Covering Problems and Core Percolations on Hypergraphs

Bruno Coelho Coutinho,[1] Ang-Kun Wu,[2] Hai-Jun Zhou,[3] and Yang-Yu Liu[4,5,*]
[1]*Instituto de Telecomunicações, Physics of Information and Quantum Technologies Group, Lisbon P-1049-001, Portugal*
[2]*Department of Physics and Astronomy, Rutgers University, Piscataway, New Jersey 08854, USA*
[3]*CAS Key Laboratory for Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China*
[4]*Channing Division of Network Medicine, Brigham and Women's Hospital and Harvard Medical School, Boston, Massachusetts 02115, USA*
[5]*Center for Cancer Systems Biology, Dana-Farber Cancer Institute, Boston, Massachusetts 02115, USA*

(Received 8 March 2019; revised manuscript received 13 April 2020; accepted 27 April 2020; published 17 June 2020)

We introduce two generalizations of core percolation in graphs to hypergraphs, related to the minimum hyperedge cover problem and the minimum vertex cover problem on hypergraphs, respectively. We offer analytical solutions of these two core percolations for uncorrelated random hypergraphs whose vertex degree and hyperedge cardinality distributions are arbitrary but have nondiverging moments. We find that for several real-world hypergraphs their two cores tend to be much smaller than those of their null models, suggesting that covering problems in those real-world hypergraphs can actually be solved in polynomial time.

DOI: 10.1103/PhysRevLett.124.248301

As a natural generalization of a graph, a hypergraph consists of vertices and hyperedges [1]. A hyperedge can simultaneously connect any number of vertices, which facilitates a more faithful representation of many real-world networks [2,3]. For example, given a set of proteins and a set of protein complexes, the corresponding hypergraph naturally captures the information on proteins that interact within a protein complex [4]. For a biochemical reaction system, the hypergraph representation indicates which biomolecules participate in a particular reaction [4,5]. In computer science, the factorization of complicated global functions of many variables can often be represented by a factor graph, which can be mapped to a hypergraph [3–8]. In social science, a collaboration network can also be represented by a hypergraph, where vertices represent individuals and hyperedges connect individuals who were involved in a specific collaboration, e.g., a scientific paper, a patent, a consulting task, or an art performance [7,8].

As in graphs, the degree of a vertex in a hypergraph is the number of hyperedges that connect to it. The number of vertices connected by a hyperedge is called the cardinality of that hyperedge. If all hyperedges have the same cardinality K, the hypergraph is said to be uniform or K uniform [9–11]. Note that a graph is just a two-uniform hypergraph.

The core of a graph—defined as the remainder of the greedy leaf removal (GLR) procedure where leaves (vertices of degree one) and their neighbors are removed iteratively from the graph—has been related to the conductor-insulator transition [12], structural controllability [13], and many combinatorial optimization problems [14]. Note that this core is fundamentally different from the k-core [15–17] that is obtained by recursively removing all vertices with degree less than k, or the giant connected component [18–21]. The GLR removes not only vertices with degree one (which resembles the 2-core) but also their neighbors regardless of their degrees [22,23]. Indeed, the core size is related to a fundamental combinatorial problem—the minimum vertex cover (MVC) problem, which aims to find the smallest set of vertices in a graph so that every edge is incident to at least one vertex in the set [24]. If the core is absent, then the MVC problem is solvable in polynomial time. Otherwise, if the core exists and is extensive in size, then the MVC problem is generally NP hard [24–26]. As the dual of the MVC problem, the minimum edge cover problem aims to find the smallest set of edges so that for every vertex in the graph there is at least one edge incident to it. Both covering problems can be defined similarly on hypergraphs. The minimum edge cover problem on graphs can be computed in polynomial time [27]. Yet, this is not true for hypergraphs, where both the minimum hyperedge cover (MHC) and the MVC problems are generally NP hard [28]. Note that the MVC and MHC on hypergraphs are related to many real-world problems, e.g., finding the optimal drug combination in pharmacology [29], searching files in storage systems [30], etc. Typically these problems can be solved using approximate algorithms, e.g., highest-degree-first [29,31] and simulated annealing [29]. Here we show that those approximate algorithms are not always necessary when the core is absent. To achieve that, we extend the concept of the core in graph to the hypergraph case, and define two cores associated with the MVC and MHC problem, respectively.

