-NNR structure as its starting point and focus on how to achieve a highly responsive system by perturbing the system through supplying leadership resources. The networks considered are also directed, emulating the communication graphs of flocks and swarms in nature, where it has been shown that the communication/sensing cost of protocols, with directed information flow, is smaller than that of their undirected counterparts [11].

An upper limit for convergence speed to consensus shall be defined by considering only the first left eigenvector of the unperturbed system. This limit is used to validate the main contribution of the paper, which is the development of semi-analytical methods that effectively and efficiently identify a network leadership perturbation. This limit could also be applied as a distributed topology optimising algorithm, where [12] presents a distributed algorithm for evaluating a network's first left eigenvector that could also be the foundation for applying the leadership perturbation algorithms in a distributed manner.

II. CONSENSUS MODEL

In [11] a theoretical framework for the analysis of consensus algorithms for multi-agent systems is presented, where a graph is used to represent a given network. Adopting this framework results in the convergent rate to consensus being represented by the smallest, non-zero, eigenvalue of a graph's Laplacian matrix [13].

The networks considered have N agents connected via local communication with a static, time-invariant, topology. This is represented with a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, that is at least weakly connected, where $\mathcal{V} = 1, 2, ..., N$ is a set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges. A uniform signal $\mathbf{u} = u \cdot \{1, 1, ..., 1\}^T \in \mathbb{R}^N$ is supplied to all agents with different positive gains c_i, where $i = 1, 2, ..., N$. The dynamics of this system are defined as

$$\dot{x_i} = \sum_{i=1}^{N} a_{ij}(x_i - x_j) + c_i(u - x_i) \tag{1}$$

where x_i is the state of the i^{th} agent and u is the scalar target value that all agents must achieve. a_{ij} is the ij entry of the graph adjacency matrix, that is 1 if there exists a directed edge from node i to j and 0 otherwise. Each connection as documented in the adjacency matrix provides a directional link for sharing information on an agents state. The resource allocation, c_i, ranges from 0 to 1, where $\sum_i c_i = 1$, and scales the comparison between the uniform input signal, u, and the current state x_i. The collective dynamics of the network can be expressed as

$$\dot{\mathbf{x}} = -L\mathbf{x} + C(\mathbf{u} - \mathbf{x}) \tag{2}$$

where L is the Laplacian matrix of graph $\mathcal{G}$ and C is the perturbation matrix where $C = \text{diag}(\mathbf{c}) = \text{diag}(c_1, ..., c_N)$. The Laplacian is defined as $L = D - A$ where $D = \text{diag}(d_1, ..., d_N)$ is the degree matrix of $\mathcal{G}$, with elements $d_i = \sum_j a_{ij}$, and A is the adjacency matrix.

Lemma 1 *[14], [15] For a directed network G with individual protocol (2), consensus is achieved if G contains a directed spanning tree, that is there exists a vertex u such that for any $v \in \mathcal{V}$, there exists a directed path from u to v.*

The essence of Lemma 1 implies that for a directed network, the sufficient condition for achievement of consensus is that each agent is reachable from the input $\mathbf{u}$ through a directed path.

Eq. (2) can be rewritten

$$\dot{\mathbf{x}} = -(L + C)\mathbf{x} + C\mathbf{u},$$

before removing the $C\mathbf{u}$ term by means of a coordinate change. Consider

$$\mathbf{y} = \mathbf{x} - (L + C)^{-1}C\mathbf{u}.$$

After applying the coordinate change the model becomes

$$\dot{\mathbf{y}} = -(L + C)\mathbf{y}. \tag{3}$$

The Laplacian matrix always has a zero eigenvalue for each connected component of the directed graph, i.e. one zero eigenvalue for a strongly connected graph, where $|\lambda_1| = 0 \leq |\lambda_2| \leq ... \leq |\lambda_N|$ with any complex eigenvalues occuring in conjugate pairs. For the perturbed negated Laplacian, $S = -L - C$, the smallest eigenvalue is non-zero and therefore the magnitude of $\lambda_1(S)$ becomes the measure of convergence rate to consensus for the system [16]. From the Perron-Frobenius theorem we have the following:

Lemma 2 *[17] The eigenvalue $\lambda_1(S)$ is a simple eigenvalue and the entries of the FLE share the same sign.*

The dominant eigenvalue for the adjacency matrix, and therefore also the Laplacian matrix, is real and, for a strongly connected graph, strictly dominant with an algebraic multiplicity of 1 [18].

