Dynamical influence driven space system design

Ruaridh A. Clark, Ciara N. McGrath, and Malcolm Macdonald

Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW,
ruaridh.clark@strath.ac.uk

Abstract. Complex networks are emerging in low-Earth-orbit, with many thousands of satellites set for launch over the coming decade. These data transfer networks vary based on spacecraft interactions with targets, ground stations, and other spacecraft. While constellations of a few, large, and precisely deployed satellites often produce simple, grid-like, networks. New small-satellite constellations are being deployed on an ad-hoc basis into various orbits, resulting in complex network topologies. By modelling these space systems as flow networks, the dominant eigenvectors of the adjacency matrix identify influential communities of ground stations. This approach provides space system designers with much needed insight into how differing station locations can better achieve alternative mission priorities and how inter-satellite links are set to impact upon constellation design. Maximum flow and consensus-based optimisation methods are used to define system architectures that support the findings of eigenvector-based community detection.

1 Introduction

Historically, constellations were composed of a few large satellites that produced simple, grid-like, communication network topologies. New small-satellite constellations present as complex data transfer networks due to the variety of orbital positions and heterogeneous capabilities of the satellites involved. This paper demonstrates how holistic assessment of these complex networks can aid space system designers.

Data transfer is a spreading process that can be represented by a network in order to detect the relative influence of nodes [3]. A network of averaged contacts over time, enables the network's adjacency matrix to provide insights into the major pathways for spread, as in [2] for the identification of influential disease spreaders. For space system flow networks, where targets are sources of data and ground stations are sinks, the eigenvectors of the adjacency matrix can detail the relative influence of ground stations in terms of receiving target data. Specifically, the concept of dynamical influence – the influence of a node's dynamical state on the rest of the network [7] – is employed to detect influence and divide the system according to the data received from targets. This form of community detection was introduced in [3], as the communities of dynamical influence (CDI), whereby communities were detected based on node alignment in a Euclidean space defined by the system's dominant eigenvectors (i.e. those associated with the largest eigenvalues).

A network-based approach is proposed herein, partly because it is intractable to evaluate a wide range of feasible architectures through the use of high fidelity simulations of data transfer. Constellation design has predominantly focused on large, latency prioritising, constellations that maintain continuous contact between targets and ground stations (referred to as *bent-pipe* systems). Examples of these systems include OneWeb and Starlink, where target-ground station geographical proximity [14, 1] has been shown to drive ground station placement and *minimum cost, maximum flow* optimisation has been used to define effective inter-satellite link topologies [14]. For many other applications involving data collection, latency is important but not critical as long as it falls within reasonable bounds. These *store-and-forward* systems – where spacecraft gather information from one location (e.g. ship AIS beacons or Earth monitoring images) and deliver it to another surface location (referred to as a ground station) – are the focus of this paper as ground station placement must account for both target coverage and data throughput.

In the past, ground station network design has relied on engineering judgement and best practices. Lacoste et al. demonstrated the difficulties in applying best-practices for selecting multiple ground stations [8]. They found it difficult to predict the contribution of a given ground station to an existing set, highlighting the need for combinatorial optimisation methods for the ground station selection problem. Our approach aims to allow designers to select stations in strategic locations, hence reducing the number of stations or the lease time they require to deliver their service.

2   Methods

This paper analyses flow networks that represent the data transfer capacities of entities in a space systems, where the sink nodes (ground stations) are connected to source nodes (targets) via intermediary spacecraft nodes that can also share inter-satellite link (ISL) connections. A toy example of a space system without ISLs is displayed in Fig. 1. Ground station selection methods shall be presented that consider data volume and target coverage. This toy example highlights how one ground station receives data from both targets and therefore – regardless of the data volume transferred along each connection – will achieve better target coverage. The *loop-back* edges ensure cycles that are necessary for spectral identification of popular pathways. *Loop-back* edge weights are far smaller weights than data transfer edges to minimise the impact of these artificial connections on assessments of target coverage.

