ExBCG-TC: Extended Borel Cayley Graph Topology Control for Ad Hoc Networks

Dongsoo Kim and Jaewook Yu
Department of ECE
Stony Brook University (SUNY)
Stony Brook, NY 11794–2350
Email: {dongsso.kim, jaewook.yu}@stonybrook.edu
http://www.ece.sunysb.edu/~dkim, jwyu

Eric Noel
200 Laurel Avenue
AT&T Labs
Middletown, NJ 07748
Email: eric.noel@att.com

K. Wendy Tang
Department of ECE
Stony Brook University (SUNY)
Stony Brook, NY 11794–2350
Email: wendy.tang@stonybrook.edu
http://www.ece.sunysb.edu/~wtang

Abstract—In our previous work, we demonstrated that the Borel Cayley Graph topology control (BCG-TC) is an efficient topology control algorithm for producing an underlying network topology resembling a 4-regular Borel Cayley Graph and assuring small diameter, short average path length and superb power efficiency. But the BCG-TC networks are unlikely to be connected below a certain radio range threshold due to isolated nodes or isolated subnetworks. In this paper, we propose a topology control algorithm, Extended BCG-TC (ExBCG-TC), that effectively improves network connectivity of network topology generated by BCG-TC. Our ExBCG-TC works by injecting a small number of nodes into the network and connecting them to nodes already in the network. Throughout the efficient rules for establishing connection (namely the Request Criterion and the Accept Criterion), the proportion of isolated nodes or isolated subnetworks is significantly reduced or totally eliminated. The resulting network topologies are more likely to be fully connected with radio range much smaller than the radio range threshold of the BCG-TC while preserving the outstanding network topological properties and power efficiency of BCG-TC network topologies.

I. INTRODUCTION

Recent development of small and energy-efficient wireless sensors that can communicate with each other via radio transceivers has resulted in the rapid growth of wireless sensor networks (WSNs) for a wide variety of applications. Among the various research issues within WSNs, topology control has been considered as one of the most important issue [1]. By controlling the topology of the graph representing the communication links between nodes, topology control algorithms reduce energy consumption and radio interference and thus preserve network lifetime of WSNs. Desirable features of the resultant communication graph include fundamental properties such as a small nodal degree (to minimize interference and to allow efficient bandwidth utilization), a small diameter, a short average path length and graph connectivity [2] [3].

Among the topology control algorithms introduced so far, the $k$-NEIGH algorithm is concerned with (1) reducing radio interference by confining the maximum number of physical neighbors and (2) generating a connected network with high probability [2]. However, the $k$-NEIGH algorithm requires more than 10 physical neighbors to guarantee a fully connected network consisting of a large number of nodes uniformly and randomly distributed in the target area. In the Cone Based Topology Control (CBTC) algorithm, a node $w$ searches for its physical neighbor with the minimum transmission power in every cone of a certain angle $\theta$ centered at $w$ [3]. Wattenhofer demonstrates that $\theta \leq 2\pi/3$ and at most 6 logical neighbors are needed for a connected network. But, the CBTC needs directional information to find a physical neighbor, which is required to estimate a direction of another node without its position information. This directional information can add more complexity to the topology control algorithm.

With a growing interest in topology control, the importance of the underlying network topology has been rising. In [4], Newman emphasizes the importance of understanding the network structure that affects the performance of a network system. In [5], we reported that Borel Cayley Graphs (BCGs) show attractive topological network properties as an underlying network topology. The original BCGs have small diameter, short average path length, and provide fast information distribution over a wide range of network sizes. However, the original BCGs assume that all nodes are within a one-hop communication range. Namely, to adopt the original BCG as an underlying network topology for wireless networks, all nodes must be within the transmission range of each other, which can entail a large power consumption for communication. To relieve this deficiency, we proposed a topology control algorithm for BCG (BCG-TC) [6]. BCG-TC generates a network topology close to a 4-regular original BCG while assuming that each node has a limited transmission range. The resulting network topology has better topological network properties and smaller energy consumption in comparison to other benchmark topology controls. However, the BCG-TC still requires a relatively long transmission range to obtain a fully connected network.