III. CONSENSUS SPEED LIMIT

In the following sections the notation $(.)_i$ is used to indicate an element of the vector $(.)$ whenever the vector's notation already includes a subscript or the vector is the result of an operation indicated in the brackets

Theorem 1. *Let L be the Laplacian of a connected, directed graph with outdegree of k for all nodes (k-outdegree) and $C = diag([c_1, ..., c_N])$ the diagonal perturbation matrix consisting of non-negative entries, where $\sum_i c_i = 1$. Then, the limit for the smallest eigenvalue of the perturbed Laplacian system, $S = -L - C$, is*

$$\lambda_1(S) > -\max_i((v_L)_i) \tag{4}$$

where $(v_L)_i \ \forall \ i \in \mathcal{V}$ is an element of the FLE of L.

Theorem 1 can be proved as follows.

Proof: The first left eigenvector of S, $\mathbf{v}_S$, is defined as $\mathbf{v}_S^\intercal S = \lambda_1(S)\mathbf{v}_S^\intercal$ from which the dominant eigenvalue can be found, when $\sum_i (v_S)_i = 1$ where $(v_S)_i$ is an element of $\mathbf{v_S}$, to be

$$\lambda_1(S) = \sum_i (\lambda_1(S)\mathbf{v}_S^\intercal)_i = \sum_i (\mathbf{v}_S^\intercal S)_i. \tag{5}$$

Since L is a Laplacian matrix, $\sum_i (\mathbf{v}_S^\intercal L)_i = 0$, the diagonal perturbation matrix, C, can be substituted into (5) as

$$\lambda_1(S) = \sum_i (\mathbf{v}_S^\intercal S)_i = -\sum_i (\mathbf{v}_S^\intercal C)_i = -\sum_i (v_S)_i c_i. \quad (6)$$

The maximum dominant eigenvalue can then be defined as

$$\max(\lambda_1(S)) = \max(-\sum_i (v_S)_i c_i) = -\max_i((v_S)_i). \quad (7)$$

For $-\max_i((v_S)_i)$ to be achieved, $c_i = 1/m$ for those i corresponding to $\max_i((v_S)_i)$ and 0 elsewhere. Note that m is the number of elements in $\mathbf{v}_S$ equal to $\max_i((v_S)_i)$ and $\sum_i c_i = 1$ is maintained.

An equation that approximately represents the shift, $\delta\lambda_1$, in a distinct eigenvalue of a generic square matrix is defined in [19] as

$$\delta\lambda_1(S) = \lambda_1(S) \approx -\frac{\mathbf{v}_L^\intercal C \mathbf{r}_L}{\mathbf{v}_L^\intercal \mathbf{r}_L}, \quad (8)$$

where $\mathbf{v}_L$ and $\mathbf{r}_L$ are the left and right eigenvectors respectively of L corresponding to the eigenvalue, $\lambda_1(L)$. Given that

$$\lambda_1(S) = \frac{\mathbf{v}_S^\intercal S \mathbf{r}_S}{\mathbf{v}_S^\intercal \mathbf{r}_S} \quad (9)$$

it can be deduced that $\mathbf{v}_S \approx \mathbf{v}_L$ and $\mathbf{r}_S \approx \mathbf{r}_L$ for small perturbations.

