Proceedings of

Graph Theory Day 74

November 11, 2017

Sponsored by

The Metropolitan New York Section of The Mathematical Association of America

Hosted by

Seidenberg School of Computer Science and Information Systems

Pace University, New York, New York

Edited by: Edgar G. DuCasse, Christelle Scharff, and Louis V. Quintas

Pace University, New York, New York

Graph Theory Day 74 was held on November 11, 2017 at The Seidenberg School of Computer Science and Information Systems, Pace University, New York. The schedule of the day's activities was as follows.

10:00 ~ 10:30 am	Registration and Welcome "Coffee and"
10:30 ~ 10:40 am	Welcoming Remarks: Dean Jonathan Hill Seidenberg School of Computer Science and Information Systems
Invited Speaker I 10:40 ~ 11:40 am	Some New Results on Counting Graphs. Miguel A. Mosteiro.
11:40 ~ 12:00 pm	Announcements
12:00 ~ 1:30 pm	Lunch
Invited Speaker II 1:30 ~ 2:30 pm	On the Shoulders of Giants (with apologies to Isaac Newton). Marty Lewinter.
2:30 – 3:00 pm	Break
More Talks 3:00 ~ 4:30 pm	Julia M. Zorluoglu ; The at most unicyclic random graph process Kevin Philips: Infinite classes of triangular graphs Anthony Delgado: Integrated graphs Michael Yatauro: Closure theorems for the binding number of a graph
4:30 ~ 5:00 pm	Open Problems Session & Closing

Papers associated with these talks can be found in what follows together with papers submitted by other participants.

We are pleased to report there was a favorable response to the formal program and much lively discussion among the participants throughout the day.

Organizing Committee:

Edgar G. DuCasse, Mathematics Department, Pace University, New York

Christelle Scharff, Computer Science Department, Pace University, New York

Louis V. Quintas, Mathematics Department, Pace University, New York

Armen Badarian, Nassau Community College, SUNY and Metro New York MAA

Table of Contents

Polynomial Counting in Anonymous Dynamic Networks

with Applications to Anonymous Dynamic Algebraic Computations

Dariusz R. Kowalski * Miguel A. Mosteiro †

Abstract

Starting with Michail, Chatzigiannakis, and Spirakis work [15], the problem of *Counting* the number of nodes in Anonymous Dynamic Networks has attracted a lot of attention. The problem is challenging because nodes are indistinguishable (they lack identifiers and execute the same program) and the topology may change arbitrarily from round to round of communication, as long as the network is connected in each round. The problem is central in distributed computing as the number of participants is frequently needed to make important decisions, such as termination, agreement, synchronization, and many others. A variety of algorithms built on top of *mass-distribution* techniques have been presented, analyzed, and also experimentally evaluated; some of them assumed additional knowledge of network characteristics, such as bounded degree or given upper bound on the network size. However, the question of whether Counting can be solved deterministically in sub-exponential time remained open. In this work, we answer this question positively by presenting Methodical Counting, which runs in polynomial time and requires no knowledge of network characteristics. Moreover, we also show how to extend Methodical Counting to compute the sum of input values and more complex functions without extra cost. Our analysis leverages previous work on random walks in evolving graphs, combined with carefully chosen alarms in the algorithm that control the process and its parameters. To the best of our knowledge, our Counting algorithm and its extensions to other algebraic and Boolean functions are the first that can be implemented in practice with worst-case guarantees.

1 Introduction

In this work, we address the standing question of whether the number of nodes of an Anonymous Dynamic Network (ADN) can be counted deterministically in polynomial time or not. We answer this question positively by presenting the Methodical Counting algorithm, and proving formally that after a polynomial number of rounds of communication all nodes know the size of the network and stop.

The problem has been thoroughly studied [15, 7, 8, 9, 6, 16, 4] because Counting is central for distributed computing. Indeed, more complex tasks need the network size to make various decisions on state agreement, synchronization, termination, and others. However, Anonymous Dynamic Networks pose a particularly challenging scenario. On one hand, nodes are indistinguishable from each other. For instance, they may lack identifiers or their number may be so massive that keeping record of them is not feasible. On the other hand, the topology of the network is highly dynamic. Indeed, the subsets of nodes that may communicate with each other may change all the time. All these features make ADN a valid model for anonymous ad hoc communication and computation.

In such a restrictive scenario, finding a way of providing theoretical guarantees of deterministic polynomial time has been elusive until now. Indeed, previous papers have either weaken the objective (e.g., computing only upper bound, only stochastic guarantees, etc.), assumed availability of network information (e.g., maximum number of neighbors, size upper bound, etc.), relied on a stronger model of communication, or provided only superpolynomial time guarantees.

*Computer Science Department, University of Liverpool, Liverpool, UK. E-mail: D.Kowalski@liverpool.ac.uk
†Computer Science Department, Pace University, New York, NY, USA. E-mail: mmosteiro@pace.edu

Methodical Counting uses no information about the network. After completing its execution, all nodes obtain the exact size of the network and stop. Moreover, they stop all at the same time, allowing the algorithm to be concatenated with other computations.

Our algorithm is based on nodes continuously sharing some magnitude, which we call *potential*,[1] resembling *mass-distribution* and *push-pull* algorithms. Unlike previous algorithms, in Methodical Counting carefully and periodically (i.e. , "methodically") some potential is removed from the network, rather than greedily doing so continuously. This approach is combined with another methodological innovation testing whether the candidate value (for the network size) is within some polynomial range of the actual network size. This complex strategy yields an algorithm in which the progress in mass-distribution can be analyzed as a sequence of parametrized Markov chains (even though the algorithm itself is purely deterministic) enhanced by mass drift and alarms controlling the process and its parameters. Our analysis approach opens the path to study more complex tasks in Anonymous Dynamic Networks applying similar techniques.

Finally, we also present a variety of extensions of Methodical Counting to compute more complex functions. Most notably, we present an extension that, concurrently with finding the network size, computes the sum of input values held at each node without asymptotic time overhead. Having a method to compute the sum and network size, more complex computations are possible in polynomial time as well. Indeed, we also describe how to compute a variety of algebraic and Boolean functions. To the best of our knowledge, ours are the first algorithms for anonymous dynamic Counting and other algebraic computations that can be implemented in practice with worst-case guarantees.

Roadmap:

The rest of the paper is organized as follows. We specify the model and notation details in Section 2. Then, we overview previous work in Section 3 and present our results in Section 4. Section 5 includes the details of Methodical Counting, and we prove its correctness and running time in Section 6. Extensions to other functions are presented in Section 7.

2 Model, Problem, and Notation

The Counting Problem:

The definition of the problem is simple. An algorithm solves the Counting Problem if, after completing its execution, all nodes have obtained the exact size of the network and stop.

Anonymous Dynamic Networks:

The following model is customary in the Anonymous Dynamic Networks literature. We consider a network composed by a set V of $n > 1$ network *nodes* with processing and communication capabilities. It was shown in [15] that Counting cannot be solved in Anonymous Networks without the availability of at least one distinguished node in the network. Thus, we assume the presence of such node called *leader*. Aside from the leader, we assume that all other nodes are indistinguishable from each other. That is, we do not assume the availability of labels or identifiers, and all non-leader nodes execute exactly the same program.

Each pair of nodes that are able to communicate define a communication *link*, and the set of links is called the *topology* of the network. The nodes in a communication link are called *neighbors*. The event of sending a message to neighbors is called a *broadcast* or *transmission*. Nodes and links are reliable, in the sense that no communication or node failures occur. Hence, a broadcasted message is received by all neighbors. Moreover, links are *symmetric*, that is, if node a is able to send a message to node b, then b is able to send a message to a.

[1] In previous related works this quantity, used in a different way, was termed *energy*. We steer away from such denomination to avoid confusion with node energy supply.

Without loss of generality, we discretize time in *rounds*. In any given round, a node may broadcast a message, receive all messages from broadcasting neighbors, and carry out some computations, in that order. The time taken by the computations is assumed to be negligible.

The set of links among nodes may change from round to round, and nodes have no way of knowing which were the neighbors they had before. These topology changes are arbitrary, limited only to maintain the network connected in each round. That is, at any given round the topology is such that there is a *path*, i.e., a sequence of links, between each pair of nodes, but the set of links may change arbitrarily from round to round. This adversarial model of dynamics was called 1-*interval connectivity* in [14].

The following notation will be used. The maximum number of neighbors that any node may have at any given time is called the *dynamic maximum degree* and it is denoted as Δ. The maximum length of a path between any pair of nodes at any given time is called the *dynamic diameter* and it is denoted as D. The maximum length of an opportunistic path between any pair of nodes over many time slots is called the *chronopath* [10] and it is denoted as $\mathcal{D}$.

algorithm	needs		computes	stops?	complexity	
	size upper bound N	dynamic maximum degree u.b. $d_{\max}$			time	space
Degree Counting [15]		✓	$O(d_{\max}^n)$	✓	$O(n)$	
Conscious [7]	✓	✓	n	✓	$O(e^{N^2}N^3) \Rightarrow$ $O(e^{d_{\max}^{2n}} d_{\max}^{3n})$ using [15]	
Unconscious [7]			n	No	No theoretical bounds	
$\mathcal{A}_{\mathcal{O}^P}$ [8]		Degree oracle for each node	n	Eventually	Unknown	
EXT [6]			n	✓	$O(n^{n+4})$	EXPSPACE
Incremental Counting [16]		✓	n	✓	$O\left(n\,(2d_{\max})^{n+1}\,\frac{\ln n}{\ln d_{\max}}\right)$	
Methodical Counting [This work]			n	✓	$O(n^5 \ln^2 n)$	PSPACE

Table 1: Comparison of Counting protocols for Anonymous Dynamic Networks.

3 Previous Work

In this section we overview previous work directly related to this paper. A comprehensive overview of work related to Anonymous Dynamic Networks can be found in a survey by Casteigts et al. [3] and references in the papers cited here. The related work overviewed, in comparison with our results, is summarized in Table 1.

With respect to lower bounds, it was proved in [5] that at least $\Omega(\log n)$ rounds are needed, even if D is constant. Also, a trivial observation is that $\Omega(\mathcal{D})$ is a lower bound as at least one node needs to hear about all other nodes to obtain the right count, and the chronopath $\mathcal{D}$ is the largest number of hops that a message from some node needs to take to reach other node in the network, possibly along multiple time slots.

Counting was already studied in [15], together with the problem of *Naming*, for dynamic and static networks. It was shown in this work that it is impossible to solve Counting without the presence of a distinguished node, even if nodes do not move. The Counting protocol presented for Anonymous Dynamic

Networks requires knowledge of an upper bound on Δ, and the count obtained is only an upper bound on the network size, which may be as bad as exponential.

An exact count is obtained by the Conscious Counting algorithm presented in [7]. However, the computation relies on knowing initially an upper bound on the network size. The running time of this protocol is exponential only if the initial upper bound is tight.

In the same work and follow-up papers [8, 9], the authors presented protocols under more challenging scenarios where Δ is not known. However, either the protocol does not terminate [7], and hence the running time cannot be bounded, or the protocol is terminated heuristically [9]. In experiments [9], such heuristic was found to perform well on dense topologies, but for other topologies the error rate was high. That is, the results only apply to dense Anonymous Dynamic Networks. Another protocol in [8] is shown to terminate eventually, without running-time guarantees and under the assumption of having for each node an estimate of the number of neighbors in each round. In [15] it was conjectured that some knowledge of the network such as the latter would be necessary, but the conjecture was disproved later in [6]. On the other hand the protocol in [6] requires exponential space.

Recently, a protocol called Incremental Counting was presented in [16]. This algorithm reduced exponentially the running time guarantees with respect to previous works developed under the same model. Incremental Counting obtains the exact count, all nodes terminate simultaneously, the topology dynamics is only limited to 1-interval connectivity, it only requires polynomial space, and it only requires knowledge of the dynamic maximum degree Δ. The superpolynomial running time proved still does not provide enough guarantee for practical application, but reducing from doubly-exponential to exponential was an important step towards understanding the complexity of Counting.

In a follow-up paper [4], Incremental Counting was tested experimentally showing a promising polynomial behavior. The study was conducted on pessimistic inputs designed to slow the convergence, such as bounded-degree trees rooted at the leader uniformly chosen at random for each round, and a single path starting at the leader with non-leader nodes permuted uniformly at random for each round. The protocol was also tested on static versions of the inputs mentioned, classic random graphs, and networks where some disconnection is allowed. The results exposed important observations. Indeed, even for topologies that stretch the dynamic diameter, the running times obtained are below Δn^3. It was also observed that random graphs, as used in previous experimental studies [9], reduce the convergence time, and therefore are not a good choice to indicate worst-case behavior. These experiments showed good behavior even for networks that sometimes are disconnected, indicating that more relaxed models of dynamics, such as (α, β)-connectivity [10, 11], are worth to study. All in all, the experiments in [4] showed that Incremental Counting behaves well in a variety of pessimistic inputs, but not having a proof of what a worst-case input looks like, and being the experiments restricted to a range of values of n far from the expected massive size of an Anonymous Dynamic Network, a theoretical proof of polynomial time remained an open problem even from a practical perspective.

In a recent manuscript [2] a polynomial Counting algorithm is presented relying on the availability of an algorithm to compute average with polynomial convergence time. Such average computation is modeled as a Markov chain with underlying doubly-stochastic matrix, which requires topology information within two hops (cf. [17]). In our model of Anonymous Dynamic Network, such information is not available, and gathering it may not be possible due to possible topology changes from round to round.

4 Our Contributions

We present and analyze a deterministic distributed algorithm to compute the number of nodes in an Anonymous Dynamic Network. We call such algorithm Methodical Counting. As opposed to previous works, our algorithm does not require any knowledge of network characteristics, such as dynamic maximum degree or an upper bound on the size. After $O(n^5 \ln^2 n)$ communication rounds of running Methodical Counting, all nodes obtain the network size and stop at the same round. To the best of our knowledge, this is the first polynomial deterministic Counting algorithm in the pure model of Anonymous Dynamic Network.

Our algorithm is based on distributing potential in a mass-distribution fashion, similarly as previous works for Counting. The main algorithmic novelty in our approach is that the leader participates in the

process as any other node, removing potential only after it has accumulated enough. This approach allowed us to leverage previous work on random walks in evolving graphs. For this approach to work, we combine it with testing whether the candidate value for the network size is polynomially close to the actual value. Our approach also opens the path to study more complex computations in Anonymous Dynamic Networks using the same analysis.

Finally, we also present extensions of Methodical Counting to compute more complex functions. Most notably, we show how to modify Methodical Counting to compute the sum of input values held by nodes at the same time than counting. Having an algorithm to compute the network size and the sum of input values, we also show how to compute other algebraic and Boolean functions.

5 Methodical Counting

In this section we present Methodical Counting. First, we give the intuition of the algorithm, the details can be found in Figures 1 and 2. (References to algorithm lines are given as $\langle figure\# \rangle.\langle line\# \rangle$.)

Initially, the leader is assigned a potential of 0 and all the other nodes are assigned a potential of 1. Then, the algorithm is composed by epochs, each of which is divided into phases composed by rounds of communication. Epoch k corresponds to a size estimate k that is iteratively increased from epoch to epoch until the correct value n is found. Each epoch is divided into p phases. The purpose of each phase is for the leader to collect as much potential as possible from the other nodes in a mass-distribution fashion as follows.

Each phase is composed by r rounds of communication. In each round, each node[2] broadcasts its potential and receives the potential of all its neighbors. Each node keeps only a fraction $1/d$ of the potentials received. The parameters p, r, and d are functions of k. The specific functions needed to guarantee correctness and saught efficiency are defined in Theorem 2. This varying way of distributing potential is different from previous approaches using mass distribution. After communication, each node updates its own potential accordingly (cf. Lines 1.11 and 2.10). That is, it adds a fraction $1/d$ of the potentials received, and subtracts a fraction $1/d$ of the potential broadcasted times the number of potentials received. Then, a new round starts.

At the end of each phase the leader "consumes" its potential. That is, it increases an internal accumulator ρ with its current potential, which is zeroed for starting the next phase (cf. Lines 1.19 and 1.20). A node stops the update of potential described, raises its potential to 1, and broadcasts an alarm in each round until the end of the epoch if any of the following happens: 1) at the end of the first phase its potential is above some threshold τ as defined in Theorem 2 (cf. Lines 1.15 and 2.14), 2) at any round it receives more than $d-1$ messages (cf. Lines 1.12 and 2.11), or 3) at any round it receives an alarm (cf. Lines 1.12 and 2.11). The alarm for case 1) allows the leader to detect that the estimate is wrong when $k^{1+\epsilon} < n$ for some $\epsilon > 0$ (Lemmas 3 and 4), the alarm for case 2) allows the leader to detect that d is too small and hence the estimate is wrong, and the alarm for case 3) allows dissemination of all alarms. In the alarm status the potential is set to 1 to facilitate the analysis, but it is not strictly needed by the algorithm.

At the end of each epoch, the leader checks the value of ρ. If $k-1-1/k \leq \rho \leq k-1$ the current estimate is correct and the leader changes its status to "done" (cf. Line 1.21). Otherwise, all its variables are reset to start a new epoch with the next estimate (cf. Line 1.23). Before starting a new epoch the network is flooded with the status of the leader for k rounds (cf. Lines 1.28 and 2.17). If $k = n$, the leader initiates message "done" and the k rounds are enough for all the nodes to receive the "done" status and after completing the k rounds stop. Otherwise, nodes will not receive the "done" status and after completing the k rounds they start a new epoch.

6 Analysis

In this section we analyze Methodical Counting. References to algorithm lines are given as $\langle figure\# \rangle.\langle line\# \rangle$. We will use the standard notation for the L_p norm of vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$

[2] As opposed to previous work, in Methodical Counting the leader also follows this procedure.

Figure 1: METHODICAL COUNTING algorithm for the leader. N is the set of neighbors of the leader in the current round. The parameters d, p, r and τ are as defined in Theorem 2.

```
1:  procedure COUNT
2:      ρ ← 0                                              // accumulator of consumed potential
3:      Φ ← 0                                                                  // current potential
4:      k ← 2                                                                   // current estimate
5:      status ← normal                                          // status=normal|alarm|done
6:      while status ≠ done do                                                 // iterating epochs
7:          for phase = 1 to p do                                              // iterating phases
8:              for round = 1 to r do                                          // iterating rounds
9:                  Broadcast ⟨Φ, status⟩ and Receive ⟨Φ_i, status_i⟩, ∀i ∈ N
10:                 if status = normal and |N| ≤ d − 1 and ∀i ∈ N : status_i = normal then
11:                     Φ ← Φ + Σ_{i∈N} Φ_i/d − |N|Φ/d                         // update potential
12:                 else                                                       // k is wrong
13:                     status ← alarm
14:                     Φ ← 1
                /* r rounds completed */
15:             if phase = 1 and Φ > τ then                                    // k is wrong
16:                 status ← alarm
17:                 Φ ← 1
18:             if status = normal then                          // prepare for next phase
19:                 ρ ← ρ + Φ
20:                 Φ ← 0
            /* p phases completed */
21:         if status = normal and k − 1 − 1/k ≤ ρ ≤ k − 1 then                 // the size is k
22:             status ← done
23:         else                                                 // prepare for next epoch
24:             ρ ← 0
25:             Φ ← 0
26:             k ← k + 1
27:             status ← normal
28:         for round = 1 to k do                                // disseminate termination
29:             Broadcast ⟨status⟩ and Receive ⟨status_i⟩, ∀i ∈ N
            /* epoch completed */
30:     return k
```

Figure 2: METHODICAL COUNTING algorithm for each non-leader node i. N is the set of neighbors of i in the current round. The parameters d, p, r and τ are as defined in Theorem 2.

```
1:  procedure COUNT
2:      Φ ← 0                                                          // current potential
3:      k ← 2                                                          // current estimate
4:      status ← normal                                                // status=normal|alarm|done
5:      while status ≠ done do                                         // iterating epochs
6:          for phase = 1 to p do                                      // iterating phases
7:              for round = 1 to r do                                  // iterating rounds
8:                  Broadcast ⟨Φ, status⟩ and Receive ⟨Φ_i, status_i⟩, ∀i ∈ N
9:                  if status = normal and |N| ≤ d − 1 and ∀i ∈ N : status_i = normal then
10:                     Φ ← Φ + Σ_{i∈N} Φ_i/d − |N|Φ/d                 // update potential
11:                 else                                               // k is wrong
12:                     status ← alarm
13:                     Φ ← 1
                /* r rounds completed */
14:             if phase = 1 and Φ > τ then                            // k is wrong
15:                 status ← alarm
16:                 Φ ← 1
            /* p phases completed */
17:         for round = 1 to k do                                      // disseminate termination
18:             Broadcast ⟨status⟩ and Receive ⟨status_i⟩, ∀i ∈ N
19:             if ∃i ∈ N : status_i = done then
20:                 status ← done
21:         if status ≠ done then
22:             k ← k + 1
23:             status ← normal
            /* epoch completed */
24:     return k
```

as $||\mathbf{x}||_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p}$, for any $p \geq 1$. Only for the analysis, nodes are labeled as $0, 1, 2, \ldots, n-1$, where the leader has label 0. The potential of a node i at the beginning of round t is denoted as $\Phi_t[i]$, and the potential of all nodes at the beginning of round t is denoted as a vector $\mathbf{\Phi}_t$. The aggregated potential is then $||\mathbf{\Phi}_t||_1$. The subindex t is used for rounds, phases, or dropped as needed. We will refer to the potential right after the last round of a phase as $\mathbf{\Phi}_{r+1}$. Such round does not exist in the algorithm, but we use this notation to distinguish between the potential right before the leader consumes its own potential (cf. Line 1.23) and the potential at the beginning of the first round of the next phase.

