communications physics

A Nature Portfolio journal

Article

https://doi.org/10.1038/s42005-025-02474-7

Assigning entities to teams as a hypergraph discovery problem

Guilherme Ferraz de Arruda [1], Wan He [2], Nasimeh Heydaribeni [3], Tara Javidi [3], Yamir Moreno [4] & Tina Eliassi-Rad [2,5]

Assigning agents to teams under strict task and effort constraints is crucial in business, science, and engineering, where disruptions can cause significant losses. Current methods do not explore hypergraph-based solutions that explicitly optimize algebraic connectivity under constraints, leaving unresolved how to systematically form robust, recoverable teams. We present a hypergraph-based team assignment algorithm where nodes represent agents and hyperedges represent tasks. The search is guided by input constraints and aims to optimize resilience and diffusion by maximizing the algebraic connectivity of an edge-dependent, vertex-weighted hypergraph. We employ constrained simulated annealing to find a satisfactory hypergraph by enforcing both the minimum effort required for task completion and the maximum effort agents can exert. We evaluate robustness by assessing solution recovery after node removal attacks. Our results demonstrate that the hypergraph formulation yields more robust solutions than the bipartite formulation and the greedy approach.

The Team Formation Problem (TFP) involves assigning individuals to one or more tasks. The optimal solution often depends on a subjective definition of the fitness function[1]. This problem was originally proposed in ref. 2 and has been studied from various perspectives[1–15]. The most studied formulations include the formation of robust and recoverable teams[1,3–6], budget and profit optimization[1,3,7–9], and single- or multi-skilled candidate optimization[1,10–15], among others. Regardless of the fitness function, the optimal solution often requires computing all possible assignments, which, due to the combinatorial nature of the problem, makes it NP-hard[1]. Therefore, heuristics are often used to obtain locally optimal solutions that balance computational cost and solution quality.

Solving the TFP is important beyond its theoretical and computational interest. Ref. 1 discusses modern applications of TFP, such as using TFP to explore what-if scenarios. For example, an organization could use TFP to consider possible reorganizations of its employees. Another application is Labor Strategy Optimization[1], where TFPs can help inform decisions about an organization's capability, location, and flexibility given a desired demand.

Moreover, the authors of a recent review article[1] propose a relationship between TFP and the N-body problem in physics. Their argument is based on the observation that aggregating a set of pairwise interactions does not capture the dynamics between groups of people. This is the same motivation for studying hypergraphs in complex systems[16–24]. In this context, the motivation to use hypergraphs — or other forms of higher-order interactions — is to analyze dynamical phenomena, such as social contagion, where one-to-many and/or many-to-many interactions occur[16–18,20,22,24]. Beyond this class of models, hypergraphs have been studied in a wide variety of problems, including cooperation in groups[25,26], percolation[27,28], random walks[29–31], and synchronization[32–34]. These works show that hypergraphs can differ significantly from graphs, both in how individuals interact and in the fragility of these structures. In particular, assuming that the failure of some nodes implies the failure of the hyperedge, the results in ref. 28 suggest that hypergraphs can be very fragile.

We focus on a TFP in which teams are robust and recoverable. We propose a hypergraph-based approach for task assignments. In our context, robustness is defined as the ability of a team to complete its tasks after the removal of an agent or a set of agents. We also aim to incorporate heterogeneity in (i) the importance of agents in the assigned tasks and (ii) the energy and budgets of tasks and agents. The rationale is that an agent may play a fundamental role in one task but a less important role in another project. For example, consider scientific collaborations where a researcher might simultaneously lead one project and play a lesser role by contributing to other projects. This is also reflected in the time the researcher spends on each project, which may have different requirements. Therefore, we propose using edge-dependent vertex weight (EDVW) hypergraphs[35], where agents (nodes) can have different weights in different tasks (groups or hyperedges), capturing the

[1]Institute of Physics Gleb Wataghin, University of Campinas (UNICAMP), Campinas, SP, Brazil. [2]Network Science Institute & Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115, USA. [3]Electrical and Computer Engineering Department & Halıcıoğlu Data Science Institute, University of California, San Diego, CA 92093, USA. [4]Institute for Biocomputation and Physics of Complex Systems (BIFI) and Department of Theoretical Physics, University of Zaragoza, Zaragoza 50018, Spain. [5]Santa Fe Institute, Santa Fe, NM 87501, USA. e-mail: tina@eliassi.org

heterogeneities of agents and tasks. Moreover, inspired by the study of resilience in graphs[36,37], we propose using the algebraic connectivity of the Laplacian matrix[35], which is an ideal candidate for summarizing robustness, as it includes all the information of an EDVW hypergraph. The advantage of this approach is that, as a by-product of optimizing robustness via algebraic connectivity, we simultaneously reduce the diffusion timescale, which may facilitate communication between agents. In addition, we explicitly propose including the energies and budgets as constraints in our optimization algorithm, which also captures this type of heterogeneity.

Our main contributions are mapping the TFP to a hypergraph discovery problem and demonstrating that the algebraic connectivity of the EDVW Laplacian quantifies the resilience of formed teams against attacks, specifically node removals. We frame TFP as a hypergraph discovery problem and systematically analyze several small hypergraph cases. We also conduct a finite-size analysis for certain hypergraph classes, showing that algebraic connectivity is constrained by the hypergraph and is nontrivially related to its connectivity. Next, we propose a constrained simulated annealing approach that searches the hypergraph space to find a hypergraph satisfying the team-formation constraints while maximizing algebraic connectivity. We measure the resilience of the optimized hypergraphs under node-removal attacks and show that, in most cases, they incur lower patching costs. Furthermore, the number of unsuccessful patches is virtually zero, whereas it can be as high as 60% in the original (observed) hypergraph.

Results

This section contains two parts: (i) results from experimental analysis and (ii) results from experiments on real-world data. Table S3 in Supplementary Note 9 lists the notations used in this paper.

Results from experimental analysis

Algebraic connectivity. The edge-dependent vertex-weighted hypergraph (EDVW)[35] is defined as $\mathcal{H} = \{\mathcal{V}, \mathcal{E}, \omega, \gamma\}$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of nodes, $\mathcal{E} = \{e_1, \ldots, e_K\}$ is the set of hyperedges, which are subsets of nodes of arbitrary size, $\omega(e_k)$ is a function weighting the hyperedges, and $\gamma(v_i, e_k)$ is a function weighting the importance of node v_i in hyperedge e_k. Note that a node can have different weights depending on the hyperedge. This type of hypergraph is particularly interesting for modeling rich data with context-dependent weights. Examples include collaboration networks[35], machine learning applications such as hypergraph neural networks[38–44], and chemical reactions[45,46].

We define the weighted degree of each agent as $d(v_i) = \sum_{e\in\mathcal{E}(v_i)}\omega(e)$, and the weighted degree of each task as $\delta(e) = \sum_{v\in e}\gamma(v, e)$. A random walk that captures all the relationships and weights in an EDVW hypergraph can be defined as a sequence of nodes where: (i) the walker in node v_i chooses a hyperedge e according to its weighted degree, i.e., $\frac{\omega(e)}{d(v_i)}$, then (ii) the walker chooses a node v_j within hyperedge e with probability proportional to its hyperedge degree, i.e., $\frac{\gamma(v_j,e)}{\delta(e)}$.

The probability transition matrix for our random walk is expressed as

$$P = D_V^{-1} W D_E^{-1} R^T, \tag{1}$$

where

- $P \in \mathbb{R}^{N\times N}$ is usually an asymmetric matrix.
- $D_V \in \mathbb{R}_+^{N\times N}$ is the diagonal node degree matrix, whose components are the weighted degree of each agent, i.e., $[D_V]_{ii} = d(v_i)$.
- $W \in \mathbb{R}_+^{N\times K}$ is the hyperedge weight matrix, whose components $W_{ik} = \omega(e_k)$ if node v_i is in the hyperedge e_k and $W_{ik} = 0$ otherwise.
- $D_E \in \mathbb{R}_+^{K\times K}$ is the diagonal hyperedge degree matrix, whose components are the weighted degree of each hyperedge, $[D_E]_{kk} = \delta(e_k)$.
- $R \in \mathbb{R}_+^{N\times K}$ is the vertex-weights matrix, whose components $R_{ik} = \gamma_{e_k}(v_i)$.

We can now define the combinatorial Laplacian matrix as[35]

$$L^H = \Pi - \frac{\Pi P + P^T\Pi}{2}, \tag{2}$$

where $\Pi_{ii} = \pi_i$ is a diagonal matrix and π_i is the stationary distribution (the left eigenvector of P, i.e., $\pi P = \pi$). This Laplacian matrix was originally defined in ref. 35 and is based on the Laplacian definition for directed graphs in ref. 47. Chitra and Raphael[35] argue that although the Laplacian matrix is a $N \times N$ symmetric object, it captures the essence of higher-order interactions. They demonstrate the existence of an EDVW hypergraph such that a random walk on that hypergraph is not equivalent to a random walk on its clique graph. Therefore, the argument is that such a random walk captures higher-order effects, as the lower-order effects are different. Note that the above definition of the Laplacian matrix is a symmetrization defined for any directed graph and is not particular to hypergraphs. Since the random walk on the EDVW hypergraph captures higher-order effects, this Laplacian definition also captures those effects while remaining symmetric and semi-positive definite.

Complementarily, since a diffusion process on this hypergraph depends on the algebraic connectivity of the Laplacian, we expect these weighting functions to capture the relationships between agents and tasks, reflecting the resilience of the team assignment. Formally, a diffusion process is defined as:

$$\frac{d\boldsymbol{x}(t)}{dt} = -L^H\boldsymbol{x}(t), \tag{3}$$

which solves as

$$\begin{aligned}\boldsymbol{x}(t) &= \exp\left(-L^H t\right)\boldsymbol{x}(0)\\ &= \sum_{i=1}^{N}\exp(-\mu_i t)\boldsymbol{v}_i\boldsymbol{v}_i^T\boldsymbol{x}(0)\\ &= \boldsymbol{v}_0\boldsymbol{v}_0^T\boldsymbol{x}(0) + \exp(-\mu_2 t)\boldsymbol{v}_i\boldsymbol{v}_i^T\boldsymbol{x}(0)\\ &\quad + \sum_{i=3}^{N}\exp(-\mu_i t)\boldsymbol{v}_i\boldsymbol{v}_i^T\boldsymbol{x}(0),\end{aligned} \tag{4}$$

where μ_i's are the eigenvalues of L^H and $\boldsymbol{v}_i$ are their associated eigenvectors, and $0 = \mu_1 < \mu_2 < \cdots \leq \mu_N$. Note that algebraic connectivity defines the timescale of our process. In other words, one can approximate the diffusion process by considering only the term dependent on μ_2 as the other terms decay faster and are (subsequently) negligible. So, the larger the algebraic connectivity, μ_2, the faster the diffusion. One might argue that the diffusion process described in Eq. (3) may not be the most appropriate model for the diffusion of information since it assumes conservation. However, it has been shown that the Laplacian matrix also captures important properties of the susceptible-infected-susceptible (SIS) epidemic spreading in a graph[48], which is not conservative. Van Mieghem[48] proposes an analogous approach to describe the spread of disease, where the eigenvalues and eigenvectors of the Laplacian matrix account for the pattern of links between individuals.