In reality, $\mathbf{v}_S \neq \mathbf{v}_L$ and $\mathbf{r}_S \neq \mathbf{r}_L$, in particular $(v_S)_i < (v_L)_i$ for i corresponding to $\max_i((v_S)_i)$ where the optimal perturbation $c_i = 1/m$, implicitly defined in (7), is applied. This can be shown by first considering $-\mathbf{v}_L^\intercal L = \lambda_1(-L)\mathbf{v}_L^\intercal = \mathbf{0}$ and Lemma 2, from which it can be noted that $(\mathbf{v}_L^\intercal S)_i = 0$ when $c_i = 0$. Observing that $(\mathbf{v}_S^\intercal S)_i = (\lambda_1(S)\mathbf{v}_S^\intercal)_i < 0$ as long as i is globally reachable. Then for S it can be seen that $(v_S)_i > (v_L)_i$ where $c_i = 0$. Therefore, given that $\sum_i (v_L)_i = \sum_i (v_S)_i = 1$, it can be seen that for the optimal perturbation case $(v_S)_i < (v_L)_i$ for i corresponding to $c_i = 1/m$. Hence, the limit for the magnitude of the smallest eigenvalue of the perturbed Laplacian is $\max_i((v_L)_i)$ and is approached when $(v_S)_i \to (v_L)_i$. ■

IV. ALGORITHMS

A. Power Optimisation

The FLE, $\mathbf{v}_L$, of the Laplacian matrix has been shown to be a good strategy for allocating resources in certain cases [8]. However a numerical optimiser, using sequential quadratic programming methods with an active-set algorithm[1], is able to consistently uncover better allocations [8]. The optimiser's search space is dependent on network size, resulting in $\mathcal{O}(n)$ operations requiring the calculation of the matrix eigenvalue $\mathcal{O}(n^3)$ [21], producing a total run time of $\mathcal{O}(n^4)$. For a k-NNR network, the Power Optimisation strategy is a semi-analytical approach that focuses resources on the most prominent nodes by raising the FLE to some power, p, according to

$$\mathbf{c} = \frac{\mathbf{v}_L^p}{\sum_i (v_L)_i{}^p} \quad (10)$$

[1]Numerical optimiser was implemented with the fmincon algorithm in MATLAB [20].

with $\mathbf{c}$ being the resource allocation vector and where $\mathbf{v}_L^p$ is an element-wise operation. The maximum magnitude for the smallest eigenvalue, $\lambda_1(S)$, is sought by locally changing p in $\mathbb{R}$ using the bisection method [22]. This method iteratively reduces the resources to less influential nodes while increasing those assigned to the most prominent until the convergence rate stops increasing, while maintaining $\sum_i c_i = 1$. The search space is dependent on the power, p, and hence does not grow with an increasing number of nodes, n, therefore the eigenvalue calculation is the dominant process resulting in $\mathcal{O}(n^3)$ for the Power Optimisation method.

In Theorem 1 the optimal perturbation matrix C is optimal in that it can approach the upper limit for the smallest eigenvalue. The Power Optimisation vector, as defined in (10), will be shown to converge towards an optimal perturbation and in doing so will reveal the conditions in which the limit for the smallest eigenvalue can be reached.

Defining the perturbation matrix as

$$C = -\text{diag}\{\mathbf{c}\} = -\text{diag}\Big\{\frac{\mathbf{v}_L^p}{\sum_i (v_L)_i{}^p}\Big\}, \quad (11)$$

Fig. 1 displays the power, p, determined to produce the maximum eigenvalue shift for 10 k-NNR networks at each node interval between 50 and 150. The networks were created by randomly distributing nodes in a plane before applying k-NNR connection rules for an outdegree of 30. It can be seen that a high power above 45 is usually required to find the optimum resource allocation. By applying this high power assumption and also considering the constraint $\sum_i c_i = \sum_i \frac{(v_L)_i^p}{\sum_i (v_L)_i^p} = 1$ applied to (11), a case is produced where $c_i \not\approx 0 \ \forall \ i$ that approximately correspond to $\max_i((v_L)_i)$ and $c_i \approx 0$ otherwise. Hence,

$$\lambda_1(S) > \sum_i (v_L)_i \Big(-\text{diag}\Big\{\frac{(v_L)_i^p}{\sum_i (v_L)_i^p}\Big\}\Big) = -m \cdot \max_i((v_L)_i) \cdot \frac{1}{m} = -\max_i((v_L)_i) \quad (12)$$

