



HAL
open science

Generation and Analysis of Dynamic Graphs

Vincent Bridonneau

► **To cite this version:**

Vincent Bridonneau. Generation and Analysis of Dynamic Graphs. Computer Science [cs]. Normandie Université, 2024. English. NNT : 2024NORMLH23 . tel-04888896

HAL Id: tel-04888896

<https://theses.hal.science/tel-04888896v1>

Submitted on 15 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

Pour obtenir le diplôme de doctorat

Spécialité **INFORMATIQUE**

Préparée au sein de l'**Université Le Havre Normandie**

Generation and Analysis of Dynamic Graphs

Présentée et soutenue par

VINCENT BRIDONNEAU

Thèse soutenue le 03/12/2024

devant le jury composé de :

M. FREDERIC GUINAND	PROFESSEUR DES UNIVERSITES - Université Le Havre Normandie (ULHN)	Directeur de thèse
M. BINH-MINH BUI-XUAN	CHARGE DE RECHERCHE (HDR) - Sorbonne Université	Président du jury
M. YOANN PIGNE	PROFESSEUR DES UNIVERSITES - Université Le Havre Normandie (ULHN)	Co-directeur de thèse
MME SABRINA GAITO	PROFESSEUR - UNIVERSITE DE MILAN	Membre du jury
M. ROBERTO INTERDONATO	CHARGE DE RECHERCHE (HDR) - CIRAD MONTPELLIER	Rapporteur du jury

Thèse dirigée par **FREDERIC GUINAND** (LABORATOIRE D'INFORMATIQUE DE TRAITEMENT DE L'INFORMATION ET DES SYSTEMES) et **YOANN PIGNE** (Université Le Havre Normandie)



Remerciements

Tout ce travail et cette thèse n'auraient jamais pu voir le jour sans le soutien de nombreuses personnes qui ont cru en moi et m'ont permis d'arriver jusqu'ici. Je tiens particulièrement à remercier Frédéric Guinand et Yoann Pigné, qui m'ont offert cette belle opportunité de travailler sur un sujet aussi riche et profond. Nos échanges ont souvent été intenses, avec des visions parfois opposées sur la manière d'étudier des problèmes et d'en concevoir de nouveaux. Mais je pense que nous avons tous les trois pu en tirer des enseignements qui nous permettent aujourd'hui de comprendre et de voir les choses avec un regard différent.

Je tiens particulièrement à remercier mes parents, Lydia et Éric Prola, ainsi qu'Olivier Bridonneau et MC, qui ont été d'un soutien indéfectible et ont cru en mon projet, même lorsque je doutais sérieusement de moi. Ils ont toujours su trouver les mots pour me rassurer et me motiver. Sans eux, cette thèse ne serait pas ce qu'elle est. Je remercie également mes chers frères et ma chère sœur, qui occupent une place toute particulière dans mon cœur. Je serai toujours là pour eux.

Je tiens aussi à remercier ce bon vieux Jason Schoeters, le J, avec qui j'ai passé des moments inoubliables : un très bon ami et un très bon vivant, par la même occasion. Un grand merci également aux Pils de Fut, avec qui j'ai passé d'excellents moments. La distance n'a pas eu raison de notre amitié, et je suis sûr que nous continuerons à bien rigoler autour d'une bonne bière !

Je tiens également à remercier chaleureusement Monsieur Stéphane Mouez, mon professeur de mathématiques en seconde, qui a cru en moi, m'a donné le goût des mathématiques et a fait naître en moi l'envie de faire de la recherche. Je remercie aussi Monsieur Olivier Rivière, qui m'a transmis la rigueur mathématique et a renforcé ma volonté d'étudier des problèmes et de les définir de façon claire et précise. Je suis reconnaissant envers l'équipe de Math.en.Jeans, qui m'a permis de découvrir dès le lycée le monde de la recherche et ce à quoi ressemblent les problèmes que l'on peut étudier. Bien que simples en apparence, ils peuvent renfermer de nombreux trésors, notamment dans les moyens que l'on met en œuvre pour les résoudre.

Un grand merci également à Mathieu Faverge et Pierre Ramet, qui m'ont accueilli dans leur laboratoire et avec qui j'ai pu contribuer pour la première fois à un projet de recherche, à ma petite échelle.

Enfin, j'aimerais remercier tous mes collègues du LITIS du Havre, avec qui j'ai pu créer de belles amitiés et partager d'agréables moments.