Problem definition. We consider the task assignment problem with N agents in the set $\mathcal{N} = \{1, \ldots, N\}$. Each agent will be assigned to one or more tasks in the set $\mathcal{K} = \{1, \ldots, K\}$. A task requires E_k units of "energy" to complete, which can be time, money, or other resources. Each agent has a total of B_i units of energy, which is allocated to a set of tasks. Task k is assigned to agent i with weight $\mathcal{B}_{ik}$, i.e., agent i will spend $\mathcal{B}_{ik}$ units of energy to complete task k. We assume that $\mathcal{B}_{ik}$ are integers. The matrix $\mathcal{B} = (\mathcal{B}_{ik})_{i\in\mathcal{N},k\in\mathcal{K}} \in \mathfrak{B} = \mathbb{N}^{N\times K}$ represents the assignments. We also define the binary matrix $\boldsymbol{X} \in \mathcal{X} = \{0, 1\}^{N\times K}$, whose elements $\boldsymbol{x}_{ik} = 1$ if agent i is assigned to task k with positive weight and $\boldsymbol{x}_{ik} = 0$ otherwise. The

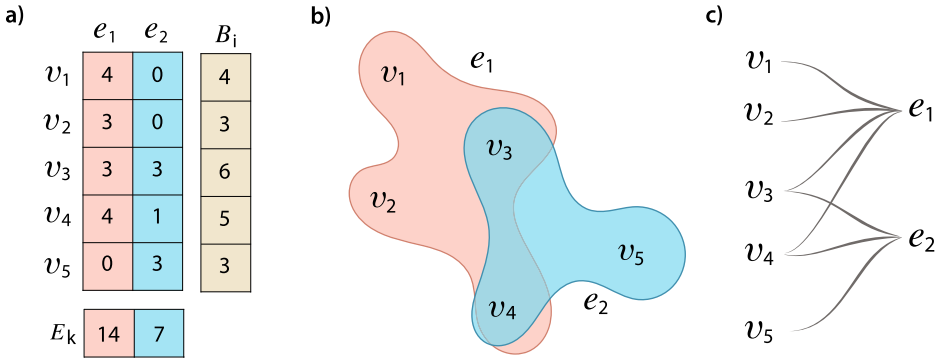

Fig. 1 | Graphical representation of the task-assignment problem. In (**a**), the task assignment is represented by $\mathcal{B}$ as well as an exemplary case of budgets, B_i's, and energies, E_k's, in (**b**) and (**c**), the hypergraph and bipartite representations of the same task assignment. Notice that the hypergraph and bipartite representations are equivalent. See Section "Mapping The Problem as a Hypergraph," for the complete definition of the objects in the figure.

total energy units of the agents are denoted by $B = \sum_{i\in\mathcal{N}} B_i$. We must have $B \geq \sum_{k\in\mathcal{K}} E_k$ to have a feasible solution. We assume that agents pay a cost $f(\mathcal{B})$ for the task assignment $\mathcal{B}$. Our goal is to choose an assignment $\mathcal{B}$ that minimizes the cost $f(\mathcal{B})$. Thus, the optimization problem can be formalized as

$$\min_{\mathcal{B}\in\mathfrak{B}} f(\mathcal{B}) \tag{4a}$$

$$s.t. \sum_{i\in\mathcal{N}} \mathcal{B}_{ik} \geq E_k \; \forall k \in \mathcal{K} \tag{4b}$$

$$\sum_{k\in\mathcal{K}} \mathcal{B}_{ik} \leq B_i \; \forall i \in \mathcal{N} \tag{4c}$$

Thus, we want to optimize the resilience of the final team assignment given the set of constraints in Eqs. (4b) and (4c). That is, we want to choose a cost function $f(\mathcal{B})$ in Eq. (4a) to capture the resilience of the final configuration. We will show later that the negative value of the algebraic connectivity of the hypergraph associated with the assignment $\mathcal{B}$ can be an appropriate cost function to provide the resilience we are looking for in the final assignment.

Problem formulation. The problem formulated in previous section can be mapped into an edge-dependent vertex-weighted hypergraph, where we weigh both the hyperedges and the nodes within each hyperedge[35]. Our optimization problem can be mapped to the EDVW hypergraph $\mathcal{H} = \{\mathcal{N}, \mathcal{E}, \omega, \gamma\}$, where $\mathcal{N}$ is the set of agents, $\mathcal{E}$ is the set of tasks (note that $|\mathcal{E}| = |\mathcal{K}|$ and the only difference between the sets $\mathcal{E}$ and $\mathcal{K}$ is the nature of the element in them). The weighting functions can be defined arbitrarily. Here, we propose to weigh the importance of a task as the energy required to complete it, i.e., $\omega(e_k) = E_k$. Also, the importance of an agent within a task is assumed to be the energy the agent spends in that task, $\gamma(v_i, e_k) = \mathcal{B}_{ik}$. Figure 1 (a) and (b) show this mapping graphically. Finally, we propose to maximize the algebraic connectivity of the Laplacian matrix in Eq. (2), i.e., μ_2, the second smallest eigenvalue of L^H.

The main advantage of formulating our problem as a hypergraph is that we can use the concepts of robustness and diffusion. By maximizing algebraic connectivity, we expect to make the hypergraph more resistant to attack and allow for faster diffusion processes. In practice, from the outset, we hope that the failure of a node or task will have minimal impact. From the second, we expect the flow of information between agents to be as fast as possible. Note that from a TFP perspective, these are some of the desirable features for a solution. In[49], the authors suggest that desirable features for TFPs include (i) reducing communication costs, (ii) being resilient with respect to the removal of an agent, (iii) reducing personnel costs, (iv) balancing workloads, and (v) incorporating unique experts, skills, and leaders. We note that our proposed mapping focuses on a robust team assignment. We refer the reader to Supplementary Note 1 for additional discussion on the team formation problem. However, algebraic connectivity maximization also reduces the timescale of the diffusion process, suggesting that communication costs are reduced as well. Moreover, with respect to personnel costs and workload balancing, these features are incorporated into our method through the constraints. Thus, one can simply restrict the space of solutions to those that satisfy a given set of personnel costs and workload balance. We also note that we did not include different expertise and leaders in our formulation. In other words, all agents in the system can perform the tasks equally well. It should be noted that this assumption may be reasonable in some scenarios. Examples include the assignment of tasks to artificial agents, especially teams of robots[1,5,50,51].

Here, we focus on connected hypergraphs. The reason for this choice is twofold. First, Laplacian matrices are semi-positive definite, so the multiplicity of zero eigenvalues is equal to the number of connected components in the hypergraph. Thus, if we optimize the algebraic connectivity in a hypergraph with multiple connected components, we can optimize only one component and neglect the others. Second, from an application point of view, we want to increase communication between agents. In this case, we need to ensure that there is a path between any two agents.

As an alternative representation, we can describe any hypergraph using a bipartite graph where both agents and tasks are represented as nodes in this graph. The adjacency matrix of this graph is given as

$$A^B = \begin{bmatrix} \mathbf{0} & (W \circ \chi) \\ (\mathcal{B} \circ \chi)^T & \mathbf{0} \end{bmatrix},$$

where $\circ$ is the Hadamard product, and χ is the incidence matrix (i.e., $\chi_{ij} = 1$ if node v_i belongs to the hyperedge e_j and $\chi_{ij} = 0$ otherwise). The first N nodes represent the nodes (agents) in the hypergraph, while the remaining K nodes represent the hyperedges (tasks). Thus, we can define the probability transition matrix as $P^B = \left(D^B\right)^{-1} A^B$, where D^B is a diagonal matrix whose diagonal elements are $D^B_{ii} = \sum_j A^B_{ij}$, and the Laplacian can be defined using the same formulation as in Eq. (2). Although the bipartite graph fully describes the hypergraph, the random walk in this structure alternates between tasks and agents. For a detailed description of the Laplacian bipartite matrix, we refer the reader to Supplementary Note 3.

Analysis on small hypergraphs. To gain more insight into the behavior of optimizing algebraic connectivity, we focus on small hypergraphs where we can study the whole set of possible hypergraphs. We focus on hypergraphs with $N = 5$ and $K = 3$. We generate all possible hypergraphs with a single connected component, and the minimum cardinality is equal to or greater than two. We set $\mathcal{B}_{ik} = 1$ for all assignments in the generated hypergraph and $\mathcal{B}_{ik} = 0$ otherwise. The budgets and energies are defined as $B_i = \sum_k \mathcal{B}_{ik}$ and $E_k = \sum_i \mathcal{B}_{ik}$ to make the process as unconstrained as possible. We compute the algebraic connectivity for all these cases. In Fig. 2, we show the highest and the lowest algebraic connectivity and two intermediate cases. In Fig. 2a–d, we show the graphically transposed incidence matrix, where the rows represent the hyperedges (tasks) and the columns represent the nodes (agents). In

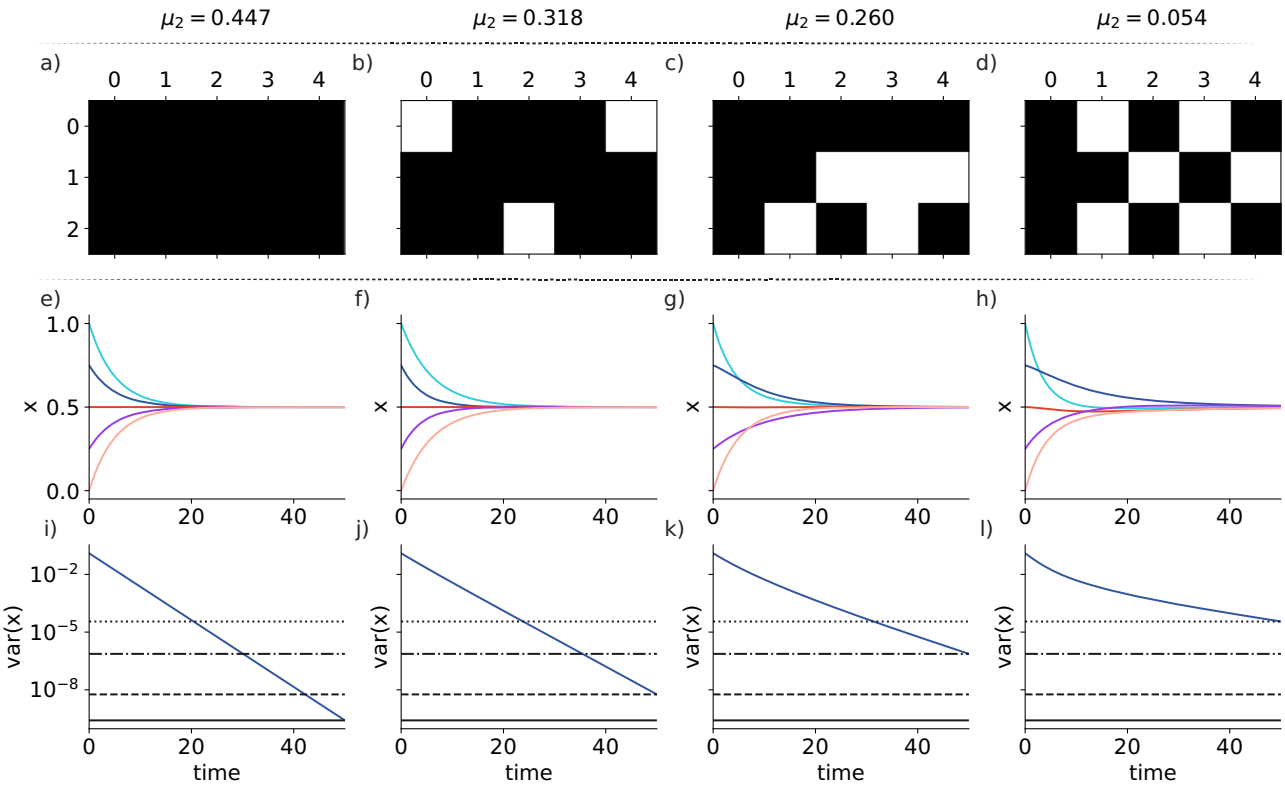

Fig. 2 | Examples of small hypergraphs. From left to right, in decreasing order of algebraic connectivity. From (**a**) to (**d**), the graphical representation of the transposed incidence matrix is used for visualization, where the rows represent the hyperedges (tasks) and the columns represent the nodes (agents). From (**e**) to (**h**), we show an example of the diffusion process obtained by solving Eq. (3) using L^H. Here, for visualization purposes, the initial condition is $\boldsymbol{x}(0) = [1.0, 0.75, 0.5, 0.0]^T$. All processes start with the same initial condition, $\mathcal{B}_{ik} = 1$ for all assignments (see incidence matrices), $B_i = \sum_k \mathcal{B}_{ik}$, and $E_k = \sum_i \mathcal{B}_{ik}$. From (i) to (l), we show the variance of the state x as a function of time. The visual difference between (**e**) and (**f**) is small, especially between the middle curves (blue, red, and purple). However, the curves at the extremes (light blue and light red) suggest that reaching the steady state takes slightly longer than in Fig. 2f. Panels (**i**)–(**l**) show the variance of the states x. The horizontal, solid, dashed, dash-dot, and dotted lines represent the final diffusion variances for each hypergraph.