2.1   Space System Definition

A space system is defined for this study based on the orbital positions and targets of the Spire Global, Inc. constellation that collects AIS data from ships globally. All 111 spacecraft that as of July 2021 were operated by Spire Global, Inc. are included in this case study. The two-line elements (TLEs) for these spacecraft are
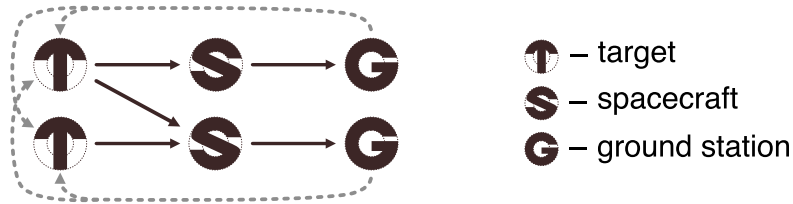
Fig. 1. Toy example of a data transfer flow network. Dashed lines indicate artificial *loop-back* edges that ensure each ground station is part of a cycle.

obtained from [6]. The Keplerian orbit elements of the spacecraft at epoch are detailed in data set [11]. All spacecraft in the constellation are in approximately circular orbits but, due to the use of rideshare launches, are at a variety of altitudes, Right Ascension of Ascending Nodes (RAANs), and inclinations. The spacecraft are well distributed in RAAN, with 74 in sun-synchronous orbits, 22 at around 51.6 degrees inclination, 8 at around 37 degrees, 4 in near-polar orbits, and 3 in near equatorial orbits.

The target locations for the case study are based on data provided by Spire Global, Inc. [15] for the 24-hour period of 11-August-2019 14:09 UTC to 12-August-2019 14:08 UTC that provides the last reported position of all ships detected in this 24-hour window. From this, 250 targets are positioned to approximate the locations of ships worldwide that cannot be seen from land with these locations detailed in data set [11].

Ninety-four ground station sites are considered for this study, including 77 detailed by Portillo et al. as possible ground station locations [13] and an additional 17 locations estimated from Spire Global, Inc. published ground station network. These locations are detailed in the data set [11].

2.2   Data Transfer Capacity Network

The *data transfer capacity* networks are graphs defined as $G = (V, E)$, where there is a set of V vertices and E edges, which are ordered pairs of elements of V when considering data transfer. The adjacency matrix, A, is a square N × N matrix where N is the number of vertices and is equal to the total number of ground station, spacecraft, and targets in the system. An adjacency matrix captures the network's connections where $(A)_{ij} > 0$ if there exists an edge connecting vertex i and j and 0 otherwise. This matrix is representative of the data transfer capacity of communication links that emerge during a defined period of time.

Data transfer capacities are calculated by simulating the motion of all spacecraft. The spacecraft initial locations and orbit paths are propagated using a fixed-step integrator based on the Livermore Solver for Ordinary Differential Equations [17]. The equations are formulated using the Gauss equations as derived from general perturbation methods [16]. Only perturbations due to the Earth's oblateness to the second order (J_2) are included.
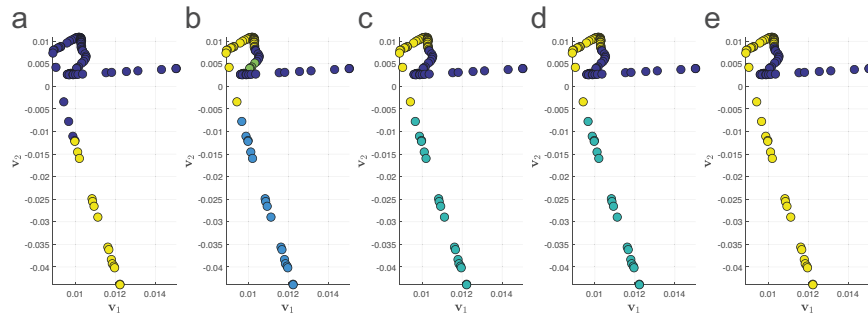
Fig. 2. Variation in CDI communities (denoted by node colour) with number of input eigenvectors from 3 to 7 (**a** to **e**). v_1 & v_2 are the first two dominant eigenvectors.

From the simulation, the cumulative time that each spacecraft, target, or ground station is in view of a spacecraft is determined. A spacecraft is considered in view of a target or a ground station if the elevation angle is greater than 15 degrees elevation. Therefore, $(A)_{ij} = c_{ij}d_{ij}$ where c_{ij} is cumulative time in view and d_{ij} is the data rate between nodes i and j.