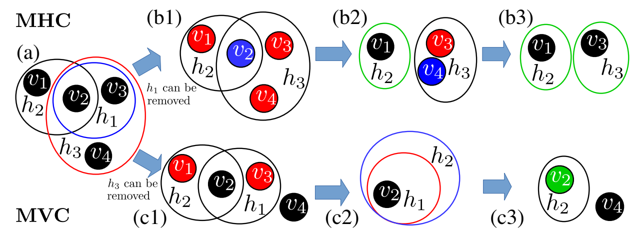

FIG. 1. Generalized greedy leaf removal helps us solve the minimum hyperedge cover and the minimum vertex cover problems on hypergraphs. Vertices are represented by dots and hyperedges by circles. The green vertices and hyperedges in (b3) and (c3) are the solution to the MHC or MVC problem, respectively.

Let us consider the MHC problem of the hypergraph $\mathcal{H}_0$ in Fig. 1(a), which has three hyperedges $\{h_1, h_2, h_3\}$ and four vertices $\{v_1, v_2, v_3, v_4\}$. The hyperedge h_3 contains all the vertices in h_1, as well as vertex v_4, thus h_1 is not necessary for the MHC and can be removed, leading to the hypergraph $\mathcal{H}_1$ shown in Fig. 1(b1). In $\mathcal{H}_1$ the vertex v_2 is contained by hyperedge h_2 that contains also vertex v_1. Hence if v_1 is covered, v_2 is also covered. We can therefore remove v_2 from $\mathcal{H}_1$. By iteratively removing vertices and hyperedges using these rules, we get the hypergraph shown in Fig. 1(b3), for which solving the MHC problem is trivial. The MVC problem is dual to the MHC problem, hence we can obtain a dual set of rules. In the hypergraph $\mathcal{H}_0$, the hyperedge h_3 contains all the vertices in h_1, as well as vertex v_4, thus if h_1 is covered h_3 is also covered. We can therefore remove h_3 and obtain the hypergraph $\mathcal{H}_2$ in Fig. 1(c1), for which we find that vertex v_1 is redundant and can be removed (since v_2 covers the same hyperedge as v_1 and also covers hyperedge h_1). By iteratively removing vertices and hyperedges using these rules, we get the hypergraph shown in Fig. 1(c3), for which the MVC problem is trivial. Similar rules have been proposed to reduce the complexity of set covering in the context of linear programming [32]. But the systematic study of the size of a core based on these rules is still missing.

The example shown in Fig. 1 prompts us to define three sets of hyperedges (or vertices): (i) S is a solution of the MHC (MVC) problem; (ii) $\tilde{S}$ contains hyperedges (vertices) that can be determined to be part of S using our approach; and (iii) the core h_{core} (v_{core}) contains the hyperedges (vertices) that cannot be determined if they belong to S or not using our approach. If the cardinality (degree) of a hyperedge (vertex) is zero, then it should not be covered. For the remaining vertices (hyperedges) we can find the vertices (hyperedges) that belong to each category based on the following rules. *(Rule-1)*: Consider hyperedges h_1 and h_2 that contain the set of vertices V_1 and V_2. If V_1 and V_2 are not empty sets, and $V_1 \subseteq V_2$, we remove h_1 (or h_2) to solve the MHC (or MVC) problem, respectively. *(Rule-2)*: Consider vertices v_1 and v_2 that are contained by the set of hyperedges H_1 and H_2. If H_1 and H_2 are not empty sets, and $H_1 \subseteq H_2$, we remove v_2 (or v_1) to solve the MHC (or MVC) problem, respectively. We repeat this process until no more vertices or hyperedges can be removed. In the final hypergraph, hyperedges (vertices) with cardinality (degree) one belong to $\tilde{S}$ [Figs. 1(b3) and 1(c3)]; hyperedges (vertices) with cardinality (degree) larger than 1 belong to h_{core} (v_{core}). Note that these two rules do not first aim to find leaf vertices or hyperedges (i.e., vertices or hyperedges of degree or cardinality one), and then delete them along with their neighbors. Instead, they first remove extra structures for MHC or MVC, "neighbors," and the leaves are just leftovers.

We emphasize that these two rules can be considered as the generalized GLR procedure on hypergraphs, which reduces to the standard GLR procedure on graphs, when we restrict $r = 2$ for all the hyperedges. And this procedure is fundamentally different from the procedure to obtain the k core in hypergraphs [33]. Note that even if the resulting h_{core} (v_{core}) is very small but nonzero, the generalized GLR procedure is better than approximation algorithms in solving the MHC (MVC) problem, because it explicitly tells us which hyperedges (vertices) belong to the solution S, which do not, and which cannot be determined.