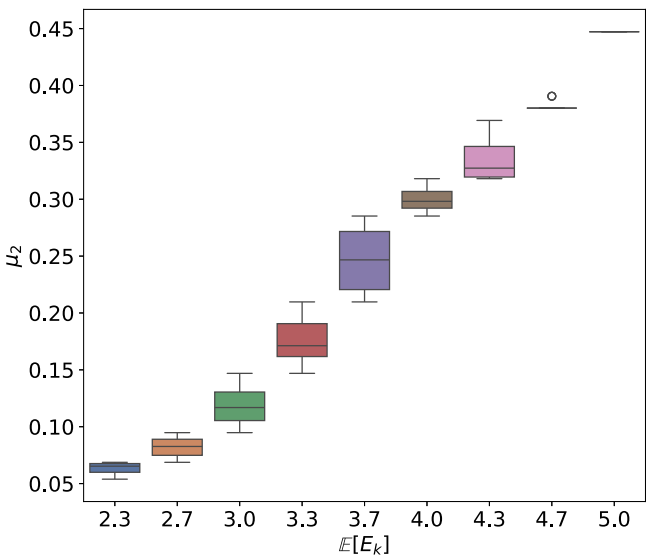

Fig. 3 | Algebraic connectivity distribution as a function of the constraints for all connected hypergraphs with $N = 5$ and $K = 3$. The figure shows a box plot of the algebraic connectivity for all connected small hypergraphs with $N = 5$ and $K = 3$ versus the average E_k. The box shows the quartiles, and the whiskers show the rest of the distribution. Outliers are defined as points above or below 1.5 times the interquartile range. The fewer constraints imposed, the higher the algebraic connectivity. Colors for visualization purposes. We refer the reader to Fig. S1 in the Supplementary Note 2 for more on the influence of constraints on a hypergraph's algebraic connectivity.

Fig. 2e–h, we show the behavior of a diffusion process, obtained by solving Eq. (3) using L^H for the studied hypergraphs. All hypergraphs have the same initial condition.

Figure 2 shows that the higher the algebraic connectivity, the faster the diffusion. We note that the visual difference between Fig. 2e, f is minimal, particularly between the middle curves (blue, red, and purple). However, note that the light blue and light red curves reach the steady state slightly later than in Fig. 2f. We emphasize that this analysis is based on the assumption that the process can be approximated by an exponential term depending on algebraic connectivity. Implicitly, we assume that the gap between μ_2 and μ_3 is large enough to allow us to neglect the remaining terms. The comparison between Fig. 2e and the others, especially (g) and (h), is clear because the timescale in (a) is significantly smaller. Notice also that there might be a dependence on the initial condition, which should be less important as we increase the system size. Finally, in Fig. 2i, j, k, we show the variance of $\boldsymbol{x}$ as a function of time. From these figures, we can see that the higher the algebraic connectivity, the faster the diffusion (i.e., it converges faster to the steady state). The analysis of this figure also suggests that unconstrained optimization of the algebraic connectivity favors denser hypergraphs. Nevertheless, we find that density does not fully explain intermediate cases. For example, Fig. 2c is less dense than Fig. 2d but has higher algebraic connectivity. We note that the constrained problem will limit the space of possible solutions, providing an opposing force to the expansion favored by maximizing algebraic connectivity. In other words, we expect to be closer to the intermediate cases than to the bounded cases (Fig. 2a, d).

Figure 3 shows how algebraic connectivity behaves as a function of constraints. The figure depicts the box plot of the algebraic connectivity versus the average value of E_k. Notice that each box is computed using a

different number of hypergraphs due to the combinatorial nature of this experiment. In this experiment, B_i and E_k are perfectly correlated as we assume that all B_{ik} are equal to one. Therefore, analyzing μ_2 as a function of one of the constraints is sufficient. As the figure shows, the fewer constraints imposed, the higher the algebraic connectivity. This behavior is not linear, but it is increasing. Furthermore, we observe that, for intermediate values of $\mathbb{E}[E_k]$, the distribution of algebraic connectivity exhibits greater variance.

Team assignment swapping. We want to understand the role of some representative small structures. So, we design a hypergraph rewiring process and a set of experiments to investigate how different structures change the algebraic connectivity under an assignment-swapping setting. We start with a set of N_C isolated communities, within which m_{c_i} hyperedges are shared by the corresponding n_{c_i} community members, and there is no communication between them.

We design algorithms that create connections between communities with different emphasis through assignment swapping while preserving edge and node weights. We assume that the initial collaboration setup satisfies the set of constraints. Since the assignment-swapping operations preserve the weight of each node and hyperedge throughout the rewiring process, each simulated hypergraph obtained through assignment swapping will satisfy the same set of constraints. Specifically, in the assignment-swapping algorithms, we fix the number of inter-community assignment swaps at $n_{swap} = N_c - 1$ for all rewiring methods. Our four rewiring methods are:

- Connected by one node: A randomly selected centroid node connects all communities through assignment swapping. In each swap operation, the centroid node exchanges one of its hyperedge participations with a randomly chosen node from a different community, ensuring that the centroid node participates in exactly one hyperedge per community. In terms of collaboration, this means that the centroid agent exchanges task assignments to work on one task in each research group, creating a hub-like structure.
- Connected by one hyperedge: A randomly selected centroid hyperedge connects all communities through assignment swapping. The algorithm moves one node from each non-centroid community into the centroid hyperedge, while simultaneously redistributing the nodes originally in the centroid hyperedge to replace the removed nodes in each community-specific hyperedge. This process ensures that the centroid hyperedge contains exactly one node from each community. In collaboration terms, the centroid task becomes a cross-community project with one agent from each research group, while the original agents from the centroid task are exchanged to work on tasks in the other communities.
- Head-to-tail connection: Communities are connected sequentially by swapping nodes between adjacent communities. For each pair of adjacent communities i and $i+1$, the algorithm selects one hyperedge from each community and swaps one node between them, creating a linear connection pattern.
- Connected by random swaps: Communities are connected through random assignment swapping between pairs of communities. In each swap operation, the algorithm randomly selects two communities, randomly chooses one hyperedge from each of these communities, and then randomly selects one node from each hyperedge to exchange with the other. This process is repeated for n_{swap} operations to create random inter-community connections.

Notice that all four rewiring methods preserve the weights of the hyperedges and nodes by design, as each swap operation exchanges node positions between hyperedges while maintaining each agent's total budget allocation and each task's total energy assignment. Although this experiment uses binary participation, the same swapping principle would preserve agent budget constraints and task energy requirements in scenarios with weighted resource allocations.

To provide an illustrative analogy for this process in terms of research collaboration, consider each community as a research lab or a collaboration project. Initially, information was contained and could only flow freely within each community. Collaboration occurs only within the communities and rarely between the labs. Information diffusion between communities could occur after the introduction of system-wide information flows while satisfying the same set of constraints on agents' budget and task energy requirements by preserving node and edge weights. Analogously, inter-group collaborations or visiting research opportunities connect different communities and contribute to the flow of information, knowledge, and skills. Note that in the hypergraph setting, additional information flow could be achieved without adding additional nodes or hyperedges.

Figure 4 shows a graphical example of the different types of hypergraphs analyzed. Given the initial setting, additional information flow could be facilitated by inter-community assignment swaps, which introduce collaboration between members of different communities. The more inter-community assignment swaps, the better the information diffusion between communities. Therefore, to fairly compare the swapping processes with different connection patterns and to evaluate their impact on algebraic connectivity, the number of inter-community assignment swaps is constant for each rewiring type.

For simplicity, we let N_c be the number of communities in $\mathcal{H}$, where n_i is the number of nodes in the community i, and m_i is the number of hyperedges in the community i. In a hub-like hypergraph connected by a single node, a centroid node is chosen to exchange with a random node in each of the other communities, and the end result would be a graph where all communities are connected through the centroid node as it participates in an edge in each community. Similarly, we can define a hub-like hypergraph connected by a single hyperedge. In this case, a centroid edge is chosen, and each community sends a node to join the centroid edge. Alternatively, a head-to-tail hypergraph can be formed by connections between each pair of consecutive communities. Finally, we also considered the case of random swaps, where connections between communities are formed by randomly swapping assignments between communities.

Figure 5 shows the behavior of the algebraic connectivity as a function of the number of nodes. The hub-like hypergraph, i.e., connection through a node or an edge, consistently outperforms the other connection schemes. Note that in this setting, the connection through an edge and the connection through a node are identical due to the graph isomorphism to the hypergraph transpose. The linear head-to-tail hypergraph has the smallest algebraic connectivity. The randomized connection outperforms the linear head-to-tail hypergraph on average. We note that we observe a positive correlation between the algebraic connectivity and the average shortest distance. When the average degree increases, we expect (1) the algebraic connectivity to increase and (2) the shortest paths to decrease. We also observe a power-law relationship between the algebraic connectivity and the size of the hypergraphs, where

$$\mu_2 \sim N_c^{-a}, a > 0 \text{ and } a_{head2tail} > a_{random} > a_{oneedge} = a_{onenode}.$$

The number of introduced connections between communities is the same for graphs of the same size but differently connected for a fair algebraic connectivity comparison.

We should note that the results in Fig. 5 also show an unusually high value for the exponents. This suggests that finite size effects play a role. However, due to the computational cost, we are unable to increase the system size to analyze large system sizes ($> 10^3$). Therefore, we decided to focus on small systems for this experiment. We should also note that the order observed between the algebraic connectivity for different systems is the main result of this analysis.

For most of the collaboration networks, we expect both the average and the maximum author budget and task energy requirements to be much smaller than the number of tasks or authors, i.e., $\mathbb{E}[\Omega]$, $max(\Omega)$, $\mathbb{E}[\Gamma]$, $max(\Gamma) \ll |\mathcal{K}|$ and $\mathbb{E}[\Omega]$, $max(\Omega)$, $\mathbb{E}[\Gamma]$, $max(\Gamma) \ll |\mathcal{N}|$, which means that under the constraints of the author's budget and task

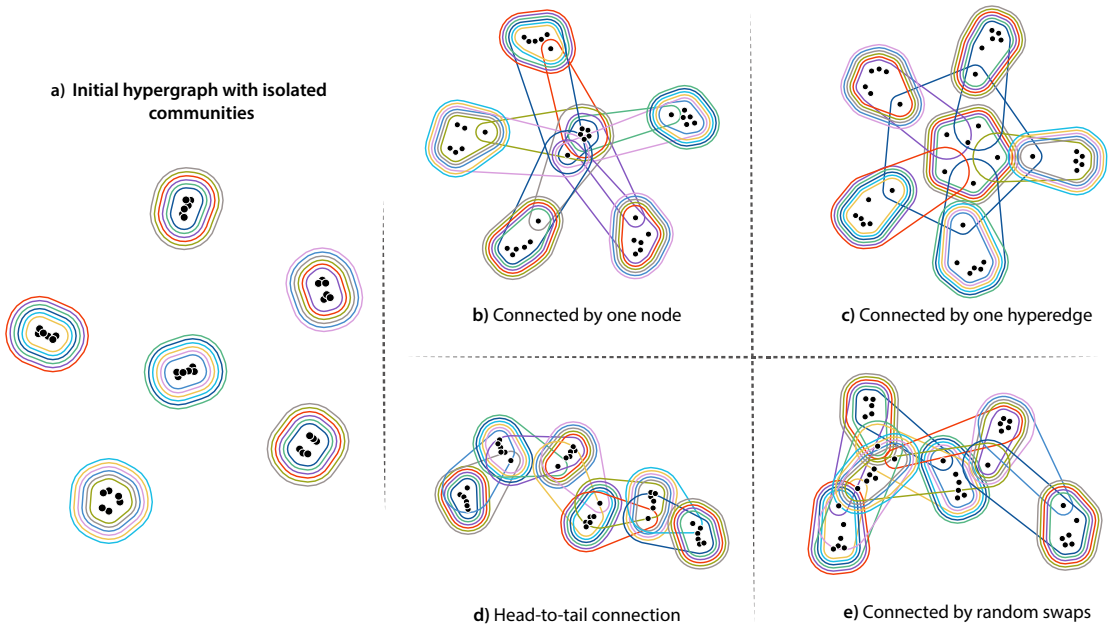

Fig. 4 | Example of hypergraph structures obtained from assignment swapping. Hypergraph assignment swapping with different structures, focusing on preserving constraints on agent budgets and task requirements. In (**a**), the initial hypergraph with the isolated communities, while in (**b**)–(**e**), we show the connection by one node, connection by one hyperedge, a head-to-tail connection scheme, and connection by random swaps, respectively. In this Example, we start with a set of isolated communities, each formed by 6 nodes sharing 6 hyperedges, i.e., 6 agents sharing 6 tasks. We introduce different connections to these isolated communities with different structures to explore properties favorable to the algebraic connectivity function. For simplicity, we let N_c be the number of communities in $\mathcal{H}$, where n_i is the number of nodes in community i, and m_i is the number of hyperedges in community i.