Artificial *loop-back* edges from all ground stations (sinks) to all targets (sources), see Fig. 1, create cycles that are captured by the system's eigenvectors. A weight of 0.001 is used for the artificial loop-back edges, where this value has a minor influence on community assignment, as described in the following section, as long as the value is far smaller than the data transfer weights.

2.3   Communities of Dynamical Influence

Communities of dynamical influence (CDI), introduced in [3], is used to provide insight into the flow pathways through the network. CDI identifies communities based on their alignment in Euclidean space defined by the system's dominant eigenvectors. The nodes in this space, which are further from the origin of the coordinate system than any of their connections, are defined as leaders of separate communities. This is assessed by comparing the magnitude of each node's position vector with the scalar projection onto this vector from all other node position vectors. Each community is then defined with respect to the leaders, by assessing which leader each node is in closest alignment with using the scalar product of position vectors.

Communities are ranked in terms of their influence by evaluating the largest entry of the first dominant eigenvector (v_1) for each community (i.e. eigenvector centrality (EC)) that is known to be a non-negative vector [9]. The community that contains the node with the largest EC value ($\mathbf{v}_1$) is ranked as the most influential community, with the other communities ranked in descending order according to their largest EC value.

Increasing the number of eigenvectors, used by CDI, equates to increasing the number of dimensions - in the eigenvector-defined Euclidean space - and

this can lead to variation in community assignment. For example, the yellow (least influential) community in Fig. 2 **b**, **c**, & **d** is part of the purple (most influential) community in **a** and grows to include the negative v_2 nodes in **e**. By incorporating more eigenvectors, a more nuanced picture of community structure can be revealed. However, the more eigenvectors included the less prominent are the most dominant eigenvectors and, hence, the greater risk that the communities no longer reflect the most popular data pathways.

ISL space systems. By including ISLs, spacecraft nodes can become more prominent (i.e. larger eigenvector entries) than ground station nodes for the first few dominant eigenvectors. Ground station community assignment using CDI is prone to error if the ground station nodes are not prominent in any of the eigenvectors used. Therefore an adaptation of CDI is required to detect consistent ground station communities. Previously 5 dominant eigenvectors were used by CDI, for systems with ISLs this is updated to include eigenvectors up to the 5th dominant eigenvector that includes a ground station node with the largest entry in magnitude.

For the example presented later in Section 3, $500\,\text{kb/s}$ ISL data rates (see Fig. 5 **a**) results in the first 33 dominant eigenvectors being used to evaluate CDI. While for $5\,\text{kb/s}$ ISL data rates (see Fig. 5 **b**) all the largest magnitude entries – for the first 5 dominant eigenvectors – belonging to ground station nodes, so the first 5 dominant eigenvectors are used by CDI.

2.4 Ground Station Selection

A few methods of ground station selection are considered. These methods use the flow network to make a ground station selection based on differing design priorities.

Maximum flow aims to maximise the data throughput from any targets to the ground stations. Mean consensus leadership performs a trade-off between data volume and coverage to achieve high data throughput and good target coverage. Minimum consensus leadership prioritises target coverage, by improving the connectivity of the least connected target.

Maximum Flow. Maximum flow is assessed using the Ford-Fulkerson algorithm [5] by considering all targets as a single source. This enables maximum flow to be calculated, from the data transfer capacity network, for each sink node (ground station) separately. Maximum flow considers the bottlenecks for data flow by considering every link's transfer capacity from source to sink.

Linear Consensus Protocol. The consensus leadership selections are identified by assessing the ability of ground station nodes to lead target nodes to consensus according to the following consensus protocol.

We consider a network where each node v_i has a state $x_i \in \mathbb{R}$ and continuous-time integral dynamics, $\dot{x}_i(t) = u_i(t)$ where $u_i \in \mathbb{R}$ is the control input for agent

i. The linear consensus protocol is $u_i(t) = \sum_{v_j \in N_i} a_{ij}(x_j - x_i)$ and describes how each node adjusts its state based on the state of its neighbours, as presented in [12], where $A_2 = [a_{ij}]$ is the weighted adjacency matrix (for paths of length 2) and the set of neighbours for node v_i is N_i. This adjacency matrix for paths of length 2, A_2, can be created by squaring the adjacency matrix, $A_2 = A^2$. (Note: This approach is only proposed for systems without inter-satellite links (ISLs) as target to ground station paths using ISL transmissions will have length greater than 2.)

