Maintaining Social Connections through Direct Link Placement in Wireless Networks

Li Qiu  
The Pennsylvania State University  
University Park, PA, USA  
lyq023@cse.psu.edu

Liang Ma  
IBM T. J. Watson Research Center  
Yorktown, NY, USA  
maliang@us.ibm.com

Guohong Cao  
The Pennsylvania State University  
University Park, PA, USA  
gcao@cse.psu.edu

Abstract—Mobile Social Network (MSN), built on interconnected mobile devices, enables the flexibility of information exchanges of individuals within a virtual community. MSN may be self-organized and/or infrastructure-less, and the communication links may frequently fluctuate. When link qualities degrade, it remains critical to maintain the connections of important social pairs when supporting all social pairs is impossible. To achieve this goal, we propose to proactively place some reliable links (e.g., satellite links or UAV links) into the underlying communication network, so as to improve the service quality of the important social pairs, referred to as the Maintaining Social Connections (MSC) problem. We formulate the MSC problem and prove it is NP-hard. As such, we first study a special case of MSC, which is submodular and solvable with high approximation-ratio. Since the general MSC problem is not submodular, we propose an efficient approximation algorithm. Specially, by carefully choosing two submodular functions to lower/upper bound the MSC problem, we are able to prove the high approximation ratio of the proposed algorithm using these selected submodular functions. We further develop evolutionary algorithms to iteratively adjust the link placement via different exploration strategies, which also yield guaranteed approximation-ratio. Extensive evaluations based on both synthetic and real-world social network traces demonstrate the effectiveness of our proposed algorithms.

I. INTRODUCTION

Mobile Social Network (MSN) is a special social network where individuals with similar interests and/or goals form a virtual community to exchange information. Unlike traditional communication networks, MSNs, composed of inter-connected mobile devices, can be organized in an infrastructure-less and self-configuring manner. Under such mobile environments, the network may experience high node mobility and transmission interference, and thus it is hard to meet the desired communication quality required by end users. Various routing techniques have been proposed [1]–[4] to improve the end-to-end communication performance. However, most of these works focus on improving the data forwarding performance for all node pairs in the network. They ignore the fact that in practical mobile network environment, it is often impossible (or with significantly high cost) to maintain the communications of all node pairs in the network. Moreover, in real MSNs, the communication between some node pairs are more important than the rest, and thus it is critical to maintain the social connections between these important social pairs rather than every pair in the network. For instance, in a battlefield scenario, for a platoon of soldiers consisting of several squads, the commander of the platoon should maintain good connections with the squad leaders, but not necessarily with all other soldiers. As another example, during disaster recovery, it is critical to maintain the social connections between the control center and the rescue team; but may not be necessary for the control center to maintain connections with all other people in the area.

Maintaining connections of important social pairs in MSN is challenging. Since data is generally forwarded via multiple unstable wireless links, the probability that the data is successfully delivered is relatively low. Although multipath routing [5] or even flooding could be used to improve the data forwarding performance, each path may still experience a high failure rate. Moreover, such redundant transmission may further degrade the communication of other social pairs; therefore, it is likely that the overall data delivery ratio cannot meet the desired quality of service requirement.

To address this problem, we propose to proactively place some reliable communication links (e.g., satellite links or UAV links) to the given MSN, so as to improve the data forwarding performance of the important social pairs, referred to as the Maintaining Social Connections (MSC) problem. Specifically, each reliable link enables the direct communication between two selected nodes in the network (even if they are far away from each other), and has a failure probability close to 0. These reliable links are referred to as shortcut edges. Besides the pairs of nodes directly connected by the shortcut edges, other social node pairs may also benefit from these shortcut edges if transmitting data via these shortcut edges improves the data forwarding performance.

Existing works on direct link placement are limited to improving the connectivity or reducing the distance of all social node pairs [6]–[9]; therefore, it is a waste of direct link resources when these solutions are applied to our problem since the important social pairs are a subset of all node pairs in the network. Our work is most related to [10] in that [10] also studies how to maintain important social connections; however, [10] is limited to maintaining the connection of only one social pair via physical layer techniques such as amplify-and-forward, while we focus on simultaneously maintaining...
social connections of multiple social pairs through direct placement of shortcut edges. Since shortcut edges are valuable and expensive resources, the real challenge of this problem lies in the effective placement of limited number of shortcut edges, such that each shortcut edge can benefit more social pairs.

To reveal the difficulty of the MSC problem, we first consider a special case, where all important social pairs share a common node. By proving that this special case is NP-hard, we show that the general MSC problem is NP-hard and non-submodular. Then, we propose an efficient approximation algorithm for the general MSC problem. Specially, by carefully choosing two submodular functions to lower/upper bound the MSC problem, we are able to prove the high approximation ratio of the proposed algorithm using these selected submodular functions. We further propose evolutionary algorithms, where the basic idea is to employ both randomized selection and strategic exploration [11] to iteratively adjust the current shortcut edge placement to approach the optimal, and the trade-off is how to balance the portions of randomization and exploration. In addition to the theoretical guarantees, evaluations based on both synthetic and real-world social network traces demonstrate the effectiveness of our proposed solutions in maintaining social connections under limited shortcut edge resources.

In many practical scenarios, MSN may exhibit high dynamicity, e.g., link conditions, network topologies, and the importance level of different social pairs may change over time, which poses additional challenges to the shortcut edge placement. We demonstrate that the above solutions can also be applied to such dynamic network environment while sustaining theoretical performance guarantees, which therefore shows the vast applicability of our developed algorithms.

The main contributions of this paper are summarized as follows:

- We propose to maintain social connections of important social pairs through direct placement of limited shortcut edges. We formalize the problem and prove it is NP-hard.
- For MSC where important social pairs share a common node, we prove it remains NP-hard but is submodular. Our proposed algorithm has an approximation ratio of \((1 - \frac{1}{m})\).
- Since the general MSC problem is non-submodular, we bound it by two submodular functions, and derive an approximation algorithm with provable approximation ratio.
- We develop evolutionary algorithms with both theoretical approximation guarantees and better performance.
- We demonstrate the applicability of the above algorithms in dynamic networks.

The rest of the paper is organized as follows. Section II reviews related work. Section III presents the problem formulation. We discuss a special case of MSC in Section IV and focus on the MSC problem in Section V to develop three efficient algorithms. We discuss the algorithm applicability in dynamic networks in Section VI. Section VII presents the performance evaluation results, and Section VIII concludes the paper.