In this paper, we propose an Extended BCG-TC (ExBCG-TC) to reduce required transmission range to generate a fully connected network while preserving attractive topological property and power efficiency of the network generated by BCG-TC. In the ExBCG-TC, a small number of nodes is randomly injected into the network generated by BCG-TC. Then these injected nodes are connected to the nodes already in a network according to a Request Criterion and an Accept Criterion. For performance evaluation, we investigated network topological properties and power consumption in information distribution of the formulated network by ExBCG-TC. For comparison, $k$-NEIGH and $k$-RANDOM are used as benchmark. In $k$-RANDOM, a node $u$ randomly selects its logical neighbors among its physical neighbors. The target network is composed of 1081 nodes uniformly and randomly distributed in an area of $100 \times 100 m^2$. The result shows that the network topology generated by ExBCG-TC has the best performance among the network topologies investigated.

This paper is organized as follows: In Section II, we review BCGs, BCG random expansion algorithm and BCG-TC algorithm. In Section III, we present the proposed ExBCG-TC and in Section IV we go over the simulation result. In Section V, we provide a conclusion and future work.
II. Related Work

In this section, we provide a brief summary of the definition and connection rule of BCGs, the BCG random expansion algorithm and the BCG-TC algorithm.

A. Borel Cayley Graphs

The BCGs are vertex-transitive, symmetric and regular pseudo random graphs derived from the Borel subgroup. The Borel subgroup is a set of triangular and nonsingular matrices of the form 

\[ A = \begin{pmatrix} a & g \end{pmatrix}, \quad a \in \mathbb{Z}_p, \quad g \in \mathbb{Z}_p^k \]

where \( a \) is a non-negative integer number ranging from \( 0 \) to \( p-1 \). The BCGs are constructed based on the mapping of elements into the graph structure.

Definition 1 (Borel Cayley Graph [7]). Assume that \( V \) and \( G \subseteq V \setminus \{1\} \) are a Borel subgroup and a generator set, respectively. \( G \) is a BCG consisting of vertices represented by \( 2 \times 2 \) matrices \( G \) and directed edges satisfying \( u = v \ast g \), where \( u \neq v \in V, g \in G \) and \( \ast \) is a modulo-\( p \) multiplication chosen as the group operation.

The generator set \( G \) excludes the identity element \( I \) to avoid self-loops.

As noted from the definition, a vertex in BCG is represented by a \( 2 \times 2 \) matrix. This matrix element needs to be mapped to a unique non-negative integer number for easy computation during network modeling (e.g. node ID assignment and routing protocol). [5] provides a function that maps \( 2 \times 2 \) matrix elements of Cayley graphs to a unique integer value by grouping the matrix elements into \( k \) classes. From the mapping function, non-negative integer number ranging from 0 to \( n-1 \) are generated where \( n \) is the number of nodes of BCG.

In terms of the connection rule in BCG, arbitrary two nodes \( u, v \) represented by \( 2 \times 2 \) matrices are logically and bidirectionally connected from \( u \) to \( v \) by multiplication of one end node \( v \) and a certain power of generator \( g \) chosen from a generator set \( G \). This relationship can be expressed as \( u = v \ast g \ast \ldots \ast g \). Since the generator set \( G \) is closed under inversion \( (g^{-1}) \in G \), the connection in opposite direction is obtained by \( u = v \ast g \ast \ldots \ast g^{-1} \). Effectively, a symmetric (undirected) connection is established in BCG. Furthermore, selecting a generator \( g \) gives each node of BCG two edges (in \( g, g^{-1} \) directions). Therefore, choice of two generators \( g_1, g_2 \) and their inverses \( g_1^{-1}, g_2^{-1} \) generates a 4-regular undirected BCG.

B. BCG random expansion algorithm

From the definition of BCG, the network size of the BCG is determined by parameters \( p \) and \( k \) \( (N = p \times k) \). This resulting size inflexibility is one of the most critical issues of BCGs. In [8], the BCG random expansion algorithm is introduced to relax this size inflexibility of BCGs without performance degradation. The BCG random expansion algorithm expands the original BCG by injecting a node between two distinct edges that are randomly selected from the original BCG to maintain the 4-regular property. If there exists any common node on the selected edges, the new edges become multiple edges which should be avoided. Once the two edges are selected, they are unwired and then the end nodes of the selected two edges are rewired to the newly added node. This rewiring preserves the 4-regular property of the BCGs. In [8], the topological properties and information distribution performance of the BCG random expansion algorithm are discussed further.