Abstract

In this thesis, we investigate iterative processes producing a flow of graphs. These processes find applications both in complex networks and time-varying graphs. Starting from an initial configuration called a seed, these processes produce a continuous flow of graphs. A key question arises when these processes impose no constraints on the size of the generated graphs: under what conditions can we ensure that the graphs do not become empty? And how can we account for the changes between successive steps of the process? To address the first question, we introduced the concept of sustainability, which verifies whether an iterative process is likely to produce graphs with periodic behaviors. We defined and studied a graph generator that highlights the many challenges encountered when exploring this notion. Regarding the second question, we designed a metric to quantify the changes occurring between two consecutive steps of the process. This metric was tested on various generators as well as on real-world data, demonstrating its ability to capture the dynamics of a network, whether artificial or real. The study of these two concepts has opened the door to many new questions and strengthened the connections between complex network analysis and temporal graph theory.

Résumé en français :

La nature et les sociétés humaines offrent de nombreux exemples de systèmes composés d'entités qui interagissent, communiquent ou sont simplement connectées les unes aux autres. La théorie des graphes offre un excellent formalisme pour modéliser ces systèmes complexes, allant des réseaux sociaux aux systèmes biologiques. La plupart des phénomènes observés dans ces réseaux peuvent s'exprimer sous forme de propriétés sur les graphes. On peut notamment citer le phénomène du « petit monde » ou les réseaux dits « sans échelle ». Comprendre les mécanismes sous-jacents à leur évolution est essentiel pour saisir les dynamiques de ces réseaux. Différents mécanismes existent pour reproduire les propriétés observées. Parmi eux, on peut citer l'attachement préférentiel, utilisé notamment par le modèle de Barabási-Albert (BA), qui permet de produire des séquences de graphes croissants sans échelle.

Dans une direction parallèle, on peut également étendre le concept de graphe en y ajoutant une dimension temporelle. Dans ce cas, les propriétés statiques des graphes sont retravaillées pour tenir compte de l'évolution des graphes dans le temps. Par exemple, on peut citer la notion de *trajet* qui, semblable à celle de chemin, traduit la possibilité de se déplacer d'un sommet à un autre en respectant des contraintes temporelles. De même que dans le cas des réseaux complexes, la capacité à générer des graphes temporels est étudiée afin de produire des graphes aux propriétés spécifiques. On peut par exemple évoquer le modèle Edge-Markovian Graph, un processus stochastique permettant de produire des graphes et d'étudier des problèmes de communication.

L'observation de ces mécanismes de génération donne naissance à la problématique de cette thèse, qui réside dans l'étude de processus itératifs de génération de graphes temporels. Lorsqu'un graphe est obtenu par itérations successives d'un tel mécanisme, on parle d'un graphe dynamique. Cette dénomination met en avant l'aspect itératif du processus pour produire une séquence ordonnée de graphes. Une question nous a particulièrement intéressés dans le cadre de ce travail : que se passe-t-il lorsqu'un générateur n'est soumis à aucune contrainte, notamment en ce qui concerne l'évolution du nombre de sommets au fil du temps ?

Cette situation soulève deux problématiques : la possibilité qu'un processus conduise à des graphes périodiques au-delà d'un certain moment et la quantification des changements entre deux étapes consécutives du processus. Pour répondre à ces interrogations, nous avons introduit deux métriques. La première, que nous avons appelé *sustainability*, et que l'on peut traduire par *pérennité*, est une mesure qualitative : un générateur est dit *sustainable* s'il produit des graphes qui ne deviennent ni vides ni périodiques. La seconde métrique, le *DynamicScore*, quantifie les changements entre deux instants successifs, à la fois au niveau des sommets (*V-DynamicScore*) et des arêtes (*E-DynamicScore*).

Pour démontrer la pertinence de la notion de *pérennité*, nous avons défini et étudié un générateur de graphes mettant en évidence les nombreux défis rencontrés lors de l'exploration de cette notion. En ce qui concerne le *DynamicScore*, nous l'avons testé sur divers générateurs ainsi que sur des données réelles, démontrant sa capacité à capturer la dynamique d'un réseau, qu'il soit artificiel ou réel. L'étude de ces deux concepts a ouvert la voie à de nombreuses nouvelles questions et renforcé les liens entre l'analyse des réseaux complexes et la théorie des graphes temporels.

keywords: Dynamic Graphs, Dynamic Graph Metrics, Dynamic Graph Generation, Complex Networks, Dynamic Graph Analysis

mots clés : Graphes dynamiques, Métriques sur les graphes dynamiques, Génération de graphes dynamiques, Réseaux complexes, Analyse des graphes dynamiques

Contents

1	Introduction	1
2	Introducing Time in Graphs	5
2.1	Graph	6
2.2	Complex Networks Analysis	7
2.2.1	Complex Networks Properties	7
2.2.1.1	The Small-World Property	8
2.2.1.2	Scale-free Networks	9
	Hyperbolic Geometry	9
2.2.1.3	Strong Clustering Coefficient	10
2.2.1.4	Triadic Closure	10
2.2.1.5	Navigability	11
2.2.2	Generation of Complex Networks	11
2.2.2.1	The Watts-Strogatz Model	11
2.2.2.2	The Barabási and Albert Model	13
2.2.2.3	BA-based models	13
	Triadic Closure	14
	Popularity vs Similarity	14
	Conclusion	15
2.3	Time in Graphs	15
2.3.1	Formalism	16
2.3.2	Problems and Properties in Graphs with Time	17
2.3.2.1	Journey	17
2.3.2.2	Connectivity	19
	Classification	20
2.3.3	Generation of Time-Varying Graphs	21
2.3.3.1	Generator of Edge-Markovian Graphs	21