Given the linear consensus protocol, the state of the network develops according to $\dot{x}(t) = -Lx(t)$ with the graph Laplacian matrix, L, defined as $L = D - A_2$ where $D = diag(out(v_1), ..., out(v_n))$ is a diagonal matrix composed of the out-degrees of each node, i.e. $out(v_i) = \sum_j a_{ij}$.

Given the definitions for the continuous-time integral dynamics and $\dot{x}_i(t)$, the discrete-time agent dynamics are given in [4] as $x_i(t + 1) = x_i(t) + \epsilon u_i(t)$ provided that $0 < \epsilon < \frac{1}{\max_i d_{ii}}$ where d_{ii} is an element of D. The choice of ϵ affects the number of steps required for nodes to reach convergence, therefore $\epsilon = 0.999 \times \frac{1}{\max_i d_{ii}}$ as the number of computational steps can be reduced while still guaranteeing convergence of the system [4]. Convergence is defined here as $\bar{x}_i > 0.99 \ \forall \ i \in \tau$, where τ is the set of all source nodes, when $x_j = 1 \ \forall \ j \in g$ with g the set of all sink nodes.

Consensus Leadership. To select ground stations, all ground station nodes are provided with variable resources that define their contribution in leading the target nodes to a new state. In fact, all nodes have resources assigned in a resource vector, $\mathbf{r} = \{r_1, ..., r_n\}$ where n is the total number of nodes in the system. However, for all non-ground station nodes these resources are set $r_i = 1 \ \forall \ i \notin g$ where g is the set of all sink nodes. For the ground station nodes, $0 \geq r_i \geq 1 \ \forall \ i \in g$, and $\sum_i r_i = n_d \ \forall \ i \in g$ where n_d is the *desired* number of sink nodes (i.e. the number to be selected).

These resources define each ground station's contribution by scaling each row of the adjacency matrix by $\mathbf{r}$,

$$A_w = A_2 \odot R \quad \text{where} \quad R = \left.\begin{bmatrix} \mathbf{r} \\ \mathbf{r} \\ \vdots \\ \mathbf{r} \end{bmatrix}\right\} n \text{ elements}$$

where $\odot$ indicates a Hadamard product, i.e. element-wise multiplication that in this case only varies the weight of elements in columns corresponding to sink nodes. This altered adjacency matrix, A_w, replaces the adjacency matrix, A_2, for assessing state dynamics using the linear consensus protocol.

The initial state of all sink nodes is set at $x_i = 1 \ \forall \ i \in g$, where g is the set of all sink nodes. Therefore, the system will reach consensus (i.e. $\bar{x}_i \approx 1 \ \forall \ i \in \tau$ with $\bar{x}$ the mean value and τ the set of all source nodes) as long as all of the source nodes have a directed path to one of the sink nodes in g, which act as network leaders.

Algorithm 1 Consensus leadership optimisation

1: **Input:** Resource vector ($\mathbf{r}$), desired no. of sink nodes (n_d)
2: Set $n_s \leftarrow 0$ (number of selected sink nodes)
3: **while** $n_d > n_s$ **do**
4:     **if** $n_d - n_s > 1$ **then**
5:         Gradient-based numerical optimisation with input *uniform* vector
6:         **if** function evaluations/iterations limit is exceeded **then**
7:             Power optimisation of resources
8:         **end if**
9:         **if** $r_i < 1 \ \forall \ i \in v$, where v are un-selected sink nodes **then**
10:             Gradient-based numerical optimisation with input *zero* vector
11:             **if** function evaluations/iterations limit is exceeded **then**
12:                 Power optimisation of resources
13:             **end if**
14:         **end if**
15:         Add i to selected nodes and set $r_i \leftarrow 1$, where $r_i = \max_i r_i \ \forall \ i \in v$
16:     **else**
17:         Brute force final node selection
18:     **end if**
19: **end while**
20: **return** Optimised resource allocation

The resource vector $\mathbf{r}$ is primarily optimised using a gradient-based numerical optimiser [10], supported by heuristic algorithms that improve efficiency and mitigate against the solver finding local minima far from the global optimum.