II. RELATED WORK

Mobile social networks (MSNs) have attracted lots of attention in recent years [12]–[14]. MSNs, composed of interconnected mobile devices, can be organized in an infrastructureless and self-configuring manner, and thus it is hard for end users to achieve the desired communication quality. To improve the end-to-end communication performance, a “carry-and-forward” strategy can be used [15], where mobile nodes physically carry the data, and forward the data when contacting a node with higher forwarding capability. Recent studies also utilize social knowledge to facilitate user communication and data forwarding [2]–[4], [16]. However, most of these existing works focused on improving the data forwarding performance for all node pairs in the network, and ignored the fact that the communication between some node pairs might be more important than others.

Although in [10], the authors investigated the problem of maintaining important social connections, their strategy is limited to maintaining only one social connection. In this paper, we aim to simultaneously maintain multiple important social connections.

To maintain multiple social connections, we propose to proactively place some reliable communication links in the network. There are some existing works on direct link placement [6]–[8], [17], however, these works have a different objective from ours. For example, Meyerson and Tagiku [7] investigated the problem of minimizing the diameter of the network (i.e., the longest distance between any two nodes in the network), by adding at most \(k\) shortcut edges. In [8], [17], the authors studied the problem of minimizing the average shortest path distance over all pairs of nodes in the network. In all these works, the shortcut edges are devoted to benefit the communication between all pairs of nodes. In this paper, we show that maintaining important social pairs is more important and challenging than maintaining all social pairs in the network, and accordingly the previous results cannot be directly applied.

III. PROBLEM FORMULATION

A. Network Model

We consider a wireless network with \(n\) nodes, where the connections of \(m\) important social pairs should be maintained \((m \leq \binom{n}{2})\). The network is modeled as an undirected graph \(G = (V, E)\), where \(V = \{v_1, \ldots, v_n\}\) is the vertex set, and edge \(e_{i,j} \in E\) represents an edge between \(v_i\) and \(v_j\). Let \(p_{i,j}\) denote the link failure probability of edge \(e_{i,j}\). Then for any path \(\Lambda = v_1, v_2, \ldots, v_q\) (where \(v_i \in V\) for \(i = 1, 2, \ldots, q\)), its failure probability is

\[
p = 1 - \prod_{i=1}^{q-1} (1 - p_{i,i+1}),
\]

where \(p_{i,i+1}\) is the failure probability of edge \(e_{i,i+1}\).
B. Important Social Pairs

Let set $S = \{u_1, w_1, \ldots, u_m, w_m\}$ denote the $m$ important social pairs, which is known to us. We aim to maintain social connections of these $m$ node pairs, i.e., for each node pair, there exists one path with the failure probability less than or equal to a threshold $p_i$. For simplicity, we consider the case when $m = 1$ (i.e., the number of important social pairs is less than the number of shortcut edges), this problem becomes trivial. This is because we can directly connect each important social pair by one shortcut edge. Hence, we consider the problem when $m > k$, where each shortcut edge may benefit the communication of more than one important social pair.

C. Problem Formulation

The goal is to maximize the number of important social pairs that meet the connectivity requirement, by adding at most $k$ shortcut edges to $G$. Note that the path failure probability is calculated based on Eq. (1). To simplify the computation, for each edge $e_{i,j}$, we define its length $l_{i,j} = -\ln(1 - p_{i,j})$ (here $\ln$ is the natural logarithm). Based on the definition of $l_{i,j}$, the failure probability of any path $\Lambda = v_1, v_2, \ldots, v_q$ can be rewritten as:

$$p = 1 - \prod_{i=1}^{q-1} (1 - p_{i,i+1}) = 1 - e^{-\sum_{i=1}^{q-1} l_{i,i+1}},$$

where $\sum_{i=1}^{q-1} l_{i,i+1}$ is defined as the length of path $\Lambda$. Then, the problem of finding the path with the lowest failure probability for the pair $\{u_i, w_i\}$ is equivalent to the problem of finding the shortest path between $u_i$ and $w_i$. For any node pair $\{u_i, w_i\}$, it meets the connectivity requirement, if and only if the length of the shortest path between $u_i$ and $w_i$ is no larger than $-\ln(1 - p_i)$. Since the failure probability of each shortcut edge is assumed to be 0, adding a shortcut edge between any two nodes is equivalent to adding an edge of length 0 between those two nodes (as its edge length is $-\ln(1 - 0)$).

In this way, our original problem is equivalent to maximizing the number of important social pairs with distance no larger than $d_i = -\ln(1 - p_i)$, i.e., the distance requirement, by adding at most $k$ shortcut edges. Given a set $F \subseteq V \times V$ of shortcut edges with length 0, we define function $\sigma(F)$ as the number of pairs in $S$ that meet the distance requirement in $G' = (V, E \cup F)$. Then, the problem of Maintaining Social Connections (MSC problem) is defined as follows:

**Objective:** Given an undirected graph $G = (V, E)$, length $l_{i,j}$ on each edge $e_{i,j} \in E$, a distance threshold $d_i$ (i.e., distance requirement), and a set of $m$ important social pairs $S$, find a set $F \subseteq V \times V$ of at most $k$ shortcut edges with length 0, such that $\sigma(F)$ is maximized.

A shortcut edge $f \in F$ and a regular link in $e \in E$ may connect to the same nodes in $V$; in this case, $f$ is of length 0, and thus potentially shortening the distance of important social pairs. On the other hand, if $m \leq k$ (i.e., the number of important social pairs is less than the number of shortcut edges), this problem becomes trivial. This is because we can directly connect each important social pair by one shortcut edge. Hence, we consider the problem when $m > k$, where each shortcut edge may benefit the communication of more than one important social pair.

IV. MSC with a Common Node

To reveal the difficulty of the MSC problem, in this section, we consider a special case: maintaining social connections with a shared common node (referred to as $\text{MSC-CN}$), and prove it is NP-hard. Since the MSC problem is at least as hard as the $\text{MSC-CN}$ problem, the MSC problem is also NP-hard. The following discussions on $\text{MSC-CN}$ also provide insights into understanding the approximation solution to the general MSC problem.

A. MSC-CN

In $\text{MSC-CN}$, all important social pairs share a common node. Let $u$ denote the shared common node. We have the following theorem.

**Theorem 1.** The $\text{MSC-CN}$ problem is NP-hard.

**Proof.** To prove the theorem, we assert that for every instance of $\text{MSC-CN}$, there exists an optimal solution $F^*$, where all shortcut edges in $F^*$ are incident to the common node $u$. For any pair $\{u, w_i\} \in S$, there exists a shortest $uw_i$-path which only uses at most one shortcut edge in $F^*$. We omit the proof for conciseness, as a similar proof can be found in Lemma 1 of [7].