2.3.3.2	Simple Time-Varying Graphs Generation	22
2.3.3.3	Generating Graph with a Given Feature	22
2.4	Discussion on Time in Graphs	23
2.5	Formalizing the Concept of Dynamic Graphs	24
3	Qualifying and Quantifying Graph Dynamics	27
3.1	Sustainability	28
3.1.1	Definition	28
3.1.2	Comparison to Existing Generative Processes	29
3.1.2.1	Barabási-Albert Model	29
3.1.2.2	Edge Markovian Graphs	29
Known Properties of EMGG	29	
Density Evolution of Edge-Markovian Graphs	30	
Sustainability	31	
3.2	DynamicScore	32
3.2.1	Definition	32
3.2.2	Specific Values of DynamicScore	34
3.2.2.1	Constant Graph Order	34
3.2.2.2	Growing Network	35
Linear Grow	35	
Exponential Growth	36	
Graph Densification	36	
3.2.3	Comparison to Generator Models	37
3.2.3.1	Sequence of Erdos-Renyi Graphs	37
3.2.3.2	Edge-Markovian Graphs	39
3.2.3.3	Barabasi-Albert Generating Graphs	40
3.3	Information Persistence Problem	42
3.3.1	Problem Formulation	43
3.3.1.1	Communication Strategy	43
Constant Flooding	43	
Simple Flooding	44	
3.3.2	Studied Problems	44
3.3.3	Remarks and First Results	45
3.3.3.1	Time Complexity of the Spreading Algorithm	45
3.3.4	Questions and Open Problems	46
3.3.4.1	Questions Related to the Simple Flooding Algorithm	46
3.3.4.2	Connection between Simple Flooding and Constant Flooding	47
3.3.4.3	Sustainability and Information Persistence Problem	47

4	Illustrative Case Study of Dynamic Graph Generators Analysis	51
4.1	Definitions and Generative Model and Definitions	51
4.2	Theoretical Analysis	54
4.2.1	First Results on Sustainability	54
4.2.2	Limit Case Analysis	54
4.2.2.1	Case: $S_S = \mathbb{N}$ and $S_C = \mathbb{N}$	55
4.2.2.2	Case: $S_S = \mathbb{N}$ and $S_C = \emptyset$	55
4.2.2.3	Case: $S_S = \emptyset$ and $S_C = \mathbb{N}$	55
4.2.2.4	Case: S_S is a non-empty finite set and $S_C = \emptyset$	55
4.2.2.5	Case: $S_S = \emptyset$ and S_C is a non-empty finite set	56
4.2.2.6	Graph Order Increase	56
4.2.2.7	Case: $S_S = \mathbb{N}$ and S_C is a non-empty finite set	57
4.2.2.8	Case: S_S is a non-empty finite set and $S_C = \mathbb{N}$	58
4.2.2.9	Summary of results	58
4.2.3	General Cases	58
4.2.3.1	Partition sets	60
4.2.4	The Redistributed Model	60
4.3	Segments	61
4.3.1	$S = \{0\}$	62
4.3.1.1	Sustainability	62
4.3.1.2	Relationship with the Information Persistence Problem	65
4.3.1.3	Study of Graph Evolution	65
4.3.1.4	Graph Evolution and Sustainability	69
	One fixed point	70
	Two fixed points	70
	Three fixed points	72
	Building the Sustainable Interval	74
4.3.2	Vertex DynamicScore	74
4.3.3	Conjecture on the Sustainability	75
4.3.4	The Purpose of the Non-redistributed Model	77
4.4	Infinite Sets and Asymptotic Graph Order Evolution	80
4.4.1	Intermediate Result	80
4.4.2	The Equivalent and First Interpretation	83
4.4.3	Generated Graphs Interpretation	83
4.4.3.1	Exponential Increasing	84
4.4.3.2	Exponential Decreasing	84
4.4.3.3	Quasi Constant Evolution	84
4.4.3.4	Sustainability of Small Generated Graphs	86