Since the hypergraph core h_{core} is closely related to the MHC problem, we study the corresponding core percolation problem on random hypergraphs [34]. To achieve that, based on the generating function formalism [20,35], we generalize the mean-field approach proposed for the graph case [22]. Consider a large uncorrelated random hypergraph $\mathcal{H}$ with nondiverging moments of vertex degree and hyperedge cardinality distributions (see Supplemental Material [21], Sec. VII for potential generalizations). It is useful to define two types of removable vertices: a vertex is (i) α removable (leftover leaves in the end) if it is or can become a leaf vertex; (ii) β removable (removed based on the two rules mentioned) if its degree is larger than one and belongs to at least one leaf hyperedge. Dually, we define two types of removable hyperedges: a hyperedge is (i) δ removable (leftover leaves in the end) if it is or can become a leaf hyperedge; (ii) ϵ removable (removed based on the two rules mentioned) if it has cardinality r and is removed because it is connected to $(r-1)$ β-removable vertices. We can determine the category of a vertex v in $\mathcal{H}$ by the categories of its neighboring hyperedges in the modified hypergraph $\mathcal{H}\backslash\{v, h\}$ with vertices v and hyperedges h removed from $\mathcal{H}$, using the following rules: (i) α-removable vertices are vertices that are removed because, in the original hypergraph $\mathcal{H}$, they are only covered by a hyperedge h. Such vertex can only be covered by hyperedge h, and therefore can be removed from the original hypergraph $\mathcal{H}$ along side with h and all vertices contained by h. A vertex v, can become an α-removable vertex in two ways: it can be a leaf vertex, or all hyperedges, except h, are removed in a way that does not remove v, defined as ϵ-removable hyperedges; (ii) β-removable vertices are removed because they are automatically covered by a hyperedge that is determined to be belong to the covering set. Therefore, at least one neighboring hyperedge is δ removable. Similarly, we can determine the category of a hyperedge h in $\mathcal{H}$ by the categories of its neighboring vertices in the modified hypergraph $\mathcal{H}\backslash\{v, h\}$ with vertex v and hyperedge h removed from $\mathcal{H}$, using the following rules: (iii) δ-removable hyperedge: at least one neighboring vertex is α removable; note that if a vertex is α removable, meaning its degree is or can become one. Therefore, the only way to cover it is using the hyperedge connected to it. (iv) ϵ-removable hyperedge: removed because they are or can become a leaf hyperedge. In such a case, all except one vertex contained in it have been removed as β-removable vertices. (See Supplemental Material [21], Fig. S2, for illustrations of those rules.) Let α (or β) denote the probability that a random neighboring vertex or hyperedge h in a hypergraph $\mathcal{H}$ is α removable (or β removable) in $\mathcal{H}\backslash h$. Let δ (or ϵ) denote the probability that a random neighbor of a random vertex v in a hypergraph $\mathcal{H}$ is δ removable (or ϵ removable) in $\mathcal{H}\backslash v$. The only way for a vertex (or hyperedge) to be α (or ϵ) removable is to have degree (or cardinality) one, or that all but one of the hyperedges (or nodes) that contains (or are contained by) it are removed as ϵ (or β) removable hyperedges (or nodes). By contrast, a vertex (or hyperedge) is β (or δ) removable if it contains (or are contained by) at least one δ (or α) removable hyperedges (or nodes). Hence, we can derive a set of self-consistent equations (see Supplemental Material [21], Sec. II, for detailed derivations and schematic diagrams):

$$\alpha = \sum_{k=1}^{\infty} Q_n(k)\epsilon^{k-1}, \tag{1}$$

$$1-\beta = \sum_{k=1}^{\infty} Q_n(k)(1-\delta)^{k-1}, \tag{2}$$

$$1-\delta = \sum_{r=1}^{\infty} Q_h(r)(1-\alpha)^{r-1}, \tag{3}$$

$$\epsilon = \sum_{r=1}^{\infty} Q_h(r)\beta^{r-1}. \tag{4}$$

Here $Q_n(k)$ $[Q_h(r)]$ is the excess degree (cardinality) distribution. The fraction of vertices in h_{core} (vertices incident to the hyperedges in h_{core}), denoted as s_h^v, is given by

$$s_h^v = \sum_{k=2}^{\infty} P_n(k) \sum_{l=2}^{k} \binom{k}{l} (1-\delta-\epsilon)^l \epsilon^{k-l}, \tag{5}$$