In this way, constructing the aforementioned optimal solution $F^*$ (with $k$ shortcut edges) is equivalent to finding $k$ distinct edge endpoints, each of which connects to $u$ to form a shortcut edge in $F$, such that $\sigma(F)$ is maximized. Let $F^* = \{f_{u,v_1}, \ldots, f_{u,v_q}\}$, where $f_{u,v_j}$ is the shortcut edge connecting $u$ and $v_j$. Let $W = \{w_i\} \{u, w_i\} \in S\}$, i.e., $W$ consists of all the nodes of the important social pairs except the common node $u$. For any node $w_i \in W$, if there exists $f_{u,v_j} \in F^*$, such that the distance between $v_j$ and $w_i$ is no larger than $d_i$, then the pair $\{u, w_i\}$ meets the distance requirement. Consider placing a shortcut edge between node $v_j$ and the common node $u$ (i.e., $f_{u,v_j}$). Let set $C_i$ denote all nodes in $W$ whose distance to $v_j$ is no larger than $d_i$. Then the $\text{MSC-CN}$ problem is equivalent to finding at most $k$ sets from $C = \{C_1, \ldots, C_n\}$ (i.e., number of nodes in the network) to cover the maximum number of nodes in $W$.

Based on the above argument, the $\text{MSC-CN}$ problem is precisely the maximum coverage problem [18]. The maximum coverage problem is proved to be NP-hard, and accordingly the $\text{MSC-CN}$ problem is also NP-hard.

With Theorem 1, we have the following corollary.
Corollary 2. The MSC problem is NP-hard.

Proof. We use contradiction to prove the corollary. Assume MSC problem is not NP-hard. Then, its sub-problem MSC-CN is not NP-hard, which contradicts Theorem 1. Thus, MSC is NP-hard. ■

B. Solution to MSC-CN

In this section, we present an approximation solution to the MSC-CN problem. We first prove that MSC-CN is submodular, and then present a greedy algorithm which can achieve \((1 - \frac{1}{e})\)-approximation of the optimal. We now give the definition of submodular functions.

Definition 3. A set function \(\psi : 2^\Omega \rightarrow \mathbb{R}\) is submodular, if for any \(X \subseteq Y \subseteq \Omega\) and any \(x \in \Omega \setminus Y\), the following inequality is always satisfied:
\[
\psi(X \cup \{x\}) - \psi(X) \geq \psi(Y \cup \{x\}) - \psi(Y). \tag{2}
\]
In other words, submodular functions suggest that if \(X\) is a subset of \(Y\), the marginal benefit of adding an element \(x\) to \(X\) is at least as large as the marginal benefit of adding \(x\) to \(Y\).

We now prove the submodularity of MSC-CN. Based on the proof of Theorem 1, we focus on the case when all the shortcut edges in \(F\) are incident to the common node \(u\).

Theorem 4. The objective function \(\sigma(F)\) in MSC-CN is submodular, where \(F \subseteq \{u\} \times V, |F| \leq k\), and \(u\) is the common node of all important social pairs.

Proof. We define \(W = \{w_i|\{u, w_i\} \in S\}\) and \(C = \{C_1, C_2, ... , C_n\}\) as those in the proof of Theorem 1. For any set of shortcut edges \(X \subseteq \{u\} \times V\), let \(Y_X\) be the set of endpoints of edges in \(X\), except the common node \(u\) (i.e., \(u \notin Y_X\)). Then \(\{u, w_i\} \in S\) satisfies the distance requirement if and only if \(\exists v_j \in Y_X\) with \(w_i \in C_j\), i.e., \(C_j\) covers \(w_i\). Let \(C_X = \bigcup_{j \in Y_X} C_j\). Then, we have \(\sigma(X) = |C_X|\), and \(C_X \subseteq C_Y\) for any \(X \subseteq Y \subseteq \{u\} \times V\). Consider any \(f \notin Y\), we can derive that the number of overlapped nodes between \(C(f)\) and \(C_X\) is less than or equal to the number of overlapped nodes between \(C(f)\) and \(C_Y\). Hence, the number of additional node pairs meeting the distance requirement by adding \(f\) to \(X\) is at least as large as that by adding \(f\) to \(Y\). Thus, we get
\[
\sigma(X \cup \{f\}) - \sigma(X) \geq \sigma(Y \cup \{f\}) - \sigma(Y);
\]
therefore, the objective function \(\sigma(F)\) in MSC-CN is submodular. ■

The property of the objective set function in MSC-CN being submodular enables us to employ greedy shortcut edge placement to solve the problem efficiently. In particular, the greedy placement algorithm works as a multi-round selection process in MSC-CN. Let \(u\) be the common node shared by all important social pairs. At each round, on top of the current shortcut edge placement \(F\) (initial value of \(F\) is \(\emptyset\)), we select \(v\) as the shortcut edge endpoint that maximizes \(\sigma(F \cup \{f_{u,v}\}) - \sigma(F)\), where \(f_{u,v}\) is the shortcut edge connecting \(u\) and \(v\). This selection process runs until \(k\) shortcut edges have been selected or all node pairs satisfy the distance requirement. The performance of such greedy placement algorithm is bounded as follows.

Theorem 5. In MSC-CN where all node pairs sharing a common node \(u\), let \(F^*\) denote the optimal solution, and let \(\tilde{F}\) denote the solution obtained by the greedy placement algorithm, where \(F^*, \tilde{F} \subseteq \{u\} \times V, |F^*| \leq k,\) and \(|\tilde{F}| \leq k\). We have
\[
\sigma(\tilde{F}) \geq \left(1 - \frac{1}{e}\right) \cdot \sigma(F^*).
\]

Proof. Let \(F_i\) denote the solution of the greedy algorithm after the \(i\)-th iteration. Let the optimal solution \(F^* = \{f_1, ..., f_k\}\). Since \(\sigma\) is monotone, we have \(\sigma(F^*) \leq \sigma(F^* \cup F_i)\). As \(\sigma\) is submodular, the following inequality holds for all \(1 \leq i \leq k\):
\[
\sigma(F^*) \leq \sigma(F_i) + (\sigma(F_i \cup \{f_i\}) - \sigma(F_i)) + \ldots + (\sigma(F^*) - \sigma(F_i \cup \{f_1, ..., f_k-1\})) \leq \sigma(F_i) + k \cdot (\sigma(F_i) - \sigma(F_i)) \leq \sigma(F_i) + k \cdot (\sigma(F_{i+1}) - \sigma(F_i)).
\]
Based on the above inequality, we use induction to show that \(\sigma(F_i) \geq (1 - (1 - \frac{1}{e})^i) \cdot \sigma(F^*)\). Apparently, the previous inequality holds for the base case of \(i = 0\). For any \(1 \leq i \leq k\), based on inequality (3), we have
\[
\sigma(F_i) \geq \sigma(F_{i-1}) + \frac{1}{k} \cdot (\sigma(F^*) - \sigma(F_{i-1})) \geq \left(1 - \left(1 - \frac{1}{e}\right)^i\right) \cdot \sigma(F^*),
\]
where the last line of the above inequality is due to the induction hypothesis. Then, we have: \(\sigma(\tilde{F}) = \sigma(F_k) \geq (1 - (1 - \frac{1}{e})^{k}) \cdot \sigma(F^*) \geq (1 - \frac{1}{e}) \cdot \sigma(F^*)\). ■