5	Analysis of Real World Networks	89
5.1	Transforming Events into Graphs	90
5.1.1	Events as Instantaneous Contacts	90
5.1.1.1	Time Interval	91
5.1.1.2	Sliding Window	92
5.1.1.3	First-to-last Vertices	92
5.1.1.4	Growing Model	93
5.1.1.5	Remarks	94
5.1.2	Events as Permanent Contacts	95
5.1.2.1	Filtering	95
5.2	Illustrative Case Study	96
5.2.1	Instantaneous Contacts	96
5.2.1.1	Email Network	96
	Remarks about the study of Leskivec	97
	Growing Network Transformation	98
	Time Interval Transformation	98
	Sliding Window Transformation	99
	First-to-last Vertices and Edges Transformation	99
5.2.1.2	Autonomous System	101
	Growing Network Transformation	101
	Time Interval Transformation	102
	Sliding Window Transformation	103
	First-to-last Transformation	103
5.2.2	Permanent Contacts	105
5.2.2.1	Data and Model	105
5.2.2.2	Objectives	106
5.2.2.3	Analysis	107
5.2.2.4	Improving Filtering Methods	109
	The Ellipse Model	109
	Filtering Branches	110
6	Conclusion	113

List of Figures

2.1	Representation of a graph with six vertices and five edges.	6
2.2	The degree distribution of a BA graph obtained with $n = 100000$ final vertices, $m_0 = 6$ initial vertices and $m = 5$ new connections every step.	14
2.3	Representation of a dynamic graph $\mathcal{G} = (G_1, G_2, G_3)$	17
2.4	Representation of a journey between vertices A and B in the dynamic graph $\mathcal{G} = (G_1, G_2, G_3)$	18
2.5	Representation of a dynamic graph $\mathcal{G} = (G_1, G_2, G_3)$ with traversal times represented by labels on edges.	19
2.6	Representation of a time connected dynamic graph $\mathcal{G} = (G_1, G_2)$	20
2.7	Representation of a round connected dynamic graph $\mathcal{G} = (G_1, G_2, G_3)$	20
2.8	Description of the states in the Edge-Markovian Graphs Generator.	21
3.1	The two different configurations where $ p + q - 1 = 1$. On the left, $p = q = 1$. This means whatever the initial configuration, the edge will stay in the same state. On the right, $p = q = 0$. This means the edge state changes at every step.	30
3.2	Illustration of the dynamic score for vertices. Here n_{t+1} is assumed to equal n_t . For the case $\mathcal{D}_t^V = \frac{2}{3}$, $-$ refers to as $V_t - V_{t+1}$, the set of non-conserved vertices. \cap refers to as $V_t \cap V_{t+1}$, the set of conserved vertices. Finally, $+$ refers to as $V_{t+1} - V_t$, the set of created vertices.	34
3.3	Average dynamic score as a function of the parameters p and q . On the left, the parameter p is set and the parameter q ranges from 0.05 to 0.95. One may notice that for a fixed value of parameter p , the average dynamicScore does not depend on q . On the right, the parameter q is set and the parameter p ranges from 0.05 to 0.95. As observed with the picture on the left, the average dynamicScore does not depend on q so all the marks are mingled.	41

4.1	Leaves of the tree represent the general cases. Rounded corners green boxes corresponds to cases for which results are presented in this Section, in Section 4.3 and in Section 4.4. Dashed boxes are cases not covered within this report.	59
4.2	Relationship between the average value of c_t and the expected value. Each point correspond to a single threshold d . d is ranging from 0.001 to 0.01 with a step of 0.0005 and from 0.01 to 0.2 with a step of 0.005	66
4.3	One fixed point	71
4.4	Two fixed points	72
4.5	Three fixed points	73
4.6	Mean value of vertex DynamicScore got from experimentation. Points represent the average over 20 run and 30000 time steps for a single m and M . The yellow surface is the plan of equation $z = \frac{2}{3}$. For all these parameters, d is set to 0.05. Red points represent DynamicScore greater than $\frac{2}{3}$. Blue points represent DynamicScore lower than $\frac{2}{3}$	76
4.7	Success rate of simulations for m ranging from 1 to 8. Each curve represent a different value of m . The x-axis is a value of M and the y-axis is the success rate.	77
4.8	Example of the evolution of a dynamic graph produced by D3G3. Parameters are $S = [13, 31]$ and $d = 0.05$	78
4.9	Scenario of exponential increase. $A = [0, 5], s = 7, n_0 = 375$. The theoretical value is given by the formula $n_t \approx \left(\frac{2 A }{s}\right)^t n_0$	84
4.10	Scenario of exponential decrease. $A = [4, 5], s = 5, n_0 = 7523$. Theoretical values are given by the formula $n_t \approx \left(\frac{2 A }{s}\right)^t n_0$	85
4.11	Simulation performed considering $s = 4, A = [0, 1]$ and $d = 0.05$. The number of steps is 5000 and the initial seed graph is a random geometric graph of order 2000.	86
4.12	Theoretical graphical representation of $f_{S,d}$ for value of n from 0 to 400. The blue curve correspond to $A = [0, 7]$ and the red one correspond to $A = [8, 15]$	87
5.1	Illustration of a sequence of event. Here, $T = 8$ and only three vertices are interacting in the whole sequence	91
5.2	Illustration of a transformation from a sequence of event to a time-varying graph. Here, $T = 8$ and $\tau = 3$. As u does not interact with another vertex, it is not present in G_1	91
5.3	Illustration of a transformation from a sequence of event to a time-varying graph. Here, $T = 8, W = 3$ and $\tau = 1$	92
5.4	Illustration of a transformation from a sequence of events to a time-varying graph using the F2LVE transformation. Here, $T = 8$ and $\tau = 3$. Dashed lines indicate edges part of the resulting graph but which are not occurring at that date.	93
5.5	Illustration of a transformation from a sequence of event to a dynamic graph. Here, $T = 8$ and $\tau = 3$	93
5.6	DynamicScores, n_t and e_t graph order evolution for the Growing Network transformation for the E-mail network.	98
5.7	DynamicScores and graph order evolution for the Time Interval transformation.	99