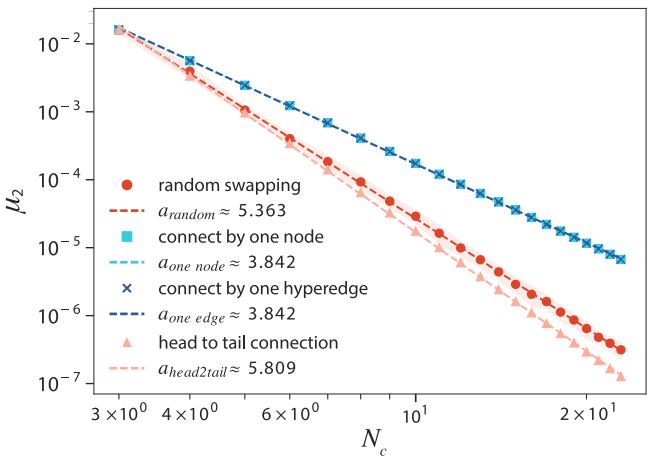

Fig. 5 | Hypergraph assignment swapping on isolated communities. Here, we tested four different systems: (i) random, (ii) connected by one node, (iii) connected by one hyperedge, and (iv) head to tail (see Fig. 4). We notice that the algebraic connectivity scales with the system size as $\mu_2 \sim N_c^{-a}$, where, in this example, $N_c = n_{c_i} = m_{c_i}, \forall c_i \in$ the set of communities, $N = n_c \times N_c = {N_c}^2, K = m_c \times N_c = {N_c}^2$. This figure is the result of 30 independent repetitions.

requirements, the collaborative network that optimizes algebraic connectivity is not expected to be centralized by a node or an edge, and is most likely to have a decentralized structure.

The analysis in this section suggests that head-to-tail structures perform worse than centralized structures (i.e., those centralized by nodes or hyperedges). This demonstrates that, even when weights are homogeneous, heterogeneity in structure can play a significant role in algebraic connectivity. This also indicates that understanding the properties that lead to higher or lower algebraic connectivity is non-trivial because the configuration space increases exponentially with the number of nodes, hyperedges, and weights.

Results from experiments on real-world data

Optimizing the algebraic connectivity. In this section, we show our results on how we managed to optimize the algebraic connectivity of the hypergraphs extracted from real datasets (see subection "Datasets" in the "Methods" Section). In Fig. 6, we present the algebraic connectivity gain, $Gain = \frac{\mu_2^{Optimized}}{\mu_2^{Real}}$, for the three optimization methods considered, the constrained simulated annealing (CSA) for the hypergraph and bipartite formulations, and the greedy approach (see the "Methods" section for the optimization methods details). Here, we consider the APS dataset with different extracted hypergraphs associated with 2-year periods from 1993 to 2021. Note that all approaches optimize the algebraic connectivity of the real collaboration hypergraph, as shown by the dashed line in Fig. 6. Notably, the CSA algorithm also provides a significantly better solution when compared to the greedy approach, where the algebraic connectivity in the optimized hypergraphs is between ten and five hundred times higher than the original hypergraph. In addition, in Fig. 6, we also show the comparison between the CSA approach using the hypergraph and the bipartite representation. Although they show approximately similar gains, we note that the comparison between the two methods is not straightforward because they represent different objects. The purpose of reporting such a comparison is to show that both systems are optimized versions of the original data. Thus, the comparisons are fair in terms of our quality measures.

We note that when applied to the 2002–2003 hypergraph, the greedy algorithm did not finish within the 500-h time limit. To improve computational efficiency, we have developed an adaptation of the greedy algorithm. The adapted approach uses random assignment of tasks to agents when the number of available agents exceeds 50 during the assignment process. The rationale behind this adaptation is that early-stage assignments have a relatively small impact on the final optimized

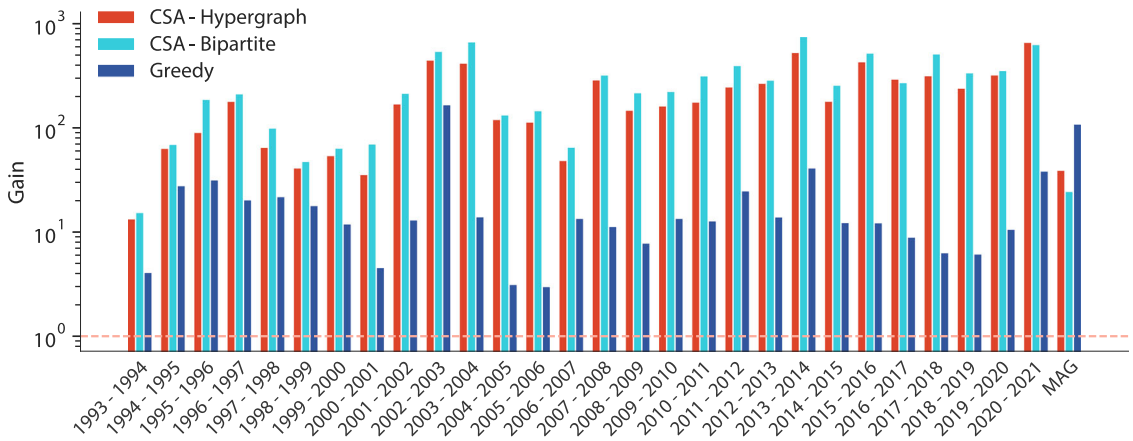

Fig. 6 | Algebraic connectivity gain for real hypergraphs. Algebraic connectivity gain for the hypergraphs extracted from the APS dataset between the years 1993–2021 and the MAG dataset, considering the original dataset (represented by the dashed pink line), the outputs of CSA for the hypergraph and bipartite formulations, and the output of the greedy approach. CSA consistently outperforms the greedy approach, except on the MAG dataset which has a significantly higher average number of papers per author and average number of authors per paper. While the bipartite representation has a slightly larger gain over the hypergraph representation, it is computationally more expensive because it has a larger Laplacian matrix. Table S2 in Supplementary Note 7 summarizes the properties of the hypergraphs extracted from the APS and MAG collaboration datasets. Fig. S5 in Supplementary Note 8 shows the evolution of these hypergraphs as they are optimized by CSA.

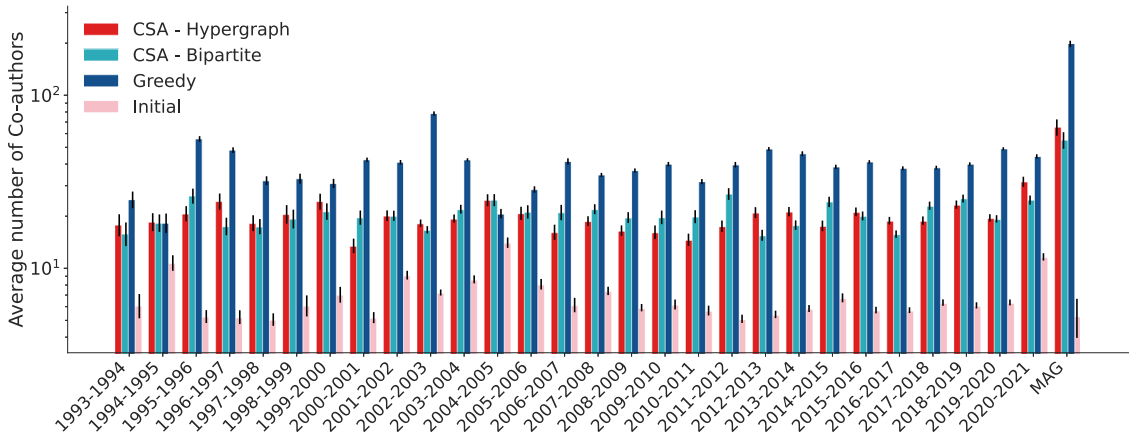

Fig. 7 | Average number of co-authors for real hypergraphs. Average number of co-authors for the extracted hypergraphs from the APS dataset between the years 1993–2021 and the MAG dataset, considering the original dataset, the CSA for the hypergraph and bipartite formulations, and the greedy approach. Error bars show 95% confidence intervals calculated from the distribution of co-author counts across all authors within each optimized solution. Specifically, for each incidence matrix (initial or optimized), we calculate the number of co-authors for every individual author, yielding a distribution across all authors in each collaboration dataset from which we compute the mean and its confidence interval. As our numerical analysis of the algebraic connectivity suggested, the optimized hypergraphs tend to increase connectivity and consequently increase the average number of co-authors compared to the initial hypergraphs. However, the CSA-Hypergraph method achieves this optimization while maintaining lower average number of co-authors than both the Greedy and CSA-Bipartite approaches. This suggests that CSA-Hypergraph can optimize algebraic connectivity with reduced communication cost.

algebraic connectivity. This small adjustment sufficiently reduces the computational cost.

As mentioned in the subsection "The Optimization Problem" in the "Mehtods" section, we are also interested in other quantities of our solutions, such as the average number of tasks assigned to an agent $\overline{T}$ and the average number of co-authors $\widehat{A}$. Specifically, $\widehat{A}$ is defined by first computing, for each author, the average number of co-authors across all of their papers. Then, this value is averaged over all authors. We use this metric instead of the average number of authors per paper, as it offers a proxy for the communication overhead each author has to deal with. Moreover, notice that the average number of authors per paper is directly related to the average number of papers per author due to the bipartite structure. This is, $N \times$ Average Number of Papers per Author $= K \times$ Average Number of Authors per Paper. Figures 7, 8, respectively, show these two quantities. We notice that the optimized versions are systematically denser, both in terms of the average number of co-authors and the average number of papers per author (tasks). Interestingly, the CSA method for the hypergraph also provides solutions with a lower average number of co-authors and a lower average number of papers per author compared to the greedy approach. This is a desirable feature, as we expect it to reduce communication costs.

Resilience against attacks. Figure 9 shows the patching cost (see subsection "Evaluation Measures," in the Methods section for its definition) of the optimized solution compared with the initial hypergraph under the four-node removal attack. The plots represent the patching costs of the hypergraphs corresponding to the APS dataset from 1993 to 2021. As the figure shows, the optimized solutions are more resilient to such attacks since the patching cost is almost always lower for the optimized solution. Similarly, we almost never have an unsuccessful patch for the optimized hypergraphs, while this is not the case for the original hypergraphs (results not shown). An "unsuccessful patch" refers to any case where at least one constraint remains unsatisfied after patching.