The optimiser attempts to minimise separate objective functions to produce either a mean or minimum consensus leadership selection. A ground station node i is selected whenever the optimiser assigns $r_i > 1$, or when the optimisation converges and r_i has the largest entry of $\mathbf{r}$ from the pool of unselected ground stations. An overview of this algorithm is provided in Algorithm 1. Note that Power Optimisation is a heuristic algorithm, described in [3], that efficiently increases the proportion of resources assigned to the largest entries of $\mathbf{r}$. For the final ground station selection it is possible to assess each node individually (brute force) rather than use a gradient-based optimiser.

Mean Consensus Leadership. The mean consensus leadership aims to maximise the mean consensus state of all target nodes, with the optimisation defined as follows,

$$
\begin{aligned}
\min \quad & \frac{\sum_{i \in \tau}(1 - x_i)}{n_\tau} \\
\text{s.t.} \quad & r_j \leq 1 \ \forall \ j \in g \\
& \sum_j r_j = n_g \ \forall \ j \in g, \ n_g \in \mathbb{Z}
\end{aligned}
\tag{1}
$$

where n_τ is the number of source nodes, n_g is the number of ground stations, and m is the total available resources. The source nodes states, x_i, are evaluated at a point prior to convergence, defined as the closest step to $0.9 \times s_{ref}$ where s_{ref} is the number of steps to convergence. Initially, s_{ref} is defined for a system with a uniform resource vector, $r_i = \frac{n_d}{n_g} \ \forall \ i \in g$ where n_d is the *desired* number of sink nodes and n_g is the total number of sink nodes considered (i.e. possible ground station locations). Note that s_{ref} and the *evaluation step* are updated during the optimisation, after each sink node selection when for a given sink node i the allocated resources become $r_i = 1$ for the first time.

Minimum Consensus Leadership. The minimum consensus leadership aims to maximise the consensus state of the target with the lowest state value at the evaluation step, with the optimisation defined as follows,

$$\begin{aligned} \min \quad & \left(1 - \min_{i \in \tau}(x_i)\right) \\ \text{s.t.} \quad & r_j \leq 1 \ \forall \ j \in g \\ & \sum_j r_j = n_g \ \forall \ j \in g, \ n_g \in \mathbb{Z}. \end{aligned} \quad (2)$$

3   Results

The division of ground stations into communities of dynamical influence (CDI), reveals the differences in ground station connectivity to spacecraft and targets. For the space system described in Section 2.1, Fig. 3 **a** & **b** shows how community assignment relates to the network embedding according to v_1, v_2, and v_3. The ability of these communities to reach targets globally is captured by v_1, where Fig. 3 **c** shows that the largest v_1 entries are attributed to the community of northern and southern ground stations (community 1).

These ground station communities can be understood by considering the spacecraft inclinations in the constellation. Inclination provide an estimate for the highest latitude ground station that will be visible to a given spacecraft, where the field of view can allow for contact to be made with high latitude ground stations. Spacecraft inclination also provides insight into the ground stations that will be seen for longest, as those will be ground station with similar latitudes to a spacecraft's inclination. Therefore, the 1st community (purple) achieves long contact times with polar orbiting spacecraft (74 sun-synchronous and 4 near-polar orbit in this constellation), with the lower latitudes in this community serviced most readily by the 22 spacecraft at 51.6 degrees. The 2nd community, in terms of influence (cyan), contains equatorial stations that are serviced by 3 equatorial orbiting spacecraft, which provide a strong connection through to equatorial shipping targets. Finally, the 3rd community (yellow) forms a band that is most connected to the 8 spacecraft at 37 degrees inclination.