5.8	DynamicScores and graph order evolution for the Sliding Window transformation.	99
5.9	DynamicScores and graph order evolution for the First-to-last transformation.	100
5.10	Comparison between the number of vertices and the number of edges of each graph. Each blue point represents a measured number of vertices and number of edges computed from the data analysis. The dashed black line represents the theoretical expectation.	102
5.11	DynamicScores, n_t and e_t evolution for the Growing Network transformation for the Autonomous System network.	102
5.12	DynamicScores, n_t and e_t evolution for the Time Interval transformation for the Autonomous System network.	103
5.13	DynamicScores, n_t and e_t evolution for the Sliding Window transformation for the Autonomous System network.	104
5.14	DynamicScores, n_t and e_t evolution for the FTLVE transformation for the Autonomous System network.	104
5.15	A short portion of the mycelium of <i>Podospora anserina</i> , and its graph model. Blue, degree-one, nodes are the apexes, yellow, degree-three, nodes are branching nodes and red, degree-two, nodes are structural nodes, distributed along the hyphae. (Courtesy of Thibault Chassereau)	105
5.16	Evolution of both \mathcal{D}_t^V and \mathcal{D}_t^E for the mycelium until the end of the experiment.	108
5.17	Comparison between the evolution of n_t and e_t for the whole growing dynamic graph of PA.	109
5.18	Evolution of \mathcal{D}_t^V through time for different cut-off values of r . A cut-off value of 0% means that network is entirely contained in the ring. The inner disk is null.	109
5.19	The complete network after adding all the vertices and the edges. This picture represents only the edges.	110

List of Tables

3.1	Dynamic Score of both the edges and vertices for the Barabasi-Albert model. The considered parameters are $ V_0 = 2$, $m_0 = E_0 = 1$ and $m = 1$	41
4.1	Order and DynamicScore evolution for the different cases. n_t denotes the order of graph G_t , \mathcal{D}_t^V its vertices DynamicScore and \mathcal{D}_t^E the edges DynamicScore.	58
4.2	Sustainability of dynamic graphs according to parameters sets of D3G3.	58
4.3	Ratio of sustainability graphs for different thresholds d . Each run has a limit of 3500 iterations.	79

CHAPTER 1

Introduction

Nature and human societies offer many examples of systems composed of entities that interact, communicate or are just connected with each other. The Internet, a transportation network, a swarm of robots, an ant colony, a social network, an urban network, or a crowd are some examples [7].

Graphs are certainly one of the best formalisms for modeling them. Every vertex in the graph models one entity. A link is added between two vertices when a particular condition about the corresponding entities is verified. For instance: two persons are talking to each other, a predator catches a prey, a virus passes from one individual to another, two actors perform in the same play, etc. The semantic of the interaction, communication or connection is proper to the system. During last two decades, many works have been dedicated to the study of networks modeling these systems. It has been shown that, unlike classical, regular or random graphs, graphs modeling complex real systems present specific statistical properties, leading researchers to introduce the term of complex networks for naming them. Among main characteristics that were highlighted are the small-world and the scale-free properties. Once the observations are done and the analyses are performed, one fundamental question remains: "which mechanisms might enable such properties to appear in these networks?" The design of generative or constructive processes aimed at producing graphs with such characteristics has been shown to be a promising approach. In 1998, Watts and Strogatz proposed a way of generating graphs owning the small-world property. Starting from a regular lattice the process randomly rewires part of the network's connections [43]. One year later, in 1999, Barabási and Albert designed an iterative process for generating graphs with both the small-world and the scale-free properties [5]. At each time step a new vertex is added to the graph and is more likely linked to high degree vertices. This mechanism or rule is known as "preferential attachment". In both cases, the approach used consists in designing processes based on appropriate mechanisms to generate graphs with the desired properties. However, such studies are more interested in the final graph obtained at the end of the process rather than in the dynamics of the graph itself.