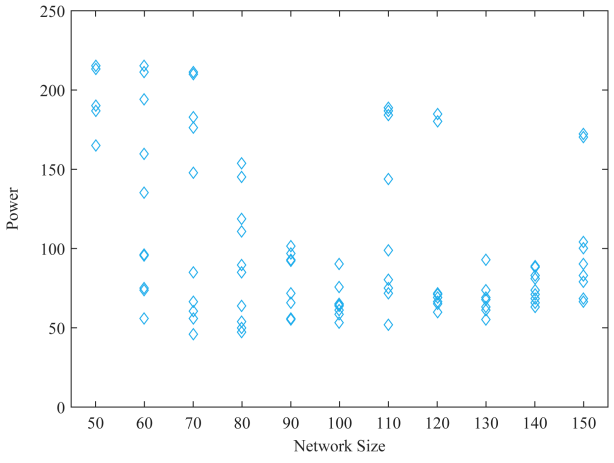

Fig. 1: Power, p, found that produces highest consensus speed for the Power Optimisation method.

where m is the number of elements for which $c_i \not\approx 0$. Noting that in (12), $\frac{(v_L)_i^p}{\sum_i (v_L)_i^p} \approx 0$ or $\frac{(v_L)_i^p}{\sum_i (v_L)_i^p} \approx \frac{1}{m}$ for large p. The Power Optimisation approach, therefore, attempts to achieve the optimal perturbation highlighted in (4). This is not possible for every graph but the method will be shown to be highly effective for many scenarios, in particular high outdegree networks.

1) Results: As stated in Theorem 1, the limit magnitude for the smallest eigenvalue of the perturbed, negated, Laplacian is equal to the largest element of the FLE, $\max_i((v_L)_i)$. The results in Fig. 2 approach this limit for outdegrees greater than 26, in a 50 node network where nodes have been randomly distributed in a plane before applying k-NNR connection rules for a range of outdegrees. The numerical optimiser is described in Section IV-A and is considered to be a near-optimal benchmark The consensus speeds for the numerical and power optimisation approaches are seen to converge with, but never exceed, the $\max_i((v_L)_i)$, achieving better results than the unmodified FLE that was used as a leadership allocation in [8].

B. Communities of Influence

The number of communities/modules detected by the Leicht-Newman algorithm for directed networks [23], from hereon referred to as Leicht-Newman modules, is compared with the network outdegree. High outdegree, k-NNR, networks are found to often be composed of fewer non-overlapping modules (e.g. 2 modules for a 100 node network with 50 outdegree) than lower outdegree scenarios where many modules are present (e.g. 10 modules for 100 node network with 5 outdegree). It is shown in section IV-B.2 that the Power Optimisation method achieves its best results at higher outdegrees. This matches the findings in [24] where a single node is seen to become an increasingly effective leader in denser networks, i.e. those with a greater number of connections. Therefore, a similar approach to Power Optimisation is taken for low outdegree cases but it is repeated for multiple influential communities instead of just focusing on the most prominent one. This method will be referred to as the Communities of Influence (CoI) approach, where the FLE of the adjacency matrix, $\mathbf{v}_a$, is calculated for manipulated versions of the network to find different communities of influential nodes. The algorithm is defined in Algorithm 1, where it is worth noting that for a k-outdegree graph $\mathbf{v}_L = \mathbf{v}_a$. However, after the deletion of a node's connections $\mathbf{v}_L \neq \mathbf{v}_a$ and $\mathbf{v}_a$ is required to reveal the other influential communities.