First, we provide a broad description of our analysis of Methodical Counting. Consider the vector of potentials $\mathbf{\Phi}_i$ held by nodes at the beginning of any given phase i. The way that potentials are updated in each round (cf. Lines 1.11 and 2.10) is equivalent to the progression of a d-lazy random walk on the evolving graph underlying the network topology [1], where the initial vector of potentials is equivalent to an initial distribution $\mathbf{p}_i$ on the overall potential $||\mathbf{\Phi}_i||_1$ and the probability of choosing a specific neighbor is $1/d$. For instance, the initial vector of potentials $\mathbf{\Phi}_0 = \langle 0, 1, 1, \ldots \rangle$, corresponds to a distribution $\mathbf{p}_0 = \langle 0, 1/(n-1), 1/(n-1), \ldots \rangle$ on the initial $||\mathbf{\Phi}_0||_1 = n - 1$.

Note that our Methodical Counting is not a simple "derandomization" of the lazy random walk on evolving graphs. First, in the Anonymous Dynamic Network model neighbors cannot be distinguished, and even their number is unknown at transmission time (only at receiving time the node learns the number of its neighbors). Second, due to unknown network parameters, it may happen in an execution of Methodical Counting that the total potential received could be bigger than 1. Third, our algorithm does not know a priori when to terminate and provide result even with some reasonable accuracy, as the formulas on mixing and cover time of lazy random walks depend on (a priori unknown) number of nodes n. Nevertheless, we can still use some results obtained in the context of analogous lazy random walks in order to prove useful properties of parts of algorithm Methodical Counting, namely, some parts in which parameters are temporarily fixed and the number of received messages does not exceed parameter d.

It was shown in [1] that random walks on d-regular explorable evolving graphs have a uniform stationary distribution, and bounds on the mixing and cover time were proved as well. Moreover, it was observed that those properties hold even if the graph is not regular and d is only an upper bound on the degree.[3]

Thus, for the cases where d is an upper bound on the number of neighboring nodes, we analyze the evolution of potentials within each phase leveraging previous work on random walks on evolving graphs. Specifically, we use the following result which is an extension of Corollary 14 in [1].

Theorem 1. *(Corollary 14 in [1].) After t rounds of a $d_{\max}$-lazy random walk on an evolving graph with n nodes, dynamic diameter D, upper bound on maximum degree $d_{\max}$, and initial distribution $\mathbf{p}_0$, the following holds.*

$$\left|\left|\mathbf{p}_t - \frac{\mathbf{I}}{n}\right|\right|_2^2 \leq \left(1 - \frac{1}{d_{\max} D n}\right)^t \left|\left|\mathbf{p}_0 - \frac{\mathbf{I}}{n}\right|\right|_2^2$$

In between phases the leader "consumes" its potential, effectively changing the distribution at that point. Then, a new phase starts.

In Methodical Counting, given that d is a function of the estimate k, if the estimate is low there may be inputs for which d is not an upper bound on the number of neighbors. We show in our analysis that in those cases the leader detects the error and after some time all nodes increase the estimate.

First, we prove correctness when $k = n$ as follows.

Lemma 1. *If $d \geq k$ and $k = n$, after running the* Methodical Counting *protocol for $p \geq \frac{k}{1-1/k} \ln(k(k-1))$ phases, each of $r \geq 4dk^2 \ln k$ rounds, the potential ρ consumed by the leader is $k - 1 - 1/k \leq \rho \leq k - 1$.*

Proof. The second inequality is immediate because the initial total potential in the network is $n - 1$ and it does not increase during the execution. So, if $k = n$, the potential consumed by the leader cannot be more than $k - 1$.

[3]Their analysis relies on Lemma 12, which bounds the eigenvalues of the transition matrix as long as it is stochastic, connected, symmetric, and non-zero entries lower bounded by $1/d$. Those conditions hold for all the transition matrices, even if the evolving graph is not regular.

For the first inequality, consider the vector of potentials $\mathbf{\Phi}_1$ at the beginning of round 1 of any phase i. As explained above, we analyze the evolution of potentials within phase i as a random walk on the evolving graph underlying the network topology. Consider the initial distribution $\mathbf{p}_i$ on the overall potential $||\mathbf{\Phi}_1||_1$. Then, using Theorem 1, we know that after a phase i of $r \geq 4dk^2 \ln k$ rounds the distribution is such that

$$\begin{aligned}\left|\left|\mathbf{p}_{r+1} - \frac{\mathbf{I}}{k}\right|\right|_2^2 &\leq \left(1 - \frac{1}{d\mathcal{D}k}\right)^r \left|\left|\mathbf{p}_1 - \frac{\mathbf{I}}{k}\right|\right|_2^2 \\ &\leq \exp\left(-\frac{r}{d\mathcal{D}k}\right) \\ &\leq \exp\left(-\frac{4dk^2 \ln k}{d\mathcal{D}k}\right), \text{ given that } k = n > \mathcal{D}, \\ &\leq \exp\left(-4 \ln k\right) \\ &= \frac{1}{k^4}.\end{aligned} \tag{1}$$

Given that $(p_{r+1}[0] - 1/k)^2 \leq \left|\left|\mathbf{p}_{r+1} - \frac{\mathbf{I}}{k}\right|\right|_2^2$, we have that $(p_{r+1}[0] - 1/k)^2 \leq 1/k^4$ and hence $p_{r+1}[0] \geq 1/k - 1/k^2$. Notice that the latter is true for any initial distribution, as the distance to uniform in Equation 1 has been upper bounded by 1. Thus, applying recursively we have that after $p \geq \frac{k}{1-1/k} \ln(k(k-1))$ phases it is

$$\begin{aligned}||\mathbf{\Phi}_p||_1 &\leq \left(1 - \frac{1}{k}\left(1 - \frac{1}{k}\right)\right)^p (k-1) \\ &\leq \exp\left(-\frac{p}{k}\left(1 - \frac{1}{k}\right)\right)(k-1) \\ &\leq 1/k.\end{aligned}$$

Thus, the claim follows. □ □

The previous lemma shows that if $\rho > k-1$ or $\rho < k-1-1/k$ we know that the estimate k is wrong, but the complementary case, that is, $k-1-1/k \leq \rho \leq k-1$, may occur even if the estimate is $k < n$ and hence the error has to be detected by other means. To prove correctness in that case, we show first that if $k < n \leq k^{1+\epsilon}$ for some $\epsilon > 0$ the leader must consume $\rho > k-1$ potential if the protocol is run long enough. To ensure that $d \geq \Delta + 1$, we restrict $d \geq k^{1+\epsilon}$.

Lemma 2. *If $1 < k < n \leq k^{1+\epsilon} \leq d$, $\epsilon > 0$, after running the* Methodical Counting *protocol for $p \geq \frac{(2+\epsilon)k^{1+\epsilon}}{1-1/k} \ln k$ phases, each of $r \geq (4+2\epsilon)dk^{2+2\epsilon} \ln k$ rounds, the potential ρ consumed by the leader is $\rho > k-1$.*

Proof. Given that $d \geq n$, we can use Theorem 1 as in Lemma 1 to show that after a phase i of $r \geq (4+2\epsilon)dk^{2+2\epsilon} \ln k$ rounds the distribution is such that

$$\begin{aligned}\left|\left|\mathbf{p}_{r+1} - \frac{\mathbf{I}}{n}\right|\right|_2^2 &\leq \left(1 - \frac{1}{d\mathcal{D}n}\right)^r \left|\left|\mathbf{p}_1 - \frac{\mathbf{I}}{n}\right|\right|_2^2 \\ &\leq \exp\left(-\frac{r}{d\mathcal{D}n}\right) \\ &\leq \exp\left(-\frac{(4+2\epsilon)dk^{2+2\epsilon} \ln k}{d\mathcal{D}n}\right), \text{ given that } k^{1+\epsilon} \geq n > \mathcal{D}, \\ &\leq \exp\left(-(4+2\epsilon) \ln k\right) \\ &= \frac{1}{k^{4+2\epsilon}}.\end{aligned}$$

Given that $(p_{r+1}[0]-1/n)^2 \leq \left|\left|\mathbf{p}_{r+1} - \frac{\mathbf{1}}{n}\right|\right|_2^2$, we have that $(p_{r+1}[0]-1/n)^2 \leq 1/k^{4+2\epsilon}$ and hence $p_{r+1}[0] \geq 1/n - 1/k^{2+\epsilon}$. The latter is true for any initial distribution, as the distance to uniform has been upper bounded by 1. So, applying recursively, we have that after $p \geq \frac{(2+\epsilon)k^{1+\epsilon}}{1-1/k} \ln k$ phases it is

$$\begin{aligned}
||\mathbf{\Phi}_p||_1 &\leq \left(1 - \left(\frac{1}{n} - \frac{1}{k^{2+\epsilon}}\right)\right)^p (n-1) \\
&\leq \exp\left(-p\left(\frac{1}{n} - \frac{1}{k^{2+\epsilon}}\right)\right)(n-1) \ , \text{ since } k^{1+\epsilon} \geq n, \\
&\leq \exp\left(-\frac{p}{k^{1+\epsilon}}\left(1 - \frac{1}{k}\right)\right)(n-1) \ , \text{ replacing } p, \\
&\leq \frac{n-1}{k^{2+\epsilon}} \ , \text{ given that } k^{1+\epsilon} > n-1, \\
&< 1/k \ .
\end{aligned}$$

Thus, the potential consumed by the leader is $\rho \geq n - 1 - 1/k > k - 1$ for any integers $n > k > 1$. □ □

It remains to show that even if $n > k^{1+\epsilon}$ Methodical Counting still detects that the estimate is low. First, we prove the following two claims that establish properties of the potential during the execution of Methodical Counting. (Recall that we use round $r+1$ to refer to potentials at the end of the phase right before the leader consumes its potential in Line 1.23.)

Claim 1. *Given an Anonymous Dynamic Network of n nodes running* Methodical Counting *with parameter d, for any round t of the first phase, such that $1 \leq t \leq r+1$, if d was larger than the number of neighbors of each node x for every round $t' < t$, then $||\mathbf{\Phi}_t||_1 = n-1$.*

Proof. For the first round the claim holds as the initial potential of each node is 1 except the leader that gets 0. That is, $||\mathbf{\Phi}_1||_1 = n-1$. For any given round $1 < t \leq r+1$ in phase 1 and any given node x, if d is larger than the number of neighbors of x, the potential is updated only in Lines 1.11 and 2.10 as

$$\Phi_{t+1}[x] = \Phi_t[x] + \sum_{i \in N_t[x]} \Phi_t[i]/d - |N_t[x]|\Phi_t[x]/d \ .$$

Where $N_t[x]$ is the set of neighbors of node x in round t. Inductively, assume that the claim holds for some round $1 \leq t \leq r$. We want to show that consequently it holds for $t+1$. The potential for round $t+1$ is

$$||\mathbf{\Phi}_{t+1}||_1 = ||\mathbf{\Phi}_t||_1 + \frac{1}{d}\sum_{x \in V}\left(\sum_{y \in N_t[x]} \Phi_t[y] - |N_t[x]|\Phi_t[x]\right) \ . \tag{2}$$

In the Anonymous Dynamic Network model, communication is symmetric. That is, for every pair of nodes $x, y \in V$ and round t, it is $x \in N_t[y] \iff y \in N_t[x]$. Fix a pair of nodes $x', y' \in V$ such that in round t it is $y' \in N_t[x']$ and hence $x' \in N_t[y']$. Consider the summations in Equation 2. Due to symmetric communication, we have that the potential $\Phi_t[y']$ appears with positive sign when the indeces of the summations are $x = x'$ and $y = y'$, and with negative sign when the indices are $x = y'$ and $y = x'$. This observation applies to all pairs of nodes that communicate in any round t. Therefore, we can re-write Equation 2 as

$$||\mathbf{\Phi}_{t+1}||_1 = ||\mathbf{\Phi}_t||_1 + \frac{1}{d}\sum_{\substack{x,y \in V: \\ y \in N_t[x]}}\Big(\Phi_t[y] - \Phi_t[x] + \Phi_t[x] - \Phi_t[y]\Big) = ||\mathbf{\Phi}_t||_1 \ .$$

Thus, the claim follows.

□ □

Claim 2. *Given an Anonymous Dynamic Network of n nodes running* Methodical Counting*, for any round t of any phase and any node x, it is $0 \leq \Phi_t[x] \leq 1$.*

Proof. If $t = 1$ the potential of the leader is $\Phi_1[0] = 0$ and the potential of any non-leader node x is $\Phi_1[x] = 1$. Thus, the claim follows. Inductively, for any round $2 < t \leq r + 1$, we consider two cases according to node status. If a node x is in alarm status at the beginning of round t, then it is $\Phi_t[x] = 1$ as, whenever the status of a node is updated to alarm, its potential is set to 1 and will not change until the next epoch (cf. Figures 1 and 2). On the other hand, if a node x is in normal status at the beginning of round t, it had its potential updated in all rounds $t' < t$ only in Lines 1.11 and 2.10 as

$$\Phi_{t'+1}[x] = \Phi_{t'}[x] + \sum_{y \in N_{t'}[x]} \Phi_{t'}[y]/d - |N_{t'}[x]|\Phi_{t'}[x]/d.$$

For all rounds $t' < t$, node x exchanged potential with less than d neighbors, because otherwise it would have been changed to alarm status in Lines 1.13 and 2.12. Therefore it is $|N_{t'}[x]|\Phi_{t'}[x]/d < \Phi_{t'}[x]$ which implies $\Phi_t[x] \geq 0$. It can also be seen that $\Phi_t[x] \leq 1$ because, for any $t' < t$, it is

$$\begin{aligned}
\Phi_{t'+1}[x] &= \Phi_{t'}[x] + \sum_{y \in N_{t'}[x]} \Phi_{t'}[y]/d - |N_{t'}[x]|\Phi_{t'}[x]/d \\
&\leq \Phi_{t'}[x] + \frac{|N_{t'}[x]|}{d} - \frac{|N_{t'}[x]|}{d}\Phi_{t'}[x] \\
&= \Phi_{t'}[x] + \frac{|N_{t'}[x]|}{d}(1 - \Phi_{t'}[x]) \\
&\leq \Phi_{t'}[x] + 1 - \Phi_{t'}[x] = 1.
\end{aligned}$$

□ □

It remains to show that even if $n > k^{1+\epsilon}$ Methodical Counting still detects that the estimate is low. We focus on the first phase. We define a threshold τ such that, after the phase is completed, all nodes that have potential above τ can send an alarm to the leader, as such potential indicates that the estimate is low. We show that the alarm must be received after $k^{1+\epsilon}$ further rounds of communication.

Lemma 3. *For $\epsilon > 0$, after running the first phase of the* Methodical Counting *protocol, there are at most $k^{1+\epsilon}$ nodes that have potential at most $\tau = 1 - 1/k^{1+\epsilon}$.*

Proof. We define the *slack* of node x at the beginning of round t as $s_t[x] = 1 - \Phi_t[x]$ and the vector of slacks at the beginning of round t as $\mathbf{s}_t$. In words, the slack of a node is the "room" for additional potential up to 1. Recall that the overall potential at the beginning of round 1 of phase 1 is $||\mathbf{\Phi}_1||_1 = n - 1$. Also notice that for any round and any node x the potential of x is non-negative as shown in Claim 2. Therefore, the overall slack with respect to the maximum potential that could be held by all the n nodes at the beginning of round 1 is $||\mathbf{s}_1||_1 = 1$.

Consider a partition of the set of nodes $\{L, H\}$, where L is the set of nodes with potential at most $\tau = 1 - 1/k^{1+\epsilon}$ at the end of the first phase, before the leader consumes its own potential in Line 1.23. That is, $\Phi_{r+1}[x] \leq \tau$ for all $x \in L$. Assume that the slack held by nodes in L at the end of the first phase is at most the overall slack at the beginning of the phase. That is, $\sum_{x \in L} s_{r+1}[x] \leq ||\mathbf{s}_1||_1 = 1$. By definition of L, we have that for each node $x \in L$ it is $s_{r+1}[x] = (1 - \Phi_{r+1}[x]) \geq 1 - \tau$. Therefore, $|L|(1-\tau) \leq \sum_{x \in L} s_{r+1}[x] \leq 1$. Thus, $|L| \leq 1/(1 - \tau) = k^{1+\epsilon}$ and the claim follows.

Then, to complete the proof, it remains to show that $\sum_{x \in L} s_{r+1}[x] \leq 1$. Let the scenario where d is larger than the number of neighbors that each node has in each round of the first phase be called "case 1", and "case 2" otherwise. Claim 1 shows that in case 1 at the end of the first phase it is $||\mathbf{\Phi}_{r+1}||_1 = n - 1$. Therefore, the slack held by all nodes is $||\mathbf{s}_{r+1}||_1 = 1$ and the slack held by nodes in $L \subseteq V$ is $\sum_{x \in L} s_t[x] \leq 1$. We show now that indeed case 1 is a worst-case scenario. That is, in the complementary case 2 where some nodes have d neighbors or more in one or more rounds, the slack is even smaller. To compare both scenarios we denote the slack for each round t, each node x, and each case i as $s_t^{(i)}[x]$.

Assume that some node x is the first one to have $d' > d-1$ neighbors. Let $1 \leq t \leq r$ be the round of the first phase when this event happened. We claim that $||\mathbf{s}^{(2)}_{t+1}||_1 \leq ||\mathbf{s}^{(1)}_{t+1}||_1$. The reason is the following. Given that more than $d-1$ potentials are received, node x increases its potential to 1 for the rest of the epoch (cf. Lines 1.12 and 2.11). That is, the slack of x is $s^{(2)}_{t+1}[x] \leq s^{(2)}_t[x] = s^{(1)}_t[x]$. Additionally, the potential shared by x with all neighbors during round t is $d'\Phi_t[x]/d > \Phi_t[x](1-1/d)$ (cf. Lines 1.11 and 2.10). That is, the potential shared by x with neighbors in case 2 is more than the potential that x would have shared in case 1. Then, combining both effects (the relative increase in potential of x and its neighbors') the overall slack is $||\mathbf{s}^{(2)}_{t+1}||_1 \leq ||\mathbf{s}^{(1)}_{t+1}||_1$. The same argument applies to all other nodes with d or more neighbors in round t.

Additionally, for any round t' of the first phase, such that $t < t' \leq r$, we have to additionally consider the case of a node y that, although it does not receive more than $d-1$ potentials, it moves to alarm status because it has received an alarm in round t'. Then, notice that the potential of y is $\Phi_{t'+1}[y] = 1 \geq \Phi_{t'}[y]$, and it will stay in 1 for the rest of the epoch (cf. Lines 1.13 and 2.12). Therefore, the slack of y is $s^{(2)}_{t+1}[y] \leq s^{(1)}_{t+1}[y]$.