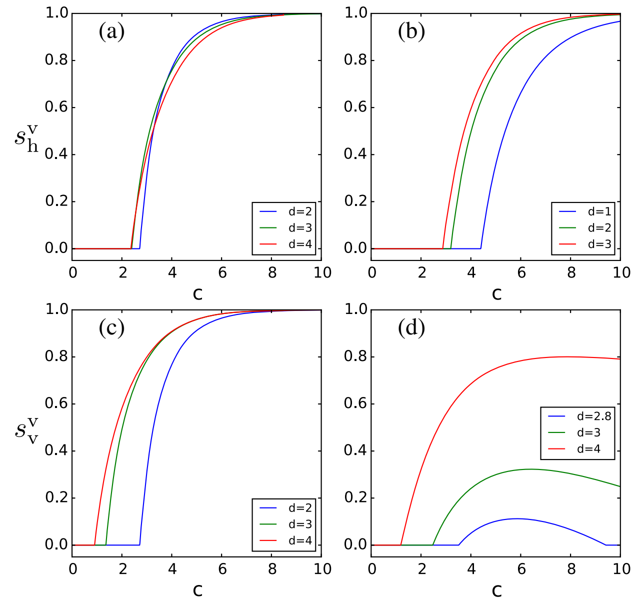

FIG. 2. The relative size of the core size hypergraphs with Poisson vertex degree distributions, (a),(b) s_h^v. (c),(d) s_v^v. In (a) and (c) we consider d-uniform hypergraphs, meaning that all hyperedge have the same cardinality d. In (b) and (d) we consider that the cardinality of the hyperedges follows a Poisson distribution with average cardinality d.

where $P_n(k)$ $[P_h(r)]$ is the degree (cardinality) distribution. (See Supplemental Material [21], Sec. III B for the formula of the fraction of hyperedges in h_{core}, denoted as s_h^h. The comparison between simulations and analytical results are shown in Supplemental Material [21], Sec. IV. For hypergraphs with Poisson vertex degree distribution and different hyperedge cardinality distributions, we find that the h_{core} emerges as a continuous phase transition [Figs. 2(a) and 2(b)], displaying the following scaling behavior:

$$s_h^v \propto (c-c^*)^{\zeta_1}, \tag{6}$$

with critical exponent $\zeta_1 = 1$ (see Supplemental Material [21] Sec. III E for details). We emphasize that this scaling behavior only represents the asymptotic behavior of the phase transition as $c - c^* \to 0^+$. Here, $c \equiv \sum_{k=0}^{\infty} kP(k)$ is the mean degree of the network. The relation between the critical mean degree c^* (percolation threshold) and the hyperedge mean cardinality d is represented in Fig. 3.

Similarly, the v_{core} percolation (associated with the MVC problem) can also be analytically studied for random hypergraphs. In this case, the equations on α and β are the same as the h_{core} percolation case, but for hyperedges we derive the following self-consistent equations:

$$\tilde{\delta} = \sum_{r=1}^{\infty} Q_h(r)\alpha^{r-1}, \tag{7}$$

$$1 - \tilde{\epsilon} = \sum_{r=1}^{\infty} Q_h(r)(1-\beta)^{r-1}. \quad (8)$$

The $v_{\rm core}$ consists of those vertices connected to at least two nonremovable hyperedges. Hence, the fraction of vertices in $v_{\rm core}$ is given by

$$s_v^v = \sum_{k=2}^{\infty} P_n(k) \sum_{l=2}^{k} \binom{k}{l} (1-\tilde{\delta}-\tilde{\epsilon})^l \tilde{\epsilon}^{k-l}. \quad (9)$$

See Supplemental Material [21] Sec. III B for the formula of the fraction of hyperedges in $v_{\rm core}$, s_h^h. For hypergraphs with Poisson vertex degree distribution and different hyperedge cardinality distributions, we find that the $v_{\rm core}$ emerges as a continuous phase transition [see Figs. 2(c), 2(d)], displaying the following scaling behavior:

$$s_v^v \sim (c - c^*)^{\zeta_2}, \quad (10)$$

with critical exponent $\zeta_2 = 1$ (see Supplemental Material [21], Sec. III D for details). Figure 2(d) shows that for a Poisson-Poisson hypergraph s_v^v starts to decrease for large c. This nonmonotonic behavior is due to the presence of hyperedges with cardinality 1 in the hypergraph (see Supplemental Material [21], Sec. IV for detailed explanations).