C. BCG topology control algorithm

BCG-TC aims at generating a network topology close to a 4-regular BCG with a more realistic constraint on the transmission range for each node. From the definition of BCGs, a 4-regular undirected BCG is created by defining two generators and their inverses (i.e. generators \( g_1, g_2, g_1^{-1} \) and \( g_2^{-1} \)). However, BCG has an assumption that all nodes are in a one-hop communication range. In [6], to overcome this limitation, we presented BCG-TC algorithm that considers a limited transmission range of network to generate a network topology resembling a 4-regular BCG. The BCG-TC assumes all nodes in a network are aware of their IDs and the IDs of their logical neighbors. After deployment, during Phase I of BCG-TC, a node searches for its logical neighbors among its physical neighbors. If the resulting node from the Phase I has less than 4 neighbors, the node performs Phase II that utilizes the Cut-Through Rewiring (CTR) algorithm [5]. Using the CTR algorithm, the node finds the next available neighbor in the same direction in which the node didn’t find a neighbor during the Phase I. As a result, the BCG-TC algorithm produces a network topology resembling the 4-regular original BCG.

III. Extended BCG-TC

Even though BCG-TC generates network topologies close to the original BCG topology, it places a stringent constraint on the node a radio range to guarantee the resulting network topology forms a fully connected graph. In [5], with 1081 nodes uniformly and randomly distributed in area \( 100 \times 100 \text{m}^2 \), it was found that the radio range must be at least 100m to guarantee a fully connected network. This large transmission range is not practical and causes undesirably large power consumption in communication. To relieve the radio range constraint of BCG-TC while maintaining a small constant degree \( (k = 4) \) and other attractive properties of BCG, we propose an EXTENDED BCG-TC (ExBCG-TC).

A. Protocol

The ExBCG-TC is aimed at eliminating isolated nodes and isolated subnetworks when the radio range is smaller than the threshold that guarantees a full connectivity of BCG-TC network topology. The isolated nodes and isolated subnetworks are eliminated by injecting a small number of nodes and connecting them to the existing nodes. For example, in [8], Request Criterion and Accept Criterion are provided for the injected node to select logical neighbors among its physical neighbors. In ExBCG-TC, we define Request Criterion and Accept Criterion as:

- **Request Criterion**: All physical neighbors receiving HELLO send REQ to an injected node. This is to prevent an isolation of injected nodes when there are no physical neighbors qualified to send REQ.
- **Accept Criterion**: The physical neighbors with a degree smaller than 4 have a higher priority to be a logical neighbor of the injected node. Thus, an isolated physical neighbor has the highest priority to become a logical neighbor of injected nodes. To limit a maximal nodal degree to 4, the injected node can select at most 4 logical neighbors. Note that these criteria are applied to the BCG-TC network topology. The Request Criterion and the Accept Criterion defined by the ExBCG-TC make it possible for the BCG-TC network topology to become fully connected with a smaller transmission range than that of the BCG-TC.
The ExBCG-TC is composed of four steps: (a) identification of physical neighbors (broadcasting HELLO and receiving REQ), (b) selection of logical neighbors, (c) transmission of connection requests (sending ACK-I) and (d) connection confirmation (receiving ACK-II). In the ExBCG-TC, we assume that each injected node is assigned an ID prior to deployment that is consistent with BCG-TC mode of operation. Thus, if two nodes are logically connected, a bidirectional connection between the two nodes is created.

Algorithm 1 ExBCG-TC

```
Procedure EXTENDED BCG-TC(I)
1: procedure EXTENDED BCG-TC(I)
2: for each w ∈ Π do
3:     V ← 0
4:     V* ← 0
5:     /* collect physical neighbors */
6:     for each v within r do
7:         V ← V ∪ {v}
8:     /* collect vertex with degree smaller than 4 */
9:     if d_v < 4 then
10:        V* ← V* ∪ {v}
11:     end if
12: end for
13: /* select neighbors and connect */
14: if |V*| = 0 then  \textcircled{Case 1}
15:    BCG Random Expansion \textcircled{Section III-C}
16: else if |V*| = 1 ∨ |V*| = 2 then \textcircled{Case 2}
17:    E(V) ← E(V) ∪ {e(w, v)}, ∀ v ∈ V*
18:    choose random e(x, y) ∈ E(V \ V*)
19:    E(V) ← E(V) \ {e(x, y)}
20:    E(V) ← E(V) ∪ {e(w, x), e(w, y)}
21: else if |V*| = 3 ∨ |V*| = 4 then \textcircled{Case 3}
22:    E(V) ← E(V) ∪ {e(w, v)}, ∀ v ∈ V*
23: else if |V*| > 4 then \textcircled{Case 4}
24:    sort V* by degree
25:    V* ← first 4 vertices in V*
26:    E(V) ← E(V) ∪ {e(w, v), ∀ v ∈ V*}
27: end if
28: end for
29: end procedure
```