Three methods of ground station selection, as described in Section 2.4, are employed to propose ground station locations for differing mission priorities. In
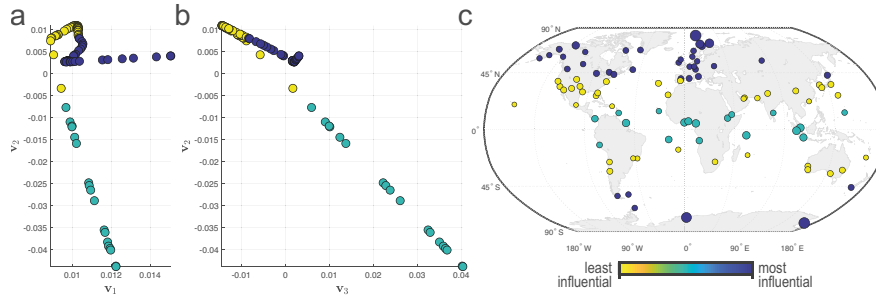
Fig. 3. Ground stations are assigned communities using CDI, colour denotes influence in respect to target connectivity. Nodes are embedded in **a** with v_1 & v_2 and in **b** with v_2 & v_3. In **c**, the nodes are placed according to their location and dot size for each node i is proportional to $(v_1)_i^3$.

Fig. 4, selections of 1, 5, 10, 20, and 30 ground stations are presented both in a Euclidean space, defined by v_1 & v_2, and on a map of the Earth with community assignment noted. Fig. 4 demonstrates how CDI and v_1 can combine to provide intuitive insights for space system designers.

Maximum flow selects (Fig. 4 **a** & **b**) exclusively from the most influential community (coloured purple) as its objective improves data throughput without considering target coverage. The selections largely align with v_1 magnitude, but v_1 is still a global measure of influence that considers connectivity to all targets. Therefore, maximum flow selects exclusively from high latitude ground stations due to the increased time in contact with spacecraft versus other locations.

The mean consensus leadership (Fig. 4 **c** & **d**) is shown to select ground stations that cover all three communities. For small selection sizes 1,5, & 10, the most influential community is still prioritised but this method achieves the most even division of ground stations across the three CDI communities for the 20 & 30 ground station selections. Resulting in 9, 12, and 9 ground stations in the communities, ordered by influence, for the 30 selection.

The minimum consensus leadership (Fig. 4 **e** & **f**) shows the same initial prioritisation of the most influential community. However, the selections then diverge from mean consensus leadership by prioritising the least influential community, which likely equates to improving connectivity to the least connected targets in the system. Ground stations are still selected from all three communities to ensure global target coverage.