Phase diagrams of the $h_{\rm core}$ and $v_{\rm core}$ percolations on hypergraphs with Poisson vertex degree distributions are shown in Fig. 3. Note that the phase diagram of $v_{\rm core}$ percolation is equal to that of $h_{\rm core}$ percolation if we interchanged the mean cardinality d with the mean degree c. This is true because the $v_{\rm core}$ of a hypergraph is the $h_{\rm core}$ of the dual hypergraph.

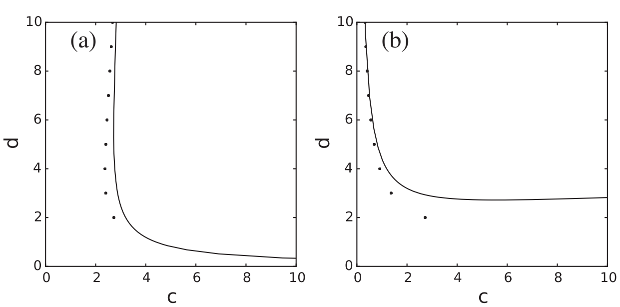

FIG. 3. Phase diagram of the $h_{\rm core}$ and $v_{\rm core}$ percolations on hypergraphs with Poisson vertex degree distributions. The black dots and black line represent the phase boundary of d-uniform hypergraphs and hypergraphs with Poisson hyperedge cardinality distribution, respectively. (a) $h_{\rm core}$. Note that, for the d-uniform hypergraph ($d > 1$) with Poisson vertex degree distribution, the critical mean degree (i.e., c^* of the $v_{\rm core}$ percolation) can be simply related to d as $c^* = e/(d-1)$, where $e = 2.71828\cdots$ [see (a) black dots, and Supplemental Material [21] Sec. III C for details]. This result was previously found in Ref. [11]. (b) $v_{\rm core}$.

We also apply the generalized GLR procedure to compute the $h_{\rm core}$ and $v_{\rm core}$ for several real-world hypergraphs: (i) APS consists of articles published in all the American Physical Society journals from 1893 to 2010 [36], where individual authors and their joint articles are considered as vertices and hyperedges, respectively. (ii) DGT consists of drugs (hyperedges) and their target genes (vertices) as listed in the DrugBank [37]. (iii) GMN consists of reactions (hyperedges) and the involved metabolites (vertices) in the genome-scale metabolic network of *E. coli* obtained from the BiGG database [38]. We find that for GMN, $h_{\rm core}$ contains 3.4% of hyperedges; while $v_{\rm core}$ contains less than 0.2% of vertices. For the other two hypergraphs (APS and DGT), both $h_{\rm core}$ and $v_{\rm core}$ contains less than 0.2% of hyperedges or vertices [Fig. 4(a)]. We also compare the size of each core with that of two null models of the real-world hypergraphs. The first null model (random1) corresponds to an ensemble of random hypergraphs with the same degree and cardinality sequences as a real hypergraph. The expected core sizes of this null model can be analytically calculated from Eqs. (1)–(5) and (7)–(9) based on the degree and cardinality sequences of the real hypergraphs. The second null model (random2) corresponds to an ensemble of random hypergraphs with Poisson degree and cardinality distributions (with the same mean degree and mean cardinality as those of a real hypergraph). The expected core sizes of this null model can again be analytically calculated from Eqs. (1)–(5) and (7)–(9) based on the Poisson degree and cardinality distributions. Note that for random1, the size of the core is always zero [blue points in Fig. 4(a)]; while for random2, the size of the core is between 30% and 100%, much bigger than that of the real hypergraph. These results suggest that the degree and cardinality distributions are main factors that explain the small cores of these real-world hypergraphs. Because $h_{\rm core}$ and $v_{\rm core}$ of those real-world hypergraphs are