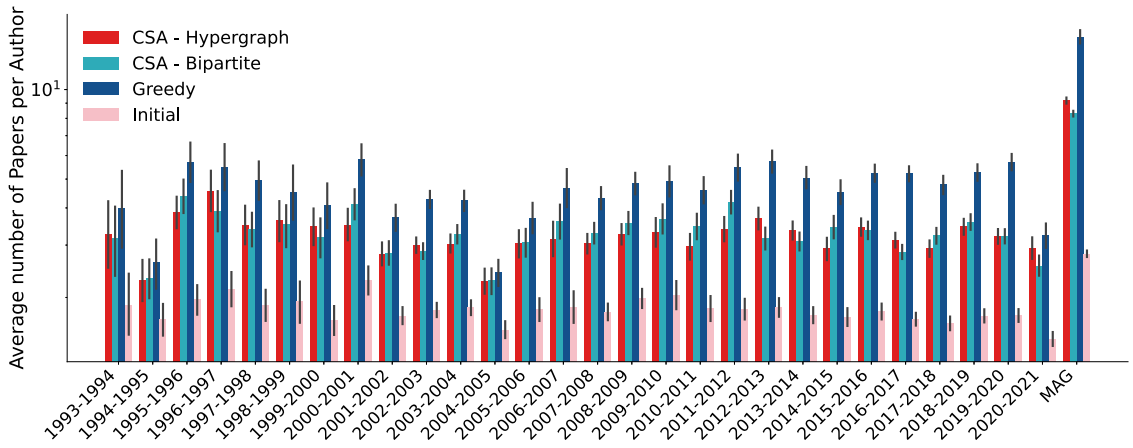

Fig. 8 | Average number of papers per author for real hypergraphs. Average number of papers per author of the extracted hypergraphs from the APS dataset between the years 1993–2021 and the MAG dataset, considering the original dataset, the CSA for the hypergraph and bipartite formulations, and the greedy approach. Error bars show 95% confidence intervals calculated from the distribution of paper counts across all authors within each solution. Specifically, for each incidence matrix (initial or optimized), we calculate the number of papers for every individual author, yielding a distribution across all authors in each collaboration dataset from which we compute the mean and its confidence interval. Similar to the co-authorship patterns in Fig. 7, optimizing for algebraic connectivity increases the average number of papers per author compared to the initial dataset. The CSA-Hypergraph method again achieves this optimization while maintaining a lower average number of papers per author than both the Greedy and CSA-Bipartite approaches. Thus, CSA-Hypergraph can optimize algebraic connectivity with reduced individual workload.

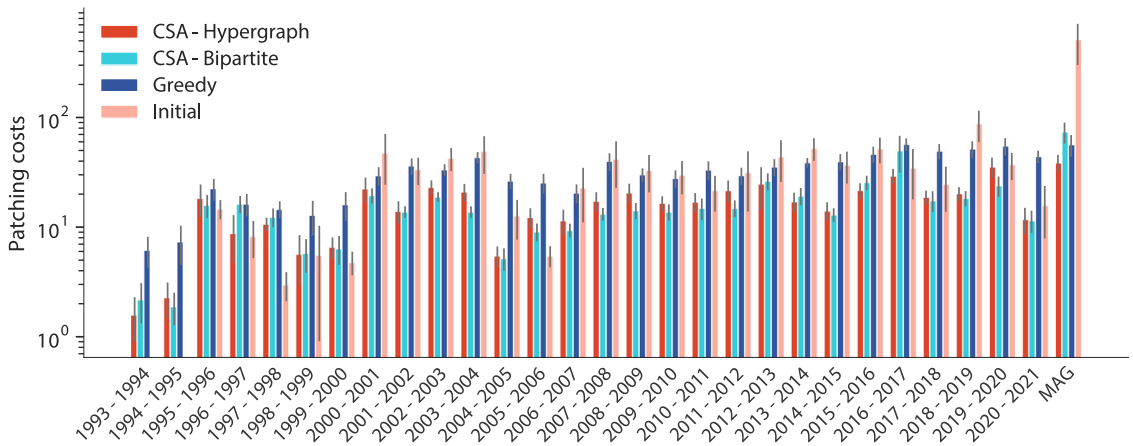

Fig. 9 | Patching costs after removing four nodes for the initial and optimized versions of the real hypergraphs. Patching costs after removing four nodes for the extracted hypergraphs from the APS dataset between the years 1993–2021 and the MAG dataset, considering the original dataset, the CSA for the hypergraph and bipartite formulations, and the greedy approach. The bars represent the average of $n_{exp} = 10$ runs, while the error bars represent the $\frac{\sigma_{exp}}{\sqrt{n_{exp}}}$. The patching costs are zero in the APS dataset for the hypergraphs of 1993–1994 and 1994–1995. Note that despite similar patching costs, the initial hypergraphs often have unsatisfied constraints (see Fig. 10). Thus, comparing only the patching costs without considering the unsatisfied constraints can be misleading.

"Unsatisfied constraints," reported in Fig. 9, are the total amount by which budget constraints are exceeded and task energy requirements are unmet. Moreover, note that any comparison between hypergraphs in which the constraints are met and those in which they are not is unfair (see also Fig. 10 and the discussion below). Finally, for comparison, we note that the greedy approach generally has a higher patching cost than the CSA approach. Both hypergraph and bipartite approaches using the CSA method produce similar results.

In Fig. 10, we show our main result, the sum of the unsatisfied constraints under the removal of four nodes and after patching the solution. We can see that in the CSA approach for the hypergraph, the removal of four nodes does not imply any violation of the constraints after patching. For the bipartite CSA case, the results are similar. The only exception is the hypergraph extracted from 2016 to 2017. However, this is not statistically sufficient to conclude that one approach is better than the other. Furthermore, we notice that the hypergraphs from 1993–1994 to 1994–1995 have zero unsatisfied constraints in the original data. This also explains why the original hypergraph has a lower patching cost than the optimized version. A similar analysis can be done for the 2005–2006 hypergraph. Moreover, we notice that the greedy approach performs well, as it does not leave unsatisfied constraints. We observe similar results when two nodes are removed. We refer the reader to Figs. S2 and S3 in Supplementary Note 5 for those results.

Finally, we report a similar analysis for the MAG dataset. Although it is difficult to directly compare these two datasets, they are similar in nature, and their main difference lies in how the data are collected, curated, and selected. For the latter, in the APS experiments, we fixed a 2-year time window and observed only the changes that occurred due to social factors, the number of researchers working in statistical mechanics, and their productivity. On the other hand, for the MAG dataset, the data were filtered using the keyword "hypergraph", and time is not a constraint. This factor may impose different constraints, as time is closely related to our notion of how many papers a researcher can produce. Note that this is translated in our models as our constraints, i.e., the energy agents can spend and the energy tasks that must be completed. Despite these differences, there are no

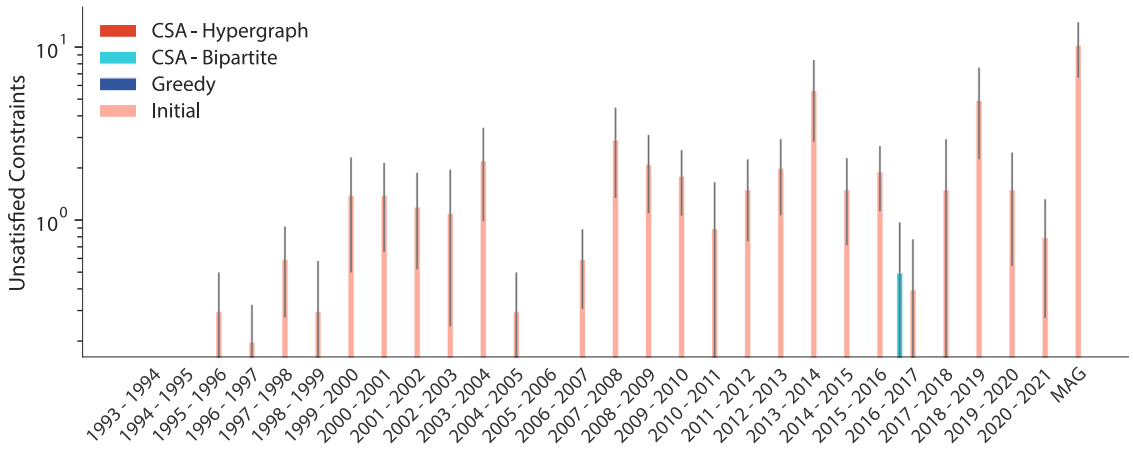

Fig. 10 | The sum of unsatisfied constraints after removing four nodes for the initial and optimized versions of the real hypergraphs. The sum of unsatisfied constraints after removing four nodes and patching the solution for the extracted hypergraphs from the APS dataset between the years 1993–2021 and the MAG dataset, considering the original dataset, the CSA for the hypergraph and bipartite formulations, and the greedy approach. The bars represent the average of $n_{exp} = 10$ runs, while the error bars represent the $\frac{\sigma_{exp}}{\sqrt{n_{exp}}}$. The unsatisfied constraints are zero for CSA hypergraph and bipartite (except for 2016–2017) and greedy for all of the cases (no bars are shown). Note that most initial hypergraphs will have unsatisfied constraints. However, only the CSA-bipartite-optimized hypergraph for 2016–2017 will have unsatisfied constraints.

methodological issues, and the comparisons are still reasonable due to their similar nature.

Regarding the MAG dataset (see the last set of bars in Figs. 6, 7, 8, 9, 10), the results are similar to those obtained for the APS, where patching costs and unsuccessful patches are significantly reduced. In the case of the MAG collaboration dataset, the algebraic connectivity gain is $Gain = \frac{\mu_2^{Optimized}}{\mu_2^{Real}} \approx 39$, which is similar to the *Gain* observed in the APS hypergraphs, as shown in Fig. 6. The same can be said for the average number of co-authors and the average number of papers per author (see Figs. 7 and 8). We observe a significant reduction in the patching cost of the optimized case, about 60% lower on average. However, we see a notable reduction in the number of unsuccessful patches and unsatisfied constraints since the numbers of unsuccessful patches and unsatisfied constraints were close to zero most of the time for the optimized solutions, as can be seen in Fig. 10. For additional results, we refer the reader to Figs. S2 and S3 in Supplementary Note 5.

Discussion

Our motivation to propose the use of algebraic connectivity in the team assignment problem arises from its applications in graph theory[36,37]. Thus, our main results concern the robustness of our assignment with respect to the patching costs and unsatisfied constraints, shown in Figs. 9, 10, respectively. From this analysis, we can see that the algebraic connectivity is indeed capturing the resilience features of the assignment. This is specifically evident when analyzing the unsatisfied constraints, Fig. 10, where the CSA approaches have no unsatisfied constraints after the four-node attack and patching the solution. Moreover, by optimizing the algebraic connectivity, we expect also to reduce the timescale of diffusion processes, as argued in Section "Results from experimental analysis." Although diffusion is just a mathematical model, in practice, we hope that such quantity can also be reflected in practical terms of information diffusion. Our main concern in this case was the average number of agents per task and the average number of co-authors. As shown in Figs. 7, 8, these two measures tend to increase as a consequence of optimizing the algebraic connectivity. This can be observed by noticing that in all the tested optimization methods, these quantities increased. However, our CSA hypergraph approach presented an increase that is lower than that of the greedy method. Thus, our methods are improving the robustness and diffusion at the same time but without unboundedly increasing the communication costs. Lastly, the algebraic connectivity alone might not be enough. Here, the constraints play a major role in our results. They reduce the space of feasible solutions and act as an opposing force, driving the solutions towards more practical solutions. We also remark that setting appropriate solutions must be key in real applications.

To our knowledge, the Laplacian matrix used in our problem definition is the only one in the literature that accounts for heterogeneity in an EDVW hypergraph. In[35], the authors argue that although this matrix is symmetric, it still captures higher-order interactions. The reasoning behind this argument is that the probability transition matrix P is generally asymmetric, whose asymmetries are induced by the context-dependent weights (i.e., different weights for the same node in different hyperedges), and thus represent a higher-order effect. Furthermore, a symmetric Laplacian is obtained using the definition in ref. 47. This technique takes into account the notion of circulation in a directed graph and can be applied to any directed graph. Thus, although it is still a symmetric projection, i.e., it can be interpreted as a graph, the higher-order structure should still be encoded in the weights of the Laplacian matrix. Note that a weighted bipartite representation of the hypergraph would also capture all the weights. However, in this case, we have two types of nodes in the bipartite representation: the nodes in the hypergraph (agents) and the nodes representing the hyperedges (tasks). By analyzing robustness using this representation, we assume that perturbations in both the nodes and the hyperedges are the same, which is against the nature of our application since we want to guarantee that we have enough resources to complete the tasks (see below a comparison with the bipartite representation).