Incorporating inter-satellite links (ISLs) with 1000 km range on every spacecraft, for the space system described in Section 2.1, alters the distribution of influential ground station as shown in Fig. 5. The updated community maps differ depending on the data rate set for the ISLs, where in **a** 500 kb/s and in **b** 5 kb/s are modelled while downlink data rates to ground stations remain at 1000 kb/s. Comparing these maps with Fig. 3 **c** indicates that the ground stations with the greatest influence continue to be at the highest latitude bands. In
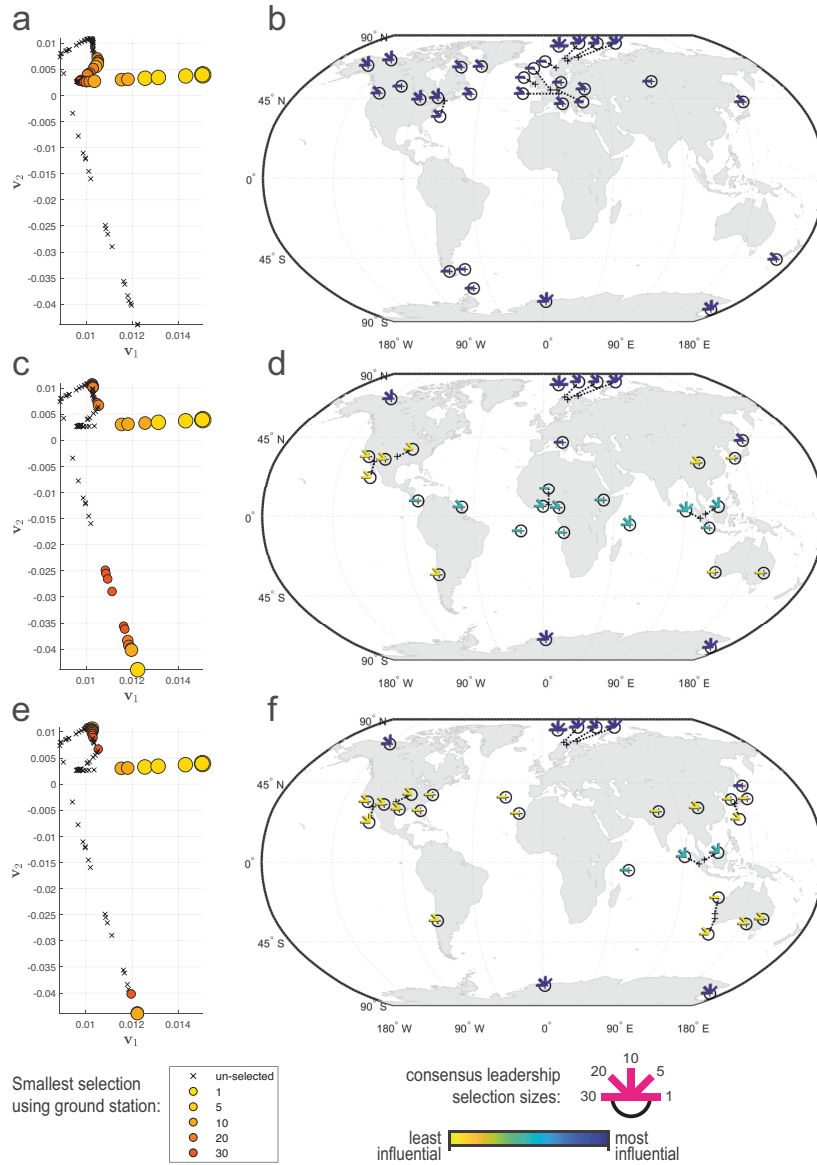
Fig. 4. Differing ground station selections are shown for various sizes. Selections: **a** & **b** maximum flow; **c** & **d** mean consensus leadership; **e** & **f** minimum consensus leadership. In **a**, **c**, & **e** the smallest selection size employing each ground station is noted, alongside the node's position in a Euclidean space defined by v_1 & v_2. In **b**, **d**, & **f** the location and community designation using CDI (see Fig. 3 **b**) are highlighted for the ground stations in each selection size. Communities are denoted by colour.
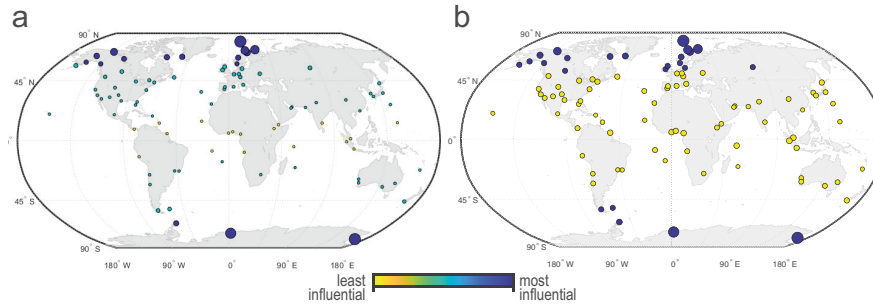
Fig. 5. Ground station community maps are presented for ISL links with data rates of **a** 500 kb/s and **b** 5 kb/s. Ground station nodes are assigned to communities using CDI, colour denotes community influence in respect to target connectivity. Nodes are embedded according to **a** v_1 & v_2 and **b** v_2 & v_3. In **c**, the nodes are placed according to their location and dot size for each node i is proportional to $(v_1)_i^3$.

contrast to Fig. 3 **c** the equatorial ground stations are less important, which is likely due to the ISLs enabling the equatorial satellites to pass equatorial target data onto polar orbiting spacecraft. This claim is also supported by the comparison between Fig. 5 **a** and **b**. The equatorial ground stations have the smallest v_1 values in **a** with 500 kb/s ISLs, but their v_1 values are equivalent or larger than many higher latitude stations in **b** when less data can be transmitted between spacecraft with only 5 kb/s ISLs. This increase in equatorial influence in **b**, also results in only one community being detected at latitudes below 50 degrees.