In another approach, graphs can be enriched with a notion of temporality, allowing for a paradigm shift. Fundamental concepts in graph theory, such as paths or connectivity, take on a new dimension in temporal graphs, broadening our understanding of static graphs. For example, the concept of a path extends to that of a journey, which represents a path connecting two vertices while respecting temporal constraints. Defining the shortest journey becomes more complex, as multiple definitions are possible, such as arriving as early as possible, minimizing the number of edges used, or minimizing the total time. Several models have been developed to generate temporal graphs with the aim of studying these new properties. Microcanonical Randomized Reference Models [19], for instance, create temporal graphs by reshuffling the edges of an existing graph while preserving certain characteristics, such as connectivity. Other models, like Edge-Markovian graphs, developed in [11], employ generative processes to produce a sequence of graphs starting from an initial condition. In this case, the presence or absence of an edge is determined by a Markovian stochastic process dependent on the current state of the edge, a model often used to study problems such as flooding.

At the intersection of these two perspectives, iterative processes are used to create evolving structures, enabling the study of either static characteristics, such as small-world properties, or temporal ones, like time-connectivity. In this context, dynamic graphs play a key role, as they are formed through the combination of an iterative process and an initial seed graph, producing a sequence of graphs that evolve over time. However, most models in the literature constrain the evolution of these sequences by fixing the size of the graphs or limiting the changes to the vertex set. A major contribution of this thesis is the exploration of dynamic graph generation when these constraints are relaxed. Specifically, we investigate what happens when the number of vertices is allowed to vary over time, raising fundamental questions: can a graph eventually become empty if left unconstrained? How can we quantify the changes that occur between consecutive steps in an iterative process?

To address these questions, two key metrics were introduced. The first, sustainability, assesses whether a graph generator produces non-empty and non-periodic graph sequences over time, thus maintaining structural complexity without collapsing or falling into repetitive cycles. The second metric, the DynamicScore, quantifies the changes between two consecutive time steps at both the vertex (V-DynamicScore) and edge (E-DynamicScore) levels, providing a granular view of the dynamics at play. Our analysis shows that these metrics provide valuable insights into the behavior of dynamic graphs. For instance, sustainable graph sequences maintain stability over time, while the DynamicScore helps to track the evolution of network dynamics. Additionally, we explore broader applications of these metrics, such as understanding how information might persist in dynamic graphs and determining how vertices must interact to ensure information propagation. This will constitute the core of Chapter 3.

For exploring the potential of these metrics a dynamic graph generator, presented in Chapter 4 has been proposed. Further investigations into dynamic graph families produced by this generator reveal that sustainability depends on the classification of parameters, and that additional tools are required to fully resolve questions related to the evolving network order. Among the different obtained results, we

have shown that for certain categories of parameters, the application of specific rules leads to a growing structure, allowing the generated graphs to remain sustainable. We also demonstrated that even when an estimate of the number of vertices is available, additional methods and tools are necessary to fully address the question of sustainability. This is investigated in Chapter 4.

To better see the interest of the DynamicScore metric the last part of the document is dedicated to the analysis of real-world networks. The DynamicScore metric is applied to these networks and the results are analyzed. In the study conducted during this thesis real-world networks are represented as sequences of events. These events either model instantaneous or permanent interactions. Instantaneous interaction refers to an interaction which does not last in time such as e-mails or two individuals having a conversation for instance. The analysis of such networks allowed us to generate dynamic profiles of networks, revealing how they evolve over time at different time scales. Permanent interaction here refers to a contact which never disappear once created. Example of such contact may be found in biology such as in organic system where an event model a new connection between two entities. For this case, we study the DynamicScore on filtered sub graphs of a bigger time-varying graph aggregating all the events and provide tools and ideas to observe what the DynamicScore may say about these filtering. This is detailed in Chapter 5.

Introducing Time in Graphs

In this chapter, we will explore graphs from various perspectives. We will demonstrate how this tool proves powerful for modeling and analyzing the properties of complex networks. Complex networks represent systems composed of entities interacting with one another. Several observations have highlighted common properties shared by different networks. These fascinating properties do not seem to result from randomness or any particular regularity. They can be expressed in terms of graph characteristics, illustrating the strength of this model. The observation of these properties led to the idea of generating graphs that exhibit them. Among the existing mechanisms, some stand out due to their iterative nature. Corresponding processes build a graph starting from an initial one and then apply rules successively to generate a sequence of graphs. This concept of iterative process also applies when adding a temporal dimension to graphs. Incorporating a temporal dimension provides new perspectives for studying graphs. Such an addition introduces new concepts that are absent from static graph theory. For example, the concept of paths in graphs becomes much richer within the paradigm of temporal graphs.

To cover the points mentioned above, this chapter is organized as follows. We will start by briefly introducing what graphs are and then present some fundamental concepts necessary for the rest of the chapter. In Section 2.2, we will apply the notion of graphs to the study of complex networks. We will also discuss the common properties of complex networks before examining the generative mechanisms that can be implemented to achieve these properties. Then, in Section 2.3, we will address the integration of the temporal dimension in graphs. We will begin by presenting the different formalisms used to describe such graphs. We will then discuss some properties and related works on these graphs before talking about the generators used to create them. Finally, we will introduce the notion of dynamics. This concept, applicable to both the generation of complex networks and temporal graphs, will be described in Section 2.5.

2.1 Graph

Graphs are fundamental tools for modeling real world networks. A graph consists of a set of entities, called vertices or nodes. These vertices can be connected by links, called edges. In such cases edges may have an orientation, meaning one end is the source while the other is the destination. In that case, edges are referred to as arcs. From a formal point of perspective a graph may be defined as follows:

Definition 1 *Graph:*

Let V be a set, called the set of vertices, and $E \subseteq V \times V$, called the set of edges. The couple $G = (V, E)$ is called a graph.

An illustration of a graph G is provided in Figure 2.1. In this example, the vertices of the graph are A, B, C, D, E and F . This graph is undirected, and its edges are $\{A, B\}$, $\{C, D\}$, $\{F, A\}$, $\{F, B\}$, and $\{F, E\}$.

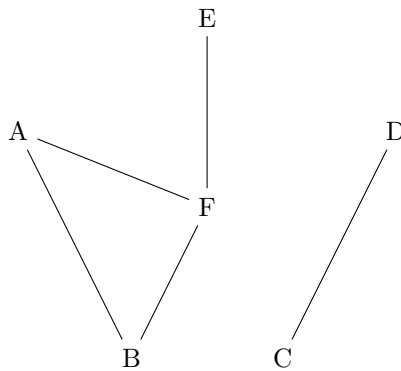


Figure 2.1: Representation of a graph with six vertices and five edges.

From the definition of a graph it is possible to define basic concepts. For instance, a **path** between two vertices u and v is a finite sequence of edges (e_1, \dots, e_k) such that it is possible to go from u to v by traversing the edges in the order provided by the sequence. For instance, a path from A to E may be $(\{A, F\}, \{F, E\})$ or $(\{A, B\}, \{B, F\}, \{F, E\})$. If a path exists between two vertices u and v we say that u can **reach** v . If every vertex may reach every other vertex, then the graph is said to be **connected**, otherwise, it is said to be **disconnected**. The graph represented on Figure 2.1 is not connected because there is no path between A and D for instance.

The **length of a path** is the number of edges used in the sequence: the length $(\{A, F\}, \{F, E\})$ is two while the length of $(\{A, B\}, \{B, F\}, \{F, E\})$ is three. A shortest path between two vertices is a path such that it is not possible to find a path having a shorter length. The **distance** between two vertices corresponds to the length of a shortest path, if one exists, otherwise the distance does not exist or is considered to be infinite.

The **neighborhood** of a vertex u , referred to as \mathcal{N}^u , corresponds all the vertices sharing a common edge with u . For instance, the neighborhood of A is $\mathcal{N}^A = \{F, B\}$. The **degree** of a vertex u , referred to as $\deg(u)$, corresponds to the size its neighborhood. For instance the degree of A is $\deg(A) = 2$. In the

case of directed graphs, the notion of degree may be divided in two : **indegree** and **outdegree**. Indegree corresponds to the number of arc pointing to a vertex, while outdegree corresponds to the number of arc exiting a vertex.

With these notions it is possible to define more sophisticated metrics like **triangles** and **clustering coefficient**. A triangle in a graph corresponds to three mutually connected vertices. The clustering coefficient represents the ratio between the number of triangles and the maximum potential number of triangles in a graph. This concept can be defined either at the level of a single node or for the entire graph. When defined at the level of a node, the clustering coefficient is referred to as the **local clustering coefficient**. For a given vertex u , this quantity is the ratio size of the number of triangles u belongs to over the maximum potential number of potential triangles u belongs to. Assuming the graph is undirected, the number of potential triangles is $\binom{\deg(u)}{2}$. It possible to define the clustering coefficient at the whole graph level taking the average local clustering coefficient for all vertices. The clustering coefficient in this case is referred to as the **average clustering coefficient**.

2.2 Complex Networks Analysis

In the exploration of real-world networks, a fundamental first step involves emphasizing their inherent properties, which serve as the foundation for understanding what is considered as a common property complex networks share. These properties are the small-world phenomenon, scale-freeness, strong clustering coefficient, triadic closure and navigability. Each of them is discussed and definitions are provided to express them in terms of graph property.