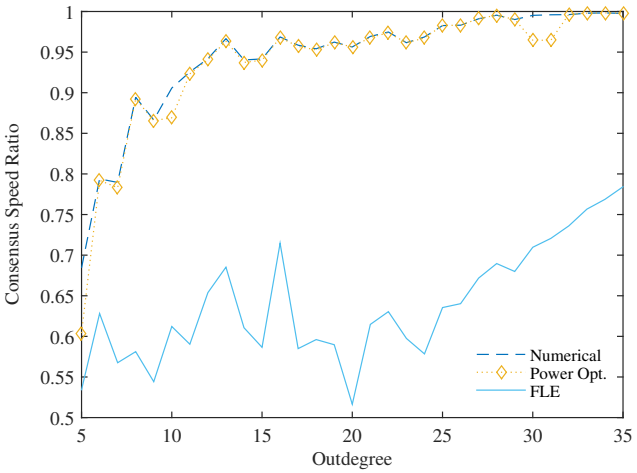

Fig. 2: Consensus speed ratio with respect to $\max_i((v_L)_i)$ for various resource allocation vectors over a range of outdegrees in a 50 node k-NNR network.

Algorithm 1 CoI

procedure FINITE LEADERSHIP OPTIMISATION
Calculate the FLE, $\mathbf{v}_a$, for the adjacency matrix, $A = (a_{ij}) \in \mathbb{R}^{N\times N}$.
for $m = 1 : n \in \mathbb{R}^+$ **do**
Define the resource vector, $\mathbf{c}_m = \frac{(\mathbf{v}_a)_m{}^p}{\sum_i (v_a)_m{}^p}$ $\forall\ i \in \mathcal{V}$, where $p = p_m \in \mathbb{R}^+$.
$\forall\ i \in \mathcal{V},\ j \in \arg\min_k((v_a)_k),\ a_{ij} = a_{ji} = 0$.
Calculate $\mathbf{v}_{m+1}$ for the updated adjacency matrix.
end for
Quasi-Newton solver[2] maximises $|\lambda_1(S)|$ where $C = \mathrm{diag}(\mathbf{c}_{CoI^n})$ with $r_m \in \mathbb{R}^+$ a weighting variable for the different resource vectors.
$\mathbf{c}_{CoI^n} = f(\mathbf{c}_1, ..., \mathbf{c}_m, p_1, ..., p_m, r_1, ...r_{m-1})$ with the function $f(...)$ defined in (13).
end procedure

In Algorithm 1, the number of communities required to find a near-optimal leadership varies depending on the topology in question. Five communities of influence were deemed sufficient for the networks examined in this paper. The resource combining function for n communities is

$$\mathbf{c}_{CoI^n} = \frac{\mathbf{c}_1 + \sum_{i=2}^{n} \frac{\mathbf{c}_i}{r_{i-1}}}{1 + \sum_{i=1}^{n-1} \frac{1}{r_i}} \tag{13}$$

where the denominator, with weighting variables $\{r_1, ..., r_{n-1}\}$, scales the combined vectors to ensure $\sum_i (c_{CoI^n})_i = 1$ and $\{\mathbf{c}_1, ..., \mathbf{c}_n\}$ is the resource vector defined in Algorithm 1. The initial guesses for the powers (p_1,...,p_n) were 50, given the results presented in Fig. 1 for the Power Optimisation method, with the weighting variables set at 1. The presence of more variables in the optimisation increases the search space and the algorithm run time when compared with the Power Optimisation, however the run time remains defined by the eigenvalue calculation. The worst case for CoI is therefore also $\mathcal{O}(n^3)$, but the difference in actual computational time is explored in the next section.