4   Conclusions

The differing contact patterns of a spacecraft constellation can result in the formation and detection of communities of dynamical influence (CDI). These CDI communities are informative when considering space system design, specifically ground station selection, as they identify both high data throughput locations and locations with different target coverage. These insights were highlighted through comparison with different ground station selection methods. For the selections that maximised data throughput – maximum flow – all ground stations were shown to belong to the most influential community. For the selections that prioritised improving connectivity of the least connected targets – minimum consensus leadership – the largest proportion of stations in the large selection sizes (20 & 30 ground stations) were based in the least influential community. For the selections that aimed to achieve a balance between high data throughput and target coverage – mean consensus leadership – there was an even division of ground station across the three CDI communities in the 20 & 30 ground station selections. Finally, it was shown using CDI that inter-satellite links (ISLs) should be accounted for by space system designers, as they will likely impact the best performing ground station locations.

References

1. Chen, Q., Yang, L., Liu, X., Guo, J., Wu, S., Chen, X.: Multiple gateway placement in large-scale constellation networks with inter-satellite links. International Journal of Satellite Communications and Networking 39(1), 47–64 (2021)
2. Clark, R.A., Macdonald, M.: Identification of effective spreaders in contact networks using dynamical influence. Applied network science 6(1), 1–18 (2021)
3. Clark, R.A., Punzo, G., Macdonald, M.: Network communities of dynamical influence. Scientific reports 9(17590) (2019), doi: 10.1038/s41598-019-53942-4
4. Di Cairano, S., Pasini, A., Bemporad, A., Murray, R.M.: Convergence properties of dynamic agents consensus networks with broken links. In: 2008 American Control Conference. pp. 1362–1367. IEEE (2008), doi: 10.1109/ACC.2008.4586682
5. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. In: Classic papers in combinatorics, pp. 243–248. Birkhäuser Boston (2009), doi: 10.1007/978-0-8176-4842-8_15
6. Kelso, T.S.: Norad two-line element sets current data. https://celestrak.com/, accessed: 07/11/2019
7. Klemm, K., Serrano, M.Á., Eguíluz, V.M., San Miguel, M.: A measure of individual role in collective dynamics. Scientific reports 2(292) (2012), doi: 10.1038/srep00292
8. Lacoste, F., Guérin, A., Laurens, A., Azema, G., Periard, C., Grimal, D.: FSO ground network optimization and analysis considering the influence of clouds. In: Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP). pp. 2746–2750. IEEE (2011)
9. Lohmann, G., Margulies, D.S., Horstmann, A., Pleger, B., Lepsien, J., Goldhahn, D., Schloegl, H., Stumvoll, M., Villringer, A., Turner, R.: Eigenvector centrality mapping for analyzing connectivity patterns in fmri data of the human brain. PloS one 5(4) (2010)
10. MathWorks: 'fmincon', constrained minimization. https://uk.mathworks.com/help/optim/ug/fmincon.html (2020)
11. McGrath, C.N., Clark, R.A.: Location of ground stations, targets and spacecraft for spire global case study (Jul 2021), https://doi.org/10.5281/zenodo.5243314
12. Olfati-Saber, R., Murray, R.M.: Consensus problems in networks of agents with switching topology and time-delays. IEEE Transactions on automatic control 49(9), 1520–1533 (2004), doi: 10.1109/TAC.2004.834113
13. del Portillo, I., Cameron, B., Crawley, E.: Ground segment architectures for large leo constellations with feeder links in ehf-bands. In: 2018 IEEE Aerospace Conference. IEEE (2018), doi: 10.1109/AERO.2018.8396576
14. del Portillo, I., Cameron, B.G., Crawley, E.F.: A technical comparison of three low earth orbit satellite constellation systems to provide global broadband. Acta Astronautica 159, 123–135 (2019), doi: 10.1016/j.actaastro.2019.03.040
15. Spire Global Ltd.: Spire maritime website. https://maritime.spire.com/, accessed: 10/12/2019
16. Vallado, D.A., McClain, W.D.: Fundamentals of Astrodynamics and Applications. Microcosm Press & Springer, third edn. (2007), pp. 615–633, pp. 642–667
17. Virtanen, P., et al.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17, 261–272 (2020)