Comparing the greedy with the original data, we notice an improvement in algebraic connectivity, Fig. 6. However, this comes at the cost of higher values for the average number of agents per task and the average number of co-authors; see Figs. 7, 8. When analyzing the patching costs, we observe inconsistent results of both an improvement and degradation of the solutions in terms of patching costs after the four-node attack experiment. However, similar to CSA, the greedy approach presents satisfying results in terms of unsatisfied constraints after node removal attacks and patching, see Fig. 10. From a computational point of view, the greedy approach is $O(N^4)$, while the computation of the algebraic connectivity alone is $O(N^3)$. Due to the stochastic nature of the CSA approaches, the comparisons may be perceived as unfair since one could run the CSA for an arbitrary number of iterations. However, the overall quality of the greedy solutions was not satisfactory. Finally, one could use mixed approaches where the greedy solution is used as the initial state for the CSA.

The CSA approaches for the hypergraph and the bipartite formulations seem to be statistically equivalent in our experiments, both in terms of algebraic connectivity gain, Fig. 6, and robustness, Figs. 9, 10. Thus, we have no evidence to advocate one approach over the other in terms of

assignments. However, the computational cost of computing the algebraic connectivity for the bipartite approach is significantly higher, $O((N+K)^3)$, versus $O(N^3)$ in the hypergraph case. Thus, we have evidence that the hypergraph approach should be preferred in practice.

The main contribution of our work is to map the team building problem as a hypergraph discovery problem and propose the use of algebraic connectivity as an optimization function to improve the resilience of formed teams. In Section "Results," we validate this claim by showing that hypergraphs with higher algebraic connectivity are more resilient to agent removal.

In this work, we assume that the agents are identical. Although this is beyond the scope of this work, we note that the incorporation of non-identical agents is possible by considering a matrix $\mathcal{C}_{ik} \in [0, 1]$ that will encode the efficiency with which agent i can perform the task k. In practice, these new constraints can be incorporated by rewriting the optimization problem in Eq. (4) as follows:

$$\max_{\mathcal{B}\in\mathfrak{B}} \ \mu_2(L^H) \tag{5a}$$

$$s.t. \sum_{i\in\mathcal{N}} \mathcal{B}_{ik}\mathcal{C}_{ik} \geq E_k \qquad \forall k \in \mathcal{K} \tag{5b}$$

$$\sum_{k\in\mathcal{K}} \mathcal{B}_{ik} \leq B_i \qquad \forall i \in \mathcal{N} \tag{5c}$$

Here, $\mathcal{B}$, E_k, and B_i are defined as in Eq. (7). The original problem is recovered when $\mathcal{C}_{ik} = 1$ for all agents and tasks. We can also set $\mathcal{C}_{ik} = 0$ to model the scenario in which agent i lacks the skill to perform task k.

A more sophisticated approach is to extend the matrix formulation to model agent specialization and heterogeneous task requirements by considering a set of ℓ_p distinct skills. In this model, we define a set of weighted incidence matrices, $\mathcal{B}^\ell$ with $\ell \in \mathcal{L} = \{1, 2, \cdots, \ell_p\}$, where each matrix corresponds to a different skill. The assignment of task k to agent i, $\mathcal{B}_{ik}$, is now further specified for each required skill, denoted by $\mathcal{B}^\ell_{ik}$. This variable represents the amount of skill effort ℓ that the agent i distributes to the task k.

These assignments are governed by skill-specific constraints. An agent's budget is no longer a constant value, but is instead specified for their capacity in each skill, denoted by B^ℓ_i. Similarly, a task's energy requirement is specified for each skill needed for its completion, denoted by E^ℓ_k. Finally, to ensure that each agent's total effort across all skills does not exceed their overall budget (B_i), we add a set of additional budget constraints.

Formally, this new optimization process can be defined as

$$\max_{\mathcal{B}\in\mathfrak{B}} \ \mu_2(L^H) \tag{6a}$$

$$s.t. \sum_{i\in\mathcal{N}} \mathcal{B}^\ell_{ik} \geq E^\ell_k \qquad \forall k \in \mathcal{K} \text{ and } \ell \in \mathcal{L} \tag{6b}$$

$$\sum_{k\in\mathcal{K}} \mathcal{B}^\ell_{ik} \leq B^\ell_i \qquad \forall i \in \mathcal{N} \text{ and } \ell \in \mathcal{L} \tag{6c}$$

$$\sum_{\ell=1}^{\ell_p} B^\ell_i \leq B_i. \tag{6d}$$

This formulation induces a multilayered hypergraph, where each skill corresponds to a separate layer. To optimize overall team resilience, this multilayered hypergraph can be projected onto a single hypergraph by adding the assignment matrices across all skill layers, $\mathcal{B} = \sum_\ell \mathcal{B}^\ell$. The objective is then to maximize the algebraic connectivity of this projected hypergraph.

Finally, we note that these are not the only approaches for including non-identical agents. These approaches highlight that the mapping can easily be generalized to other problems and sets of constraints by changing or adding new constraints. However, note that identical agents are the least restrictive case, thus providing the largest feasible solution set. In other words, more restrictive approaches imply a feasible set of solutions that is a subset of the feasible set for identical agents. Our results mainly concern the algebraic connectivity, Eq. (5a), whose properties should not change due to more restrictive constraints. In the latter case, we would expect the computational time to increase because of the reduction in the size of the feasible set.

The potential impact of our proposed approach on complex systems and higher-order network goes beyond team formation. Our approach is particularly helpful when studying classes of processes described by the EDVW Laplacian matrix. For example, He et al.[52] use a reweighted version of the EDVW Laplacian matrix to better cluster single-cell RNA sequencing data.

Our approach can generate optimized hypergraphs (that are resilient to node removal attacks), which can then be used for comparison with real systems. A straightforward application is to study the robustness of real and artificial systems, where our approach can provide a measure of how closely real and optimized systems align. A similar argument can be made for the study of random walks[35] and diffusion processes, as they are closely related. Another example is the possible generalization of coupled oscillators with edge and node weights. This application is not limited to specific problems and can be used to analyze different classes of problems, such as social, biological, or artificial systems. Moreover, the assumption of identical agents in such a general system is probably the most reasonable assumption.

While our work contributes to the fields of complex systems, we also foresee its impact on the study and design of hypergraph neural networks, as discussed in[35]. In this context, our results on the relationship between algebraic connectivity and robustness may help select neural networks less susceptible to adversarial node-level attacks. Furthermore, a better understanding of the mathematical properties of the EDVW Laplacian will be useful in studying and designing new hypergraph neural networks that accurately and efficiently encode higher-order interactions.

Conclusion

We propose a team assignment algorithm based on discoverey a hypergraph whose structure is optimized for resilience and diffusion. Specifically, we represent each agent's effort on a task as an edge-dependent, vertex-weighted hypergraph. Our method optimizes the algebraic connectivity of this hypergraph. In our formulation, we consider two constraints: the energy an agent can expend and the energy required to complete a task. These constraints reduce the feasible region and counterbalance the algebraic connectivity objective. We use constrained simulated annealing to find the optimal solution. We systematically evaluate all connected small hypergraphs with $N = 5$ agents and $K = 3$ tasks to validate algebraic connectivity. This experiment shows that algebraic connectivity favors densely connected hypergraphs. Additionally, we conduct a finite-size analysis considering four classes of hypergraphs: (i) connected by a node, (ii) connected by a hyperedge, (iii) head-to-tail (a.k.a. a chain of hyperedges), and (iv) randomly swapped hyperedges. This analysis confirms that head-to-tail structures scale worse (with a larger exponent) than centralized structures (i.e., those centralized by nodes or hyperedges). We use two scientific collaboration datasets to evaluate the robustness of our assignments using an attack-based evaluation, where nodes are removed. We estimate the cost of moving the assignment into the feasible region. We verify that our optimized hypergraphs are significantly more resilient to task completion than the original data. We also compare our constrained simulated annealing optimization approach with a greedy approach, finding that constrained simulated annealing yields a significant improvement in algebraic connectivity. Furthermore, we compare our hypergraph representation of the problem with a bipartite representation that captures similar properties. Specifically, the random walk defined on the hypergraph is equivalent to a two-step random walk on the bipartite graph. We find that both representations yield similar results in terms of attacks and unsuccessful patching costs. However, the computational cost of the bipartite representation is significantly higher than that of the hypergraph. We hope our results motivate further exploration of algebraic connectivity in the team assignment problem. Additionally, the proposed hypergraph representation and optimization of algebraic connectivity under constraints can be used to analyze other systems, such as financial systems in refs. 53,54.

Methods

The optimization problem

Motivated by applications in projects where tasks are interconnected and information diffusion between agents is an important factor, we propose to maximize the algebraic connectivity of L^H as a quality metric. The rationale behind this measure is that higher-order interactions are well captured by the Laplacian. More specifically, they are captured by the algebraic connectivity of L^H. At the same time, by maximizing the algebraic connectivity, we simultaneously optimize robustness and information flow. However, the algebraic connectivity alone could drive the optimization algorithm to solutions where the agents are overworked. This problem is avoided by constraining the solution space. Thus, we rewrite the problem in Eq. (4) as

$$\max_{\mathcal{B}\in\mathfrak{B}} \mu_2(L^H) \tag{7a}$$

$$s.t. \sum_{i\in\mathcal{N}} \mathcal{B}_{ik} \geq E_k \, \forall k \in \mathcal{K} \tag{7b}$$

$$\sum_{k\in\mathcal{K}} \mathcal{B}_{ik} \leq B_i \, \forall i \in \mathcal{N}. \tag{7c}$$

Note that L^H is a function of $\mathcal{B}$ and that only the first equation differs from the formal definition in Eq. (4).

Constrained simulated annealing

One of our solutions to the optimization problem (7) is constrained simulated annealing based on the penalty method. Specifically, we add a penalty function to our objective to penalize infeasible solutions generated by the simulated annealing optimization. The penalty function corresponding to the constraints in (7b) is $-\lambda_k(E_k - \sum_{i\in\mathcal{N}}\mathcal{B}_{ik})^+$ for all $k \in \mathcal{K}$, and the penalty function corresponding to the constraints in (7c) is $-\eta_i(\sum_{k\in\mathcal{K}}\mathcal{B}_{ik} - B_i)^+$ for all $i \in \mathcal{N}$, where $(\alpha)^+ = \alpha$ if $\alpha \geq 0$ and $(\alpha)^+ = 0$ if $\alpha < 0$. We keep the weights λ_k and γ_i fixed throughout the optimization. This method is implemented in Alg. 1.

We numerically observe that to maximize the algebraic connectivity of the hypergraph, it is best to use all of the agents' budgets. Therefore, we initialize $\boldsymbol{X}$ so that $\sum_{k\in\mathcal{K}} x_{ik} = B_i$. Then, to use simulated annealing to maximize algebraic connectivity, one only needs to swap the energy units in $\boldsymbol{X}$ (subtract and add energy units to and from x_{ik}'s) to generate new perturbations.

We also find that, because the search space is large compared to the feasible region (see Supplementary Note 6 for an analysis of the size and entropy of the solution space), adding the penalty function is insufficient to guide the simulated annealing algorithm to the feasible region. Therefore, we use a guided perturbation approach to push the samples toward the feasible region. The guided perturbation method is conducted by swapping the energy units in the incidence matrix so that we have a smaller number of constraint violations after each round. The swapping of energy units is performed according to two subroutines, which are chosen randomly (with adjustable probabilities). In the first subroutine, we randomly choose a task with extra energy, where tasks with more extra energy are more likely to be chosen, and one energy unit from its assigned agents is transferred to a task that needs more energy. The second task is also randomly selected, where the tasks with a greater energy shortage are more likely to be chosen. In the second subroutine, we randomly swap the energy units of the agents between two random tasks.