V. GENERAL MSC PROBLEM

In this section, we study the general MSC problem. Since its objective function is proved to be non-submodular, the results in Theorem 5 cannot be directly applied. We exploit other submodular functions to bound or approximate the non-submodular MSC problem, and then propose shortcut edge placement algorithms with guaranteed approximation ratios.

A. Submodularity of MSC

We prove that the objective function of the general MSC problem is not submodular. Consider a network \(G = (V, E)\), where \(V = \{v_1, v_2, v_3\}\), and \(E = \emptyset\). Let the set of important social pairs \(S = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}\}\), and distance threshold \(d_t = 1\). Consider the following counter-example, where \(x = f_{1,2}\) (i.e., the shortcut edge between \(v_1\) and \(v_2\)), \(X = \emptyset\), and \(Y = \{f_{2,3}\}\). Then, we have \(\sigma(X \cup \{x\}) - \sigma(X) = 1 < \sigma(Y \cup \{x\}) - \sigma(Y) = 2\), which contradicts the definition of submodular functions in Eq. (2).
B. Approximation Algorithm

Since the general MSC problem is not submodular, we exploit the sandwich approximation strategy [19] to propose an approximation algorithm. Specially, by carefully choosing two submodular functions to lower/upper bound the MSC problem, we are able to prove the high approximation ratio of the proposed algorithm using these selected submodular functions.

Suppose we have two submodular functions \( \mu \) and \( \nu \), which upper and lower bounds function \( \sigma \) everywhere, i.e., \( \mu(F) \leq \sigma(F) \leq \nu(F) \) for all \( F \subseteq V \times V \) (recall \( \sigma \) is the mapping between solution \( F \) and the number of pairs in \( S \) that meet the distance requirement) with cardinality constraint \( k \) on \( F \) (i.e., \( |F| \leq k \)). Apparently, for problems with objective function \( \mu \) or \( \nu \), we can derive an efficient approximation algorithm for \( \sigma \) (i.e., the non-submodular MSC problem).

Let \( F_\mu \), \( F_\sigma \) and \( F_\nu \) be the solutions generated by greedy algorithms for functions \( \mu \), \( \sigma \) and \( \nu \), respectively (i.e., each solution is obtained by iteratively selecting a shortcut edge maximizing the marginal gain of the corresponding objective function). Our solution to the MSC problem is as follows:

\[
F_{\text{app}} = \arg \max_{F \in \{F_\mu, F_\sigma, F_\nu\}} \sigma(F).
\]

The performance of \( F_{\text{app}} \) can be theoretically bounded as follows [19]:

\[
\sigma(F_{\text{app}}) \geq \max \{ \frac{\sigma(F_\sigma)}{\nu(F_\sigma)}, \frac{\mu(F_\nu)}{\sigma(F_\sigma)}, \sigma(F_\sigma) \} \cdot (1 - \frac{1}{e}) \cdot \sigma(F_\sigma),
\]

where \( F_\sigma^* \) is the optimal solution maximizing \( \sigma \) subject to cardinality constraint \( k \).

In the above equation, the data-dependent approximation ratio involves the optimal solution \( F_\sigma^* \), which may not be obtained in polynomial time. Hence, in practice, we have \( \sigma(F_{\text{app}}) \) bounded by the term \( \frac{\sigma(F_\sigma)}{\nu(F_\sigma)} \) rather than \( \max \{ \frac{\sigma(F_\sigma)}{\nu(F_\sigma)}, \frac{\mu(F_\nu)}{\sigma(F_\sigma)}, \sigma(F_\sigma) \} \), that is \( \sigma(F_{\text{app}}) \geq \frac{\sigma(F_\sigma)}{\nu(F_\sigma)} \cdot (1 - \frac{1}{e}) \cdot \sigma(F_\sigma) \).

The real challenge of solving our MSC problem lies in finding the submodular function \( \nu \) and \( \mu \) that upper and lower bounds \( \sigma \) everywhere. Furthermore, the achieved theoretical approximation ratio is also closely related to how tightly \( \sigma \) is bounded by \( \nu \) and \( \mu \). We now present the upper and lower bound function in the next two subsections.

1) Lower Bound Function: Recall that function \( \sigma \) maps \( F \) to the number of pairs in \( S \) that meet the distance requirement. To construct the lower bound function \( \mu \), we add one additional restriction on top of \( \sigma \), i.e., each path can use at most one shortcut edge. After adding this restriction, for the pairs that originally go through multiple shortcut edges in their shortest path, their pairwise distance increases. Thus, by this extra restriction, less number of important social pairs meet the distance requirement. Accordingly, \( \mu \) lower bounds \( \sigma \) everywhere.

Under the constraint that each pair in \( S \) can use at most one shortcut edge, for an arbitrary shortcut edge \( f_{i,j} \), let set \( S_{i,j} \) denote the pairs in \( S \) that satisfy the distance requirement after adding \( f_{i,j} \) to \( G \). For any set of shortcut edges \( X \), let \( S_X \) denote the set of pairs that meet the distance requirement after adding \( X \) to \( G \). Then, we have \( |X| = |S_X| \) and \( S_X = \bigcup_{f_{i,j} \in X} S_{i,j} \). Accordingly, for any two sets of shortcut edges \( X \) and \( Y \) with \( X \subseteq Y \subseteq V \times V \), we have \( S_X \subseteq S_Y \). Consider adding one more shortcut edge \( f \in V \times V \). The number of overlapped pairs between \( S_X \) and \( S_{\{f\}} \) is less than or equal to the number of overlapped pairs between \( S_Y \) and \( S_{\{f\}} \). Hence, we have \( \mu(\{f\} \cup X) - \mu(X) \geq \mu(\{f\} \cup Y) - \mu(Y) \), and \( \mu \) is submodular.

2) Upper Bound Function: The upper bound function is selected based on the maximum coverage problem. Specifically, consider a set of shortcut edges \( F \), and a pair \( \{u_i, w_i\} \in S \). We say node \( u_i \) (\( w_i \)) is covered by \( F \), if there exists an endpoint \( v_j \) of a shortcut edge in \( F \) such that the distance between \( v_j \) and \( u_i \) (\( w_i \)) is less than \( d_i \). Recall that the distance between any pair in \( S \) is greater than \( d_i \). Then for any pair \( \{u_i, w_i\} \in S \) that meets the distance requirement, both \( u_i \) and \( w_i \) must be covered by \( F \). For any pair \( \{u_i, w_i\} \in S \), define the weight of \( u_i \) (\( w_i \)) to be half of the number of times \( u_i \) (\( w_i \)) appears in \( S \). For example, when \( S = \{\{u_1, w_1\}, \{u_1, w_2\}\} \), \( u_1 \) has a weight of 1, while both \( u_1 \) and \( w_2 \) have a weight of 0.5. Intuitively, the weights are defined in this way, such that when the nodes of the social pairs in \( S \) are all distinct, the sum weight of all nodes in node pairs that meet the distance requirement is exactly the number of pairs that meet the distance requirement. Therefore, function \( \nu(F) \) is defined as the sum weight of all nodes in important node pairs that are covered by \( F \).

We first show that \( \nu \) upper bounds \( \sigma \) everywhere. We begin by considering a very simple case: suppose for one specific \( F \), only one pair \( \{u_i, w_i\} \in S \) meets the distance requirement. Then at least \( u_i \) and \( w_i \) are covered by \( F \), and \( \nu(F) \) must be larger than or equal to the sum weight of \( u_i \) and \( w_i \). That is, \( \nu(F) \geq 0.5 + 0.5 = 1 \). Since only one pair meets the distance requirement, we have \( \nu(F) = 1 \) and \( \nu(F) \geq \sigma(F) \).

Furthermore, consider an arbitrary set of shortcut edges \( F \). For all the pairs that meet the distance requirement, let \( \kappa(F) \) denote the sum weight of all nodes in these pairs. Based on the definition of node weight, we have \( \kappa(F) \geq \sigma(F) \). Also recall \( F \) at least covers the nodes in the pairs that meet the distance requirement; therefore, \( \nu(F) \geq \kappa(F) \geq \sigma(F) \).

Then we show function \( \nu \) is submodular. Note that function \( \nu \) corresponds to the weighted maximum coverage problem. In Subsection IV-B, we have proved the submodularity of the maximum coverage problem (or equivalently the MSC-CN problem). Since the maximum coverage problem is a special case of the weighted maximum coverage problem, the previous proof can be easily extended here. Hence, function \( \nu \) is submodular.

Therefore, we can use submodular function \( \mu \) and \( \nu \) to derive a performance-guaranteed solution to solve non-submodular MSC.
Algorithm 1: Evolutionary Algorithm (EA)

input: Graph $G$, set of important social pairs $S$, number of shortcut edge placements $k$
output: Shortcut edge placement $F$

1: $F \leftarrow \emptyset$, $P \leftarrow \{F\}$;
2: $\tau \leftarrow 0$;
3: while $\tau \leq \tau_0$ ($\tau$: number of iterations) do
4:   foreach $f_{i,j}$ with $i, j \in V$ and $i \neq j$ do
5:      if $f_{i,j} \in F$ then
6:         Remove $f_{i,j}$ from $F$ with probability $\frac{2}{n(n-1)}$;
7:      else
8:         Add $f_{i,j}$ to $F$ with probability $\frac{2}{n(n-1)}$;
9:   end
10:  $F' \leftarrow P \cup \{F\}$;
11:  foreach $F' \in P$ do
12:     if $\sigma(F') \leq \sigma(F)$ and $|F'| \geq |F|$ then
13:        $P \leftarrow P \setminus \{F\}$;
14:   end
15:  end
16:  $\tau \leftarrow \tau + 1$;
17:  $F \leftarrow \arg \max_{F \in P, |F| \leq k} \sigma(F)$;

C. Evolutionary Algorithm (EA)

In this section, we present alternative approximation strategies based on evolutionary algorithms [11]. Generally speaking, evolutionary algorithms belong to randomized metaheuristic optimization algorithms. They start from several randomized solutions, and iteratively improve upon existing solutions to approach the optimal solution.

Specifically, we consider a Global Simple Evolutionary Multi-objective Optimizer (GSEMO) [20] for non-submodular problems. We modify GSEMO, and apply it to our MSC problem. As the name suggests, the number of objectives is usually more than 1. To apply GSEMO, we target the following two objective functions: (i) maximize $\sigma(F)$ without the cardinality constraint on $F$, and (ii) maximize $-|F|$. Clearly, for such multi-objective optimization, our cardinality constraint on $F$ is transformed to Objective-(ii). Then the proposed algorithm works as follows. Initially, let the candidate solution set contain $F = \emptyset$ (i.e., $F$ contains no shortcut edges which is the best solution for Objective-(i)). Based on existing solutions in the candidate solution set, the algorithm iteratively generates new solutions, and add this solution to the candidate solution set if this solution is not worse than an existing solution in $P$ from the perspective of both Objective-(i) and Objective-(ii). This evolutionary algorithm is presented in Algorithm 1.

In Algorithm 1, the edge placement (lines 4–19) is adjusted $r$ times, and line 22 outputs the best solution from the candidate set as the final output. Apparently, increasing $r$ improves the performance of the algorithm. In each iteration, a random solution $F$ is selected from the solution set $P$ by line 4. Then, for any shortcut edge $f_{i,j}$, if $F$ already contains $f_{i,j}$, it is removed from $F$ with probability $\frac{2}{n(n-1)}$; otherwise, $f_{i,j}$ is added to $F$ with probability $\frac{2}{n(n-1)}$. With this new solution $F$ obtained by lines 5–11, we then update the solution set $P$. If there exists a solution $F' \in P$ that satisfies $\sigma(F') \geq \sigma(F)$ and $|F'| \leq |F|$, i.e., better than $F$ in terms of both objectives, $F$ is discarded. Otherwise, $F$ is added to the solution set $P$, and any $F' \in P$ that is worse than $F$ (i.e., $\sigma(F') \leq \sigma(F)$ and $|F'| \geq |F|$) is removed from $P$.

Recall that $\mu(F)$ and $\nu(F)$ are the lower and upper bound functions of $\sigma$. We have the following two theorems that can theoretically bound the performance of Algorithm 1.

Theorem 6. For any shortcut edge placement $X, Y$ where $X \subseteq Y$, and any shortcut edge $f \notin Y$, let $\epsilon$ be

$$\epsilon = \max_{X,f} \left( \nu(X \cup f) - \nu(X) - \mu(Y \cup f) \right).$$

Then $\epsilon \geq 0$, and the expected number of iterations in Algorithm 1 (i.e., $r$) to obtain a feasible solution $F$ that satisfies $\sigma(F) \geq (1 - \frac{1}{\epsilon})(\sigma(F^*) - \epsilon k)$ is $O(n^2k)$, where $F^*$ is the optimal solution to MSC.

Proof. Since $\sigma(F) \leq \sigma(X \cup \{f\})$, $\sigma(X) \leq \mu(X \cup \{f\}) - \nu(X)$. Combining the above inequality with the fact that $\mu$ and $\nu$ are submodular leads to

$$\sigma(X \cup \{f\}) - \sigma(X) \geq \mu(X \cup \{f\}) - \nu(X).$$

Since $\nu(Y \cup \{f\}) - \mu(Y) \geq \sigma(Y \cup \{f\}) - \sigma(Y)$ always holds, combining this inequality with inequality (6) leads to

$$\sigma(X \cup \{f\}) - \sigma(X) \geq \sigma(Y \cup \{f\}) - \sigma(Y) - \epsilon.$$

If $\epsilon < 0$, then $\sigma(F)$ becomes submodular, contradicting the fact that $\sigma(F)$ is not submodular; therefore, $\epsilon \geq 0$. Based on the results in Theorem 2 of [20], and note that Algorithm 1 starts from the initial solution $F = \emptyset$ rather than a random solution, it can be derived that we can achieve the desired performance in the theorem after $O(n^2k)$ iterations on average.

Alternatively, we can bound the performance of Algorithm 1 in a different form.

Theorem 7. For any feasible shortcut edge placement $F$, let $\epsilon$ be

$$\epsilon = \max_{F} \left( \frac{\nu(F) - \mu(F)}{\nu(F) + \mu(F)} \right).$$

Then $0 \leq \epsilon \leq 1$, and the expected number of iterations in Algorithm 1 to achieve a feasible solution $F$ that satisfies $\sigma(F) \geq \frac{1}{1 + \epsilon} \left( 1 - e^{-\frac{1}{\epsilon^2}} \right)^k \cdot \sigma(F^*)$ is $O(n^2k)$, where $F^*$ is the optimal solution to MSC.

Proof. Let $g(F) = \frac{\mu(F) + \nu(F)}{2}$, where $\mu(F)$ and $\nu(F)$ are defined in Section V-B. Then, we have $(1 - \epsilon) \cdot g(F) \leq \sigma(F) \leq (1 + \epsilon) \cdot g(F)$. Note that the sum of two submodular functions is submodular, hence function $g(F)$ is submodular.
Then, combining the value of $\epsilon$ with Theorem 4 in [20] yields the results in the theorem.

Theorems 6 and 7 bound the performance of EA by two different approaches. The parameter $\epsilon$ (with different forms) in both Theorems 6 and 7 quantifies the extent that $\sigma(F)$ is far way from being submodular, i.e., smaller $\epsilon$ indicates higher closeness to a submodular problem. As an extreme case, if $\epsilon = 0$ (although not achievable), both Theorems 6 and 7 show that $\sigma(F)$ is reduced to a submodular problem, achieving $(1 - e^{-1})$-approximation ratio via greedy search.

D. Adaptive Evolutionary Algorithm (AEA)

Although the performance of Algorithm 1 can be theoretically bounded, it may not perform well in practice. To be more specific, since the shortcut edge placement is randomly adjusted in each iteration, it may take many iterations for the existing solutions to be improved. In other words, the exploration policy for a new solution is randomized in Algorithm 1 without any preference. To address this issue, we present an Adaptive Evolutionary Algorithm (AEA), to balance the randomized and strategic solution generations by a tunable parameter to improve the performance.

AEA is presented in Algorithm 2. Similar to Algorithm 1, lines 4–21 iteratively adjust the shortcut edge placement, and line 24 outputs the best solution from the candidate set. Algorithm 2 is different from Algorithm 1 in the way how new solutions are explored. Specifically, a link placement $F$ with $|F| = k$ is initially generated, and the candidate solution set $P$ is set to $P = \{F\}$ (lines 1–2). Then, the algorithm iteratively generates a new solution based on an arbitrary solution $F$ in $P$ by either one of the following two different approaches. (1) lines 11–12. With probability $\delta$ (a tunable parameter, $\delta$ is close to 0), the new solution is generated randomly, where one random shortcut edge is removed from $F$, and another random shortcut edge is added to $F$. (2) lines 7–9. With probability $1 - \delta$ (higher probability), the new solution is generated as follows: the shortcut edge $f \in F$ that maximizes $\sigma(F \setminus \{f\})$ is removed from $F$, and the shortcut edge $f' \notin F$ that maximizes $\sigma(F \cup \{f'\})$ is added to $F$. Unlike Algorithm 1, the newly generated solutions in Algorithm 2 always satisfies the cardinality constraint $k$, thereby yielding only one objective in AEA (but two objectives in EA), i.e., the objective of MSC, and thus all generated solutions are feasible solutions. Moreover, the candidate solution set $P$ in AEA only contains at most $l$ (also a tunable parameter) feasible solutions, a new solution replaces a solution $F'$ in $P$ if it maintains more social connections than $F'$ (lines 17–20). Note that in lines 17–20, at most one solution $F'$ is replaced by the new solution. We have $P$ contain $l$ feasible solutions to enable the diversity of the solutions, as a new solution generated based on an inferior solution in $P$ may also outperform many other solutions.

In general, AEA (Algorithm 2) improves EA (Algorithm 1) in two aspects. First, the adjustment of shortcut edge placement in AEA mainly follows a greedy manner, where the shortcut edge that maintains the minimum number of social pairs is removed, and the shortcut edge that maintains the maximum number of social pairs is added. Hence, the link adjustment process in AEA is more efficient than the random adjustment in EA. Second, AEA only focuses on feasible solutions (i.e., set $P$ only contains feasible solutions), which helps avoid the unnecessary iterations on infeasible solutions in EA.

VI. HANDLING DYNAMIC NETWORKS

In practice, wireless link conditions, network topologies, and important social pairs may change over time. In this section, we discuss how to derive robust shortcut edge placement algorithms for such dynamic networks. Regarding network dynamicity, there are existing works that predict topological changes using node mobility patterns [21], [22], and social pair changes using social network evolvement [23]. Given the prediction capabilities, a dynamic network can then be modeled as a series of topologies $G_d = \{G_1, G_2, \ldots, G_T\}$ for time instances from 1 to $T$, each associated with its own set of important social pairs. Depending on the network dynamic level, neighboring time instances do not necessarily correspond to a fixed time interval. We assume that such dynamic topologies and social pairs are given by the above prediction techniques, and improving the prediction accuracy is beyond the scope of this paper. With such dynamic network model, we show that the algorithms proposed in previous sections can be applied to dynamic networks, while sustaining theoretical performance guarantees.

---

**Algorithm 2:** Adaptive Evolutionary Algorithm (AEA)

**input**: Graph $G$, set of important social pairs $S$, number of shortcut edges $k$

**output**: Shortcut edge placement $F$

1. $P \leftarrow \{F\}$, $\tau = 0$
2. while $\tau \leq \tau$ (r: number of iterations) do
3.   Choose $F$ from $P$ uniformly at random;
4.   $\eta \leftarrow$ random number between 0 and 1;
5.   if $\eta \leq 1 - \delta$ (\(\delta\) is a probability, close to 0) then
6.     Find shortcut edge $f \in F$ that maximizes $\sigma(F \setminus \{f\})$;
7.     $F \leftarrow F \setminus \{f\}$;
8.   else
9.     Find shortcut edge $f' \notin F$ that maximizes $\sigma(F \cup \{f'\})$;
10.    $F \leftarrow F \cup \{f'\}$;
11. end
12. Remove $f \in F$ from random;
13. Add $f \notin F$ to random;
14. end
15. if $|P| < l$ (l: size of the candidate solution set) then
16.   $P \leftarrow P \cup \{F\}$;
17. else
18.   Find $F' \in P$ that minimizes $\sigma(F')$;
19.   if $\sigma(F') < \sigma(F)$ then
20.     $P \leftarrow P \setminus \{F\}$, $P \leftarrow P \cup \{F'\}$;
21. end
22. $\tau \leftarrow \tau + 1$;
23. end
24. $F \leftarrow \arg \max_{F \in P} \sigma(F)$;
1) Objective in Dynamic Networks: Given the series of topologies \( G_i \), let function \( \sigma_i(F) \) denote the number of maintained social connections in topology \( G_i \) under edge placement \( F \). Then the objective function \( \sigma(F) \) is extended as the total number of maintained social connections across all time instances, i.e., \( \sigma(F) = \sum_{i=1}^{T} \sigma_i(F) \). Since the previous edge placement in a single network is a special case of that in dynamic networks, MSC in dynamic networks is also NP-hard.

2) Approximation Algorithm: For dynamic networks, the approximation algorithm can be reapplied as follows: Let \( \mu_i(F) \) and \( \nu_i(F) \) denote the lower and upper bound function for \( \sigma_i(F) \) in topology \( G_i \), based on the definitions in Sections V-B1 and V-B2. Let \( \mu(F) = \sum_{i=1}^{T} \mu_i(F) \) and \( \nu(F) = \sum_{i=1}^{T} \nu_i(F) \). It is easy to show under the extended definitions of \( \sigma, \mu, \) and \( \nu \), we still have \( \mu(F) \leq \sigma(F) \leq \nu(F) \). Since the sum of submodular functions is still submodular, both \( \mu \) and \( \nu \) are submodular functions. In this way, the results in Eq. (5) also apply here.

3) Evolutionary and Adaptive Evolutionary Algorithms: Similarly, under the extended objective in dynamic networks, EA and AEA can be reapplied. Moreover, using the extended \( \mu \) and \( \nu \), the theoretical results for EA remain valid.

VII. PERFORMANCE EVALUATIONS

A. Evaluation Setup

1) Dataset: To evaluate the performance of the proposed approximation algorithms for the MSC problem, we perform extensive evaluations on synthetic graphs as well as a real-world social network dataset from the SNAP project [24]. For synthetic graphs, we select the Random Geometric (RG) model, which resembles a social network by spontaneously demonstrating the community structure and displaying the degree assortativity. In a RG graph, nodes are uniformly distributed in a unit square. Two nodes are connected by an edge if their distance is smaller than a threshold.

For the real-world social network dataset, we use the dataset collected from a location-based social network called Gowalla [24], where users share their locations by check-ins. The dataset contains 196,591 users, 950,327 undirected friendships, and 6,442,890 check-ins of these users over the period of Feb. 2009 - Oct. 2010. In our simulations, we focus on the nodes that have a check-in between 6pm and midnight on Oct. 1st 2010, near Austin, Texas. Two nodes are connected by an edge, if their distance is less than 200 meters based on the locations of their check-ins.

2) Topologies for MSC in Dynamic Environment: For MSC in dynamic networks, we validate the performance of the proposed algorithms based on a tactical network. Specifically, we perform simulations based on mobility traces generated at the Network Science Research Laboratory of the US Army Research laboratory [25]. The traces record the locations of 90 nodes belonging to 7 groups during a tactical operation, where each node update their locations periodically.

3) Link Failure Model and Social Pair Selection: For all network topologies described above, the failure probability of each edge is set to be proportional to the geographical distance between the two endpoints of this edge. The important social pairs are randomly selected from the node pairs with path failure probability larger than the threshold \( p_t \).

TABLE I: \( \frac{\sigma(F)}{\nu(F)} \) for Random Geometric graph

<table>
<thead>
<tr>
<th>( p_t )</th>
<th>0.04</th>
<th>0.08</th>
<th>0.11</th>
<th>0.14</th>
<th>0.18</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \nu )</td>
<td>0.3636</td>
<td>0.1334</td>
<td>0.3000</td>
<td>0.2727</td>
<td>0.4348</td>
</tr>
<tr>
<td>4</td>
<td>0.2353</td>
<td>0.0909</td>
<td>0.2222</td>
<td>0.1935</td>
<td>0.3750</td>
</tr>
<tr>
<td>6</td>
<td>0.1905</td>
<td>0.0769</td>
<td>0.1935</td>
<td>0.1714</td>
<td>0.3333</td>
</tr>
<tr>
<td>8</td>
<td>0.1600</td>
<td>0.0667</td>
<td>0.1714</td>
<td>0.1538</td>
<td>0.3000</td>
</tr>
<tr>
<td>10</td>
<td>0.1379</td>
<td>0.0588</td>
<td>0.1538</td>
<td>0.1500</td>
<td>0.3000</td>
</tr>
</tbody>
</table>

TABLE II: \( \frac{\sigma(F^*)}{\nu(F^*)} \) for Gowalla Dataset

<table>
<thead>
<tr>
<th>( p_t )</th>
<th>0.23</th>
<th>0.27</th>
<th>0.31</th>
<th>0.35</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \nu )</td>
<td>0.2128</td>
<td>0.2577</td>
<td>0.3625</td>
<td>0.5707</td>
</tr>
<tr>
<td>4</td>
<td>0.1954</td>
<td>0.2269</td>
<td>0.3291</td>
<td>0.5330</td>
</tr>
<tr>
<td>6</td>
<td>0.1762</td>
<td>0.2269</td>
<td>0.3291</td>
<td>0.5330</td>
</tr>
<tr>
<td>8</td>
<td>0.1683</td>
<td>0.2067</td>
<td>0.3135</td>
<td>0.5128</td>
</tr>
<tr>
<td>10</td>
<td>0.1683</td>
<td>0.2024</td>
<td>0.3077</td>
<td>0.5081</td>
</tr>
</tbody>
</table>

B. Approximation Ratio of Approximation Algorithm

We present results on the theoretical approximation ratio of proposed approximation algorithm (Section V-B), i.e., tightness of the performance bound on \( F_{mv} \). Specifically, recall that \( F_{mv} \) achieves at least \( \frac{\sigma(F)}{\nu(F)} \cdot (1 - \frac{1}{e}) \)-approximation of the optimal; thus we compute the value of term \( \frac{\sigma(F)}{\nu(F)} \).

Table I presents the theoretical approximation ratio on the RG graph, where \( n = 100, m = 17 \). As reported in Table I, the value of \( \frac{\sigma(F)}{\nu(F)} \) is mostly greater than 0.1, where the largest value is around 0.43 (marked in bold). Table II reports the approximation ratio based on the Gowalla dataset, where the network contains 134 nodes, 1886 edges, 63 important social pairs. As shown in Table II, the value of \( \frac{\sigma(F)}{\nu(F)} \) is greater than 0.2 for most cases, and can be as large as 0.57 (marked in bold). In both Tables I and II, with the increase of \( k \), the approximation ratio decreases. This is because as \( k \) increases, the MSC problem becomes more complicated, and it is more likely for functions \( \mu \) and \( \nu \) (i.e., the lower and upper bound function) to deviate from \( \sigma \).

C. Comparing Approximation Algorithm against Random Selection Solution

In this subsection, we compare the proposed Approximation Algorithm (AA) against a random selection solution. For the random selection solution, we repeat the process of randomly adding \( k \) shortcut edges to the graph for 500 times, and choose the one that maintains the maximum number of social connections as the final solution.

Fig. 1 presents the shortcut edge placement of Approximation Algorithm and the random selection solution based on
D. Comparison of Proposed Solutions

Fig. 3 shows the evaluation results of the proposed algorithms under different \( p_t \) and \( k \). In Fig. 3, the number of iterations \( r = 500 \) for both EA and AEA, the size of the candidate solution set \( l = 10 \) and probability \( \delta = 0.05 \) for AEA.

Fig. 3(a) presents the results on the RG graph when \( n = 100 \) and \( m = 80 \). As expected, the number of maintained social connections increases with the number of shortcut edges. As \( p_t \) increases, the number of maintained social connections also increases. When comparing the proposed three algorithms, AEA and AA significantly outperform EA, while AEA generally performs better than AA, which demonstrates the effectiveness of AEA.

Fig. 3(b) shows the evaluation results on the Gowalla dataset when \( n = 134 \) and \( m = 76 \). Similar to Fig. 3(a), the number of maintained social connections increases with \( p_t \) and \( k \). Note that even a small number of shortcut edges can maintain many important social connections. This is because for this real-world dataset, groups of people may share the same location if they are participating in the same activity (e.g., having dinner in the same restaurant). Then, connecting a shortcut edge between two groups of people can simultaneously maintain several important social connections.

Fig. 4 presents the evaluation results of AEA and EA based on the number of iterations (i.e., \( r \)), given different \( k \). The results of AA, although irrelevant to \( r \), are also shown in the figure for comparison. Fig. 4(a) shows the results on RG graph, when \( n = 100 \), \( m = 80 \), and \( p_t = 0.14 \). For both AEA and EA, increasing \( r \) improves the performance.
of both algorithms. Although AEA performs worse than AA for small $r$, it outperforms AA as $r$ grows. Fig. 4(b) shows the results on the Gowalla Dataset ($n = 134$, $m = 76$ and $p_t = 0.23$). Similar to Fig. 4(a), AEA performs better than AA for large $r$. These results indicate that, while AA is an efficient approximation algorithm with satisfactory performance, AEA performs better at the cost of longer computation time (i.e., larger $r$).

E. Dynamic Networks

We evaluate the performance of the proposed algorithms in dynamic networks based on traces from a tactical network. For parameter settings, $r = 500$ for EA, and $r = 500$, $l = 10$, and $\delta = 0.05$ for AEA.

Fig. 5 compares the performance of the proposed algorithms in terms of the number of maintained social connections when $n = 50$, $m = 30$, and $T = 30$ (i.e., number of time instances). Fig 5(a) shows the effect of the number of shortcut edges, given different $p_t$. Similar to the results of the single topology in Fig. 3, AEA generally performs better than AA, while both AEA and AA significantly outperform EA. For the special case of $k = 20$ and $p_t = 0.11$, AEA and AA perform identically, as both algorithms can maintain almost all social connections in the network.

Under different $k$, Fig 5(b) presents the number of maintained social connections as a function of the number of time instances $T$, when $n = 50$, $p_t = 0.12$, and $m = 30$. As shown in Fig 5(b), the total number of maintained social connections increases with $k$ and $T$. However, when averaged over all time instances, the number of maintained connections in each
topology decreases with increasing $T$. This is because for MSC in dynamic networks, when $T$ increases, the total number of important social pairs across all time instances grows. Then, it is more difficult to maintain a much larger number of important social pairs across all time instances using the same shortcut edge budget. Hence, the average number of maintained social connections in each time instance decreases as $T$ grows.

VIII. CONCLUSIONS

In this paper, we proposed to maintain social connections of important social pairs by proactively placing shortcut edges into the network. We formalized the MSC problem and proved its NP-hardness. Then, we proposed an efficient approximation algorithm by carefully choosing two submodular functions to lower/upper bound the MSC problem. We also proposed two evolutionary algorithms, where the basic idea is to employ both randomized selection and strategic exploration to iteratively adjust the current shortcut edge placement to approach the optimal, and the trade-off is how to balance the portions of randomization and exploration. Extensive evaluations based on both synthetic and real-world social network traces demonstrated the effectiveness of our proposed algorithms.

Although our algorithms were proposed to solve the problem of maintaining social connections, these algorithms could also provide insights into the general shortcut edge addition problems in any graphs.

REFERENCES