In the ExBCG-TC algorithm, the transmission range is denoted by r for the rest of this paper and w ∈ Π denotes an injected node in the network generated by BCG-TC. V represents the set of physical neighbors of w, while V* is the set of nodes with a degree smaller than 4 out of V. We use d_v to represent the degree of node v. In the next section, we discuss the operation of ExBCG-TC.

B. ExBCG-TC mode of operation

The ExBCG-TC is used to establish a connection between the injected nodes and their physical neighbors while limiting the maximum nodal degree to 4. In Figure 1, the area surrounded by the dashed line corresponds to the transmission range of the injected node, w, and the white nodes represent existing nodes. Once w is injected, it advertises its existence by broadcasting HELLO with its ID and starts the timer, ReqTimer, to manage reception of REQ (see Figure 1(a)). In Figure 1(b), all physical neighbors send REQ with their IDs, degree values and neighbor lists following the Request Criterion of ExBCG-TC. When the ReqTimer expires, w starts the procedure to select logical neighbors from the collected REqs based on the the Accept Criterion of the ExBCG-TC. Note that w is assumed to receive at least one REQ since we regard the target network as a very dense wireless network and thus there is at least one physical neighbor. w will face one of following four cases:

- Case 1: All physical neighbors sending REQ have degree 4 (|V*| = 0). For this case, the injected node performs the BCG random expansion discussed in the next section.
- Case 2: If the number of the nodes of degree d < 4 is 1 or 2, w randomly selects one edge based on the responding nodes IDs and neighbor lists from the REQ received. Note that the selected edge should not have any nodes in common with the node(s) of degree d < 4. Then w sends ACK-I to the node(s) of degree d < 4 and to the two end nodes of the selected edge. As a result, the selected edge is disconnected, and the two end nodes of the selected edge together with the selected node(s) (of degree d < 4) are connected to w.
- Case 3: When there are 3 or 4 physical neighbors of degree d < 4, w sends ACK-I to each of them.
- Case 4: When more than four physical neighbors of degree d < 4 exist, w selects the four physical neighbors with the smallest degree. Then, w sends ACK-I to the selected four nodes.

ACK-I includes the new degree value and neighbor list for each selected logical neighbor. After the injected node sends ACK-I, Ack-II-Timer is initiated to manage a collection of
ACK-II. The physical neighbors receiving ACK-I update their degree and neighbor list and reply with ACK-II to w for confirmation (see Figure 1(d)). Once Ack-Il-Timer expires, the next injected node, if any, performs ExBCG-TC.

C. **BCG random expansion in ExBCG-TC**

BCG random expansion [8] is an algorithm that connects a newly injected node to nodes already in the original BCG while preserving a constant maximum nodal degree. When a node is injected into the original BCG, two distinct edges with no common end node are randomly selected. Each of the selected edges is unwired and then rewired to the injected node. In [8], since it is assumed that all nodes are within a one-hop from one another, selecting two edges is not restricted by the transmission range. However, for the BCG random expansion in ExBCG-TC, two pairs of end nodes of the selected two edges should be in the transmission range of the injected node. Figure 2 shows an example of the BCG random expansion in ExBCG-TC. When all physical neighbors have degree 4, the injected node w randomly selects two edges within its transmission range based on the REQs received from its physical neighbors. Figure 2(a) corresponds to the case where edges e(u, v) and e(x, y) are randomly selected, and w sends ACK-I to two end nodes of the selected edge. Then, the two edges are unwired and w is wired to u, v, x, and y in Figure 2(b).