1) Communities: In Fig. 3, a sample analysis of a 50 node network depicts the influential nodes from four separate communities of influence. The influence of each node for a specific community is proportional to the radius of the associated circle. The resource allocations, CoI[4] and Numerical,

[2]Numerical optimiser was implemented with the fminunc algorithm in MATLAB [25]

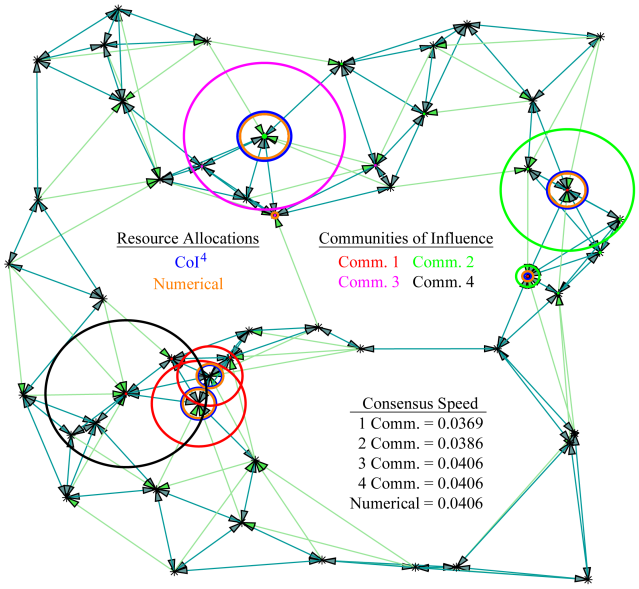

Fig. 3: 50 node, 5 outdegree, k-NNR network. Two-way connections are depicted in dark green with one-way in light green. Coloured circles are centred on nodes, with the circle radius proportional to the resource allocation.

are also detailed in the figure; the CoI^4 vector uses the CoI method for four communities to achieve a consensus speed that is on par with the numerically optimised result.

Figure 4 displays the modules detected, for the topology shown in Fig. 3, by the Leicht-Newman algorithm [23]. By comparing these plots it can be seen that the nodes selected by CoI are based in different modules to facilitate the whole graph in reaching consensus. Comm. 1 to 4 are all located in separate modules, which shows some modules to be more influential than others, with nodes from the black, magenta and cyan modules in Fig. 4 not required for achieving fast consensus.

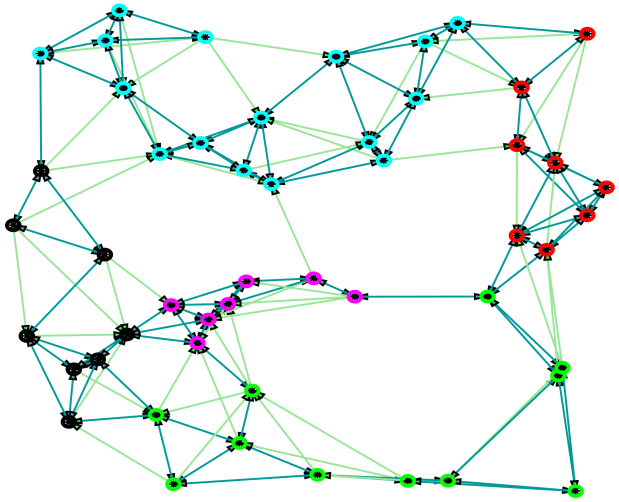

Fig. 4: Five modules, as determined by the Leicht-Newman algorithm [23], highlighted by different coloured circles.

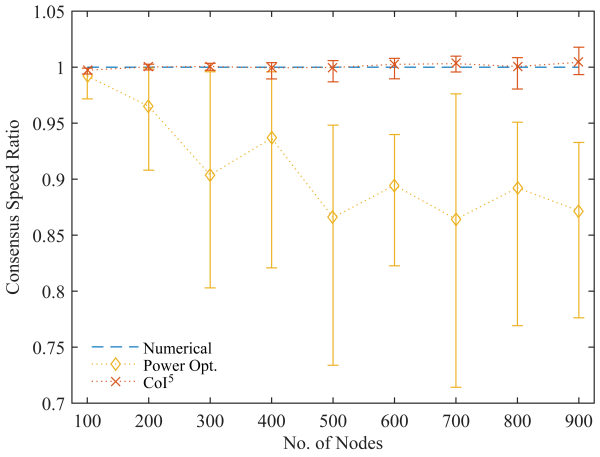

Fig. 5: Consensus Speed Ratio for k-NNR networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean.