Combining all the effects studied over all rounds, the slack at the end of the first phase is $||\mathbf{s}^{(2)}_{r+1}||_1 \leq ||\mathbf{s}^{(1)}_{r+1}||_1$. Given that $L \subseteq V$, it is $\sum_{x\in L} s^{(2)}_{r+1}[x] \leq ||\mathbf{s}^{(2)}_{r+1}||_1 \leq ||\mathbf{s}^{(1)}_{r+1}||_1 \leq 1$ which completes the proof. □ □

In our last lemma, we show that if $k^{1+\epsilon} < n$ the leader detects the error.

Lemma 4. *If $k^{1+\epsilon} < n$, $\epsilon > 0$, and $r \geq (4 + 2\epsilon - 2\ln(k^\epsilon - 1)/\ln k)dk^2 \ln k$, within the following $k^{1+\epsilon}$ rounds after the first phase of the* Methodical Counting *protocol, the leader has received an alarm message.*

Proof. Using Theorem 1, we know that after phase 1 of $r \geq (4 + 2\epsilon - 2\ln(k^\epsilon - 1)/\ln k)dk^2 \ln k$ rounds, if $k = n$, the distribution is such that

$$\begin{aligned}
\left|\left|\mathbf{p}_{r+1} - \frac{\mathbf{I}}{k}\right|\right|_2^2 &\leq \left(1 - \frac{1}{d\mathcal{D}k}\right)^r \left|\left|\mathbf{p}_1 - \frac{\mathbf{I}}{k}\right|\right|_2^2 \\
&\leq \exp\left(-\frac{r}{d\mathcal{D}k}\right) \\
&\leq \exp\left(-\frac{(4 + 2\epsilon - 2\ln(k^\epsilon - 1)/\ln k)dk^2 \ln k}{d\mathcal{D}k}\right), \text{ given that } k = n > \mathcal{D}, \\
&\leq \exp\left(-(4 + 2\epsilon - 2\ln(k^\epsilon - 1)/\ln k)\ln k\right) \\
&= 1/k^{4+2\epsilon-2\ln(k^\epsilon-1)/\ln k}.
\end{aligned}$$

Given that for any node j, it is $(p_{r+1}[j] - 1/k)^2 \leq \left|\left|\mathbf{p}_{r+1} - \frac{\mathbf{I}}{k}\right|\right|_2^2$, we have that $(p_{r+1}[j] - 1/k)^2 \leq 1/k^{4+2\epsilon-2\ln(k^\epsilon-1)/\ln k}$. Hence, it is $p_{r+1}[j] \leq 1/k + 1/k^{2+\epsilon-\ln(k^\epsilon-1)/\ln k}$ for any node j. Moreover, if $k = n$ the total potential in the network would be $k-1$ (cf. Claim 1) and no individual node should have potential larger than $(k-1)(1/k + 1/k^{2+\epsilon-\ln(k^\epsilon-1)/\ln k})$. We show that the latter is at most $\tau = 1 - 1/k^{1+\epsilon}$ as follows.

$$\begin{aligned}
(k-1)(1/k + 1/k^{2+\epsilon-\ln(k^\epsilon-1)/\ln k}) &\leq 1 - 1/k^{1+\epsilon} \\
(k-1)/k^{2+\epsilon-\ln(k^\epsilon-1)/\ln k} &\leq (k^\epsilon - 1)/k^{1+\epsilon} \\
k^{1-\ln(k^\epsilon-1)/\ln k} &\geq (k-1)/(k^\epsilon - 1) \\
\left(1 - \frac{\ln(k^\epsilon - 1)}{\ln k}\right)\ln k &\geq \ln(k-1) - \ln(k^\epsilon - 1) \\
\ln k &\geq \ln(k-1).
\end{aligned}$$

And the latter is true for any $k > 1$.

Consider a partition of the set of nodes $\{L, H\}$, where L is the set of nodes with potential at most $\tau = 1 - 1/k^{1+\epsilon}$ at the end of the first phase. At the end of the first phase, the size of L is at most $k^{1+\epsilon}$ (cf. Lemma 3), and the size of H is at least 1 because $n > k^{1+\epsilon}$. Thus, there is at least one node changing to

alarm status in Line 2.15 in round 1 of phase 2, and due to 1-interval connectivity at least one new node moves from L to H in each of the following rounds. Thus, the claim follows.

□ □

Based on the above lemmata, we establish our main result in the following theorem.

Theorem 2. *Given an Anonymous Dynamic Network with n nodes, after running* Methodical Counting *for each estimate $k = 2, 3, \ldots, n$ with parameters*

$$
\begin{aligned}
d &= k^{1+\epsilon}, \\
p &= \left\lceil \frac{(2+\epsilon)k^{1+\epsilon}}{1-1/k} \ln k \right\rceil, \\
r &= \left\lceil \left(4 + 2\epsilon + \max\left\{ 0, -\frac{2\ln(k^\epsilon - 1)}{\ln k} \right\} \right) dk^{2+2\epsilon} \ln k \right\rceil, \\
\tau &= 1 - 1/k^{1+\epsilon},
\end{aligned}
$$

where $\epsilon > 0$, all nodes stop after $\sum_{k=2}^{n}(pr + k)$ rounds of communication and output n.

Proof. Notice that the above parameters fulfill the conditions of the previous lemmas.

First we prove that Methodical Counting is correct. To do so, it is enough to show that for each estimate $k < n$ the algorithm detects the error and moves to the next estimate, and that if otherwise $k = n$ the algorithm stops and outputs k. We consider three cases: $k = n$, $k < n \leq k^{1+\epsilon}$, and $k^{1+\epsilon} < n$, for a chosen value of $\epsilon > 0$.

Assume first that $k < n \leq k^{1+\epsilon}$. Then, even if the leader does not receive an alarm during the execution, as shown in Lemma 2, at the end of the epoch in Line 1.21 the leader will detect that ρ is out of range and will not change its status to done. Therefore, no other node will receive a termination message (loop in Line 1.28), and all nodes will continue to the next epoch.

Assume now that $k^{1+\epsilon} < n$. Lemma 4 shows that within the following $k^{1+\epsilon}$ rounds after the first phase the leader has received an alarm message, even if no node has more than $d - 1$ neighbors during the execution and alarms due to this are not triggered. For the given value of p and $k \geq 2$, the epoch has more than one phase. Therefore, within $k^{1+\epsilon}$ rounds into the second phase the leader will change to alarm status in Line 1.13, will not change its status to done later in this epoch, and no other node will receive a termination message. Hence, all nodes will continue to the next epoch.

Finally, if $k = n$, Lemma 1 shows that the accumulated potential ρ will be $k - 1 - 1/k \leq \rho \leq k - 1$. Thus, in Line 1.21 the leader will change its status to done, and in the loop of Line 1.28 will inform all other nodes that the current estimate is correct. The number of iterations of such loop are enough due to 1-interval connectivity.

The claimed running time can be obtained by inspection of the algorithm, either for the leader or non-leader since they are synchronized. Refer for instance to the leader algorithm in Figure 1. The outer loop in Line 1.5 corresponds to each epoch with estimates $k = 2, 3, \ldots, n$. For each epoch, Line 1.6 starts a loop of p phases followed by k rounds in Line 1.28. Each of the p phases has r rounds. Thus, the overal number of rounds is $\sum_{k=2}^{n}(pr + k)$.

□ □

Choosing $\epsilon = \log_k 2$, the following holds.

Corollary 1. *The time complexity of* Methodical Counting *is $O(n^5 \log^2 n)$.*

$$\begin{aligned}
\sum_{k=2}^{n}(pr+k) &= \sum_{k=2}^{n}\left(\left\lceil\frac{(2+\epsilon)k^{2+\epsilon}}{k-1}\ln k\right\rceil\left\lceil\left(4+2\epsilon+\max\left\{0,-\frac{2\ln(k^{\epsilon}-1)}{\ln k}\right\}\right)k^{3+3\epsilon}\ln k\right\rceil+k\right)\\
&= \sum_{k=2}^{n}\left(\left\lceil\frac{2(2+\log_k 2)k^2}{k-1}\ln k\right\rceil\left\lceil(4+2\log_k 2)\,2^3k^3\ln k\right\rceil+k\right)\\
&\leq \sum_{k=2}^{n}\left(\left\lceil\frac{6k^2}{k-1}\ln k\right\rceil\left\lceil 48k^3\ln k\right\rceil+k\right)\\
&\in O(n^5\log^2 n).
\end{aligned}$$

7 Extensions

We argue that Methodical Counting can be extended to compute the sum of values stored in the nodes, and thus also the average (as it computes the number of nodes n), and other functions. Assume that each node of the Anonymous Dynamic Network initially stores a value, represented as a sequence of bits. W.l.o.g. we could assume that the value stored at the leader is zero; otherwise, the nodes could compute the sum of other initial values (with the leader value set up to 0), and later the leader could propagate its actual initial value appended to the message "done" at the end of the execution to be added to the computed sum of other nodes.

The modified Methodical Counting prepends the potential to the sequence. Instead of sending potential by the original Methodical Counting, each node transmits its current sequence (in which the potential stands in the first location). Changes at each position of the sequence are done independly by the same algorithm as used for the potential, cf. Figures 1 and 2. Re-setting the values, in the beginning of each epoch, means putting back the initial values of the sequence. It means that the modified algorithm maintains potential in exactly the same way as the original Methodical Counting, regardless of the initial values. At the end of some epoch, with number corresponding to the number of nodes n, all nodes terminate. When it happens, each node recalls the sequence stored in it at the end of the first phase of the epoch, multiplies the values stored at each position of the sequence by the epoch number n, and rounds each of the results to the closest integer; then it sums up the subsequent values multiplied by corresponding (consecutive) powers of 2. Note that such "recalling" could be easily implemented by storing and maintaining the sequence after the first phase of each epoch.

We argue that the computed value is the sum of the initial values. It is enough to analyze how the modified algorithm processes values at one position of the sequence, as positions are treated independently; therefore, w.l.o.g. we assume that each node has value 0 or 1 in the beginning. Consider the last epoch before the leader sends the final sequence (in our case, representing one value). In the beginning of the epoch, the values are re-set to the original one, and manipulated independently according to the rules in Figures 1 and 2. Therefore, let us focus on the first phase of this epoch. Since we already proved that the estimate of the last epoch is equal to the number of nodes, the value of d in this epoch (and thus also in its first phase) is an upper bound on the node degree. Thus, the mass distribution scaled down by the sum of the initial values behaves exactly the same as the probabilities of being at nodes in the corresponding round of the lazy random walk, with parameter d and starting from initial distribution equal to the initial values divided by the sum. Since the length of the phase is set up to guarantee that the distribution is close to the stationary uniform within error $1/n$, and the sum of bits is not bigger than n, at the end of the phase the value stored by each node is close to the sum (i.e., scaling factor) divided by n by at most $1/n^4$ (cf. Equation 1). Therefore, after multiplying it by n, each node gets value of sum within error of at most $1/n^3$, which after rounding will give the integer equal to the value of the sum.

Once having the number n and the sum, each node can compute the average. As argued in [12], the capacity of computing the sum of the input values makes possible the computation of more complex functions. Moreover, as opposed to [12] where the computation only converges, our approach outputs the exact

sum. Therefore, the extension to database queries that can be approximated using *linear synopses* [4] is straightforward. Boolean functions $f : \{0,1\}^n \to \{0,1\}$, such as AND (sum $= n$), OR (sum > 0), and XOR (sum $= 1$), as well as their complementaries NAND (sum $\neq n$), NOR (sum $= 0$), and XNOR (sum $\neq 1$), can also be implemented having n and the sum. This applies also to other "symmetric" Boolean functions, i.e., which do not depend on the order of variables, as they could be computed based on computed sum of ones and n [13]. Maximum (L_∞ norm) and minimum can be computed subsequently by flooding. That is, each node broadcasts the maximum and minimum input values seen so far. Due to 1-interval connectivity within n rounds all nodes have the answers.

Note that all these computations, including the METHODICAL COUNTING, could be done using only polynomial estimates of values, that is, with messages of length $O(\log n)$, multiplied by the maximum number of coordinates of any of the initial values. This could be also traded for time: we could use only messages of length $O(\log n)$ with time increased by the maximum number of coordinates of any initial value (which is still polynomial in the size of the input,[5] which in this case is at least n plus the maximum number of coordinates).

8 Open Directions

Straightway questions emerging from our work include existence of polynomial (in n) lower bound and improvement of our upper bound. One of the potential ways could be through investigating bi-directional relationships between random processes and computing algebraic functions in Anonymous Dynamic Network. Extending the range of polynomially computable functions is another intriguing future direction. Finally, generalizing the model by not assuming connectivity in every round or dropping assumption on synchrony could introduce even more challenging aspects of communication and computation, including group communication and its impact on the common knowledge about the system parameters.

Acknowledgments

The authors would like to thank Michal Koucký and Alessia Milani for useful discussions.

References

[1] C. Avin, M. Koucký, and Z. Lotker. How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). In *Automata, languages and programming*, pages 121–132. Springer, 2008.

[2] R. Baldoni and G. A. Di Luna. Counting on anonymous dynamic networks: Bounds and algorithms. manuscript, 2016.

[3] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.

[4] M. Chakraborty, A. Milani, and M. A. Mosteiro. Counting in practical anonymous dynamic networks is polynomial. In *Proceedings of the 4th International Conference on Networked Systems*, volume 9944 of *Lecture Notes in Computer Science*, pages 131–136, 2016.

[5] G. A. Di Luna and R. Baldoni. Investigating the cost of anonymity on dynamic networks. *CoRR*, abs/1505.03509, 2015.

[4] Additive functions on multisets, e.g. $f(A \cup B) = f(A) + f(B)$.

[5] The input in this case is distributed among the nodes, and each node possesses at least one bit

[6] G. A. Di Luna and R. Baldoni. Non trivial computations in anonymous dynamic networks. In *Proceedings of the 19th International Conference on Principles of Distributed Systems*, Leibniz International Proceedings in Informatics, 2015. To appear.

[7] G. A. Di Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. Conscious and unconscious counting on anonymous dynamic networks. In *Proceedings of the 15th International Conference on Distributed Computing and Networking*, volume 8314 of *Lecture Notes in Computer Science*, pages 257–271. Springer Berlin Heidelberg, 2014.

[8] G. A. Di Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. Counting in anonymous dynamic networks under worst-case adversary. In *Proceedings of the 34th International Conference on Distributed Computing Systems*, pages 338–347. IEEE, 2014.

[9] G. A. Di Luna, S. Bonomi, I. Chatzigiannakis, and R. Baldoni. Counting in anonymous dynamic networks: An experimental perspective. In *Proceedings of the 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, volume 8243 of *Lecture Notes in Computer Science*, pages 139–154. Springer Berlin Heidelberg, 2014.

[10] M. Farach-Colton, A. Fernández Anta, A. Milani, M. A. Mosteiro, and S. Zaks. Opportunistic information dissemination in mobile ad-hoc networks: adaptiveness vs. obliviousness and randomization vs. determinism. In *Proc. of the 10th Latin American Theoretical Informatics Symposium*, volume 7256 of *Lecture Notes in Computer Science*, pages 303–314. Springer-Verlag, Berlin, 2012.

[11] A. Fernández Anta, A. Milani, M. A. Mosteiro, and S. Zaks. Opportunistic information dissemination in mobile ad-hoc networks: the profit of global synchrony. *Distributed Computing*, 25(4):279–296, 2012.

[12] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. of the 44th IEEE Ann. Symp. on Foundations of Computer Science*, pages 482–491, 2003.

[13] E. Kranakis, D. Krizanc, and J. Vandenberg. Computing boolean functions on anonymous networks. *Information and Computation*, 114(2):214 – 236, 1994.

[14] F. Kuhn, N. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 513–522. ACM, 2010.

[15] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Naming and counting in anonymous unknown dynamic networks. In *Stabilization, Safety, and Security of Distributed Systems*, pages 281–295. Springer, 2013.

[16] A. Milani and M. A. Mosteiro. A faster counting protocol for anonymous dynamic networks. In *Proceedings of the 19th International Conference on Principles of Distributed Systems*, volume 46 of *Leibniz International Proceedings in Informatics*, pages 1–13, 2015.

[17] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.

On the Shoulders of Giants (with apologies to Isaac Newton)

Marty Lewinter

marty5040@gmail.com

Dedicated to the memory of Mike Gargano.

I had the good fortune to publish with Frank Harary, Mike Gargano, and Lou Quintas. I learned much from them and thank them for my own modest success as a graph theorist and number theorist. This is a survey of papers from each collaboration, containing results and open questions.

1. Connectivity

A graph is *traceable* if it has a spanning path. If, in addition, any vertex can be chosen as the starting point, the graph is called *homogeneous connected*. Any *hamiltonian* graph (a graph with a spanning cycle), for example, is homogeneous connected.

Note, however, that a graph need not be hamiltonian to be homogeneous connected, as can be seen by analyzing the Peterson Graph.

If both endvertices of the spanning path of a graph can be chosen randomly, the graph is called *hamilton-connected* [1]. The wheel, W_n, for example, is hamilton-connected. A bipartite graph cannot have this property, since the colors of the endvertices depend on the parity of the order of the graph. That is, an even path has oppositely colored endvertices, while the endvertices of an odd path have the same color.

2. Hypercubes

The *hypercube* Q_n is defined recursively by

1. $Q_1 = K_2$
2. $Q_n = Q_{n-1} \times K_2$.

The second equation says that we get Q_n by taking the union of two copies of Q_{n-1} and rendering corresponding vertices adjacent. In Figure 1, the two vertices of Q_1 are labeled 0 and 1. As Q_2 contains two copies of Q_1, we label the vertices in the first copy 00 and 01, while the vertices in the second copy are labeled 10 and 11.

Let's do this one more time before we generalize. Since Q_3 contains two copies of Q_2, we attach a 0 to the labels of the vertices of the first copy of Q_2 and a 1 to the labels of the those of the second copy, in which case $V(Q_3) = \{000, 001, 010, 011\} \cup \{100, 101, 110, 111\}$.

See Figure 1, in which Q_3 contains labeled "inner" and "outer" copies of Q_2. The outer labels start with 0's while the inner ones start with 1's. On the other hand, we can view Q_3 as a left copy of Q_2 whose labels have a 0 in the second place, and a right copy of Q_2 whose labels have a 1 in the second place. Finally, we can view Q_3 as a lower copy of Q_2 whose labels have a 0 in the third place, and an upper copy of Q_2 whose labels have a 1 in the third place.

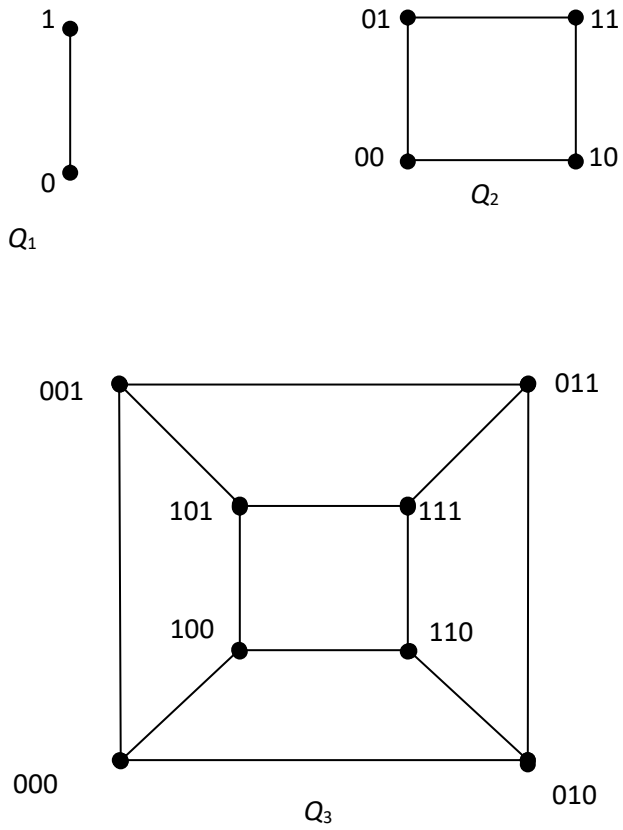


Figure 1

The vertex set of Q_n consists of all of the binary strings of length n. Since each digit is a 0 or a 1, there are 2^n such strings, implying that $|V(Q_n)| = 2^n$. By the recursive definition, we see that the degree of each vertex of Q_n is n, that is, Q_n is n-regular.