To summarize, we described how ExBCG-TC and BCG random expansion in ExBCG-TC work. When the network generated by BCG-TC includes isolated nodes or isolated subnetworks, ExBCG-TC can eliminate such isolations by randomly injecting nodes and connecting them to their physical neighbors. In ExBCG-TC, an injected node advertises its existence, finds physical neighbors, requests connections, receives acknowledgements and constructs connections. In the next section, we discuss the performance of ExBCG-TC by evaluating the topological properties and power consumption in information distribution of the resulting network topology.

### IV. **Performance Evaluation**

In this section, we evaluate the network connectivity, network topological properties and power consumption of the network topology generated by ExBCG-TC. We compare the resulting network topology from our algorithm to the network topologies of BCG-TC, k-NEIGH and k-RANDOM. We note that the number of injected nodes required by ExBCG-TC beyond that of BCG-TC is negligible (less than 4% at most) and had little impacts on our benchmark metrics.

#### A. Setup

In the simulation, all networks under consideration are based on 1081 nodes uniformly and randomly distributed in an area of $100 \times 100\text{m}^2$. For each radio range (10m, 20m, ..., 150m), 100 network samples were collected. Table I shows the detail of the parameters. The maximum nodal degree of BCG-TC, ExBCG-TC and k-RANDOM is constrained to 4. Note that it is impossible to generate a fully connected network with maximum nodal degree 4 in k-NEIGH. Therefore, we increased the nodal degrees of k-NEIGH to 10 [2]. In k-RANDOM, a node randomly selects the logical neighbors among its physical neighbors. Also, note that the injected nodes into ExBCG-TC are uniformly and randomly injected into the target area.

#### B. **Network connectivity**

By connecting isolated nodes to injected nodes, we reduce the probability for a BCG-TC network to form a disconnected graph when $r$ is below the connected graph threshold. Figure 3(a) shows the ratio of average isolated nodes versus $r$. The network topologies of the BCG-TC and the ExBCG-TC with $r = 10\text{m}$ have almost the same ratio of average isolated nodes, but the difference between the two ratios becomes larger as $r$ increases. Particularly, for $r \geq 60\text{m}$, the network topologies of the ExBCG-TC with $N_{in} = 10, 20, 30$ and 40 have no isolated node, while the BCG-TC generates isolated nodes up to $r = 90\text{m}$.

We also considered the ratio of fully connected networks as a function of $r$. The ratio for each $r$ is obtained by dividing the number of fully connected networks by the number of network samples ($100$). As demonstrated in the figure, the network topologies resulting from ExBCG-TC are more likely to produce connected networks than the network topologies of BCG-TC. Specifically, with $r = 30\text{m}$, only 4 out of 100 network samples generated by BCG-TC are fully connected while 21 fully connected networks out of 100 network samples are generated by ExBCG-TC. In particular, the number of fully connected networks of ExBCG-TC with $N_{in} = 40$ and $r = 30\text{m}$, 99% of network samples are fully connected. In the next section, we discuss the network topological properties of ExBCG-TC with $N_{in} = 40$ quickly grows from 0 to 99 when $r$ increases from 20m to 30m. It is said that by performing ExBCG-TC we obtained at least 5 times more connected networks than with BCG-TC of same radio range.

#### C. **Network Topological Properties**

In this section, we discuss the network topological properties of the resulting network topologies (diameter and average path length). The diameter is the greatest number of hops between two arbitrary nodes in the network. Thus, the shorter

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>area dimension</td>
<td>$100\times 100\text{m}$</td>
</tr>
<tr>
<td>number of nodes (N)</td>
<td>1081</td>
</tr>
<tr>
<td>number of injected nodes ($N_{in}$)</td>
<td>10, 20, 30, 40</td>
</tr>
<tr>
<td>radio range (r)</td>
<td>10, 20, ..., 150m</td>
</tr>
<tr>
<td>node distribution</td>
<td>uniformly and randomly distributed</td>
</tr>
<tr>
<td>sample size</td>
<td>100 samples per radio range</td>
</tr>
<tr>
<td>BCG generating parameters</td>
<td>$p = 47$, $c = 23$, $a = 2$</td>
</tr>
<tr>
<td>BCG generator set</td>
<td>$t_1 = 3$, $t_2 = 7$</td>
</tr>
</tbody>
</table>
diameter allows the more efficient information distribution. Figure 4(a) shows the average diameter of the network topologies investigated. The network topologies of the $k$-NEIGH have a constant average diameter of 33 over all $r$. In case of the network topologies of $k$-RANDOM, the diameter decreases from 25 to 10 in the range of $10 \leq r \leq 40$. However, it increases to 19 with $r \geq 50$ again. On the other hand, the diameters of network topologies of BCG-TC and ExBCG-TC are even smaller than the ones of $k$-NEIGH independent of $r$. The network topologies generated by BCG-TC and ExBCG-TC have almost the same diameter ranging from 14 to 9 as $r$ increases.