Notice that we cannot use the existing perturbation results for the undirected graph Laplacian (e.g.,[55]). The algebraic connectivity of an undirected graph behaves monotonically as new edges are added[55]. However, although the EDVW Laplacian matrix is symmetric, a perturbation in the hypergraph changes the random walk stationary distribution. This change is reflected in the Laplacian matrix and prevents us from using existing results such as the one in ref. 55. This illustrates the nontrivial behavior of the algebraic connectivity of the EDVW Laplacian matrix.

Constrained Simulated Annealing (CSA)

Input: Energy Requirements E, Budget Constraints B, The Initial Assignment $\mathcal{B}^0$. Optimization Parameters: T^0, a_c, T_{th}, t_{max}.

Set the temperature $T = T^0$.

Set $t = 0$.

while $T > T_{th}$ or $t < t_{max}$ **do**

Evaluate $\mathcal{B}^t$ and get penalty P^t, and $\widetilde{E}^t$ from Alg. 2.

Perturb $\mathcal{B}^t$ according to Alg. 3 and get $\mathcal{B}^{t+1}$.

$t = t + 1$

CSA: Assignment Evaluation

Input: $\mathcal{B}$, E, B, η, λ.

Output: P, $\widetilde{E}$.

Calculate the algebraic connectivity, e, for the assignment $\mathcal{B}$.

Calculate penalty function:

$P = e - \eta_i(\sum_{k\in\mathcal{K}}\mathcal{B}^t_{ik} - B_i)^+ - \lambda_k(E_k - \sum_{i\in\mathcal{N}}\mathcal{B}^t_{ik})^+$

Calculate constraint violations:

Set $\widetilde{E}_k$ = extra required energy for task k ($\widetilde{E}_k$ is negative if task k has more than enough energy assigned to it) **return** P and $\widetilde{E}$.

CSA: Assignment Perturbation

Input: $\mathcal{B}$, $\widetilde{E}$, N_s.

for $s = 1$: N_s **do**

Find tasks that require more energy, $H_p = \{k : \widetilde{E}_k > 0\}$.

Find tasks that have more than enough energy assigned to them, $H_n = \{k : \widetilde{E}_k < 0\}$.

if $H_p! = \varnothing$ and $H_n! = \varnothing$ **then**

Choose a task h_p from H_p with probabilities proportional to $\widetilde{E}$.

Choose a task h_n from H_n with probabilities proportional to $|\widetilde{E}|$.

Assign one energy unit of an agent assigned to task h_n to task h_p.

Update $\mathcal{B}$ and $\widetilde{E}$.

else

Choose two tasks randomly.

Choose one agent from each task randomly. Swap one of their energy units assigned to the chosen tasks with each other.

Update $\mathcal{B}$ and $\widetilde{E}$.

end if

end for

The computational complexity of Algorithm 1 is determined by the computation of the algebraic connectivity, which is the second smallest eigenvalue of the Laplacian matrix ($N \times N$). The complexity of such a computation is $O(N^3)$. In each round of the CSA algorithm, the algebraic connectivity is computed once. Since the total number of rounds is bounded from above by a constant number t_{max}, (see Algorithm 1), the total computational complexity of CSA is $O(t_{max}N^3)$. Producing a solution with a lower computational complexity is beyond the scope of this paper and is part of our future work.

Other factors. Although we optimize the algebraic connectivity of the hypergraphs to get robust solutions, there are some other important factors to consider when evaluating a particular solution. The factors we considered are the average number of tasks an agent is assigned to, $\bar{T}$, and the average number of teammates an agent has, $\widehat{A}$. It is not desirable to have solutions with large $\bar{T}$ and $\widehat{A}$ because coordination and collaboration become more difficult as these factors increase. Therefore, we evaluate our solutions from this perspective and try to generate solutions with controlled levels of $\bar{T}$ and $\widehat{A}$. We try two approaches to optimize the algebraic connectivity while having a controlled level of these quantities. The first approach is to penalize the objective function as these quantities increase. The second approach is to assign energy units to tasks in packs. That is, we can only assign nP energy units of each agent to a task, where P is the size of the energy pack and n is an integer.

Hyperparameter selection. There are several hyperparameters in Alg. 1 (Constrainted Simulted Annealing). In this section, we will explain the sensitivity of our algorithm to them and how one should choose their values. Some of the hyperparameters, such as the initial temperature (T_0) and the cooling schedule factor (a_c), are related to simulated annealing.

These two hyperparameters can be chosen similarly to any other simulated annealing algorithm. One must be careful not to choose very small values for them to allow exploration. On the other hand, very large values for these two parameters will cause the algorithm to produce subpar solutions. We have two other hyperparameters, T_{th} and t_{max}, which determine when the algorithm should stop. T_{th} is a lowerbound on temperature. That is, when the temperature is less than T_{th}, the algorithm stops. t_{max} is the upperbound on the number of iterations. Decreasing T_{th} and increasing t_{max} allows the algorithm to explore a larger portion of the solution space, increasing the chances of finding a good solution. Note that while more iterations can lead to better solutions, they also come with higher computational cost. We set T_{th} to $1 \times e^{-5}$ and $t_{max} \in [2000, 5000]$ in our experiments. In addition, we have some hyperparameters that act as penalty coefficients for the penalty functions that we add. We add two types of penalty functions to our objective function. The first penalty function is to penalize the infeasible solutions. The second penalty function is to control the two important factors of $\bar{T}$ (average number of tasks per agent), and $\widehat{A}$ (the average number of teammates per agent) during the optimization. The first penalty function does not play a major role in the optimization process because of the guided perturbation approach used. Regarding the second penalty function, with a higher coefficient, we get a smaller final algebraic connectivity, but also smaller $\bar{T}$ and $\widehat{A}$. One can adjust this coefficient depending on how much one can tolerate large values of these factors.

The exact values of all of the hyperparameters used to produce the results of this paper can be found in the configuration files in the paper's Github repository.

Alternative optimization methods. Although we use Constrained Simulated Annealing (Alg. 1) as the main optimization method in this study, we refer the reader to Supplementary Note 4 for a discussion of alternative optimization methods.

Baseline: the greedy approach

A greedy optimization approach was implemented as a baseline solution. Inspired by the results presented in Section "Results from experimental analysis," which show that more centralized systems are favored when optimizing algebraic connectivity, the greedy algorithm starts with an initial hypergraph assignment that connects all tasks with the minimum number of hubs, where the hubs correspond to the highest-budget agents. These hubs are then connected by adding shared tasks among these highest-budget agents.

The greedy optimization can be divided into 2 phases. In phase 1 (see Algorithm 4), we start from the centralized initial assignment and then assign agents to tasks by filling the tasks with the agents that could lead to the maximum increase in the objective function per unit input energy until each task is full. In phase 2 (see Algorithm 5), we further optimize the objective by using up all the energy left in the authors after the assignment given in phase 1. To do this, we start with the assignment computed by phase 1 that satisfies the task completion requirement. Then, analogous to phase 1, each agent is assigned to the tasks that result in the most increase in goal per unit input until the agent's budget is exhausted. If the total task energy requirements are equal to the total agent budgets, there would be no excess agent budget available if all task requirements were met. As a result, there would be no need for phase 2 operation.

Regarding the suitability of applying simulated annealing to the greedy approach, since a viable solution has already been obtained in phase 1, it could be specified in phase 2 of the optimization whether the greedy optimization should be performed by rejecting assignments that lead to a negative change in the objective function or accepting such assignments with a probability.

The computational complexity of the greedy approach is higher than that of the CSA algorithm we use. The reason is that in each round of assignment in the greedy approach, the algebraic connectivity order must be computed N times (the number of agents with available budget). The total number of rounds is of the order of the total number of tasks. Assuming that the number of tasks does not grow with N, the computational complexity of the greedy approach is $O(N^4)$.

Greedy Knapsack Phase 1: Task Fulfillment
Input E, B, h.
Output $\mathcal{B}$.
$h \in \mathbb{R}$: The energy packet size that each energy assignment must be multiples of.
Start with an empty initial assignment $\mathcal{B}^0$
while Tasks are not all fulfilled **do**
for each unfulfilled task i **do**
Potential energy spent by agent j on task i: $e_{ij} = min(B_j, E_i, h)$.
Compute change in the objective function per unit energy input by agent j when assigning e_{ij} to task i.
Assign task i to the agent that results in the maximum increase in objective function per unit input of energy.
Update assignment $\mathcal{B}$.
end for
Update the list of unfulfilled tasks.
end while
Greedy Knapsack Phase 2: Improve algebraic connectivity by using up agent energy
Input E, B, $\mathcal{B}_{phase1}$, h.
Continue with the assignment $\mathcal{B}_{phase1}$ optimized in phase 1
while Agent budget is not all used up
for each available agent j
Potential energy spent by agent j on task i: $e_{ij} = min(B_j, h)$.
Compute change in the objective function per unit energy input by agent j when assigning e_{ij} to task i.
Assign task i to the agent that results in the maximum increase in objective function per unit input of energy.
end for
Update the list of available agents.
end while

Datasets. We used two collaboration datasets in our experiments: the Microsoft Academic Graph (MAG)[56] and the American Physical Society (APS)[57]. The Microsoft Academic Graph (MAG) contains scientific publication records, citations, and other information. More information is described on MAG's website[58]. The Collaborative Archive & Data Research Environment (CADRE) project at Indiana University[59] provided MAG's raw data. From the MAG dataset, we filtered the papers with the word "hypergraph" in their title and extracted the giant connected component of the authorship hypergraph (authors as nodes and papers as hyperedges).

The American Physical Society (APS) dataset contains the basic metadata of all APS journal articles from 1993 to 2021. From the APS dataset, we considered one journal (Physical Review E (PRE)), divided the dataset into 2-year intervals, and extracted the giant connected component of the authorship hypergraphs. We then optimized each of the extracted hypergraphs using the CSA and the greedy approach (described in the following).

We define one unit of energy as a single page in a paper. We then create the matrix $\mathcal{B}$ for the hypergraphs as follows: Each row corresponds to a paper. The numbers in a given row are calculated by dividing the number of pages in a paper by the number of its authors. Here, we assume uniform contributions. The energy requirement of each paper is its page count, and each author's budget is the number of pages to which they have contributed. Thus, $\mathbb{E}[E_k]$ and $\mathbb{E}[B_i]$ refer to these energy and budget constraints, respectively. Note that these quantities differ from the average number of papers per author and authors per paper, which count the number of papers and authors without considering page number contributions.

Evaluation measures

We evaluate the resilience of our optimized hypergraphs against agent (i.e., node) removal attacks. An agent removal attack involves removing a set of

agents from the system, which results in their assigned tasks becoming incomplete. Our evaluation focuses on the ease (or feasibility) of patching the attacked hypergraph to restore it as a viable solution. Here, patching refers to adjusting the hyperedges to reestablish feasibility. This adjustment mimics a real-world scenario in which agents must collaborate to complete tasks: If an agent fails, the remaining agents step in to cover its assignments. The patching process assumes the following: agents assigned to incomplete tasks first attempt to cover for the removed agent. If they lack additional budget, they will reach out to other agents. However, they can only communicate with their teammates. If those teammates lack budget, they will continue asking their teammates, and so forth.

To evaluate this process, we use the following metric: the more hops it takes to replace a removed agent, the less resilient the original solution is to node removal attacks. We define patching cost based on the number of hops required to find replacements and whether a feasible solution is achieved. In addition, we measure unsuccessful constraints, which are the number of constraints still unsatisfied after patching. Thus, we use patching cost and unsuccessful constraints as our evaluation metrics, applying these measures to both the original hypergraphs derived from real datasets and the optimized ones produced by our algorithms.

Data availability

We refer the reader to[56] and[57] for more information on the data sets (e.g., how they were collected, any preprocessing performed at the source, and links to download the data).

Code availability

The code associated with this work is available at https://github.com/nasheydari/Task-Assignment-Hypergraph-Discovery.