We can use Euler's theorem to obtain the size of the edge set. The degree sum of a regular graph is the product of the degree and the number of vertices. For hypercubes, this is $n2^n$. Since the degree sum is twice the number of edges, $|E(Q_n)| = (n2^n)/2 = n2^{n-1}$.

Since the hypercube Q_n has the same number of black and white vertices, we show by induction in [2], that given any two vertices x and y in Q_n of opposite color, there exists a spanning x-y path.

3. Binary Meshes

Given graphs G and H with spanning subgraphs U and V, respectively, it is easy to see that U × V spans G × H. Now let the positive integers n, r, and s satisfy $n = r + s$. Then it is not hard to see that $Q_n = Q_r \times Q_s$.

Since hypercubes are traceable, the paths P_{2^r} and P_{2^s} span Q_r and Q_s, respectively. Then we show in [3] that the mesh, $M(2^r, 2^s) = P_{2^r} \times P_{2^s}$ spans Q_n. In particular, when $r = 1$, we find that the ladder $M(2, 2^{n-1})$ spans Q_n. Meshes of the form $M(2^r, 2^s)$ are termed *binary meshes*. They are the only meshes that span Q_n.

4. Equipartion Sets of Hypercubes

Consider the set S of vertices in Q_3 given by {100, 011, 111}. Note that no matter how we cut Q_3 into two copies of Q_2 (and there are three ways to do this), two vertices of S will be in one copy, and one vertex of S will be in the other. In [4], we call this a (2,1) EPS (equipartition set) for Q_3. This can be verified directly, or we can make the three labels the rows of a 3 by 3 binary matrix

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

and observe that each of the columns of B contain two 1's and one 0. Note that B has three columns because we are in Q_3. So the labels have three digits. There is no problem if two columns are identical, but the rows must be distinct so that we have three distinct vertices.

Observe that we can add copies of the last column of B to obtain a (2,1) EPS for any Q_n for which $n > 3$. This becomes a theorem in [4] to the effect that if Q_m contains an (r,s) EPS for some m, so does any Q_n where $n > m$. Now given r and s, let $f(r,s)$ be the smallest m for which Q_m has an (r,s) EPS. We obtain bounds for this function in [4] and find $f(r,s)$ for certain interesting values of r and s. Nevertheless, the elusive function $f(r,s)$ is yet to be be found.

5. *k*-long Numbers and *k*-long Graphs

Given a positive integer k, a number is called *k-long* in [5], if it can be written $n(n + k)$. This generalizes *oblong* numbers, that is, numbers of the form $n(n + 1)$, which are 1-long. Among the various properties of k-long numbers that are presented in [5], a particularly important one asserts that the product of any pair of consecutive k-long numbers is k-long.

In [6], a graph is called k-long if

1. Its vertices are labeled by distinct k-long numbers.
2. Each edge is labeled by the product of the labels of its endvertices.
3. The edge labels are distinct k-long numbers.

(A given vertex and a given edge can share the same k-long label.)

As an example, the path P_4 of order 4 is 2-long, as can be seen by labeling the vertices by 3, 8, 15, and 24 in that order. The edge labels are 24, 120, and 360, all of which are 2-long.

Using the Pell Equation and its generalization, we produce infinite classes of k-long graphs, including all trees and various cycles. Many questions about k-long graphs remain unanswered.
(i) We ask for an infinite class of graphs that are not k-long.
(ii) Are there k-long graphs that have various different labels and edge weights?

6. Paintable Graphs

In [7] and [8], we call a graph on n vertices *paintable*, if its vertices can be labeled using the integers $1, 2, \ldots, n$, such that for each $i = 1, 2, \ldots, n - 1$, the vertices i and $i + 1$ are not adjacent. Paths of order at least 4 and cycles of order at least 5 are paintable. We show in [7] that a graph is paintable if and only if its complement is traceable. Complete graphs, by the way, yield an infinite class of graphs that are not paintable.

A graph is *homogeneous paintable*, if any vertex can be labeled 1. Cycles of order at least 5 are homogeneous paintable. The path P_4 is not. One cannot label an endvertex 1.

If a graph is not paintable, how many vertices can be painted before the process fails? P_3 cannot be painted, but we can label the endvertices using the labels 1 and 2 successfully. On the other hand, an attempt at painting K_n will fail after we label any vertex with a 1. Given a graph, is there an efficient algorithm to find the maximum j such that j vertices can be painted before the process fails?

7. Graph Theory and Pascal's Triangle

In [9] we use the edge set of Q_n to prove a well known identity involving the combinatorial coefficients in the rows of Pascal's Triangle:

$$\binom{n}{1}+2\binom{n}{2}+3\binom{n}{3}+\cdots+n\binom{n}{n}=n2^{n-1} \qquad (1)$$

Notice that the right side of equation (1) is the number of edges of Q_n. We show that the left side counts these edges too, so we have equality.

Let $v = (0, 0, 0, \ldots, 0)$. Now for $k = 0, 1, 2, \ldots, n$, let D_k be the set of all vertices whose distance from v is k. Clearly, $D_0 = \{v\}$. Furthermore, D_k is the set of all vertices whose labels contain exactly k 1's. Then $|D_k| = \binom{n}{k}$.

All the edges of Q_n have endvertices in D_{k-1} and D_k, for some $k = 1, 2, \ldots, n$. Now a vertex in D_k is adjacent to the k vertices of D_{k-1} obtained by changing any of its 1's to a 0. Then there are $k\binom{n}{k}$ edges that have endvertices in D_{k-1} and D_k. Then the number of edges in Q_n is indeed the left side of equation (1).

The standard calculus proof of equation (1) starts with the binomial theorem expansion:

$$(1+x)^n=\binom{n}{0}+\binom{n}{1}x+\binom{n}{2}x^2+\binom{n}{3}x^3+\cdots+\binom{n}{n}x^n$$

Differentiating both sides yields

$$n(1+x)^{n-1}=\binom{n}{1}+2\binom{n}{2}x+3\binom{n}{3}x^2+\cdots+n\binom{n}{n}x^{n-1}$$

Letting x =1 yields equation (1).

Are there other combinatorial identities involving Pascal's Triangle that can be proven using hypercubes?

The author thanks Anthony Delgado, doctoral candidate in Mathematics Education at Columbia University, for his help in writing this paper. He is the first author on [4], [5], and [6].

References

1. F.Buckley, M.Lewinter, *A friendly introduction to graph theory*. Prentice-Hall 2003.
2. F.Harary, M.Lewinter, Hypercubes and other recursively defined hamilton-laceable graphs. *Cong. Num*. 60, (81-84) 1987.
3. F.Harary, M.Lewinter, Spanning subgraphs of a hypercube III: Meshes. *Intern.J.Computer Math*., 25, (1-4) 1988.
4. A.Delgado, M.Lewinter, L.V.Quintas, Equipartition sets of hypercubes II, *Bulletin of the ICA*, 63, (51-59) 2011.
5. *A.Delgado,* M.Gargano, M.Lewinter, J.Malerba, Introducing *k*-long numbers. *Cong. Num*. 189, (15-23) 2008.
6. *A.Delgado,* M.Lewinter, L.V.Quintas, *k*-long graphs. *Cong.Num*.196, (95-106) 2009.
7. M.Gargano, M.Lewinter, J.Malerba, Paintable graphs. *Cong. Num*. 148, (169-175) 2001.
8. M.Gargano, M.Lewinter, J.Malerba, Paintable graphs II: trees and product graphs. *Cong. Num*. 148, (169-175) 2001.
9. M.Gargano, M.Lewinter, J.Malerba, Hypercubes and Pascal's triangle: A tale of two proofs. *Math. Magazine*, Vol.76, 3 (216-217) 2003.

SOME RANDOM GRAPH PROCESSES AND ASSOCIATED PROBLEMS

Edgar G. DuCasse[1], Louis V. Quintas[1], and Julia M. Zorluoglu[2]

Pace University
[1]Mathematics and [2]Mathematics and Philosophy Departments
One Pace Plaza, New York, NY 10038, U.S.A.
educasse@pace.edu lquintas@pace.edu jz06163p@pace.edu

January 10 2018

Abstract

Let $\mathcal{U}(n)$ denote the *set of unlabeled graphs of order n* and U the graph with vertex set $\mathcal{U}(n)$ such that two vertices *G* and *H* in U are adjacent if and only if *G* and *H* differ by exactly by one edge. By introducing different $\mathcal{U}(n)$ and defining probabilities corresponding to one-edge transformations between vertices of U, Markov processes are introduced which yield a variety of random graph processes. The latter are fruitful areas to explore to determine the properties of $\mathcal{U}(n)$, U, and in particular the properties of these random graph processes.

1. Introduction

Let $\mathcal{U}(n)$ denote a specified *set of unlabeled graphs of order n*. Define a *one-edge transformation* on the graphs of $\mathcal{U}(n)$ such that this transformation is probabilistically based and defines the *transition probability* between any two elements of $\mathcal{U}(n)$. Starting at any element of $\mathcal{U}(n)$ proceed on a *random walk* via this one-edge transformation and its transition probabilities. The union of all such random walks defines the *transition digraph* T(*n*) for this *random graph process on* $\mathcal{U}(n)$. The *node set of* T(*n*) is the set $\mathcal{U}(n)$ and its *arcs* (*G*, *H*) are between any pair of graphs *G*, *H* in $\mathcal{U}(n)$ for which *H* is a one-edge transformation of *G*.

Investigations in this context consist of determining the properties of T(*n*), the elements of $\mathcal{U}(n)$, and of $\mathcal{U}(n)$ itself.

We now explicitly define the concept of a one-edge transformation consisting of adding an edge to a given graph *G* in $\mathcal{U}(n)$. Namely, among the *N* possible edges $\{u, v\}$ in the complement of *G* such that $G \cup \{u,v\}$ is in $\mathcal{U}(n)$ select one such edge with probability 1/*N*. This set of *N* edges is called the *admissible edges* for *G*. That is, select an edge to add to *G* with equal probability from among the set of admissible edges for *G*.

In what follows a variety of explicit examples with comments on each are provided.

2. The bounded random graph process

Let $\mathcal{U}(n)$ denote the set of all unlabeled graphs of order n having no vertex of degree greater than f. The admissible edges here are those edges that when added to a graph G in $\mathcal{U}(n)$ do not introduce a vertex of degree greater than f. This defines the *random f-graph process.* The case $f = n - 1$ is the unrestricted case and is known simply as the *random graph process*. This model has been the subject of much study with results found in the books **[1][2,** p. 38**]**. Theoretical and implementation techniques for the bounded degree cases are provided in **[1]** and are applicable to the random graph processes given in the following **Sections 3** to **6**.

The smallest order for which some of the properties of this random graph process can be visualized is $n = 4$. In **Figure 2.1** the transition digraph T(n) for the random 3-graph process for $n = 4$, i.e., the ($n - 1$) case is shown. Note that the graph U in the abstract is the underlying graph for T(n). Displayed in **Figure 2.2**, is the transition digraph for the random 2-graph process for $n = 4$.

3. The at most unicyclic random graph process

Let $\mathcal{AU}(n)$ denote the set of unlabeled graphs of order n having at most one cycle (such graphs are called *at most unicyclic*). Studies of this random graph process can be found in **[3][4]**. In this process the admissible edges are those that when added to a graph in $\mathcal{AU}(n)$ do not create a graph with two or more cycles. The transition digraph for the *at most unicyclic random graph process* for $n = 4$ is shown in **Figure 3.1**.

4. The unicyclic random graph process

Let $\mathcal{U}(n)$ denote the set of unlabeled graphs of order n having exactly one cycle. Initial studies of this random graph process can to be found in **[5]**. This random process has the property that its transition digraph T(n) has $n - 2$ components. See **Figure 4.1** for the case $n = 5$.

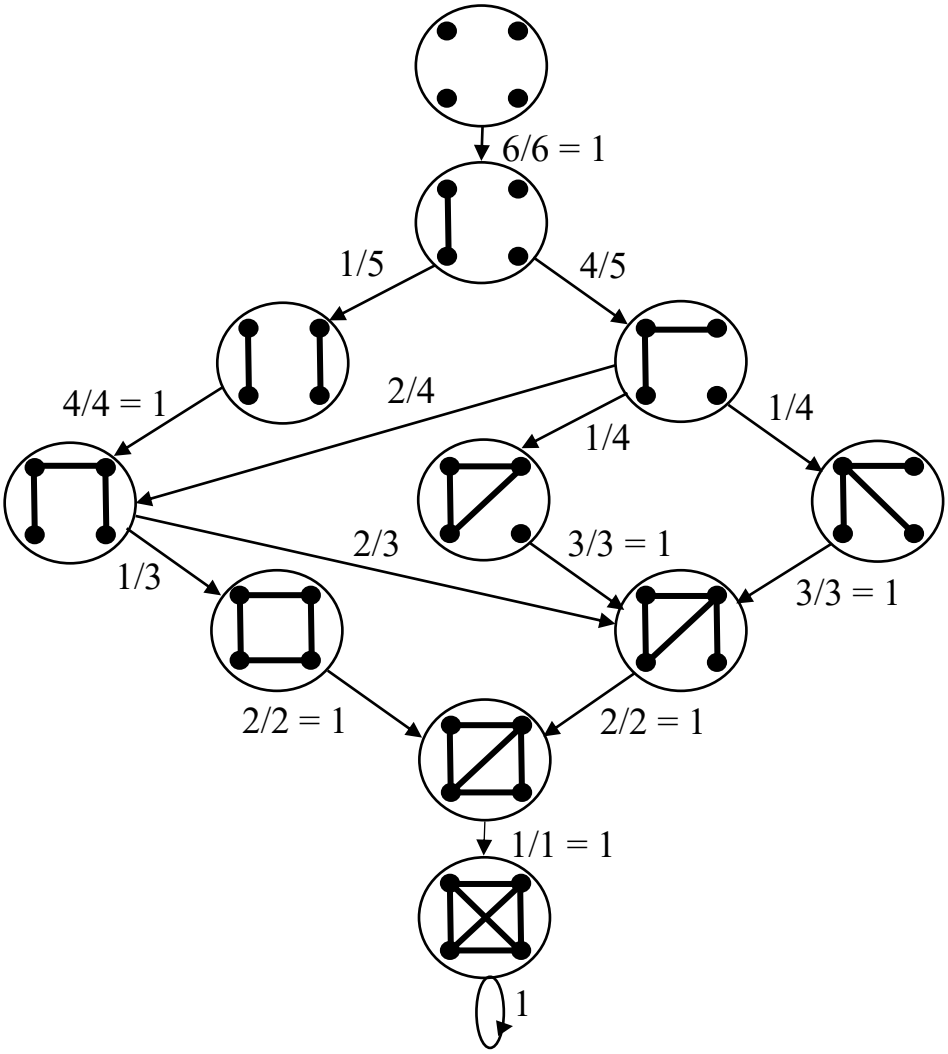


Figure 2.1 The transition digraph for the random graph process for $n = 4$

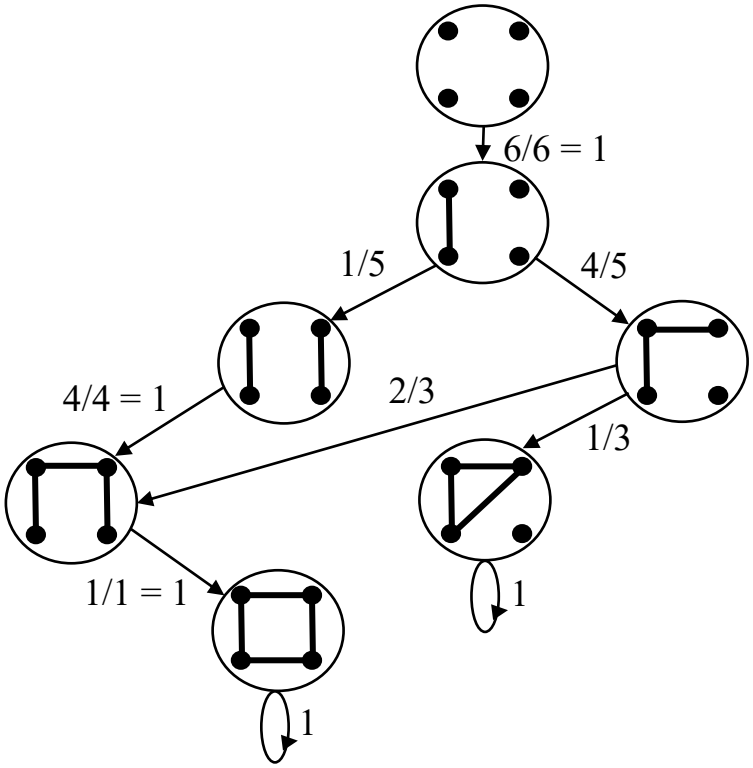


Figure 2.2 The transition digraph for the random 2-graph process for $n = 4$

5. The at most uni-*g*-cyclic random graph process

Let $\mathcal{C}\mathcal{C}(n)$ denote the set of unlabeled graphs of order n having at most one cycle and that cycle has order g. Further combining this with the additional condition that no vertex have degree greater than f provides applications in chemistry. For such a chemistry application, using only the degree $f = 4$ condition, see **[6][7]**. It is as an exercise for the reader to draw the transition digraph for the at most uni-3-cyclic random graph process for $n = 6$.

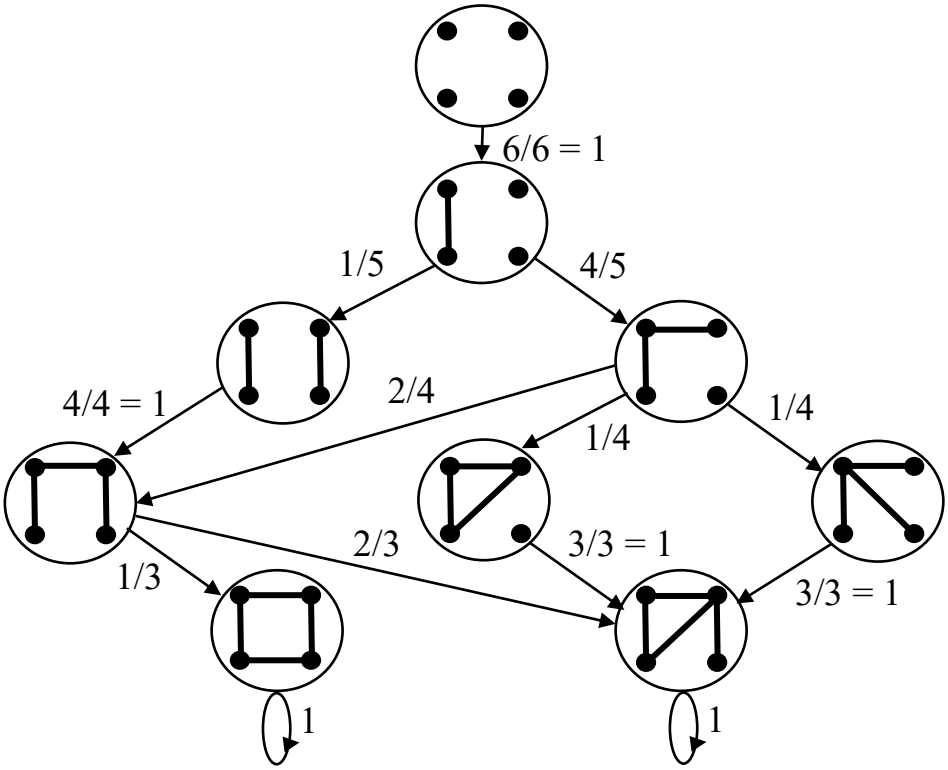


Figure 3.1 The transition digraph for the at most unicyclic random graph process for $n = 4$

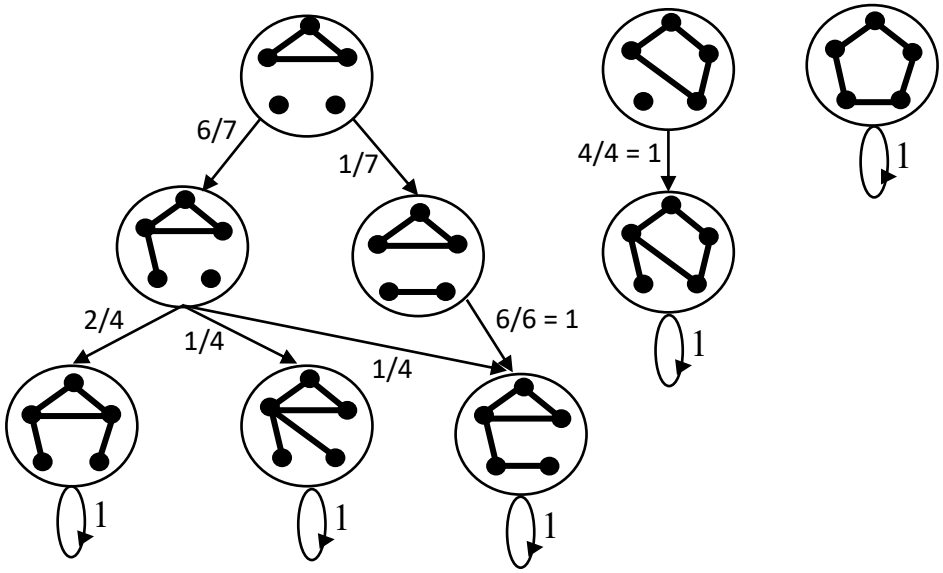


Figure 4.1 The transition digraph for the unicyclic random graph process for $n = 5$

6. The reversible random graph process

Let $\mathcal{U}(n)$ denote the set of all unlabeled graphs of order n. The distinguishing feature of the random graph process defined here is that the one-edge transformations permit both adding and deleting edges to a given graph, see **[8][9]**. The admissible edges consist of both the edges in a given graph G and those in the complement of G. An admissible edge is selected with equal probability and if not in G, the edge is added to G or if the selected edge is in G, the edge is deleted from G. Note that the number N of admissible edges at each step is C(n, 2) and each selection is chosen with probability 1/ C(n, 2).

The transition digraph for the *reversible random graph process* for $n = 4$ is shown in **Figure 6.1**.

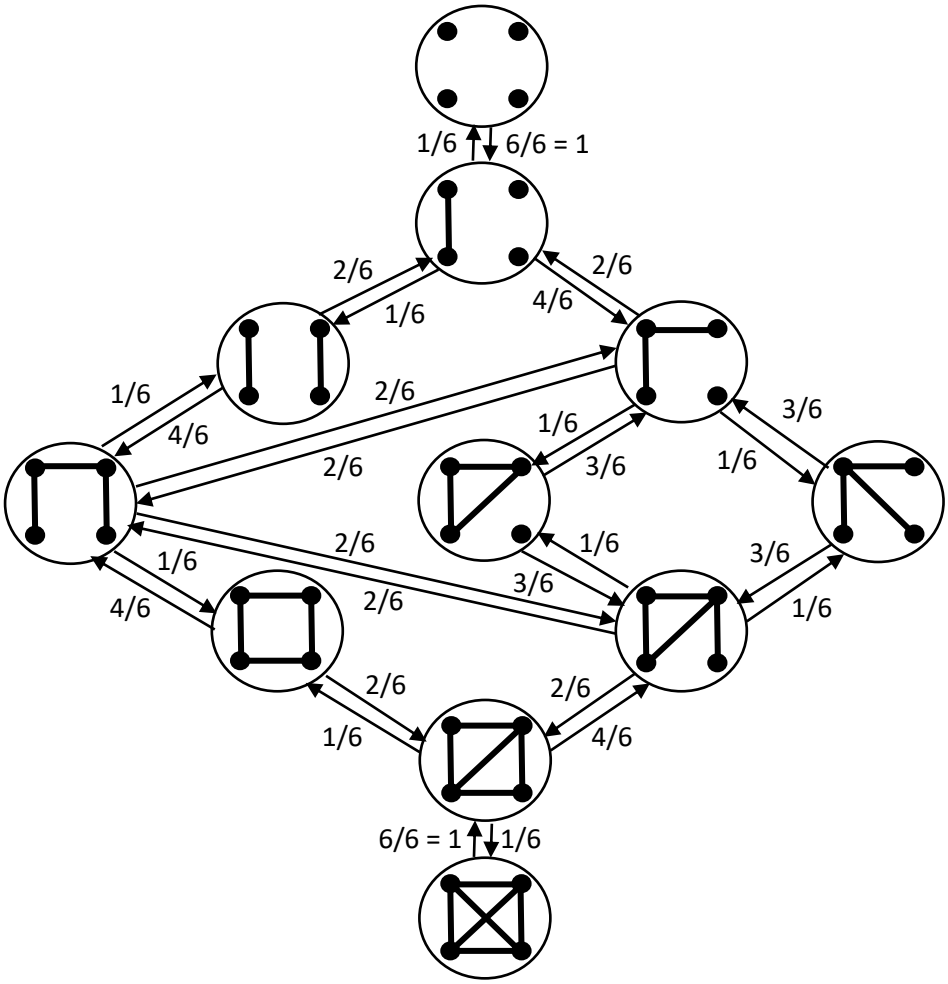


Figure 6.1 The transition digraph for the reversible random graph process for $n = 4$

7. Initial graphs, terminal graphs, and levels

A graph G in $\mathcal{CC}(n)$ is called an *initial graph*, if the deletion of any edge of G yields a graph not in $\mathcal{CC}(n)$. If the empty graph nK_1 is in $\mathcal{CC}(n)$, then nK_1 is defined to be an initial graph.

A graph G in $\mathcal{CC}(n)$ is called a *terminal graph*, if the addition of any edge to G yields a graph not in $\mathcal{CC}(n)$. If the complete graph K_n is in $\mathcal{CC}(n)$, then K_n is defined to be a terminal graph.

A graph G in $\mathcal{CC}(n)$ with L edges in a random graph process defined here is said to be on *level L*. Also note that each of the transition digraphs T(n) discussed here are bipartite graphs.

8. Traceability

A path that contains every vertex of a graph is called a *Hamilton path*. A graph is called *traceable* if it contains a Hamilton path (see **[10]**). For directed graphs the Hamilton path required must be a directed path. In the case of the reversible random graph process, the transition digraph is traceable is equivalent to its underlying graph is traceable.

9. Problems

For the above defined random graph processes study and resolve the following problems.

P1 What is the order of $\mathcal{CC}(n)$ = T(n)?

P2 What is the size of T(n)?

P3 What are the initial graphs of $\mathcal{CC}(n)$?

P4 What are the terminal graphs of $\mathcal{CC}(n)$?

P5 What is the probability distribution for the graphs on level L? or What is the probability distribution for the terminal graphs when not all of them are on the same level?

P6 For the reversible random graph process is the underlying graph of T(n) traceable?

References

[1] K.T. Balińska and L.V. Quintas, *Random Graphs with Bounded Degree*, Publishing House, Poznań University of Technology, Poznań (2006).

[2] B. Bollobás, *Random Graphs*, Academic Press, New York (1985).

[3] E.G. DuCasse, L.V. Quintas, and J.M. Zorluoglu, The At Most Unicyclic Random Graph Process, Mathematics Department, Pace University, New York, No. 1 (2017).

[4] E.G. DuCasse, L.V. Quintas, and J.M. Zorluoglu, The at most unicyclic random graph process: Order, reversibility, and traceability, (to appear).

[5] E.G. DuCasse and L.V. Quintas, The unicyclic random graph process, (to appear).

[6] S-H. Cha, E.G. DuCasse, K. Kravette, D.M. Mendoza, and L.V. Quintas, A graph model related to chemistry: Conversion of methane to higher alkanes, *International J. Chemical Modeling* **7** (2015) 17-29.

[7] S-H. Cha, E.G. DuCasse, K. Kravette, D.M. Mendoza, and L.V. Quintas, Graph Theory and the conversion of methane to higher order alkanes, *Congressus Numerantium* **224** (2015) 75-90.

[8] K.T. Balińska and L.V. Quintas, The reversible random *f*-graph process, CSIS *Pace University Technical Report* **132** (1998).

[9] K.T. Balińska, M.L. Gargano, and L.V. Quintas, The reversible random *f*-graph process with loops, *Proceeding of a Festschrift Held on the Occasion of the Retirement of Professor Robert Bumcroft, Department of Mathematics, Hofstra University* (2007) 118-128.

[10] E.G. DuCasse, L.V. Quintas, and M.O. Zimmler, Graphs whose vertices are forests with bounded degree: Traceability, *Bulletin of the Institute of Combinatorics and its Applications* **66** (2012) 65-71.

Infinite Classes of Triangular Graphs

A.Delgado, M.Lewinter, and K.Phillips

marty5040@gmail.com

Abstract

The *n*-th triangular number, $t_n = \frac{n(n+1)}{2}$, where *n* is a natural number. A nontrivial graph is triangular if:

1) its vertices are labeled by distinct triangular numbers,

2) each edge is weighted by the product of the labels of its end vertices, and

3) each edge weight is a distinct triangular number.

Similar graphs in which the vertex labels are oblong, (of the form $n(n+1)$), and *k*-long (of the form $n(n+k)$, where k is a fixed positive integer) have been studied. [4, 5, 6]. Using Pell equations, we obtain various infinite classes of triangular graphs.

I. Introduction

The *n*-th triangular number t_n equals 1 + 2 + 3 + ... + *n*. See [1, 2]. The closed form is given by

$$t_n = \frac{n(n+1)}{2}$$

The great 19th century mathematician, Gauss, proved that every positive integer can be expressed as the sum of three or fewer triangular numbers. See [1, 2].

It should be noted that there are infinitely many triangular numbers such as $t_8 = 36$ that are squares, but their density in the set of triangular numbers is zero. Note, also, that a given number m is triangular if and only if $1 + 8m$ is a square. These facts will be required later on.

We call t_n and t_m *compatible*, if the product $t_n t_m$ is triangular. For example, $t_2 = 3$ and $t_5 = 15$ are compatible since $3 \times 15 = 45 = t_9$.

II. *k*-long Graphs

In [3], a number is called *k-long* if it is of the form $n(n+k)$. This concept generalizes oblong numbers, that is, numbers of the form $n(n+1)$. It is shown there that the product of two consecutive *k*-long numbers is *k*-long. In [4, 5], a graph G is called *k*-long if

1. The vertices of G can be labeled with distinct *k*-long numbers.
2. The weights of the edges are the products of their end vertices.
3. The weights are distinct *k*-long numbers.

It should be noted that the set of labels and the set of weights need not be disjoint.

III. Triangular Graphs

In the spirit of section II, a *triangular graph* is defined as follows. A graph G, is triangular if

1. The vertices of G can be labeled with distinct triangular numbers.
2. The weights of the edges are the products of the labels of their end vertices.
3. The weights are distinct triangular numbers.

As was the case with *k*-long graphs, the set of labels and the set of weights need not be disjoint.

We suspect that the product of consecutive triangular numbers (not including 1) is never triangular, thereby making triangular graphs more difficult to construct. A computer search involving the first ten million triangular numbers strengthened our suspicion.

The path P_5 is triangular, as can be seen by labeling the vertices sequentially 1, 3, 15, 66, and 406. These labels generate the weights (sequentially) 3, 45, 990, and 26796. The vertex labels are t_1, t_2, t_5, t_{11}, and t_{28}, and the edge weights are t_2, t_9, t_{44}, and t_{231}.

IV. The Pell Equation and its Generalization

It is well known [7] that the Pell equation $x^2 - ky^2 = 1$ is solvable in positive integers provided that *k* is not a square.

It will be useful to convert the solution $x = a$ and $y = b$ into the formal expression $a + b\sqrt{k}$. If *b* is the smallest (positive) solution, then all solutions are the coefficients of $\left(a + b\sqrt{k}\right)^n$.

The generalized Pell equation $x^2 - ky^2 = j$ may or may not be solvable. $x^2 - 3y^2 = 2$ has no solution in integers as can be seen by writing it mod 3, yielding $x^2 \equiv 2 \pmod 3$.

On the other hand, if the generalized Pell equation $x^2 - ky^2 = j$ has a solution, $x = c$ and $y = d$, then it has infinitely many solutions expressed formally as $(c + d\sqrt{k})(a + b\sqrt{k})^n$. See [7].

Example: Given the generalized Pell equation $x^2 - 3y^2 = 4$, with solution $x = 4$, $y = 2$, first solve the associated ordinary Pell equation $x^2 - 3y^2 = 1$ and obtain its formally expressed solutions $(2 + \sqrt{3})^n$. Then the given generalized Pell equation has the formal solutions

$$(4 + 2\sqrt{3})(2 + \sqrt{3})^n$$

$n = 2$, for example, yields $(4 + 2\sqrt{3})(7 + 4\sqrt{3}) = 52 + 30\sqrt{3}$, from which one obtains the solution $x = 52$ and $y = 30$.

V. Finding Compatibles

The Pell equations will now be employed to find infinitely many compatibles, t_r, for a given non-square triangular number k. That is, indices, r, must be found for which kt_r is triangular. (Triangular numbers that are squares can easily be avoided in the construction of triangular graphs, as they have zero density in the set of triangular numbers.)

Using the fact that a given number m is triangular if and only if $1 + 8m$ is a perfect square, the following sequence of equations are obtained.

$$8kt_r + 1 = s^2$$

$$8k\frac{r(r+1)}{2} + 1 = s^2$$

$$4kr(r+1) + 1 = s^2$$

$$k(4r^2 + 4r + 1) + (1 - k) = s^2$$

$$k(2r+1)^2 + (1 - k) = s^2$$

$$s^2 - k(2r+1)^2 = 1 - k$$

Letting $x = 2r + 1$, results in the generalized Pell equation $s^2 - kx^2 = 1 - k$, which can be written as

$$\bar{s}^2 - k\bar{x}^2 = 1 - k \tag{1}$$

to distinguish its solutions from those of the associated Pell equation $s^2 - kx^2 = 1$.

Since $\bar{x} = 2r + 1$, it must be odd in order to obtain integer values of *r*. Observe that (1) has the formally expressed solution $1 + \sqrt{k}$, since $\bar{s}_0 = \bar{x}_0 = 1$ is a solution.

Two lemmas are required.

Let $s_1 + x_1\sqrt{k}$ be a formal solution to the Pell equation, $s^2 - kx^2 = 1$. Then the *n*-th solution, $s_n + x_n\sqrt{k} = (s_1 + x_1\sqrt{k})^n$. It follows that

$$s_{n+1} + x_{n+1}\sqrt{k} = (s_n + x_n\sqrt{k})(s_1 + x_1\sqrt{k}).$$

Equating coefficients of 1 and $\sqrt{k}$ yields Lemma 1.

Lemma 1:

$$s_{n+1} = s_n s_1 + k x_n x_1$$
$$x_{n+1} = s_n x_1 + x_n s_1$$

The *n*-th formal solution to (1), $\bar{s}_n + \bar{x}_n\sqrt{k} = (s_n + x_n\sqrt{k})(1 + \sqrt{k})$. Equating coefficients yields Lemma 2.

Lemma 2:

$$\bar{s}_n = s_n + k x_n$$
$$\bar{x}_n = s_n + x_n$$

It has now been established that the generalized Pell equation, $\bar{s}^2 - k\bar{x}^2 = 1 - k$ has infinitely many solutions. Recall, however, that $\bar{x}$ must be odd, since $\bar{x} = 2r + 1$, where *r* is the index of a triangular number compatible with *k*.

In light of the second equation of Lemma 2, this translates into the requirement that s_n and x_n have opposite parity. To show this, two cases must be considered.

Case 1: *k* is odd.

It can be seen from the Pell equation $s^2 - kx^2 = 1$, that x^2 and s^2 have opposite parity, from which it follows that *s* and *x* have opposite parity, and we are done.

Case 2: *k* is even.

It can be seen from the Pell equation that s_n is always odd. Then the second equation of Lemma 1 implies that $x_{n+1} \equiv x_1 + x_n \pmod 2$. If x_1 is even, this becomes $x_{n+1} \equiv x_n \pmod 2$, implying that x_n is always even, and we are done. If x_1 is odd, then $x_{n+1} \equiv 1 + x_n \pmod 2$, so the parity of $x_n$will alternate. This will yield infinitely many solutions in which s_n and x_n have opposite parity, and the lemma is proven.

In summary, every non-square triangular number has infinitely many compatibles. Infinite classes of triangular graphs will now be obtained.

VI. Infinite Classes of Triangular Graphs

Theorem 1: All paths are triangular.

Proof: Given n, let S_n be a strictly increasing sequence of n triangular numbers such that any two consecutive members of S_n are compatible. Such a sequence exists, for all n, by the result of the previous section. Then the path P_n is indeed triangular. Simply label its vertices consecutively using the members of S_n. □

Theorem 2: All cycles are triangular.

Proof: Given n, let $S_{n-1} = (s_1, s_2, s_3, \ldots, s_{n-1})$, be a strictly increasing sequence of n - 1 triangular numbers such that any two consecutive members of S_{n-1} are compatible, and $s_1 = 1$. Now given the cycle, C_n, label any $n - 1$ consecutive vertices with the triangular numbers of S_{n-1}. Since these triangular numbers are strictly increasing as we go around the cycle, we will obtain a strictly increasing sequence of weights, $w_1, w_2, \ldots, w_{n-2}$.

Denote the label of the final vertex by t. Choose $\underline{t}$ so that it is compatible with, and greater than, s_{n-1}. (Note that any t is compatible with 1.) Then the penultimate weight, w_{n-1}, is strictly greater than w_{n-2} since $t > s_{n-1}$. It follows that the weights, $w_1, w_2, \ldots, w_{n-1}$ are distinct since they form a strictly increasing sequence. Unfortunately, the final weight, $w_n = t < w_{n-1}$, since $w_{n-1} = ts_{n-1}$. To ensure that w_n does not equal any of the previous weights $w_1, w_2, \ldots, w_{n-2}$, increase t if necessary so that it is greater than w_{n-2}, which can be done since s_{n-1} has infinitely many compatibles. □

Theorem 3: All trees are triangular.

Proof: Let T be a rooted tree. Label the root with a triangular number which is denoted by a_{11}. Label the second row of vertices from left to right by the strictly increasing triangular numbers a_{21}, a_{22}, a_{23}… where all of these labels are compatible with a_{11}. If we denote the weights of the edges

linking the first and second rows of vertices (from left to right) by w_{11}, w_{12}, w_{13}…, it follows that these weights form a strictly increasing sequence.

Choose the triangular number, a_{31}, that is the first vertex of the third row, so that it is compatible with its parent vertex label, and is greater than the maximum of the weights of all previous edges. This will ensure that the edge weights are distinct. As was the case with the second row of vertices, the sequence of labels of the vertices of the third row, a_{31}, a_{32}, a_{33}, is strictly increasing. This guarantees that the vertex labels and edge labels thus far are distinct. Since each triangular number (except squares which are not considered) has infinitely many compatibles, arbitrarily large trees can be accommodated by continuing this algorithm.

The algorithm terminates when the desired labeling is achieved. □

Theorem 4: All unicyclic graphs are triangular.

Proof: Label the cycle using the algorithm of Theorem 2. Let j be the smallest index for which deg (a_j) > 2. Treating a_j as the root of the pendant tree attached to it, label the first vertex of the second row of its pendant tree so that it is strictly greater than all the weights thus far. The remainder of the labeling using the algorithm of Theorem 3 will ensure the distinctness of the vertex labels and edge weights thus far. Note that a_j is the initial vertex in the subsequence of vertices with degrees greater than 2. Continue the algorithm until the pendant trees of these vertices are fully labeled. □

Theorem 5: All bicyclic graphs such that the two cycles share exactly one vertex are triangular.

Proof: Label the shared vertex by 1, and apply several of the above algorithms. □

Corollary 1: Let graph G contain an arbitrary number of cycles all containing a vertex, v, and such that the intersection of the vertices of any pair of cycles is v. G may contain tree structures. Then G is triangular.

Proof: Assign the label 1 to v. Then apply the algorithms of Theorems 3 and 4. □

Theorem 6: Let graph G be constructed as follows. Begin with the cycle C_n and label one of its vertices, v, with 1. Then add $n – 3$ edges so that deg(v) = $n – 1$. Then G is triangular.

Proof: Label the vertices of the initial C_n using the algorithm of Theorem 2. Note that the added edges of the second step of the construction have the same weights as the vertices they are attached to that are not labeled 1. Hence, they will all be distinct. □

VII. Conclusion

We close with the following definition and open questions.

Definition: The distinct triangular numbers t_n and t_m, are *mutually compatible*, if there exists a triangular number other than 1 that is compatible with both of them.

Example: $t_2 = 3$ and $t_{11} = 66$ are mutually compatible, since they are both compatible with $t_5 = 15$.

Question 1: Are any given pair of distinct triangular numbers mutually compatible? A positive answer would dramatically enhance the discovery of new classes of triangular graphs, for example, the wheel, W_n.

Question 2: As was previously indicated, with the exception of 1 and 3, we have not found a pair of *consecutive* compatible triangular numbers among the first ten million triangular numbers, and we conjecture that there are no such pairs. We seek a proof or counter example.

References

[1] M. Lewinter, J. Meyer, *Elementary Number Theory with Programming*, Wiley & Sons. 2015.

[2] M. Lewinter, W. Widulski, *The Saga of Mathematics: A Brief History*. Prentice-Hall, 2002.

[3] A. Delgado, M. Gargano, M. Lewinter, J.Malerba, Introducing *k*-long numbers. *Cong.Num*.189, (15-23) 2008.

[4] A. Delgado, M. Lewinter, L.V. Quintas, *k*-long graphs. *Cong.Num*.196, (95-106) 2009.

[5] A.Delgado, M.Lewinter, B.Phillips, *k*-long graphs II. GTN of NY. LXVI (29-34) 2014

[6] A. Delgado, Oblong graphs. *Graph Theory Notes of New York* LVII, (10-13) 2009.

[7] K. Conrad, Pell's Equation, I.
http://www.math.uconn.edu/~kconrad/blurbs/ugradnumthy/pelleqn1.pdf

Integrated Graphs

A.Delgado* and M.Lewinter

*Doctoral Candidate, Columbia University

anthony.delgado12@gmail.com

In [1] a graph is called *integrated* if the vertices are colored either black or white, and at least half of the neighbors of each vertex have the opposite color of that vertex. It follows that all bipartite graphs can be integrated using any valid bipartite color assignment, since all the neighbors of any vertex in a bipartite graph have the opposite color of that vertex. In this sense, integrated graphs generalize bipartite graphs.

While graphs are bipartite if and only if they contain no odd cycles, it is shown in [1] that all graphs can be integrated. We present the proof after the following definition.

Definition: Let the vertices of a graph G be colored using the colors black and white. Then an edge of G is called *balanced* if its endvertices have opposite color.

Now given a graph G, there are many ways to color its vertices black or white. Since, however, there are only finitely many ways to color it, there is some coloring of G that has the maximum number of balanced edges. This graph must be integrated! If not, one can find a vertex, v, such that the majority of its neighbors have the same color as v. But if we change the color of v, we can produce more balanced edges, contradicting the assumption that the original coloring has the maximum number of balanced edges.

The algorithm that assigns a color to each vertex is not obvious, except for certain classes of graphs.

Example 1: To integrate the complete graph, K_{2n}, color any n vertices white, and the remaining n vertices black. To integrate the complete graph, K_{2n+1}, color any n vertices white, and the remaining $n + 1$ vertices black.

Example 2: To color the wheel, $C_{2n} + v$ (that is, W_{2n+1}), give C_{2n} a bipartite color assignment and give v either color.

It is well-known that if G and H are bipartite graphs, so is G × H. [2] This is done by coloring a vertex in G × H white if its parent vertices in G and H have the same color, and coloring a vertex in G × H black if its parent vertices in G and H have opposite color. Using this same algorithm, it can be shown that if graphs G and H are integrated, we can integrate G × H with ease.

Definition: A graph G is *strictly* integrated, if the vertices are colored either black or white, and more than half of the neighbors of each vertex have the opposite color of that vertex. It follows that all bipartite graphs can be strictly integrated using any valid bipartite color assignment, since all the neighbors of any vertex in a bipartite graph have the opposite color of that vertex.

Not all graphs, however, can be strictly integrated, as can be seen by integrating K_3.

Question 1: Characterize those graphs that cannot be strictly integrated.

Question 2: Devise an efficient algorithm for integrating any graph.

References

[1] Paul Zeitz, *The Art and Craft of Problem Solving*. Wiley, 2006.

[2] F.Buckley, M.Lewinter, *A Friendly Introduction to Graph Theory*. Prentice-Hall, 2003.

Closure Theorems for the Binding Number of a Graph

M. Yatauro
Department of Mathematics, Penn State Brandywine, Media, PA, USA
mry3@psu.edu

Let G be a finite, simple graph. For a subset S of V (G), let N(S) be the neighbor set of S. The binding number of G, introduced by Woodall and denoted by bind(G), is the minimum value obtained by the ratio jN(S)j=jSj when considering all possible subsets S of V (G). If bind(G) _ b, we say that G is b-binding. We will discuss two theorems, one for 0 < b < 1 and one for b _ 1, which each provide a number k so that if deg(x) + deg(y) _ k for some non-adjacent vertices x; y of G, then G is b-binding if and only if G + xy is b-binding. It is also shown that our k values are best possible.

A GRAPHICAL PERSPECTIVE ON REARRANGEMENTS OF THE SIMPLE RANDOM WALK

MARINA SKYERS

Abstract. Let S_n be the random walk defined on $(0,1)$ and let S_n^* be the quantile of S_n. The S_n have been the subject of intense study; their definition is immediately intuitive. Nevertheless, they are quite disorderly and this disorder is mirrored by the fact that, pointwise, $\left(\frac{S_n}{\sqrt{n}} | n \in \mathbb{Z}^+\right)$ diverges. In this paper we will see how to effectively rearrange S_n to arrive at S_n^* and thus achieve almost sure convergence. This is done via a suitable choice of permutation $F : \{0,1\}^n \to \{0,1\}^n$ such that $S_n^* = S_n \circ F$. We will describe how to minimize the graph-theoretic complexity of these permutations and also show that they satisfy some additional nice properties.

1. Introduction

For $x \in C := \{0,1\}^{\mathbb{N}^+}$ excluding the two constant sequences, identify x with $\sum_{i=1}^{\infty} \frac{x_i}{2^i} \in (0,1)$. For dyadic rationals, choose the representation with a tale of zeros. Define for $1 \leq i \leq n$, $R_i(x) := (-1)^{1+x_i}$ and $S_n(x) := \sum_{i=1}^{n} R_i(x)$. Define $\text{Weight}_n(x)$ as the sum of the first n coordinates of x. Notice that $S_n(x) = -n + 2\text{Weight}_n(x)$.

Obviously, $S_n(x)$ and $\text{Weight}_n(x)$ depend only on the first n coordinates of x. So, for binary sequences $\boldsymbol{r}$ of length n, we can define

$$S_n(\boldsymbol{r}) := S_n(x) \text{ for any } x \in C \text{ such that } x \supseteq \boldsymbol{r}$$

$$\text{Weight}(\boldsymbol{r}) := \text{Weight}_n(x) \text{ for any } x \in C \text{ such that } x \supseteq \boldsymbol{r}.$$

Observe that $S_n(\boldsymbol{r}) = -n + 2\text{Weight}(\boldsymbol{r})$. We can see this in the graph of $S_n(x)$, at each level n. The graphs of $S_n(x)$, for $n = 5, 6, 7$, will be illustrated below.

In 1733, de Moivre postulated the first version of the central limit theorem for independent random variables that take on values ± 1. It is an important special case of the central limit theorem that the $\frac{S_n}{\sqrt{n}}$ converge in distribution to the standard normal on $(0,1)$. Well-known results ([4], [5] and [6]) show this cannot possibly be improved to almost sure convergence.

The S_n have been the subject of intense study. The definition is accessible and intuitive and each S_n is canonically represented as the sum of an i.i.d. family of n irreducibly simpler random variables (the $R_i(x)$). Nevertheless the S_n are quite disorderly (as we can see in the below graphs for S_n, as n increases).

The quantile of S_n turns out to be a very orderly, non-decreasing step function, which we will call S_n^*, and it can be explicitly defined as follows. Define steps $A_{n,i}$, $i = 0, \ldots, n$, where

$$A_{n,i} = \left(\frac{1}{2^n} \sum_{j=0}^{i-1} \binom{n}{j}, \frac{1}{2^n} \sum_{j=0}^{i} \binom{n}{j} \right].$$

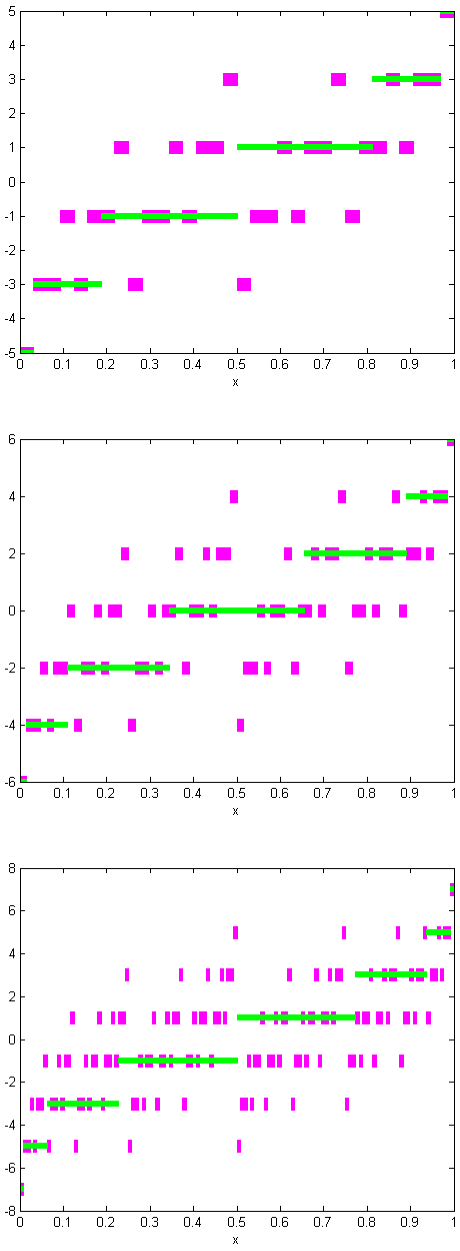
For such i, and for all $x \in A_{n,i}$ we define $S_n^*(x) = -n + 2i$. For fixed $n \in \mathbb{N}^+$, let $\kappa = \kappa_n = \kappa_n(x)$ be the following integer: $\kappa = \sum_{i=1}^{n} x_i 2^{n-i}$. Then $x \in \left[\frac{\kappa_n(x)}{2^n}, \frac{\kappa_n(x)+1}{2^n} \right)$.

We can compute $S_n^*(x)$ by identifying the step $A_{n,i}$ that includes the interval $\kappa_n(x)$. Note that, for each $n \in \mathbb{N}^+$ and for $\kappa \in [0, 2^n) \cap \mathbb{N}$, $-n \leq S_n(\kappa), S_n^*(\kappa) \leq n$ and S_n, S_n^* satisfy the dualization equations

$$S_n(\kappa) = -S_n(2^n - 1 - \kappa),$$

$$S_n^*(\kappa) = -S_n^*(2^n - 1 - \kappa).$$

Below are the graphs for S_n and $S_n^*(x)$ when $n = 5, 6, 7$.



In this paper we will investigate representations of the quantile of S_n that are as close as possible to the canonical representation for S_n, via permutations $F : \{0,1\}^n \to \{0,1\}^n$ such that $S_n^* = S_n \circ F$. In fact, we will see that $(F_n | n \in \mathbb{N}^+)$ is uniformly primitive recursive ([1],[3]).

2. Representation results

Skorokhod proved the following in [7].

Theorem 1. *Suppose that on a probability space, we have random variables X_n, $n \in \mathbb{N}^+$, and suppose the X_n converge weakly to X. Then on $([0,1], B([0,1]), \lambda)$, there are random variables Y_n, $n \in \mathbb{N}^+$, and Y, with the same distributions as the X_n and X, respectively, and such that the Y_n converge almost surely to Y.*

If in Skorokhod's Theorem, we start from $X_n = \frac{S_n}{\sqrt{n}}$, then the Y_n that result are exactly $\frac{S_n^*}{\sqrt{n}}$. So for each $n \in \mathbb{N}^+$, $\frac{S_n^*}{\sqrt{n}}$ has the same distribution as $\frac{S_n}{\sqrt{n}}$ and, more importantly, the $\frac{S_n^*}{\sqrt{n}}$ converge almost surely to the standard normal on $(0,1)$. An important question that arises here is, are there representations of S_n^* similar

to the canonical representation for S_n? And if so, how close can they be to the canonical representation for S_n? We can answer these questions as follows [8]. (For additional work related to the following results, see [2].)

Theorem 2. *For any n, there is a canonical one to one correspondence between permutations $F : \{0,1\}^n \to \{0,1\}^n$ such that $S_n^* = S_n \circ F$, and representations $S_n^* = \sum_{i=1}^n R_i^*$, where $\left(R_i^* \middle| 1 \leq i \leq n\right)$ is an i.i.d. family of random variables on $(0,1)$ such that each R_i^* depends only on the first n coordinates of x and takes on values $-1, 1$ with equal probability.*

In addition, the following theorem shows there are many such permutations.

Theorem 3. *For each n, there are exactly $\prod_{i=0}^n \left(\binom{n}{i}!\right)$ permutations $F : \{0,1\}^n \to \{0,1\}^n$ such that $S_n^* = S_n \circ F$.*

Corollary 4. *For each n, there are exactly $\prod_{i=0}^n \left(\binom{n}{i}!\right)$ families $\left(R_i^* \middle| i = 1, \ldots, n\right)$ as above.*

Additional criteria make some of these permutations more natural than (and therefore preferable to) others. We say $(F_n | n \in \mathbb{N}^+)$ is suitable if and only if for all n, F_n is a permutation of $\{0,1\}^n$ satisfying $S_n^* = S_n \circ F$ and such that:

(a) $(F_n | n \in \mathbb{N}^+)$ is explicitly and naturally definable, uniformly and highly effectively in n,

(b) if $\boldsymbol{r} \in \{0,1\}^n$ and $S_n^*(\boldsymbol{r}) = S_n(\boldsymbol{r})$, then $F_n(\boldsymbol{r}) = \boldsymbol{r}$,

(c) F_n is as close as possible to being self-inverse (even for fairly small n (such as $n = 5, 6, 7$), it is impossible for F_n to literally be self-inverse).

3. Rearrangements of the random walk

We first look at a variant, $(G_n | n \in \mathbb{N}^+)$, satisfying only the first two criteria, (a) and (b), as well as the composition equation, $S_n^* = S_n \circ G_n$. So each G_n will map Step to Weight ($\mathrm{Step}_n(\kappa) = \mathrm{Weight}_n(G_n(\kappa))$), and further, the mapping will be in an order-preserving fashion (except as ruled out by criterion (b)). This means that for all $0 \leq \kappa < 2^n$,

(i) If $\mathrm{Step}_n(\kappa) = \mathrm{Weight}_n(\kappa)$, then $G_n(\kappa) = \kappa$,

(ii) If $\mathrm{Step}_n(\kappa) \neq \mathrm{Weight}_n(\kappa)$, and, if further, $\kappa < m < 2^n$ and $\mathrm{Step}_n(\kappa) = \mathrm{Step}_n(m) \neq \mathrm{Weight}_n(m)$, then $G_n(\kappa) < G_n(m)$.

Lemma 5. *(i) and (ii) define a unique sequence $(G_n | n \in \mathbb{N}^+)$ satisfying the composition equations $S_n \circ G_n = S_n^*$.*

Proof. We have $A_{n,i} = \{\kappa | \mathrm{Step}_n(\kappa) = i\}$ and we define $B_{n,i} := \{\kappa | \mathrm{Weight}_n(\kappa) = i\}$. Further, let

$$A_{n,i}^1 := A_{n,i} \smallsetminus B_{n,i} = A_{n,i} \smallsetminus (A_{n,i} \cap B_{n,i}),$$

$$B_{n,i}^1 := B_{n,i} \smallsetminus A_{n,i} = B_{n,i} \smallsetminus (A_{n,i} \cap B_{n,i}).$$

These are the sets of things that are out of place on the i^{th} step, or of the i^{th} weight, respectively. We have the following equation:

$$\mathrm{card}\left(A_{n,i}^1\right) = \binom{n}{i} - \mathrm{card}\left(A_{n,i} \cap B_{n,i}\right) = \mathrm{card}\left(B_{n,i}^1\right).$$

$G_n \restriction A_{n,i}^1$ is simply the order-preserving bijection between $A_{n,i}^1$ and $B_{n,i}^1$.

□

In fact $(G_n | n \in \mathbb{N}^+)$ is uniformly primitive recursive in the following precise sense: there exists a single primitive recursive function $G(n, \kappa)$ such that for all n, $G(n, \cdot) \restriction \{0, \ldots, 2^n - 1\} = G_n$. Simply take $G(n, \kappa)$ to be equal to $G_n(\kappa)$, when $0 \leq \kappa < 2^n$, and supply a suitable default value (e.g., $G(n, \kappa) = 0$, or $G(n, \kappa) = \kappa$), when $\kappa \geq 2^n$ or $n = 0$, then we have defined a unique function $G : \mathbb{N}^2 \to \mathbb{N}$. It is not very difficult to show that G primitive recursive.

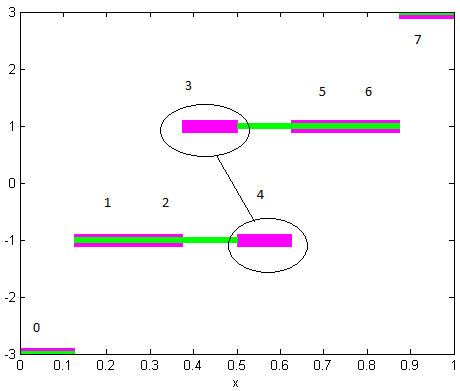
We also have an additional property for G. Each G_n satisfies the dualization equation $G_n(2^n - 1 - \kappa) = 2^n - 1 - G_n(\kappa)$.

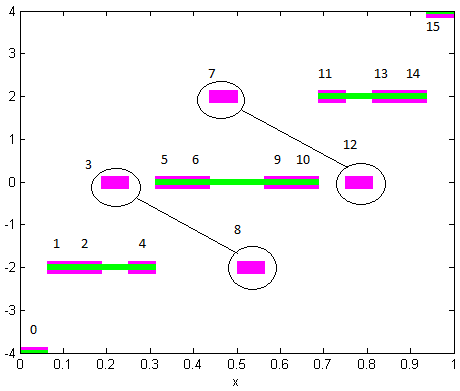
For $n = 3, 4, 5, 6, 7$ and each κ such that $\mathrm{Step}(n, \kappa) \neq \mathrm{Weight}(\kappa)$ (i.e., κ is out of place at level n), the orbit of κ under G_n is given in the following table.

n	Orbits under G_n
3	$\{3,4\}$
4	$\{7,3,8,12\}$
5	$\{16,7,3,8,5\}, \{15,24,28,23,26\}, \{11,17\}, \{13,18\}, \{14,20\}$
6	$\{32,42,15,34,49,30,19,40,56,60,55,58,47,27,13,24,11,6\}$, $\{31,21,48,29,14,33,44,23,7,3,8,5,16,36,50,39,52,57\}$
7	$\{64,15,34,21,48,73,46,69,39,25,68,30,11,5,16,36,22,65,23,66,$ $27,80,57,84,99,31,13,6,32,14,33,19,40,26,72,45,67,29,7\}$, $\{63,112,93,106,79,54,81,58,88,102,59,97,116,122,111,91,105,62,104,$ $61,100,47,70,43,28,96,114,121,95,113,94,108,87,101,55,82,60,98,120\}$, $\{3,8\}, \{51,74\}, \{53,76\}, \{124,119\}$

Our construction of $(F_n | n \in \mathbb{N}^+)$, which will satisfy all three criteria (a), (b) and (c), takes place within the general framework implicit in the construction of $(G_n | n \in \mathbb{N}^+)$. While G was implicitly constructed in two stages, F will be built in three. As before, the first stage is that F is the identity on the κ's that are in place: $F_n(\kappa) = \kappa$ if $\text{Step}_n(\kappa) = \text{Weight}_n(\kappa)$. Then identify which κ's are part of a two-cycle and pair them up. After we have maximized two-cycles (this satisfies criterion (c)), removing those from $A^1_{n,i}, B^1_{n,i}$ leaves us with sets $A^2_{n,i}, B^2_{n,i}$ of equal cardinality and we map $A^2_{n,i} \to B^2_{n,i}$ in an order-preserving fashion. As before, each F_n satisfies $F_n(2^n - 1 - \kappa) = 2^n - 1 - F_n(\kappa)$.

As noted in the table above, the first time there are values of κ that are out of place is when $n = 3$. Below are the graphs of F_n for $n = 3$ and $n = 4$.

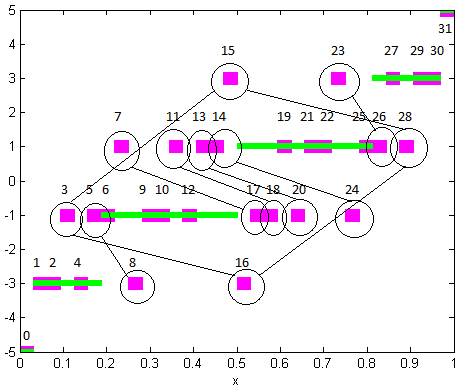


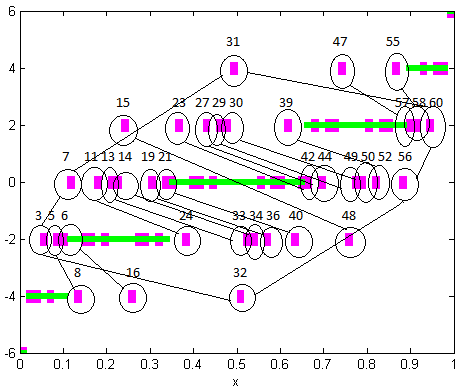


The first time there are values of κ that are not part of a two-cycle is when $n = 5$. For $n = 5, 6, 7$, the table below presents the orbits under F_n for those values of κ at level n.

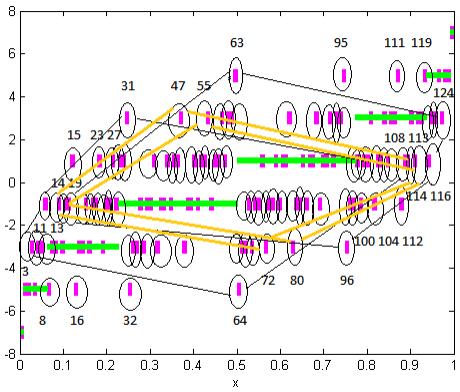
n	Orbits under F_n
5	$\{16, 28, 15, 3\}$
6	$\{32, 56, 60, 31, 7, 3\}$
7	$\{64, 108, 31, 11, 3\}, \{13, 72, 113, 47\},$ $\{14, 80, 114, 55\}, \{63, 19, 96, 116, 124\}$

The resulting cycles of these values of κ, corresponding to each of the rows of the table, are illustrated in the graphs below. We have a single four-cycle at $n = 5$ and a single six-cycle at $n = 6$.





At $n = 7$ we have two four-cycles and two five-cycles. Because the graph of F_n is rather complicated by $n = 7$, we will leave out the swaps from the graph and only illustrate the cycles. The four-cycles are highlighted in the figure below.

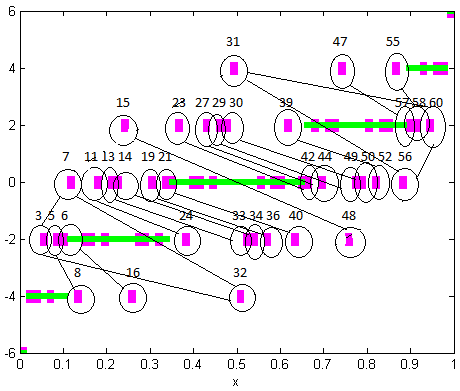


4. Graph-theoretic complexity of the permutations

The results we have presented do indeed narrow the distance between the S_n and the S_n^* with respect to the important issue of representation. The form of the composition equation that we have used so far, $S_n^* = S_n \circ F_n$, emphasizes the point of view of providing suitable representations of the S_n^*. But this equation could just as well be written in the form $S_n = S_n^* \circ F_n^{-1}$, which would emphasize the point of view of seeking to tame the disorder of the S_n. This is related to the rearrangement idea that is illustrated in the above graphs of F_n: we rearrange S_n to get S_n^*, and thus achieve almost sure convergence. The question remains how much rearranging of the S_n is optimal.

One point of view involves attempting to minimize the graph-theoretic complexity of the function F. As described in the construction of F above, F maximizes the number of two-cycles (with the proper choice of the two-cycles), but then will act just as the function G on the remaining κ's which are not part of a two-cycle. Of course we know by Theorem 3 that there are many other possible variants for the function F.

One may add some additional stages to the construction of F. In stage three (which might no longer be the terminal stage), we would seek to maximize the number of three-cycles just as we maximized the number of two-cycles in stage two, and fixed all the κ's which were in place (thereby maximizing the number of one-cycles) in stage one. If some κ remain outside the domain, proceed to stage four and continue. The goal would be to minimize lengths of cycles which could be viewed as one way of seeking to minimize the graph-theoretic complexity of the permutations. This idea is illustrated below for $n = 6$.



The existence of values of κ that are not part of two-cycles at level n, starting at $n = 5$, is the last phenomenon to create complications in the definition of the function F. It is conceivable that further interesting phenomenon (which do not create additional complications for the definition of F) first occur for some n larger than 5, and it is far from certain whether there are finitely many such n.

References

[1] Nigel Cutland, *Computability*, Cambridge University Press (1980).
[2] Dobrić, V., Garmirian, P., Skyers, M. and Stanley, L.J., *Polynomial time relatively computable triangular arrays for almost sure convergence*, preprint (2017).
[3] Herbert B. Enderton, *Computability Theory*, Academic Press (2010).
[4] W. Feller, *The general form of the so-called Law of the Iterated Logarithm*, Transactions of the American Mathematical Society, 54 (1943) 373-402.
[5] A. Khintchine, *Über dyadische brüche*, Math. Zeit., 18 (1923) 109-116.
[6] Kolmogoroff, A. *Über das gesetz des iterierten logarithmus*, Math. Ann., 101 (1929) 126-135.
[7] A. V. Skorokhod, *Limit theorems for stochastic processes*, Theory of Probability and its Applications, 1 (1956) 261-290.
[8] Skyers, M., *A tale of two sequences: a story of convergence, weak and almost sure*, Dissertation, Lehigh University (2012).

Department of Mathematics, Penn State, Brandywine Campus, Media, PA 19063, USA

Intersection and transformations of digraphs survey

Christina MD Zamfirescu[1]
Hunter College and Graduate Center
City University of New York
New York, NY 10065 and 10036, USA
zamfichris@gmail.com

Abstract. The intersection number of a digraph D is the minimum size of a set U, such that D is the intersection digraph of ordered pairs of subsets of U. The paper describes much of the work done in the area of intersection graphs and digraphs. Connections with applications in survey design can be drawn. The paper proves two main results:

Theorem 1 The intersection number of the line digraph of D equals the number of vertices of D that are neither sources nor sinks.

Theorem 2 If D contains no loops, the intersection numbers of total digraph, middle digraph and subdivision digraph of D are all equal to the number of vertices of D that are not sources, added to the number of vertices of D that are not sinks.

Keywords: intersection digraph, intersection number, line digraph, subdivision digraph, middle digraph, total digraph.

Introduction

A great deal of research has been done in the area of transformations on graphs and digraphs, found in connection with work done in groups on graphs.

The best known and most thoroughly studied among these transformations has been the line graph, that was officially introduced as such by Whitney [40] in 1932, and by 1970 has been completely characterized by Krausz [28], van Rooij and Wilf [34] and Beineke [3]. The middle graph, was introduced independently by Chikkodimath and Sampathkumar [10], and Hamada and Yoshimura [20]. Middle graphs have been characterized in several ways by Akiyama, Hamada and Yoshimura [1]. The total graph, was introduced in 1967 and studied by Behzad [2].

For over half a century transformations on digraphs, introduced as analogues of the corresponding transformations on graphs, have also received a great deal of attention. We refer to the line, total, and middle digraph, which have been introduced in 1960 by Harary and Norman [24], in 1964 by Chartrand and Stewart [9], and in 1981 (1977 in her PhD thesis) by Zamfirescu [42], respectively. Characterizations have been given by Heuchenne [25] for the line digraph, by Zamfirescu [42] for the middle digraph, and by Skowronska, Syslo and Zamfirescu [36] for the total digraph. In addition, a lot of research has been done studying these transformations in various contexts [1-44].

Using intersections of sets belonging to a family of sets, in order to define the edge connections in a graph is so natural that it arose independently in a number of areas in connection with both pure and applied mathematics, and has been studied for over 7 decades. Let U be a set, and $\mathcal{F} = \{F_i\}_i$ a finite family of non-empty subsets of U. The intersection graph $\Omega(\mathcal{F})$ is the graph with the vertex set $\mathcal{F}$ in which $\{F_i, F_j\}$ is an edge if and only if the intersection of the sets F_i and F_j is non-empty. At the same time, if $G = \Omega(\mathcal{F})$ then $\mathcal{F}$ is called a *set representation* of the graph G. As far as we know, the first person to formulate this definition in such a broad fashion, without restricting the nature of either the set U or of the family $\mathcal{F}$ appears to have been Marczewski [30] in 1945. He also established that every graph is the intersection graph of some family of subsets of a finite set.

A lot of research has been done on various concepts that represent certain types of intersection graphs. Among these is the interval graph, $\Omega(\mathcal{F})$, where $U = \Re$, the real line, and each set F_i in $\mathcal{F}$ is an interval; certain interval graphs with various sorts of restrictions, such as unit-interval graphs, and multiple interval graphs; n-dimensional interval graph; circular-arc graph, etc. The monograph written by Mc Kee and Mc Morris [31] on Intersection Graph Theory is an excellent resource, as well as a good reference for most notations used in this paper. For other ones, not defined here, please use Harary's Graph Theory [22].

On the other hand, the study of similar concepts for digraphs has just started. Beineke and Zamfirescu [4] and Sen, Das, Roy and West [35] introduced and studied in different contexts a natural analogue of the intersection graph model for digraphs. Beineke and Zamfirescu [4] made for the first time a connection between these new intersection digraphs and transformations on digraphs.

[1]Support for Christina Zamfirescu's work was provided by PSC-CUNY Awards, jointly funded by The Professional Staff Congress and The City University of New York.

Definitions

A digraph $D = (V, A)$ has V as vertex set, and A as arc set. We may also use the notations $V(D)$ and $A(D)$, to denote V and A, respectively. Note that, unless otherwise specified, from now on D may have loops but no multiple arcs, D is weakly connected, and has at least two points.

Let's consider a family of ordered pairs of subsets of a set U, and to each ordered pair let's assign a vertex $v \in V$. Let S_v (source set) be the first set in the ordered pair assigned to v, and T_v (terminal set) be the second one. The *intersection digraph* of this family of ordered pairs of sets, $\mathcal{F} = \{(S_v, T_v)\}_{v \in V}$, is the digraph D that has V as vertex set and $uv \in A$ iff $S_u \cap T_v \neq \varnothing$.

In [4], and [35], it was shown that every digraph is the intersection digraph of ordered pairs of subsets of some set U. In [44] and [43], it was shown that the line, middle, total, and subdivision digraph of a digraph D can all be generated as intersection digraphs of ordered pairs of subsets of a universal set of symbols U, which contains only vertices and arcs of the digraph D, the digraph to be transformed.

This type of intersection digraph representation, using only elements of the transformed digraph, could make possible a unique computer treatment of all these transformations of the same digraph.

Let the *intersection number*, $i\#(D)$, of a digraph D be the minimum size of a set U, such that D is the intersection digraph of ordered pairs of subsets of U. We are raising here the problem of expressing the intersection number of a transformed digraph as a function of the size of the vertex set or arc set of the original digraph that was transformed, and we solve this problem for most transformation digraphs we mentioned here.

The transformations on digraphs we consider in this paper are all based on the concept of *directed adjacency*, which throughout this paper will simply be called adjacency. This *adjacency* can be between two points (x is adjacent to y, iff xy is an arc), two arcs (α is adjacent to β, iff the ending point of the first arc is the starting point of the second, e.g. $\alpha = xy$ and $\beta = yz$), and one of each (x is adjacent to any arc $\alpha = xz$ having x as starting point, and any arc $\alpha = xz$ is adjacent to its ending point, in this case z). Furthermore, x is called a *source* (sink), iff there are no points adjacent to (from) x, and x is called a *carrier* iff it is adjacent both to exactly one other vertex, and from exactly one other vertex.

The transformations of the digraph D express adjacencies within D in various ways: The line digraph reflects the adjacencies among the arcs in D, the original digraph. The total digraph reflects the adjacencies between all elements of the original digraph: between vertices, between arcs, and between vertices and arcs (meaning one of each). The middle digraph reflects the adjacencies in D between arcs, between vertices and arcs, but not the adjacencies between vertices. The well-known subdivision digraph reflects only the adjacencies in D, that exist between vertices and arcs.

Next we will define these 4 transformations for a digraph $D = (V, A)$, and mention theorems given in [44], that generate these transformations as intersection digraphs, using $U = V \cup A$, which means that the universal set U, of the intersection digraph consists only of elements of D.

The *line digraph*, denoted $\mathcal{L}(D)$, of the digraph D has as vertex set A, the arc set of D, and there is an arc in $\mathcal{L}(D)$ from one vertex $\widehat{uv}$ [NB: $\widehat{uv}$ will denote the vertex in $\mathcal{L}(D)$, that represents the arc uv in D] to another vertex $\widehat{wz}$ iff $v \equiv w$, i.e. the adjacency of the arcs in D is preserved for the corresponding vertices in $\mathcal{L}(D)$.

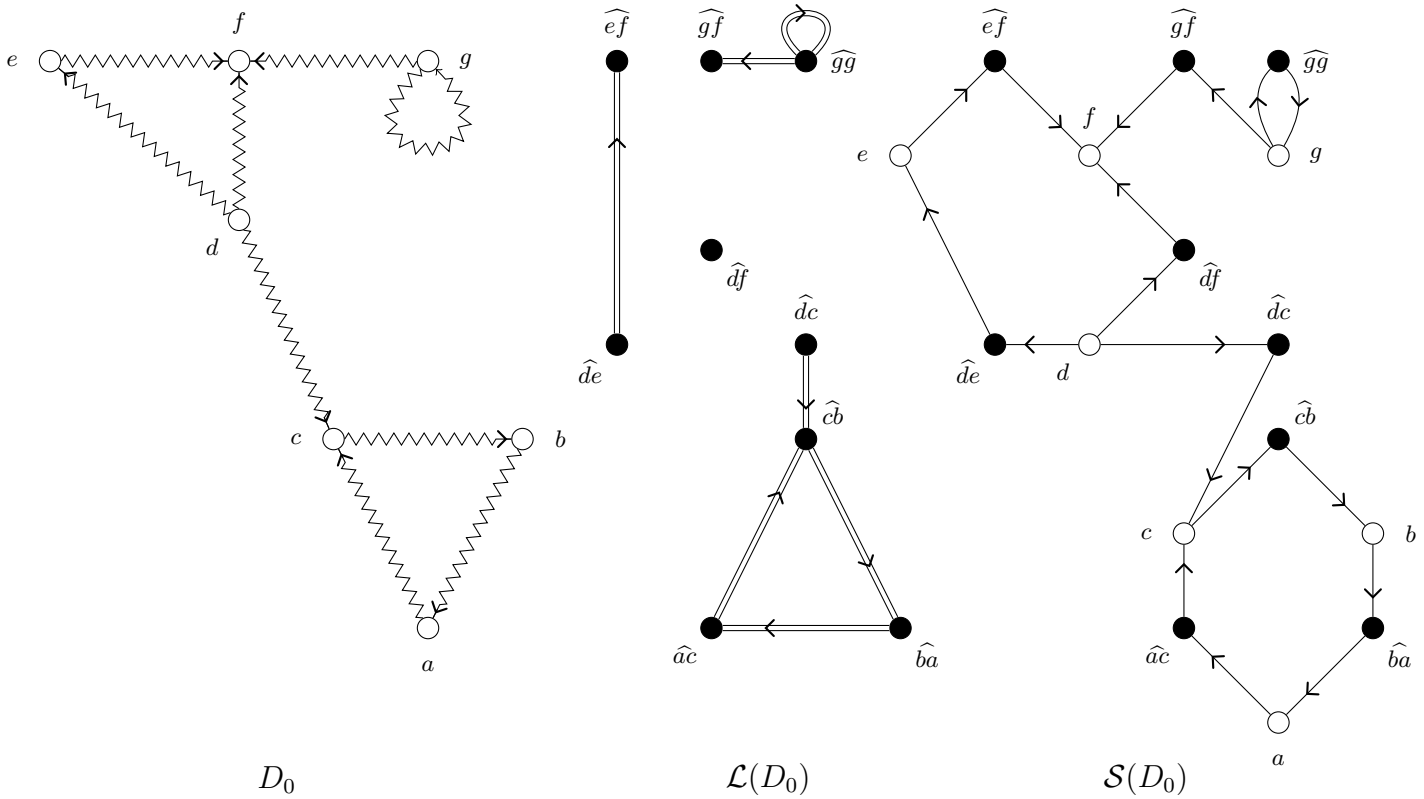
The *total digraph*, denoted $\mathcal{T}(D)$, of the digraph D has as vertex set $V \cup A$, and two such elements are connected by an arc in $\mathcal{T}(D)$ iff the corresponding elements in D are adjacent in D.

The *middle digraph*, denoted $\mathcal{M}(D)$, of the digraph D, has as vertex set $V \cup A$, and two such vertices in $\mathcal{M}(D)$ are connected by an arc in $\mathcal{M}(D)$ iff they are not both vertices in D, and the corresponding elements in D are adjacent in D.

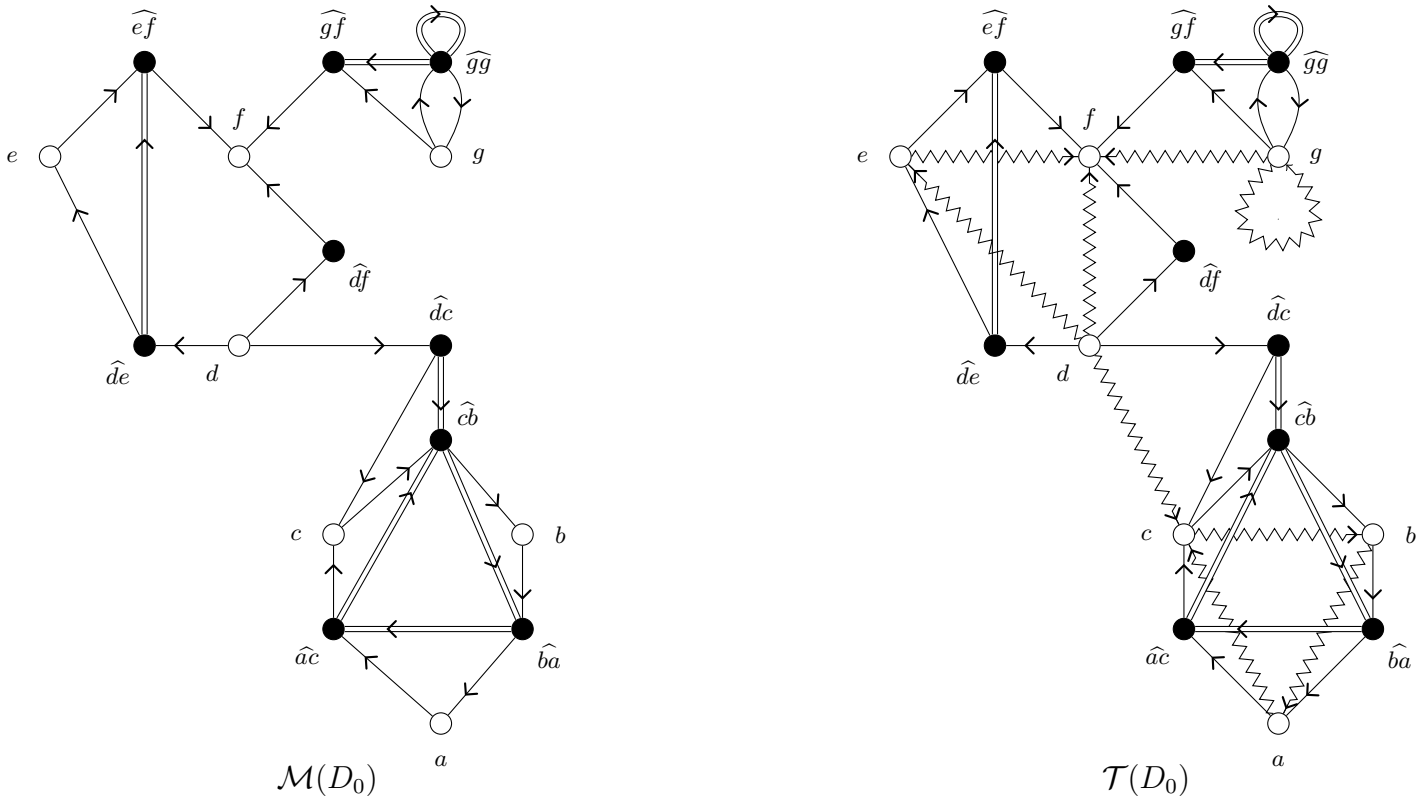
The *subdivision digraph*, denoted $\mathcal{S}(D)$, of the digraph D, has as vertex set $V \cup A$, and two such elements are connected by an arc in $\mathcal{S}(D)$ iff one of them is an arc and the other one a vertex of D, and they are adjacent in D. This is equivalent to the more common definition of a subdivision digraph, which defines it is as the digraph we obtain from D by attaching one extra point on each arc of D and thus subdivide each arc into two new arcs in $\mathcal{S}(D)$.

In the Figure below we exemplify all these transformations for a digraph D_0, with the vertex set $V(D_0) = \{a, b, c, d, e, f, g\}$, where the two types of vertices and the three types of arcs of the transformed digraphs are marked in such a way that they intuitively show their provenience: The empty (bold) points represent the vertices of the original digraph D_0, (respectively those vertices corresponding to arcs in D_0), while the wavy (double) [plain] arcs in any transformed or original digraph represent the adjacencies that exist between vertices in the original digraph D_0 (represent the adjacencies that exist between arcs in D_0) [represent the adjacencies that exist between vertices and arcs and arcs and vertices in D_0].

Figure



D_0 $\mathcal{L}(D_0)$ $\mathcal{S}(D_0)$



$\mathcal{M}(D_0)$ $\mathcal{T}(D_0)$

Results

Theorem A [44]
$\mathcal{L}(D)$, the line digraph of $D = (V, A)$, is the intersection digraph of the family $\mathcal{F}$ of ordered pairs of subsets of the universal set $U = V$, defined by:
$\mathcal{F} = \{(S_{\widehat{uv}}, T_{\widehat{uv}})\}_{\widehat{uv} \in A(\mathcal{L}(D))}$, where $S_{\widehat{uv}} = \{v\}$, and $T_{\widehat{uv}} = \{u\}$.

Theorem B [44]
$\mathcal{T}(D)$, the total digraph of $D = (V, A)$, is the intersection digraph of the family $\mathcal{F} = \{(S_\varepsilon, T_\varepsilon)\}_{\varepsilon \in U}$, of ordered pairs of subsets of the universal set $U = V \cup A$, defined by:
$S_\varepsilon = \{\varepsilon\} \bigcup \{\widehat{\varepsilon u} : \widehat{\varepsilon u} \in A\}$ and $T_\varepsilon = \{\widehat{u\varepsilon} : \widehat{u\varepsilon} \in A\}$, for all $\varepsilon \in V$,
$S_\varepsilon = S_{\widehat{uv}} = \{\widehat{uv}, v\}$ and $T_\varepsilon = T_{\widehat{uv}} = \{u\}$, for all $\varepsilon \in A, \varepsilon = \widehat{uv}$

Theorem C [44]
$\mathcal{M}(D)$, the middle digraph of $D = (V, A)$, is the intersection digraph of the family $\mathcal{F} = \{(S_\varepsilon, T_\varepsilon)\}_{\varepsilon \in U}$, of ordered pairs of subsets of the universal set $U = V \cup A$, defined by:
$S_\varepsilon = \{\varepsilon\}$ and $T_\varepsilon = \{\widehat{u\varepsilon} : \widehat{u\varepsilon} \in A\}$, for all $\varepsilon \in V$,
$S_\varepsilon = S_{\widehat{uv}} = \{\widehat{uv}, v\}$ and $T_\varepsilon = T_{\widehat{uv}} = \{u\}$, for all $\varepsilon \in A$, $\varepsilon = \widehat{uv}$

Theorem D [44]
$\mathcal{S}(D)$, the subdivision digraph of $D = (V, A)$, is the intersection digraph of the family $\mathcal{F} = \{(S_\varepsilon, T_\varepsilon)\}_{\varepsilon \in U}$, of ordered pairs of subsets of the universal set $U = V \cup A$, defined by:
$S_\varepsilon = \{\varepsilon\}$ and $T_\varepsilon = \{\mu \in U : \varepsilon$ adjacent to μ, and exactly one of ε, μ is an arc$\}$

Next, we will aim at minimizing the number of symbols in the universal set, U.
The *intersection number*, i#(G), of an undirected graph G is the minimum size of a set U, such that G is the intersection graph of subsets of U. For the undirected case, Erdös, Goodman, and Posa [12] showed that the intersection number of G equals the minimum number of complete subgraphs needed to cover its edges. Sen, Das, Roy, and West [35] proved an analogous result for digraphs. They defined the *generalized complete bipartite subdigraph* (abbreviated GBS) to be the subdigraph generated by vertex sets X, Y, the arcs of which are all xy such that $x \in X$, and $y \in Y$. Note that X and Y need not be disjoint (this is how loops are covered) which justifies the "generalized" term. If K is a GBS we shall call $X(K)$ and $Y(K)$ its X, respectively Y, sets. They gave the following:

Theorem E ([35]). The intersection number of a digraph equals the minimum number of GBSs required to cover its arcs.

We shall further give results that will express the intersection numbers of transformation digraphs of a digraph D, as functions of the numbers of vertices of D, that are sinks, sources or not sinks or not sources.
We will study the case of the line digraph separately, as it has an additional property: it satisfies the *Heuchenne Condition*, abbreviated here as H condition.
We say that a digraph fulfills the *H condition* iff for every four of its vertices, call them u, v, w and z, not necessarily distinct, the existence of the arcs uv, wv, wz implies the existence of the arc uz.

Theorem F ([25]). A digraph is a line digraph iff the H condition is fulfilled.

Let D satisfy the H condition, and let $\mathcal{C} = \{K_\sigma\}_{1 \leq \sigma \leq i\#(D)}$ be a set of minimum size, of GBSs that cover all arcs in D.
In D, let uv be an arc in some $K_\sigma \in \mathcal{C}$. It is easy to see that, given the H condition, all arcs adjacent from u, and all arcs adjacent to v in D, must also belong to K_σ since $\mathcal{C}$ is of minimum size. We can now define an equivalence relation R on the arc set $A(D)$ by stating that two arcs are related iff one of the following is fulfilled: (a) they have the same starting point; (b) they have the same ending point; (c) there is an arc in $A(D)$ from the starting point of one arc to the ending point of the other. It is easy to see that the set of GBSs induced by the equivalence classes generated by R is of minimum size, and we proved the following lemma.

Lemma 1 If H condition holds then $\mathcal{C}$ is uniquely determined in D.

Next, we can see that, if we apply Lemma 1 to $\mathcal{L}(D)$, which by Theorem F fulfills condition H, then each GBS, $K_\sigma \in C$, induced in $\mathcal{L}(D)$ by the relation R defined above, corresponds to exactly one vertex in D. That vertex in D is 1) adjacent from all arcs of D that correspond to the vertices in $X(K_\sigma)$, which means that it is not a source, and 2) adjacent to all arcs of D that correspond to the vertices in $Y(K_\sigma)$, which means that it is not a sink. Since K_σ contains at least one arc, that vertex in D must be neither a source nor a sink. This proves the next lemma and theorem.

Lemma 2 There is a one-to-one correspondence between the set $\mathcal{C}$ of GBSs and the set of all vertices in D, that are neither sources nor sinks.

Theorem 1 $i\#(\mathcal{L}(D))$, the intersection number of $\mathcal{L}(D)$ equals the number of vertices of D that are neither sources nor sinks.

Let's consider now the subdivision digraph, $\mathcal{S}(D)$, of the digraph D. It is easy to see that, since in $\mathcal{S}(D)$ in every semipath (NB: walk in the graph without following the directions of the arcs, see [22]), every second vertex is a carrier, $\mathcal{S}(D)$ satisfies the H condition. From Lemma 1 we know that $\mathcal{S}(D)$ has a unique minimum set of GBSs that cover all its arcs, and each such GBS is induced by the arc set of one of the equivalence classes generated by the equivalence relation R, defined for Lemma 1. In fact, the point (c) in the definition of R cannot occur in $\mathcal{S}(D)$, and thus each GBS in $\mathcal{S}(D)$ is a star (see [22]), which (a) has a source as the center, and any remaining vertex is a sink, or (b) has a sink as the center, and any remaining vertex is a source. We can attach these GBSs to only those vertices in $\mathcal{S}(D)$, that correspond to vertices in D. To each source (sink) will correspond exactly one GBS, consisting in a star with n arms, where n is the out-degree (in-degree) of the source (sink) in D. To each of the other vertices, we will attach exactly two GBSs, one for the in-coming arcs, and the other for the out-coming arcs. We thus proved:

Lemma 3 $i\#(\mathcal{S}(D))$ equals the number of vertices of D that are not sources, added to the number of vertices of D that are not sinks.

From now on, let's consider that D contains no loops.

Neither $\mathcal{M}(D)$ nor $\mathcal{T}(D)$ satisfies the H condition, generally.

We will show next that, in the case of both $\mathcal{M}(D)$ and $\mathcal{T}(D)$, although the covering of the arcs by a set of GBSs of minimum size may not be unique, their intersection numbers are equal to the intersection number of $\mathcal{S}(D)$. We will do this by extending the GBSs we formed for the $\mathcal{S}(D)$ to also cover all the arcs that are in $\mathcal{M}(D)$ or $\mathcal{T}(D)$ but not in $\mathcal{S}(D)$, by allocating each such arc, say xy, new to $\mathcal{S}(D)$, to the GBS that contains all arcs in $\mathcal{S}(D)$ adjacent to y. Similarly, we could allocate xy to x, instead of to y, thus defining a generally different set of GBSs, that cover all arcs in $\mathcal{M}(D)$ or $\mathcal{T}(D)$.

In order to prove that this new set of GBSs is of minimum size it is enough to show that we cannot construct a GBS in $\mathcal{M}(D)$ or $\mathcal{T}(D)$ that contains two arcs α and β, that belong to two different GBSs in $\mathcal{S}(D)$. Any arc in $\mathcal{S}(D)$ joins a vertex that represents a vertex in D with (i.e. to or from) a vertex that represents an arc in D. The latter must also be a carrier in $\mathcal{S}(D)$. If α and β have a common endpoint, then this can only represent a vertex in D, as it is not a carrier. In this case they must be in the same GBS in $\mathcal{S}(D)$. If the starting point of α is the same point as the ending of β, then by the definition of the GBS, we would need to have a loop at that point, which is not allowed in $\mathcal{S}(D)$, even if D had loops. If α and β do not have a common endpoint, say α is the arc xz and β is the arc yt, with all endpoints distinct, then the GBS must also contain the arcs xt and yz. Let's assume, without loss of generality, that x and t represent vertices, while y and z represent arcs in the original D, that we transformed. In addition, note that the arc in D represented by y must be adjacent to the arc in D represented by z. A contradiction follows from the fact that y and z must both be carriers in $\mathcal{S}(D)$, and D may not contain a loop. We therefore proved the following results.

Lemma 4 No GBS in $\mathcal{M}(D)$ or $\mathcal{T}(D)$ may contain two arcs that belong to two different GBSs in $\mathcal{S}(D)$.

Theorem 2 If D contains no loops, $i\#(\mathcal{T}(D)) = i\#(\mathcal{M}(D)) = i\#(\mathcal{S}(D))$, that is the intersection numbers of $\mathcal{T}(D)$, $\mathcal{M}(D)$ and $\mathcal{S}(D)$ are all equal to the number of vertices of D that are not sources, added to the number of vertices of D that are not sinks.

We would like to note here, that Lemma 4 is no longer true when D has loops, as the number of GBSs covering all arcs in $\mathcal{T}(D)$ might be reduced from the one covering $\mathcal{S}(D)$. For instance, the subgraph induced by the vertex set $\{g, \widehat{gg}, \widehat{gf}\}$ in $\mathcal{T}(D_0)$ in our Figure forms one GBS, while in $\mathcal{S}(D)$ and $\mathcal{M}(D)$ the same subgraph must be covered by two GBSs, due to the lack of the loop at the vertex g in $\mathcal{S}(D)$ and $\mathcal{M}(D)$.

The problem of finding equivalent results for other transformations of digraphs, such as various power digraphs, remains also open.

References

[1] J. Akiyama, T. Hamada and I. Yoshimura: On characterizations of the middle graphs, TRU Math. 11 (1975) 35-39.

[2] M. Behzad: A criterium for the planarity of the total graph, Proc. Cambridge Philos. Soc. 63 (1967), 697-681.

[3] L. W. Beineke: Characterization of derived graphs, J. Comb. Theory B 9 (1970) 129-135.

[4] L.W. Beineke, C. M. Zamfirescu: Connection digraphs and second order line digraphs, Discrete Math 39 (1982) 237-254.
[5] JC Bermond, JMS Simões-Pereira, C. Zamfirescu: On non-hamiltonian homogeneously traceable digraphs, Math. Japan 24(1979) 423-426.
[6] S. Bertz, W.F. Wright: The graph theory approach to synthetic analysis: definition and application of molecular complexity and synthetic complexity, Graph Theory Notes of New York XXXV (1998) 32-18.
[7] S. Bertz, C. Zamfirescu: New complexity indices based on edge covers, Match 42 (2000) 39-70.
[8] P. Buneman: A characterization of rigid circuit graphs, Disc. Math. 9 (1974) 205-212.
[9] G. Chartrand, M. J. Stewart: Total digraphs, Canadian Math. Bull 9 (1966) 171-176.
[10] Chikodimath, E. Sampathkumar: Semi-total graphs-II, Graph Theory Research Report, Karnath University No. 2 (1973) 5-9.
[11] J. E. Cohen: Food Webs and Niche Space, Princeton University Press, Princeton 1978.
[12] P. Erdös, A. Goodman, L. Posa: The representation of a graph by set intersections, Can. J. Math 18 (1966) 106-112.
[13] D.R. Fulkerson, O.A. Gross: Incidence matrices and interval graphs, Pacific J. Math. 15 (1965) 835-855.
[14] F. Gavril: The intersection graphs of subtrees in trees are exactly the chordal graphs, J. Comb. Theory B 16 (1974) 47-56.
[15] P.C. Gilmore, A.J. Hoffman: A characterization of comparability graphs and the interval graphs, Canadian J. Math 16 (1964) 539-548.
[16] J.R. Griggs, D.B. West: Extremal values of the interval number of a graph, SIAM J. Alg. Disc. Math. 1 (1979) 1-7.
[17] J.R. Griggs: Extremal values of the interval number of a graph, II, Disc. Math. 28 (1979) 37-47.
[18] H. Hadwiger, H. DeBrunner, V. Klee: Combinatorial Geometry in the Plane, Holt, Rinehart and Winston, New York, 1964.
[19] G. Hajos: ber eine Art von Graphen: Intern. Math Nachr. 2 (1957) 65.
[20] T. Hamada, I Yoshimura: Traversability and connectivity of the middle graph of a graph, Discr. Math 14 (1976) 247-255.
[21] P. Hanlon: Counting interval graphs, J. London Math Soc (1979)
[22] F. Harary: Graph Theory, Addison-Wesley Series in Mathematics, Addison-Wesley, 1969
[23] F. Harary, J.A. Kabell: Infinite-interval graphs, Calcutta Mathematical Society. Diamond- cum-Platinum Jubilee Commemoration Volume (1908-1983) Part I, Calcutta Math. Soc., Calcutta (1984) 27-31.
[24] F. Harary, R.Z. Norman: Some properties of line-digraphs, Rend. Circ. Mat. Palermo 9 (1960) 161-168.
[25] C. Heuchenne: Sur une certaine correspondence entre graphs, Bull. Soc. Roy. Lige 33 (1964) 743-753.
[26] D.G. Kendall: Incidence matrices, interval graphs and seriation in archaeology, Pacific J. Math, 28 (1969) 565-570.
[27] V. Klee: What are the intersection graphs of arcs in a circle, Amer. Math. Monthly 76 (1969) 810-813.
[28] J. Krausz: Dmonstration nouvelle d'un thorme de Whitney sur les rseaux, Mat. Fiz. Lapok 50 (1943) 75-89.
[29] C.B. Lekkerkerker, J.C. Boland: Representation of a finite graph by a set of intervals on the real line, Fund. Math. 51 (1962) 45-64.
[30] E. Marczewski: Sur deux proprits des classes d'ensembles, Fund. Math 33 (1945) 303-307.
[31] T.A. McKee, F.R. Mc Morris: Topics in Intersection Graph Theory, SIAM Monographs on Discr. Math and Applications, Philadelphia, 1999.
[32] F.S. Roberts: On the boxicity and the cubicity of a graph, Recent Progress in Combinatorics (W.T. Tutte, ed.) Academic Press, New York 1969, 301-310.
[33] F.S. Roberts: Discrete Mathematical Models, Prentice Hall, Englewood Cliffs, NJ, 1976.
[34] Van Rooij, H.S. Wilf: The interchange graphs of a finite graph, Acta Math. Acad, Sci Hungar. 16 (1965) 263-269.
[35] M. Sen, S. Das, A.B. Roy, D.B. West: Interval digraphs - An analogue of interval graphs, J. Graph Th. 13 (1989) 189-202.
[36] M. Skowronska, M.M. Syslo, C. Zamfirescu: An Algorithmic Characterization of Total Digraphs, J. Algorithms 7 (1986) 120-133.
[37] F.W. Stahl: Circular genetic maps, J. Cell Physiology 70 Sup. 1 (1967) 1-12.
[38] K.E. Stoffers: Scheduling of traffic lights - a new approach, Transportation Research 2 (1968) 199-234.
[39] W.T. Trotter Jr., F. Harary: On double and multiple interval graphs, J. Graph Theory 3 (1979) 205-211.
[40] H. Whitney: Congruent graphs and the connectivity of graphs, Amer. J. Math. 54 (1932) 150-168.
[41] C. Zamfirescu: Cyclic and cliquewise connectedness of line graphs, Discrete Math 170 (1997) 293-297.
[42] C.M. Zamfirescu: Local and global characterizations of middle digraphs, Theory and Applications of Graphs (G. Chartrand, ed.) J. Wiley, New York (1981) 595-607.
[43] C.M. Zamfirescu, E. and D. Celenti: Transformation of Digraphs and Intersection Digraphs: Connections and Results, Graph Theory Notes 43 (2002) 39-47.
[44] C.M. Zamfirescu, E.S. Klein: Transformed digraphs represented as intersection digraphs - a uniform way of expression and search for optimality, Congressus Numerantium 110 (1995) 137-144.