2) Results: The algorithms developed were validated through comparisons with the numerical optimiser, described in Section IV-A, that is considered to be a near-optimal benchmark. In Fig. 5 the Power Optimisation and CoI^5 methods were compared with the numerical optimiser, for forty k-NNR networks with randomly distributed nodes at each node interval between 100 and 900.The Consensus Speed Ratio being defined in reference to the numerical approach where a ratio value greater than 1 indicates a faster consensus speed than the numerical result. Fig. 5 shows an improvement with respect to the purely numerical approach as the networks grow larger. The worst case run times are reiterated in Table I and compared with the trend line for data in Fig. 5. The coefficient of determination, R^2, is also displayed in the table to show the accuracy of the trend line fit.

TABLE I: Algorithm run time comparison for n nodes.

Algorithm	**Run Time**	**Actual Trend [s]**	**R^2**
Numerical	$\mathcal{O}(n^4)$	$5\times10^{-7}n^{3.5}$	0.998
CoI^5	$\mathcal{O}(n^3)$	$1\times10^{-4}n^{2.1}$	0.9898
Power Opt.	$\mathcal{O}(n^3)$	$7\times10^{-6}n^{2.1}$	0.9912

C. Large Networks

For large networks ($N \geq 1000$) a comparison of the power optimisation and CoI methods with the numerical optimiser benchmark was not feasible, due to the computational time required. Therefore the upper limit for consensus speed, $\max_i((v_L)_i)$ as stated in Theorem 1, was used to show in Fig. 6 that near-optimal results were still being achieved. In the case of a high outdegree network, the CoI^5 vector only requires the contribution of one community and, hence, closely matches the resource allocation generated by the Power Optimisation approach. This is the case in Fig. 6 where an outdegree of 30 is large enough for Power Optimisation analysis to find a near-optimum resource allocation. It is worth noting that the Power Optimisation outperforms the CoI^5 vector, due to the CoI method relying on a numerical optimiser, that finds suboptimal local minima when analysing

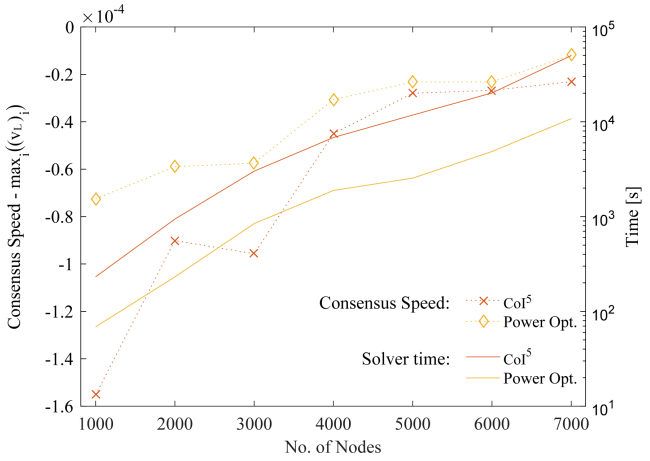

Fig. 6: Solver time and difference between optimised consensus speed and $\max_i((v_L)_i)$ for large k-NNR networks with outdegree set at 30 connections. Seven networks are analysed (one at each 1000 node interval) where $\max_i((v_L)_i) = \{31, 14, 10, 7, 5, 4, 4\} \times 10^{-4}$ for the analysed networks.

such a large network. The Power Optimisation method is effective even with very large networks (10^3 nodes) where the calculation times for the numerical optimiser would be extremely long, with the trend in Table. I predicting that a 7000 node network would take approximately 166 days to evaluate.