We also evaluated the average path length of the network topologies. The average path length is defined as the average number of hops along the shortest paths for all pair of nodes in a network. This metric is a good indicator of how efficient information distribution is over the network. In Figure 4(b), the network topologies of $k$-NEIGH have an invariant average path length of 13.28, which is larger than the one for the network topologies of BCG-TC and ExBCG-TC. The average path length of the network topologies of BCG-TC and ExBCG-TC is equal to or less than 7 over all $r$ where the connected network is generated.

\[ P_t = S(\delta_1 + \delta_2 d(u, v)), \quad (1) \]

\[ P_r = S \tau, \quad (2) \]

\[ P_{total}(\lambda) = \lambda S \sum_{\forall (u,v) \in E} (\delta_1 + \delta_2 d(u, v) + \tau), \quad (3) \]

Where $d(u, v)$ is a distance between node $u$ and $v$, $\tau$ is the number of steps to reach a consensus and $E$ corresponds to
Fig. 5: Average power consumption of node to reach the consensus

In Figure 5, we displayed the average power consumption for each node to reach a consensus in the network topologies under test. It is obtained by dividing $P_{\text{total}}$ by the number of nodes in the network. The average nodal power consumption in the network topologies generated by $k$-NEIGH has a constant value 10.7688J independent of $r$. On the other hand, the network topology of BCG-TC consumes less power than one of $k$-NEIGH over the whole transmission range. Further, we found that the network of ExBCG-TC requires a slightly different amount of power from that the BCG-TC needs. More specifically, with $r = 50m$ each node in the network of BCG-TC needs 0.1665J to reach the consensus and one of ExBCG-TC is 0.1673J with $N_{in} = 10$. Note that as the number of injected nodes increases, the needed power to reach the consensus is reduced. Among the network topologies of BCG-TC and ExBCG-TC with $N_{in} = 10, 20, 30$ and $40$, the network topology of ExBCG-TC with $N_{in} = 10$ requires the least power to reach a consensus.

Overall, we observed that the proposed ExBCG-TC generates more fully connected networks with smaller transmission range than BCG-TC networks. Moreover, ExBCG-TC did not degrade the network topological properties such as diameter and average path length. When it comes to the power consumption in information distribution, we found ExBCG-TC to need almost the same or slightly less amount of power to reach a consensus than BCG-TC.

V. CONCLUSION AND FUTURE WORK

In conclusion, the proposed ExBCG-TC is a powerful topology control algorithm to improve a connectivity of the network topology generated by BCG-TC. In ExBCG-TC, a small number of nodes (e.g., 10, 20, ..., 40 out of 1081) is randomly injected and connected to nodes already in the network. Connections of these injected nodes are established based on a set of the Request Criterion and the Accept Criterion in order to reduce or eliminate isolated nodes or isolated subnetworks that cause a disconnected network. We simulated the ExBCG-TC algorithm starting from the network topology generated by BCG-TC with 1081 nodes uniformly and randomly distributed in the area of $100 \times 100m^2$. Using the ExBCG-TC, we found the resulting network topologies are more likely to be fully connected than those of BCG-TC and $k$-NEIGH while preserving the superior network topological properties (small diameter and average path length) and power efficiency in communication.

For the future work, we plan to provide an analytical model to identify the exact number of injected nodes needed to guarantee a full connectivity of BCG-TC. Further, we will expand our algorithm to other topology controls such as $k$-NEIGH and CBTC.

VI. ACKNOWLEDGEMENT

The authors are partially supported by the National Science Foundation under Grant No. CNS 0829656 and IIP 0917956. Any opinions, findings and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES