Integrating Networks of Equipotent Nodes

Anastasia Moskvina[1] and Jiamou Liu[2(✉)]

[1] Auckland University of Technology, Auckland, New Zealand
anastasia.moskvina@aut.ac.nz
[2] The University of Auckland, Auckland, New Zealand
jiamou.liu@auckland.ac.nz

Abstract. When two social groups merge, members of both groups should socialize effectively into the merged new entity. In other words, interpersonal ties should be established between the groups to give members appropriate access to resource and information. Viewing a social group as a network, we investigate such integration from a computational perspective. In particular, we assume that the networks have equipotent nodes, which refers to the situation when every member has equal privilege. We introduce the network integration problem: Given two networks, set up links between them so that the integrated network has diameter no more than a fixed value. We propose a few heuristics for solving this problem, study their computational complexity and compare their performance using experimental analysis. The results show that our approach is a feasible way to solve the network integration problem by establishing a small number of edges.

1 Introduction

All social groups evolve through time. When two social groups merge, new relations need to be set up. Take, as an example, a merger between two companies. The success of mergers and acquisitions of companies often hinges on whether firms can socialize employees effectively into the merged new entity [1]. Therefore a big challenge faced by the top managers of both companies is how to *integrate* the two companies to ensure coherence and efficient communication. This paper approaches this challenge from a computational perspective. To motivate our formal framework, we make three assumptions: (1) the integration takes place assuming *equipotency* of nodes; (2) creating weak ties between the networks can be encouraged and forced; and (3) structural properties such as distance provide a measure of effective communication and resource accessibility.

The first condition assumes the networks follow *peer-to-peer* relational dynamic, which refers to social structures where information and resources are distributed. In such a social structure, as discussed by Baker in [3], members have no formal authority over each other, and have equal privileges regardless their roles [5]. Examples of such social groups include volunteer organizations, teams of scientists, and companies that embrace a holacracy management style [16]. Baker claims that in order for such a peer-to-peer network to operate efficiently, there must be

clear and open communication; moreover each individual should be aware of the resources available from other nodes.

The second condition arises from the nature of interpersonal relations. Social networks are usually the result of complex interactions among autonomous individuals whose relationships cannot be simply controlled and forced. Ties between people differ by strength; while strong ties denote frequent interactions which form a basis for trust, weak ties plays an important role in information flow. In business networks, although a firm is seldom in control of strong relationships among its employees [15], it can normally prepare the ground for future weak ties: conferences and meetings, group assignments, special promotions etc. can be instruments of bringing people together.

The third condition discusses how the integrated network provides members with appropriate access to resource and information. *Distance* is an important factor of information dissemination in a network [8]: a network with a small diameter means that members are in general close to each other and information could be passed from one person to any others within a small number of steps [17]. This argument has been used to explain how small-world property – the property that any node is reachable from others via only a few hops – becomes a common feature of most real-world social networks [2].

Extending these ideas, we define *network integration* as the process when one or more edges are established across two existing networks in such a way that the integrated network has a bounded diameter Δ. Furthermore, a new edge always costs effort and time to establish and maintain. Thus, we also want to minimize the number of new edges to be created during the integration process. We propose two heuristics to perform network integration. The first is a naive greedy method that iteratively creates edges to minimize the diameter of the resulting network. The second method separately discusses two cases: (1) When Δ is at least the diameter of the original networks, we create edges by considering center and peripheral nodes in the networks. (2) When Δ is smaller than the original diameter of the original networks, we first reduce the distance between nodes in the respective networks and then apply the procedure in case (1). The experiments verify that, our second heuristic significantly outperforms the first, both in terms of running time, and in terms of the output edge set.

The rest of the paper is organized as follows: Sect. 2 presents the formal framework of network integration and shows that it is a computationally hard problem. Section 3 presents a naive greedy heuristic Naive. Section 4 discusses our Integrate algorithm. Section 5 presents experimental results on our algorithms using both generated and real-world data. Section 6 discusses related works before conclusion in Sect. 7.

2   Preliminaries and Problem Setup

We define a *network* as an undirected unweighted connected graph $G = (V, E)$ where V is a set of nodes and E is a set of (undirected) edges on V. We write an edge $\{u, v\}$ as uv. A *path* (of *length k*) is a sequence of nodes $u_0, u_1, \ldots, u_k$ where

$u_i u_{i+1} \in E$ for any $0 \leq i < k$. The *distance* between u and v, denoted by $\mathsf{dist}(u, v)$, is the length of a shortest path from u to v. The *eccentricity* of u is $\mathsf{ecc}(u) = \max_{v \in V} \mathsf{dist}(u, v)$. The *diameter* of the network G is $\mathsf{diam}(G) = \max_{u \in V} \mathsf{ecc}(u)$. The *radius* $\mathsf{rad}(G)$ of G is $\min_{u \in V} \mathsf{ecc}(u)$. For two sets V_1, V_2, we use $V_1 \otimes V_2$ to denote the set of all edges $\{uv \mid u \in V_1, v \in V_2\}$.

Definition 1. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two networks. Fix a set of edges $E \subseteq V_1 \otimes V_2$. We define the integrated network of G_1, G_2 with edge set E as the graph $G_1 \oplus_E G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup E)$.*

When integrating two organizations, each person normally has constraints over who he or she may connect to; this is determined largely by *privilege*, i.e., the type of social inequality created from difference in positions, titles, ranks, etc. [5]. In this paper we focus on the simpler case of social networks with equipotent nodes, and therefore assume all nodes have unbounded and equal privilege. Take an integer $\Delta \geq 1$, we propose the *network integration problem* $\mathsf{NI}_\Delta(G_1, G_2)$:

INPUT. Two networks $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ where $V_1 \cap V_2 = \varnothing$.

OUTPUT. A set $E \subseteq V_1 \otimes V_2$ such that $\mathsf{diam}(G_1 \oplus_E G_2) \leq \Delta$.

In the rest of the paper we investigate $\mathsf{NI}_\Delta(G_1, G_2)$ on two networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ where $V_1 \cap V_2 = \varnothing$. The problem naturally depends on the value of Δ. When $\Delta = 1$, it is easy to see that $\mathsf{NI}_\Delta(G_1, G_2)$ has a solution if and only if both networks G_1, G_2 are complete. When $\Delta \geq 2$, since $G_1 \oplus_{V_1 \otimes V_2} G_2$ has diameter 2, $\mathsf{NI}_\Delta(G_1, G_2)$ guarantees to have a solution.

Throughout, we assume $\Delta \geq 2$. We are interested in a solution E to the problem $\mathsf{NI}_\Delta(G_1, G_2)$ that contains the least number of edges; such an E is called an *optimal solution*. The brute-force way of finding optimal solutions for $\mathsf{NI}_\Delta(G_1, G_2)$ examines all possible sets of edges until it finds a required solution set E. This will take time $2^{O(|V_1| \cdot |V_2|)}$. In fact, obtaining optimal solutions is computationally-hard; the following theorem implies that this problem is unlikely to be polynomial-time solvable.

Theorem 2. *The problem of deciding, given two graph G_1, G_2 and an integer $k > 0$, whether $\mathsf{NI}_\Delta(G_1, G_2)$ has a solution set E with cardinality $\leq k$, is hard for $\mathsf{W}[2]$, the second level of the W-hierarchy.*

Proof. Let $G = (V, E)$ be a graph. A *distance-r dominating set* of G is a set S of nodes such that for all $u \in V$, there is some $v \in S$ with $\mathsf{dist}(u, v) \leq r$. As shown in [11], finding the smallest distance-r dominating set in G with diameter $r + 1$ is complete for $\mathsf{W}[2]$ (for any fixed r). In fact, the $\mathsf{W}[2]$-hardness also holds if G is *diametrically uniform*, i.e., if all nodes have the same eccentricity.

Now let $G_1 = (V_1, E_1)$ be a diametrically uniform graph with diameter $\Delta \geq 2$ and let G_2 be a graph that contains only a single node $\{u\}$. For any distance-$(\Delta - 1)$ dominating set $S \subseteq V_1$, the set of edge $S \otimes \{u\}$ is a solution of $\mathsf{NI}_\Delta(G_1, G_2)$. Conversely, suppose $S \subseteq V_1$ is not distance-$(\Delta - 1)$ dominating. Then there is a node $w \in V_1$ that is at distance at least Δ away from any node $v \in S$. This means that $\mathsf{dist}(w, u)$ in the integrated network is at least $\Delta + 1$ and S is not

a solution of $\mathsf{NI}_\Delta(G_1, G_2)$. Thus $\mathsf{NI}_\Delta(G_1, G_2)$ has a size-k solution if and only if G_1 has a size-k distance-$(\Delta - 1)$ dominating set. □

3 The Naive Greedy Method

We thus turn to heuristics for finding small solution sets of $\mathsf{NI}_\Delta(G_1, G_2)$. As a first step we introduce a greedy heuristic that approximates a solution.

Definition 3. *A set of edges $E \subseteq V_1 \otimes V_2$ is called a* naive greedy set *if we can write it as $\{e_1, \ldots, e_\ell\}$ such that for all $1 \leq i \leq \ell$, and $e' \in (V_1 \otimes V_2) \setminus \{e_1, \ldots, e_{i-1}\}$, $\mathsf{diam}(G_1 \oplus_{\{e_1, \ldots, e_i\}} G_2) \leq \mathsf{diam}(G_1 \oplus_{\{e_1, \ldots, e_{i-1}, e'\}} G_2)$. A naive greedy solution to $\mathsf{NI}_\Delta(G_1, G_2)$ is a solution that is a naive greedy set.*

As its name suggests, a naive greedy set can be constructed incrementally using a greedy strategy that locally optimizes diameter in the integrated network. Naive greedy solutions to $\mathsf{NI}_\Delta(G_1, G_2)$ are not necessarily optimal, and vice versa:

Example. Let both G_1 and G_2 be paths of length 5, i.e., G_1 contains nodes $a_1, \ldots, a_5$ while G_2 contains nodes $b_1, \ldots, b_5$ with edges $a_i a_{i+1}$, $b_i b_{i+1}$ for any $1 \leq i < 5$. Suppose $\Delta = 3$. The only optimal solution E contains four edges, i.e., $E = \{a_1 b_3, a_3 b_1, a_3 b_5, a_5 b_3\}$. However, for any edge $e \in E$, $\mathsf{diam}(G_1 \oplus_{\{e\}} G_2) = 7$, while $\mathsf{diam}(G_1 \oplus_{\{a_3 b_3\}} G_2) = 5$. Thus E is not a naive greedy solution, nor will any naive greedy solution be optimal.

Theorem 4. *There exists an algorithm $\mathsf{Naive}_\Delta(G_1, G_2)$ that runs in time $O(n^6)$ and computes a naive greedy solution for $\mathsf{NI}_\Delta(G_1, G_2)$ where $n = |V_1 \cup V_2|$.*

Proof. The algorithm $\mathsf{Naive}_\Delta(G_1, G_2)$ iteratively adds edges $e_1, e_2, \ldots$ to the solution set E. It also computes a matrix $\mathsf{D}: (V_1 \cup V_2)^2 \to \mathbb{N}$ that represents the distance between nodes in the current integrated graph. See Procedure 1

Since $\Delta \geq 2$, the algorithm will terminate. Furthermore, the set of edges created by the algorithm is a naive greedy solution. At each iteration, computing each matrix D_e takes time $O(n^2)$; computing F takes $O(n^2)$. Since there are $O(n^2)$ edges in $(V_1 \otimes V_2) \setminus E_i$, this iteration runs in $O(n^4)$. Since there are at most n^2 iterations, the algorithm takes times $O(n^6)$. □

We remark that when $\Delta > 2$, the maximum number of edges required is $O(n)$, and hence $\mathsf{Naive}_\Delta(G_1, G_2)$ will take $O(n^5)$. The algorithm $\mathsf{Naive}_\Delta(G_1, G_2)$ is still too inefficient in most practical cases and hence in subsequent sections we discuss more efficient heuristics for $\mathsf{NI}_\Delta(G_1, G_2)$.

4 Efficient Algorithms for $\mathsf{NI}_\Delta(G_1, G_2)$

We separately discuss two cases: (1) when the integrated network's diameter is at least the diameters of the given networks, i.e. $\Delta \geq \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\}$; and (2) when we improve the diameter, i.e. $\Delta < \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\}$.

Procedure 1. $\mathsf{Naive}_\Delta(G_1, G_2)$; Output E

Initialize D (for the disjoint union of G_1, G_2); Set $E \coloneqq \varnothing$
while $\mathsf{diam}\left(G_1 \oplus_E G_2\right) > \Delta$ **do**
    for $e \coloneqq xy \in (V_1 \otimes V_2) \setminus E$ **do**
        for $(u, v) \in (V_1 \cup V_2)^2$ **do**   ▷ define a temporary $D_e : (V_1 \cup V_2)^2 \to \mathbb{N}$
             $D_e(u, v) \coloneqq \min\{D_i(u, v), D_i(u, x) + D_i(y, v) + 1\}$
        end for
        Set $\mathsf{diam}_e \coloneqq \max\{D_e(u, v) \colon (u, v) \in (V_1 \cup V_2)^2\}$.
    end for
    Set $F \coloneqq \{e \in (V_1 \otimes V_2) \setminus E_i \mid \mathsf{diam}_e \leq \mathsf{diam}_{e'} \text{ for all } e' \in (V_1 \otimes V_2) \setminus E_i\}$.
    Pick a random edge $e_i \in F$ and set $D \coloneqq D_{e_i}$, $E \coloneqq E \cup \{e_i\}$
end while

4.1   The Case When $\Delta \geq \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\}$

Firstly, when integrating two networks, it makes sense to establish a link between the most central persons in the networks, as they have the closest proximity to other nodes. Furthermore, if x, y are nodes that are furthest apart in the integrated network, they are unlikely to communicate effectively thanks to their shear distance; this, in a certain sense, represents a form of *structural hole* [4]. Hence in integrating these two networks, it makes sense to connect x, y by an edge. Formally, the *center* $C(G)$ of a graph $G = (V, E)$ is the set of all nodes that have the least eccentricity, i.e., $C(G) = \{v \in V \mid \mathsf{ecc}(v) = \mathsf{rad}(G)\}$. A pair of nodes (x, y) in G forms a *peripheral pair*, denoted by $(x, y) \in \mathsf{P}(G)$, if $\mathsf{dist}(x, y) = \mathsf{diam}(G)$. Our heuristic first creates an edge between two nodes that are in $C(G_1)$ and $C(G_2)$ respectively, and then iteratively "bridges" peripheral pairs.

Definition 5. *A set $E \subseteq V_1 \otimes V_2$ is called a* center-periphery set *if we can write it as $\{e_0, \ldots, e_\ell\}$ such that (1) $e_0 \in C(G_1) \otimes C(G_2)$; and (2) for all $1 \leq i \leq \ell$, $e_i \in \mathsf{P}\left(G_1 \oplus_{\{e_0, e_1, \ldots, e_{i-1}\}} G_2\right)$. A* center-periphery solution *is a solution to $\mathsf{NI}_\Delta(G_1, G_2)$ that is also a center-periphery set.*

Clearly, if $\Delta > \mathsf{rad}(G_1) + \mathsf{rad}(G_2)$, then for any $uv \in C(G_1) \otimes C(G_2)$, we have $\mathsf{diam}\left(G_1 \oplus_{\{uv\}} G_2\right) \leq \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2), \mathsf{rad}(G_1) + \mathsf{rad}(G_2) + 1\} \leq \Delta$. Thus $\{uv\}$ forms a solution of $\mathsf{NI}_\Delta(G_1, G_2)$. In this case, center-periphery solutions coincide with optimal solutions.

Theorem 6. *There exists an algorithm $\mathsf{CtrPer}_\Delta(G_1, G_2)$ that has $O(n^4)$ running time and computes a center-periphery solution for $\mathsf{NI}_\Delta(G_1, G_2)$ assuming $\Delta \geq \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\}$, where $n = |V_1 \cup V_2|$.*

Proof. The $\mathsf{CtrPer}_\Delta(G_1, G_2)$ algorithm also maintains a matrix D: $(V_1 \cup V_2)^2 \to \mathbb{N}$ such that $D(u, v)$ is the distance between u, v. The eccentricity of each node can be easily extracted from D allowing the algorithm to identify the centers $C(G_1)$ and $C(G_2)$, respectively. The algorithm then iteratively adds edges that connect peripheral pairs in the integrated graph until its diameter becomes at most Δ. See Procedure 2.

Suppose the algorithm creates a set $E \subseteq V_1 \otimes V_2$ and $\mathsf{diam}\,(G_1 \oplus_E G_2) > \Delta$. The algorithm will update the matrix D and then picks (u, v) with the largest $D(u, v)$. By definition of D, (u, v) forms a peripheral pair in $G_1 \oplus_E G_2$. We need to show that uv is a valid edge to add, that is, u, v cannot both lie in one of V_1 and V_2. Indeed, if $\{u, v\} \subseteq V_1$ or $\{u, v\} \subseteq V_2$, then $\mathsf{dist}(u, v) \leq$ $\max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\} \leq \Delta < \mathsf{diam}\,(G_1 \oplus_E G_2)$. Thus $uv \in V_1 \otimes V_2$. Now either $E \cup \{uv\}$ is a solution, or $\mathsf{diam}\,(G_1 \oplus_{E \cup \{uv\}} G_2) > \Delta$. In the latter case the algorithm repeats the iteration to find another peripheral pair. Thus the algorithm will terminate and produce a center-periphery solution to $\mathsf{NI}_\Delta(G_1, G_2)$.

It takes $O(n^3)$ to initialize the matrix D using Floyd-Warshall algorithm. At each iteration, the algorithm takes $O(n^2)$ to update D and finds a peripheral pair. Since there are at most n^2 iterations, the algorithm takes time $O(n^4)$.   □

Procedure 2. $\mathsf{CtrPer}_\Delta(G_1, G_2)$: $\Delta \geq \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\}$; Output E

Initialize the matrix D so that $D(u, v) = \mathsf{dist}(u, v)$ in the un-integrated graphs
Take a node $u \in C(G_1)$ and a node $v \in C(G_2)$
Set $e := uv$, $E := \{e\}$
while $(\mathsf{diam}\,(G_1 \oplus_E G_2) > \Delta)$ **do**
    for $(x, y) \in (V_1 \cup V_2)^2$ **do** $\triangleright$ define $D' : (V_1 \cup V_2)^2 \to \mathbb{N}$
        $D'(x, y) := \min\{D(x, y), D(x, u) + D(v, y) + 1\}$ where $e = uv$
    end for
    $D := D'$ $\triangleright$ update matrix D
    Pick (u, v) with the largest $D(u, v)$. Set $e := uv$ and $E := E \cup \{e\}$
end while

4.2   The Case When $\Delta < \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\}$

When the diameter bound Δ is less than the diameters of the two component networks G_1, G_2, the goal is to *improve* the connectivity of each original network through integration. In other words, the integration should "bring people closer". In this case $\mathsf{CtrPer}_\Delta(G_1, G_2)$ no longer applies as it is possible for both nodes in a peripheral pair to lie in the same component graph G_1 or G_2, forbidding us to create the edge xy. We therefore need to first decrease the distance between nodes in each G_1 and G_2. Suppose a, b are two people in an organization with large distance. When their organization merges with another organization, a and b can be brought closer if they both know a 'third person' c in the other organization, i.e., the ties ac and bc allows a, b to be only 2 steps away.

Definition 7. *Let $E \subseteq V_1 \otimes V_2$ be a set of edges and $i \in \{1, 2\}$. The* diameter *of G_i relative to E is the maximum distance between any two nodes in V_i in the integrated network $G_1 \oplus_E G_2$; we denote this by $\mathsf{diam}_E(G_i)$. A set of edges $E \subseteq V_1 \otimes V_2$ is a Δ-bridge if $\mathsf{diam}_E(G_i) \leq \Delta$ for both $i \in \{1, 2\}$.*

Theorem 8. *For any $\Delta \geq 2$, there exists an algorithm $\mathsf{Bridge}_\Delta(G_1, G_2)$ that runs in time $O(n^4)$ and computes a Δ-bridge E, where $n = |V_1 \cup V_2|$.*

Proof. The algorithm has two phases. In phase $i \in \{1, 2\}$, it makes $\mathsf{diam}_E(G_i) \leq \Delta$. Phase i consists of several iterations; at each iteration, the algorithm takes a pair $(u, v) \in V_i$ with maximum distance and a node $w \in V_{3-i}$, and builds two edges uw and vw. See Procedure 3. Throughout, the algorithm computes and maintains a matrix $D \colon (V_1 \cup V_2)^2 \to \mathbb{N}$ such that $D(u, v)$ is the current distance between nodes u, v. When a pair of new edges uw, vw are added, the new distance $D'(x, y)$ between any pair of nodes $(x, y) \in V_i^2$ is calculated as follows:

$$D'(x, y) = \min\{D(x, y), D(x, u) + D(w, y) + 1, D(x, u) + D(v, y) + 2,$$
$$D(x, v) + D(w, y) + 1, D(x, v) + D(u, y) + 2\} \quad (1)$$

In the worst case, the algorithm adds edges uw, vw for any pair $(u, v) \in V_i^2$ where $i \in \{1, 2\}$. Thus the algorithm terminates in at most n^2 iterations. Finding nodes u, v, w and updating the matrix D at each iteration takes time $O(n^2)$. Therefore the total running time is $O(n^4)$. □

Procedure 3. $\mathsf{Bridge}_\Delta(G_1, G_2)$: $\Delta < \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\}$; Output E

  Initialize the matrix D so that $D(u, v) = \mathsf{dist}(u, v)$ in the un-integrated graphs
  Initialize $E := \varnothing$
  for $i = 1, 2$ **do**   ▷ The two phases
     while $\mathsf{diam}_E(G_i) > \Delta$ **do**
        Take a pair of nodes $u, v \in V_i$ with maximum $D(u, v)$
        Take a node w in V_{3-i}
        $E := E \cup \{uw, wv\}$
        for $(x, y) \in (V_1 \cup V_2)^2$ **do**   ▷ define $D' : (V_1 \cup V_2)^2 \to \mathbb{N}$
           Compute $D'(x, y)$ as in (1)
        end for
        $D := D'$ ▷ update matrix D
     end while
  end for

Remark. Suppose the $\mathsf{Bridge}_\Delta(G_1, G_2)$ algorithm adds edges uw, vw. Here w plays the role as a *bridging node* that links u and v. Naturally, the choice of w affects the performance of the algorithm: by carefully choosing the bridging node w, we may reduce the number of new ties that need to be created. Imagine that G_1, G_2 represent two organizations.

1. To allow smooth flow of information between the two organizations and avoid *information gate keepers*, we should have many bridging nodes in G_2.
2. A node with a higher degree means it has better access to resource and information, and thus is a more appropriate bridging nodes.

Therefore, we introduce the following heuristics to $\mathsf{Bridge}_\Delta(G_1, G_2)$: Suppose the algorithm has selected a set E of edges and picked $(u,v) \in V_i$ where $i \in \{1,2\}$ with the largest $D(u,v)$. To pick a bridging node w:

Heuristic 1. For any node $w \in V_{3-i}$, let $b(w) = |\{v \mid wv \in E\}|$. The chosen bridging node w is taken from $B_i = \{w \in V_{3-i} \mid b(w) \leq b(w') \text{ for all } w' \in V_{3-i}\}$.

Heuristic 2. The chosen bridging node w has the highest degree in B_i.

We now extend $\mathsf{Bridge}_\Delta(G_1, G_2)$ to an algorithm that solves $\mathsf{NI}_\Delta(G_1, G_2)$.

Algorithm 4. $\mathsf{Integrate}_\Delta(G_1, G_2)$; Output E

Run $\mathsf{Bridge}_\Delta(G_1, G_2)$ to obtain a set $E \subseteq V_1 \otimes V_2$

Run $\mathsf{CtrPer}_\Delta(G_1, G_2)$ to add edges to E (instead of building E from scratch)

Theorem 9. *The $\mathsf{Integrate}_\Delta(G_1, G_2)$ algorithm runs in time $O(n^4)$ and computes a solution to $\mathsf{NI}_\Delta(G_1, G_2)$ for any networks G_1, G_2 and $\Delta \geq 2$, where $n = |V_1 \cup V_2|$.*
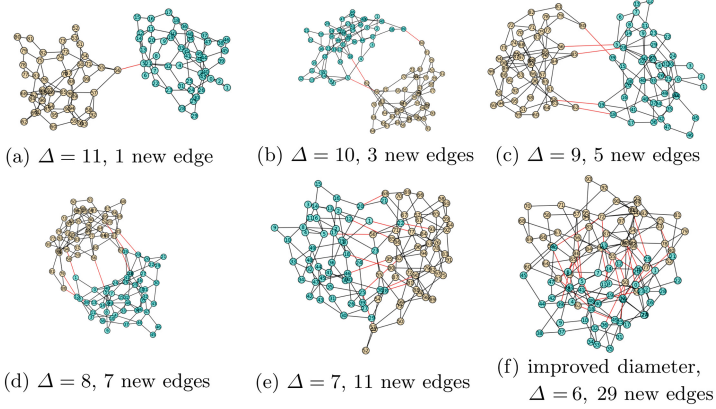
5   Experiments and Case Studies

To test the algorithms, we generate two types of random graphs: the first (NWS) is Newman-Watts-Strogatz's small-world networks, which have small average path lengths and high clustering coefficients [13]. The second (BA) is Barabasi-Albert's preferential attachment model which produces scale-free graphs whose degree distribution of nodes follows a power law [2]. In Figs. 1 and 2, we integrate two graphs of each type using the $\mathsf{Integrate}_\Delta(G_1, G_2)$ algorithm. The statistics for each graph is shown in the table below. For the NWS graphs, Δ ranges from 6 to 11, while for the BA graphs, Δ ranges from 4 to 7. The figures show how the two networks dissolve into each other with decreasing Δ: when very few edges link the two networks, the network exhibits a clear community structure; this, however, becomes less clear as more edges are created.
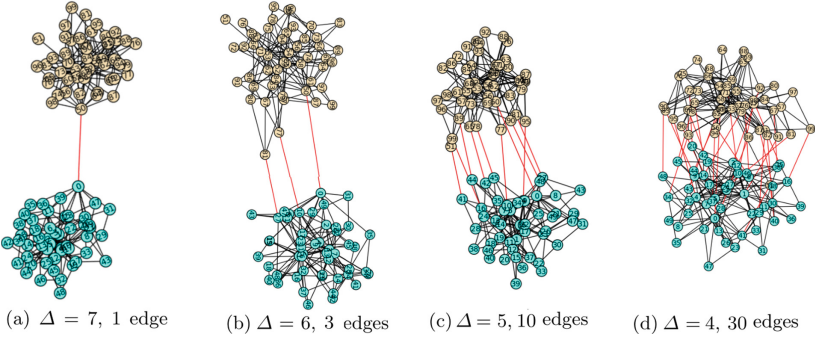
	NWS Graph 1	NWS Graph 2	BA Graph 1	BA Graph 2
Number of nodes/edges	50/77	50/78	50/141	50/141
Diameter/radius	7/5	8/5	4/3	4/3

Experiment 1 (Running times). We implement both algorithms and record their running times on 300 generated NWS and BA networks. The results indicate that $\mathsf{Integrate}_\Delta(G_1, G_2)$ outperforms $\mathsf{Naive}_\Delta(G_1, G_2)$ significantly, with the former runs more than 3000 times faster on networks with 1000 nodes to add a single edge in the solution set. Figure 3 plots how much longer (on average) $\mathsf{Naive}_\Delta(G_1, G_2)$ takes

(a) $\Delta = 11$, 1 new edge   (b) $\Delta = 10$, 3 new edges   (c) $\Delta = 9$, 5 new edges

(d) $\Delta = 8$, 7 new edges   (e) $\Delta = 7$, 11 new edges   (f) improved diameter, $\Delta = 6$, 29 new edges

Fig. 1. Integrating two NWS networks with different Δ

(a) $\Delta = 7$, 1 edge   (b) $\Delta = 6$, 3 edges   (c) $\Delta = 5$, 10 edges   (d) $\Delta = 4$, 30 edges

Fig. 2. Integrating two BA networks with different Δ

to add a single edge to the solution set compared to $\mathsf{Integrate}_\Delta(G_1, G_2)$, against the number of nodes in the networks.
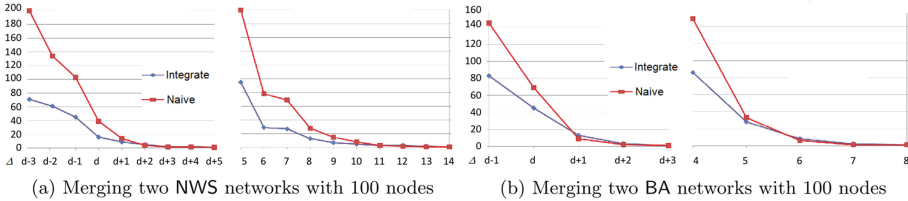
Experiment 2 (Solution size). We compare the output of $\mathsf{Integrate}_\Delta(G_1, G_2)$ against the $\mathsf{Naive}_\Delta(G_1, G_2)$ algorithm. While $\mathsf{Naive}_\Delta(G_1, G_2)$ may output smaller solutions when Δ is large, $\mathsf{Integrate}_\Delta(G_1, G_2)$ is more likely to produce smaller solutions as Δ decreases. Figure 4 plots the percentage of the cases where $\mathsf{Integrate}_\Delta(G_1, G_2)$ returns smaller sets. Note that $\mathsf{Integrate}_\Delta(G_1, G_2)$ almost always returns smaller sets whenever $\Delta < \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\}$. Figure 5 plots the average output size of $\mathsf{Integrate}_\Delta(G_1, G_2)$ and $\mathsf{Naive}_\Delta(G_1, G_2)$, against absolute and relative values of Δ. Here, each graph consists of 100 nodes. Even though $\mathsf{Naive}_\Delta(G_1, G_2)$ may outperform $\mathsf{Integrate}_\Delta(G_1, G_2)$ when Δ is large, the difference is not very significant; as Δ decreases, the advantage of $\mathsf{Integrate}_\Delta(G_1, G_2)$ becomes increasingly significant.
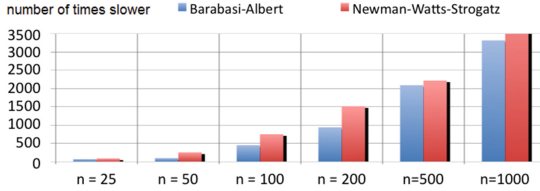
Fig. 3. The number of times $\mathsf{Naive}_\Delta(G_1, G_2)$ runs slower than $\mathsf{Integrate}_\Delta(G_1, G_2)$ (Color figure online)
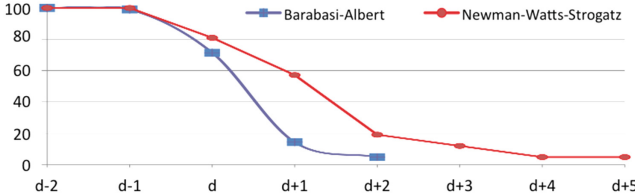
Fig. 4. The probability that $\mathsf{Integrate}_\Delta(G_1, G_2)$ outputs smaller sets with varying $\Delta \in \{d-2, \ldots, d+5\}$ where $d = \max\{\mathsf{diam}(G_1), \mathsf{diam}(G_2)\}$ (Color figure online)

(a) Merging two NWS networks with 100 nodes   (b) Merging two BA networks with 100 nodes

Fig. 5. Comparing the $\mathsf{Integrate}_\Delta(G_1, G_2)$ algorithm and the $\mathsf{Naive}_\Delta(G_1, G_2)$ algorithm: average number of edges with different parameter Δ (Color figure online)

6 Related Works

This paper studies the integration between two social networks of equipotent nodes. This problem relates to several established topics in network science:

Firstly, *strategic network formation* aims to explain how a network evolves in time [7]. A well known example along this line is on the rise of the Medici Family in the XV century [14], which explains how inter-family ties shape political structures. In a certain sense, the network integration problem can be regarded as network formation between two established networks. However, the network formation models are typically about the transformation within a single network, while this paper initiates the perspective of integrating several different networks.

Secondly, the topic of *interdependent networks* aims to model a complex environment where multiple networks interact and form a type of *network of networks* [6]. The networks in such a complex environment are non-homogeneous, i.e., the networks are of different types. For example, one may be interested in

the interdependence between a telecommunication network and a transportation networks and how such interaction affects robustness of the entire infrastructure. The focus here is on robustness of the combined structure: how does a failure in one network affects the other network. Compared to interdependent networks, the problem in this paper involves homogeneous networks and concerns a type of dynamic that 'dissolves' the two networks into one. This is more suitable for the social context discussed in this paper.

A third related area is *link prediction*, which aims to infer potential ties between nodes of a network [9]. Here, most approaches take into account surrounding contexts such as homophily and maximum likelihood.

7   Conclusion and Future Works

This paper amounts to our effort to study integration of social networks from a computational perspective, and is a continuation of our earlier work on network socialization [12], where we study how an individual joins an established network, in order to take an advantageous position in the network.

The simple formulation of the problem means that several natural limitations exist: Firstly, the equipotency assumption restricts us to a special class of social networks. In practice, individuals may have different constraints (e.g. titles, roles, positions, etc.) forbidding certain ties to be created. Hence as a future work we plan to enrich our framework by introducing privileges to nodes and study how networks are integrated with privileged-based constraints on new edges to be forged. Secondly, the paper focuses on optimising the number of new edges between networks, which may not be the most crucial factor when merging social groups. Indeed, every edge is established with certain cost; it may thus be an interesting future work to develop a cost model for the establishment of ties in a social network. Thirdly, the paper concerns with diameter of the integrated network, which is a strong measure on access to resources and information; it may make sense to consider other weaker notions. For example, a more relevant measure of integration may be the distance from any node in one network to any node in the other network, or the average distance between nodes. Lastly, we would like to extend our notion of network integration to more elaborated forms of networks. For example, in [10], a framework of hierarchical networks is defined which incorporates both formal ties in an organization and information ties. This framework allows the definition of a notion of *power* in a hierarchical network. It is then natural to ask how power is affected during integration of two hierarchical networks.

References

1. Aguilera, R., Dencker, J., Yalabik, Z.: Institutions and organizational socialization: integrating employees in cross-border mergers and acquisitions. Int. J. Hum. Resour. Manag. **15**(8), 1357–1372 (2006)

2. Albert, R., Barabasi, A.-L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**, 47 (2002)
3. Baker, M.: Peer-to-Peer Leadership: Why the Network Is the Leader. Berrett-Koehler Publishers, San Francisco (2013)
4. Burt, R.: Structural holes and good ideas. Am. J. Soc. **110**(2), 349–399 (2004)
5. Casella, E.: The Archaeology of Plural and Changing Identities: Beyond Identification. Springer, New York (2005)
6. Danziger, M., Bashan, A., Berezin, Y., Shekhtman, L., Havlin, S.: An introduction to interdependent networks. In: Mladenov, V.M., Ivanov, P.C. (eds.) NDES 2014. Communications in Computer and Information Science, vol. 438, pp. 189–202. Springer, Berlin (2014)
7. Jackson, M.: A survey of models of network formation: stability and efficiency. In: Group Formation in Economics, pp. 11–57. Cambridge University Press, Cambridge (2005)
8. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: densification and shrinking diameters. ACM TKDD **1**(1), 2 (2007)
9. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: Proceedings of CIKM, pp. 556–559 (2003)
10. Liu, J., Moskvina, A.: Hierarchies, ties and power in organisational networks: model and analysis. In: Proceedings of ASONAM, pp. 202–209 (2015)
11. Lokshtanov, D., Misra, N., Philip, G., Ramanujan, M.S., Saurabh, S.: Hardness of r-DOMINATING SET on graphs of diameter $(r + 1)$. In: Gutin, G., Szeider, S. (eds.) IPEC 2013. LNCS, vol. 8246, pp. 255–267. Springer, Heidelberg (2013)
12. Moskvina, A., Liu, J.: How to build your network? a structural analysis. In: Proceedings of IJCAI 2016, AAAI Press (2016, to appear)
13. Newman, M., Watts, D., Strogatz, S.: Random graph models of social networks. Proc. Natl. Acad. Sci. USA **99**, 2566–2572 (2002)
14. Padgett, J., Ansell, C.: Robust action and the rise of the medici, 1400–1434. Am. J. Sociol. **98**(6), 1259–1319 (1993)
15. Ritter, T., Wilkinson, I., Johnston, W.: Managing in complex business networks. Ind. Mark. Manage. **33**(3), 175–183 (2004)
16. Robertson, B.: Holacracy: The New Management System for a Rapidly Changing World Hardcover. Henry Holt and Co., New York City (2015)
17. Wang, X., Chen, G.: Complex networks: small-world, scale-free and beyond. IEEE Circ. Syst. Mag. **3**(1), 6–20 (2003). First Quarter