V. CONCLUSIONS

The newly presented semi-analytical methods (Power Optimisation and Communities of Influence) leverage the first left eigenvector (FLE) of a graph's adjacency matrix and manipulated versions of this matrix to effectively identify the best candidates to be supplied with leadership resources in k-nearest neighbour networks. The leadership resources considered are finite and can be allocated to produce a near-optimal consensus speed that can, in certain high outdegree scenarios, approach the upper limit for consensus speed driven by a finite perturbation. For a k-outdegree network this limit is shown to be the largest element of the FLE of the Laplacian or adjacency matrix. The methods presented have a reduced time complexity, when compared with a conventional numerical method, making these methods attractive when considering larger networks where taking a numerical approach can be computationally exhaustive.

VI. ACKNOWLEDGMENTS

This work was supported in part by the Engineering and Physical Sciences Research Council Doctoral Training Grant to the University of Strathclyde.

REFERENCES

[1] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 2371–2378.

[2] M. Ji, A. Muhammad, and M. Egerstedt, "Leader-based multi-agent coordination: Controllability and optimal control," in *American control conference*, 2006, pp. 1358–1363.

[3] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[4] M. Fardad, F. Lin, and M. R. Jovanović, "Algorithms for leader selection in large dynamical networks: Noise-free leaders," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 7188–7193.

[5] S. Patterson and B. Bamieh, "Leader selection for optimal network coherence," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 2692–2697.

[6] K. Fitch and N. E. Leonard, "Information centrality and optimal leader selection in noisy networks," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 7510–7515.

[7] Y. Wang, W. Yang, and X. Wang, "Optimal leader selection for fast consensus via consensus centrality," in *Control Conference (CCC), 2014 33rd Chinese*. IEEE, 2014, pp. 1482–1487.

[8] G. Punzo, G. Young, N. Leonard, and M. Macdonald, "Limited resources allocation for agile swarms," *Scientific reports*, 2016, *Accepted, unpublished*.

[9] K. Klemm, M. Á. Serrano, V. M. Eguíluz, and M. San Miguel, "A measure of individual role in collective dynamics," *Scientific reports*, vol. 2, 2012.

[10] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini *et al.*, "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proceedings of the national academy of sciences*, vol. 105, no. 4, pp. 1232–1237, 2008.

[11] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, 2004.

[12] Z. Qu, C. Li, and F. Lewis, "Cooperative control with distributed gain adaptation and connectivity estimation for directed networks," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 3, pp. 450–476, 2014.

[13] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[14] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988–1001, 2003.

[15] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[16] H. Shao and M. Mesbahi, "Degree of relative influence for consensus-type networks," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 2676–2681.

[17] R. Bapat, S. Kirkland, and S. Pati, "The perturbed laplacian matrix of a graph," *Linear and Multilinear Algebra*, vol. 49, no. 3, pp. 219–242, 2001.

[18] R. Dhal, S. Roy, Y. Wan, and A. Saberi, "Majorisations for the eigenvectors of graph-adjacency matrices," *International Journal of Control*, vol. 87, no. 12, pp. 2604–2614, 2014.

[19] A. Deif, "Rigorous perturbation bounds for eigenvalues and eigenvectors of a matrix," *Journal of computational and applied mathematics*, vol. 57, no. 3, pp. 403–412, 1995.

[20] MathWorks, "Constrained nonlinear optimization algorithms," http://www.mathworks.se/help/optim/ug/constrained-nonlinear-optimization-algorithms.html, 2015, [Online; accessed 2-March-2015].

[21] G. Stewart, "Matrix algorithms volume I: Basic decompositions," 2000.

[22] A. Kaw, E. Kalu, and D. Ngyen, "Numerical methods with applications," 2008.

[23] E. A. Leicht and M. E. Newman, "Community structure in directed networks," *Physical review letters*, vol. 100, no. 11, p. 118703, 2008.

[24] W. Ying, Y. Wen, and W. Xiaofan, "Optimal leader selection for fast consensus via consensus centrality," in *Control Conference (CCC), 2014 33rd Chinese*. IEEE, 2014, pp. 1482–1487.

[25] MathWorks, "fminunc unconstrained minimization," http://uk.mathworks.com/help/optim/ug/fminunc-unconstrained-minimization.html, 2015, [Online; accessed 2-March-2015].