Received: 26 April 2025; Accepted: 17 December 2025;
Published online: 08 January 2026

References

1. Juárez, J., Santos, C. P. & Brizuela, C. A. A comprehensive review and a taxonomy proposal of team formation problems. *ACM Comput. Surv.* **54**, 153–115333 (2021).
2. Kuhn, H. W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2**, 83–97 (1955).
3. Ahmed, F., Deb, K. & Jindal, A. Multi-objective optimization and decision making approaches to cricket team selection. *Appl. Soft Comput.* **13**, 402–414 (2013).
4. Okimoto T. et al. How to form a task-oriented robust team. In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '15, pp. 395-403. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2015).
5. Okimoto, T., Ribeiro, T., Bouchabou, D., & Inoue, K. Mission oriented robust multi-team formation and its application to robot rescue simulation. In: *International Joint Conference on Artificial Intelligence* (2016).
6. Demirović, E., Schwind, N., Okimoto, T., & Inoue, K. Recoverable team formation: Building teams resilient to change. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '18, pp. 1362-1370. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2018)
7. Kargar, M., Zihayat, M., & An, A. Finding affordable and collaborative teams from a network of experts, pp. 587–595. https://doi.org/10.1137/1.9781611972832.65 (2023).
8. Golshan, B., Lappas, T., & Terzi, E. Profit-maximizing cluster hires. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '14, pp. 1196–1205. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2623330.2623690 (2014).
9. Liu, Q., Luo, T., Tang, R., & Bressan, S. An efficient and truthful pricing mechanism for team formation in crowdsourcing markets. In: *2015 IEEE International Conference on Communications (ICC)*, pp. 567–572. https://doi.org/10.1109/ICC.2015.7248382 (2015).
10. ZAKARIAN, A. & KUSIAK, A. Forming teams: An analytical approach. *IIE Trans.* **31**, 85–97 (1999).
11. Fitzpatrick, E. & Askin, R. G. Forming effective worker teams with multi-functional skill requirements. *Comput. Ind. Eng.* **48**, 593–608 (2005).
12. Feng, B., Jiang, Z.-Z., Fan, Z.-P. & Fu, N. A method for member selection of cross-functional teams using the individual and collaborative performances. *Eur. J. Oper. Res.* **203**, 652–661 (2010).
13. Chen, S.-J. & Lin, L. Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. *IEEE Trans. Eng. Manag.* **51**, 111–124 (2004).
14. Kargar, M., & An, A. Discovering top-k teams of experts with/without a leader in social networks. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. CIKM '11, pp. 985-994. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2063576.2063718 (2011).
15. Lappas, T., Liu, K., & Terzi, E. Finding a team of experts in social networks. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09, pp. 467–476. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/1557019.1557074 (2009).
16. Iacopini, I., Petri, G., Barrat, A. & Latora, V. Simplicial models of social contagion. *Nat. Commun.* **10**, 1–9 (2019).
17. Jhun, B., Jo, M. & Kahng, B. Simplicial SIS model in scale-free uniform hypergraph. *J. Stat. Mech.: Theory Exp.* **2019**, 123207 (2019).
18. Arruda, G. F., Petri, G. & Moreno, Y. Social contagion models on hypergraphs. *Phys. Rev. Res.* **2**, 023032 (2020).
19. Battiston, F. et al. Networks beyond pairwise interactions: Structure and dynamics. *Phys. Rep.* **874**, 1–92 (2020).
20. Barrat, A., Arruda,G., Iacopini,I., & Moreno, Y. In: Battiston, F., Petri, G. (eds.) Social Contagion on Higher-Order Structures, pp. 329–346. Springer, Cham (2022) .
21. Battiston, F. et al. The physics of higher-order interactions in complex systems. *Nat. Phys.* **17**, 1093–1098 (2021).
22. Arruda, G., Petri, G., Rodriguez, P. M. & Moreno, Y. Multistability, intermittency, and hybrid transitions in social contagion models on hypergraphs. *Nat. Commun.* **14**, 1375 (2023).
23. Boccaletti, S. et al. The structure and dynamics of networks with higher order interactions. *Phys. Rep.* **1018**, 1–64 (2023).
24. Arruda, G., Aleta, A., & Moreno, Y. Contagion dynamics on higher-order networks. *Nat. Rev. Phys.* https://doi.org/10.1038/s42254-024-00733-0 (2024).
25. Alvarez-Rodriguez, U. et al. Evolutionary dynamics of higher-order interactions in social networks. *Nat. Hum. Behav.* **5**, 586–595 (2021).
26. Civilini, A., Sadekar, O., Battiston, F., Gómez-Gardeñes, J. & Latora, V. Explosive cooperation in social dilemmas on higher-order networks. *Phys. Rev. Lett.* **132**, 167401 (2024).
27. Bradde, S. & Bianconi, G. The percolation transition in correlated hypergraphs. *J. Stat. Mech. Theory Exp.* **2009**, 07028 (2009).
28. Bianconi, G. & Dorogovtsev, S. N. The theory of percolation on hypergraphs. *Phys. Rev. E* **109**, 014306 (2024).
29. Schaub, M. T., Benson, A. R., Horn, P., Lippner, G. & Jadbabaie, A. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Rev.* **62**, 353–391 (2020).
30. Di Gaetano, L., Carugno, G., Battiston, F. & Coghi, F. Dynamical fluctuations of random walks in higher-order networks. *Phys. Rev. Lett.* **133**, 107401 (2024).
31. Traversa, P., Arruda, G. F. & Moreno, Y. From unbiased to maximal-entropy random walks on hypergraphs. *Phys. Rev. E* **109**, 054309 (2024).
32. Millán, A. P., Torres, J. J. & Bianconi, G. Explosive higher-order kuramoto dynamics on simplicial complexes. *Phys. Rev. Lett.* **124**, 218301 (2020).

33. Ghorbanchian, R., Restrepo, J. G., Torres, J. J. & Bianconi, G. Higher-order simplicial synchronization of coupled topological signals. *Commun. Phys.* **4**, 120 (2021).
34. Zhang, Y., Lucas, M. & Battiston, F. Higher-order interactions shape collective dynamics differently in hypergraphs and simplicial complexes. *Nat. Commun.* **14**, 1605 (2023).
35. Chitra, U., & Raphael, B.J. Random walks on hypergraphs with edge-dependent vertex weights. In: Chaudhuri, K., Salakhutdinov, R. (eds.) *Proceedings of the 36th International Conference on Machine Learning*, ICML 2019, 9-15 June 2019, Long Beach, California, USA. Proceedings of Machine Learning Research, vol. 97, pp. 1172-1181. PMLR, Long Beach, California, USA (2019).
36. Jamakovic, A., & Van Mieghem, P. On the robustness of complex networks by using the algebraic connectivity. In: Das, A., Pung, H.K., Lee, F.B.S., Wong, L.W.C. (eds.) NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, pp. 183–194. Springer, Berlin, Heidelberg (2008).
37. Cozzo, E., Arruda, G. F., Rodrigues, F. A. & Moreno, Y. Layer degradation triggers an abrupt structural transition in multiplex networks. *Phys. Rev. E* **100**, 012313 (2019).
38. Feng, Y., You, H., Zhang, Z., Ji, R., & Gao, Y. Hypergraph neural networks. In: AAAI, pp. 3558-3565. https://doi.org/10.1609/aaai.v33i01.33013558 (2019).
39. Yadati, N. et al. HyperGCN: A New Method of Training Graph Convolutional Networks on Hypergraphs. Curran Associates Inc., Red Hook, NY, USA (2019).
40. Ji, S. et al. Dual channel hypergraph collaborative filtering. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '20, pp. 2020-2029. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3394486.3403253 (2020).
41. Dong, Y., Sawin, W., Bengio, Y. Hnhn: Hypergraph networks with hyperedge neurons. Graph Representation Learning and Beyond Workshop at ICML 2020 (2020).
42. Huang, J., & Yang, J. Unignn: A unified framework for graph and hypergraph neural networks. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, IJCAI-21 (2021).
43. Gao, Y., Feng, Y., Ji, S. & Ji, R. Hgnn+: General hypergraph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 3181–3199 (2023).
44. Li, M., Zhang, Y., Li, X., Zhang, Y., & Yin, B. Hypergraph transformer neural networks. *ACM Trans. Knowl. Discov. Data* **17**(5) https://doi.org/10.1145/3565028 (2023).
45. Jost, J. & Mulas, R. Hypergraph Laplace operators for chemical reaction networks. *Adv. Math.* **351**, 870–896 (2019).
46. Mulas, R. & Zhang, D. Spectral theory of Laplace operators on oriented hypergraphs. *Discret. Math.* **344**, 112372 (2021).
47. Chung, F. Laplacians and the cheeger inequality for directed graphs. *Ann. Combinatorics* **9**, 1–19 (2005).
48. Van Mieghem, P. Approximate formula and bounds for the time-varying susceptible-infected-susceptible prevalence in networks. *Phys. Rev. E* **93**, 052312 (2016).
49. Rehman, M. Z. et al. A novel state space reduction algorithm for team formation in social networks. *PLOS ONE* **16**, 1–18 (2021).
50. Liemhetcharat, S., & Veloso, M. Modeling and learning synergy for team formation with heterogeneous agents. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1. AAMAS '12, pp. 365-374. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2012).
51. Liemhetcharat, S. & Veloso, M. Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents. *Artif. Intell.* **208**, 41–65 (2014).
52. He, W., Bolnick, D. I., Scarpino, S. V. & Eliassi-Rad, T. Hypergraph representations of scrna-seq data for improved clustering with random walks. arXiv preprint arXiv:2501.11760v4 (2025).
53. Huang, X., Vodenska, I., Havlin, S. & Stanley, H. E. Cascading failures in bi-partite graphs: Model for systemic risk propagation. *Sci. Rep.* **3**, 1219 (2013).
54. Caccioli, F., Shrestha, M., Moore, C. & Farmer, J. D. Stability analysis of financial contagion due to overlapping portfolios. *J. Bank. Financ.* **46**, 233–245 (2014).
55. Fiedler, M. Algebraic connectivity of graphs. *Czechoslovak Math. J.* **23**, 298–305 (1973).
56. Sinha, A. et al. An overview of microsoft academic service (mas) and applications. In: *Proceedings of the 24th International Conference on World Wide Web*, pp. 243–246 (2015).
57. Society, A.P. APS data sets for research. https://journals.aps.org/datasets
58. MAG: Microsoft academic graph. https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/. [Online; accessed 13-October-2025] (2025).
59. Mabry, P. L. et al. CADRE: A collaborative, cloud-based solution for big bibliographic data research in academic libraries. *Front. Big Data* **3**, 42 (2020).

Acknowledgements

We thank M. Clarin for help with the figures. G.F.A. acknowledges Henrique Ferraz de Arruda for fruitful discussions and his help with the collaboration datasets. W.H., N.H., T.J., and T.E-R. acknowledge the support of the AccelNet-MultiNet program, a project of the National Science Foundation (Award #1927425 and #1927418) and also ARO MURI program (Award #W911NF-21-1- 0322). Y.M. was partially supported by the Government of Aragón, Spain, and ERDF "A way of making Europe" through grant E36-23R (FENOL), and by Grant No. PID2023-149409NB-I00 from Ministerio de Ciencia, Innovación y Universidades, Agencia Española de Investigación (MICIU/AEI/10.13039/501100011033). G.F.A. was partially supported by the São Paulo Research Foundation (FAPESP), through grants 2024/16711-8 and 2025/04409-8. The funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author contributions

G.F.A., W.H., and N.H. contributed equally to this work. They wrote the code, conducted the experiments, and prepared the first draft of the paper. T.E.-R. worked closely with G.F.A., W.H., and N.H. on editing the various versions of this paper. All authors—G.F.A., W.H., N.H., T.J., Y.M., and T.E-R.—contributed to the problem definition, solution, experimental design, and editing of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42005-025-02474-7.

Correspondence and requests for materials should be addressed to Tina Eliassi-Rad.

Peer review information *Communications Physics* thanks Michele Coscia and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.