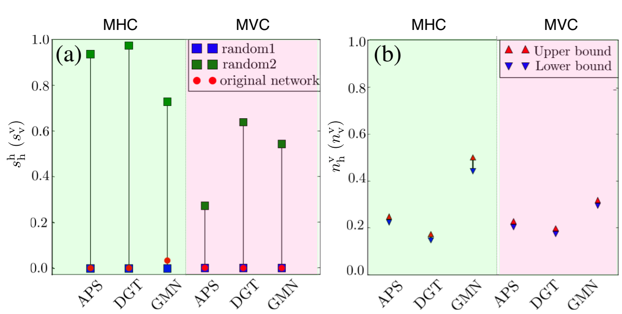

FIG. 4. (a) Fraction of hyperedges s_h^h (or vertices s_v^v) associated with the MHC (MVC) problem for three real-world hypergraphs: APS, DGT, and GMN; and their respective null models computed from Eqs. (5) and (9). This set represents those hyperedges (vertices) that cannot be covered optimally using our generalized GLR procedure. (b) Fraction of vertices n_h^v (hyperedges n_v^h) necessary to cover all the hyperedges (vertices) for the three hypergraphs.

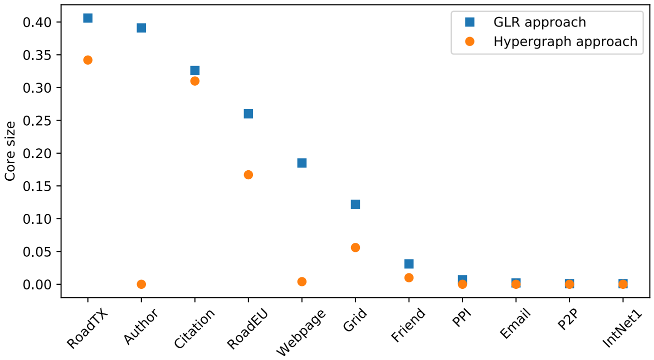

FIG. 5. Relative size of the core associated with the MDS problem for eleven real-world networks, using the GLR procedure introduced in Ref. [40] (square) and using our generalized GLR approach (circles).

really small, the MHC and MVC problems are effectively solvable. Indeed, as shown in Fig. 4(b), the upper and lower bound of the fraction of vertices (hyperedges) n_h^v (n_v^v) that are necessary to cover all the vertices (hyperedges) are very close to each other.

It turns out that our method can also be used to solve another classical NP-hard combinatorial problem: finding the minimum dominating set (MDS) for graphs. The MDS of a graph is the smallest set of vertices that needs to be occupied so that all unoccupied vertices are adjacent to at least one occupied vertex [39]. Our basic idea is as follows. We consider a hypergraph that has the same set of vertices as the original graph, and one hyperedge i contains all vertices adjacent to a vertex v_i (including v_i itself). Solving the MHC problem on this hypergraph is then equivalent to solve the MDS problem of the original graph. Our method offers a much more general approach than the greedy algorithm introduced in Ref. [40]. Indeed, the two leaf-removal rules introduced in Ref. [40] can be considered as special cases of our generalized GLR rules (see Supplemental Material [21], Sec. III A for details). In Fig. 5, we show the size of the cores associated with the dominating set for the eleven real-world networks analyzed in Ref. [40]. For most of those networks, our method shows a considerable improvement over the previous method. For some of the networks, our method actually finds no core left.

In physics it is common that a more abstract or general approach actually makes certain complicated problems easier to solve. This is often not true in social systems, biological systems, or complex systems in general. Our results suggest that generalizing graph to hypergraphs is one of the few cases where a small generalization makes a very hard problem easier to solve. Indeed, we show that our generalized GLR procedure and the corresponding hypergraph cores can help us solve various NP-hard covering problems in a systematic and universal way. If we aim to find a simple solution for complex problems, this is really an exciting result, indicating that hypergraphs might be the right way to represent complex networked systems. Our results open a new set of tools to analyze complex networked systems. It also raises a very important question: why do hypergraph cores of real systems tend to be small or absent? We anticipate our work will trigger more research activities in addressing this intriguing question.

B. C. thanks the support from Fundação para a Ciência e a Tecnologia (Portugal), namely through Project No. UIDB/EEA/50008/2020. Y.-Y. L. was supported by grants from the John Templeton Foundation (Award No. 51977) and National Institutes of Health (R01AI141529, R01HD093761, UH3OD023268, U19AI095219, and U01HL089856).

*Corresponding author.
yyl@channing.harvard.edu

[1] A. Bretto, *Hypergraph Theory: An Introduction* (Springer International Publishing, Switzerland, 2013).
[2] O. T. Courtney and G. Bianconi, Phys. Rev. E **93**, 062311 (2016).
[3] G. Bianconi and C. Rahmede, Phys. Rev. E **93**, 032315 (2016).
[4] S. Klamt, U.-U. Haus, and F. Theis, PLoS Comput. Biol. **5**, e1000385 (2009).
[5] E. Ramadan, A. Tarafdar, and A. Pothen, in *Proceedings of the 18th International Parallel and Distributed Processing Symposium, 2004*, pp. 189–196 (IEEE, Piscataway, 2004).
[6] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, IEEE Trans. Inf. Theory **47**, 498 (2001).
[7] C. Taramasco, J.-P. Cointet, and C. Roth, Scientometrics **85**, 721 (2010).
[8] L. Krumov, C. Fretter, M. Müller-Hannemann, K. Weihe, and M.-T. Houtt, Eur. Phys. J. B **84**, 535 (2011).
[9] C. Goldschmidt, Ann. Probab. **33**, 1573 (2005).
[10] C. Cooper, Random Struct. Algorithms **25**, 353 (2004).
[11] S. Takabe and K. Hukushima, Phys. Rev. E **89**, 062139 (2014).
[12] M. Bauer and O. Golinelli, Phys. Rev. Lett. **86**, 2621 (2001).
[13] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabasi, Nature (London) **473**, 167 (2011).
[14] R. M. Karp and M. Sipser, in *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science, 1981. SFCS '81* (IEEE, Piscataway, 1981) pp. 364–375.
[15] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, Phys. Rev. Lett. **96**, 040601 (2006).
[16] J. I. Alvarez-hamelin, A. Barrat, and A. Vespignani, in *Advances in Neural Information Processing Systems 18* (MIT Press, Cambridge, 2006) pp. 41–50.
[17] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, Nat. Phys. **6**, 888 (2010).
[18] P. Erdős and A. Rényi, Publ. Math. Debrecen **6**, 290 (1959).
[19] P. Erdős and A. Rényi, Publ. Math. Inst. Hung. Acad. Sci. **5**, 17 (1960).
[20] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Phys. Rev. E **64**, 026118 (2001).

[21] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.124.248301 for detailed explanations of the analytical calculations and simulation details.
[22] Y.-Y. Liu, E. Csóka, H. Zhou, and M. Pósfai, Phys. Rev. Lett. **109**, 205703 (2012).
[23] R. M. Karp and M. Sipser, in *Proceedings of 22nd Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Los Alamitos, 1981), p. 364.
[24] M. Weigt and A. K. Hartmann, Phys. Rev. Lett. **84**, 6118 (2000).
[25] M. R. Garey and D. S. Johnson, *Computers and Intractability* (W.H. Freeman, New York, 2002), Vol. 29.
[26] A. K. Hartmann and M. Weigt, J. Phys. A **36**, 11069 (2003).
[27] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1st ed. Series of Books in the Mathematical Sciences (W. H. Freeman, New York, 1979).
[28] V. Repiský, Covering edges of a hypergraph: Complexity and applications, Master's Thesis, Comenius University, 2007.
[29] A. Vazquez, BMC Syst. Biol. **3**, 81 (2009).
[30] W. Kautz and R. Singleton, IEEE Trans. Inf. Theory **10**, 363 (1964).
[31] D. S. Johnson, J. Comput. Syst. Sci. **9**, 256 (1974).
[32] J. E. Beasley, Eur. J. Oper. Res. **31**, 85 (1987).
[33] A. Dembo and A. Montanari, Ann. Appl. Probab. **18**, 1993 (2008).
[34] G. Ghoshal, V. Zlatić, G. Caldarelli, and M. E. J. Newman, Phys. Rev. E **79**, 066118 (2009).
[35] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, Rev. Mod. Phys. **80**, 1275 (2008).
[36] R. Sinatra, D. Wang, P. Deville, C. Song, and A.-L. Barabasi, Science **354**, aaf5239 (2016).
[37] D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey, Nucl. Acids Res. **34**, D668 (2006).
[38] J. Schellenberger, J. O. Park, T. M. Conrad, and B. Ø. Palsson, BMC Bioinf. **11**, 213 (2010).
[39] T. W. Haynes, P. J. Slater, and S. T. Hedetniemi, *Fundamentals of Domination in Graphs* (Marcel Dekker, New York, 1998).
[40] J.-H. Zhao, Y. Habibulla, and H.-J. Zhou, J. Stat. Phys. **159**, 1154 (2015).