In a second time, mechanisms that make properties emerging from real-world networks will be studied, shedding light on the processes that give rise to their unique features. The study of such mechanisms is made in order to explain how properties are obtained. Our investigation centers around two pivotal models: the Watts-Strogatz model, introducing a 'rewiring' mechanism in order for the small-world phenomenon to rise and the Barabási-Albert model, which employs preferential attachment, resulting in a scale-free degree distribution of node degrees. Derived models based on the Barabási-Albert model are then presented to propose mechanisms inducing new properties. Each model offers nuanced perspectives on network formation. Through a systematic exploration of these models, we endeavor to highlight the intricate dynamics governing the evolution of real-world networks, providing insights into the underlying processes that give rise to their observed properties.

2.2.1 Complex Networks Properties

The observation of real world networks reveal few common properties that are not present in random or regular graphs [7]. Many reasons motivate the search for such features, with the aim of understanding how they are produced in real life. Building generative dynamic graph models is one approach to deciphering the underlying mechanisms that lead to their emergence. This section reviews a short list of these features.

In the following, we focus on a few key properties: the small-world phenomenon, where networks exhibit both highly clustered nodes and short distances between them; scale-freeness, which refers to the presence of a few highly connected nodes alongside many vertices with small degrees; strong clustering, that corresponds to networks with high clustering coefficient values; triadic closure, which describes how two vertices connected to a common vertex tend to form a triangle; and navigability, which refers to how easily one can move from one vertex to any other using short paths. With these characteristics in mind, the aim is to understand the essence of complex networks.

2.2.1.1 The Small-World Property

A common property real complex networks share is the small world property. It translates the idea that, in a real network, only a few hops are necessary to join two different nodes taken at random. In terms of path length, it means that the average distance between two nodes is "small" compared to the size of the network. When a formal definition of how small this distance must be to considered a network as a small world, it is common to use a the logarithm of the network size. However, considerations on the average path length are not enough to state whether or not a network has the small-world property. Indeed, Watts and Strogatz in [43] have identified another feature networks must satisfy to be qualified as small world. This other feature is the clustering coefficient. In their original article, this quantity referred to as the average clustering coefficient and is defined as the average clustering coefficient of every node. See section 2.2.1.3 for more details. This quantity must be high enough for networks to be qualify as small world. Gathering these two metrics, the clustering coefficient and the average shortest path, a definition of the small world property may be as follow:

Definition 2 *Let $G = (V, E)$ be a network. Let n denote the size of the graph (i.e., $n = |V|$). Then, G satisfies the small world property if and only if, the average shortest path between every pair of nodes is proportional to $\ln(n)$ and if the average clustering coefficient C_G is high enough and not depending on the network size.*

It is important to notice that not all networks share this property. One way to realize it is to look at simple theoretical network topologies. Let us then consider different network topologies and see whether or not they satisfy the small world property.

As a first topology, let us consider the lattice or the grid topology. These topologies are very regular and therefore easy to study. In such networks, the average shortest path is proportional to n . As a consequence, they are not qualified as small-world networks. However, their average clustering coefficient is high. In the case considered by Watts and Strogatz, a regular lattice of n nodes, each connected to its k nearest neighbors, the clustering coefficient of each node is $\frac{3}{4}$.

Another topology one may consider is the topology of a random graph. The way the graph may be draw being wide, the study is restricted to graph of size n and where an edge between any two vertices has a probability p to exist or not. This way to obtain a graph is known as the Erdős-Rényi (ER) model

[17]. It turns out that using such a model to build up graphs often results in a network having a small average shortest path. However, the clustering coefficient of this family of networks is very low. Thus, completely random topologies are not small world.

In order to find a satisfying model exhibiting the small world phenomenon, Watts and Strogatz provided a generative process that takes place between a regular and a random network. This model is further studied in section 2.2.2.1. Without giving all the details, the idea behind the generation mechanism is to rewire a lattice graph so that, after a certain amount of rewiring, the graph becomes a small-world network. This model has different properties ER does not have and are investigated in the corresponding section.

2.2.1.2 Scale-free Networks

In their article [5] Barabási and Albert have shown that most real world networks, the World Wide Web for instance, exhibit a property called *scale-free*. The term scale free means that the degree distribution exhibits a power law distribution. The origin of the scale-free property may also be explained by the topological structure of these networks. Indeed, it has been observed that a few nodes, called hubs, gathered a lot of connections towards nodes, while nodes with few connections with others are more current. The following definition formalizes the property:

Definition 3 *A network is said to be scale free if there exists a and $\gamma > 2$ such that the probability $P(k)$ for a given node to have a degree k satisfies:*

$$P(k) = ak^{-\gamma}$$

The just stated definition involves the probability mass function. Some authors also sometimes restricts the range γ belongs to: $2 < \gamma < 3$. This comes from observations on real networks such as the Internet or social networks. Different models trying to produce scale free networks have emerged last decades. Among others the most popular one is the Barabási-Albert model. This model will be more precisely defined in Section 2.2.2.2. It has popularized preferential attachment as a mechanism for a new comer in a network to connect with others. This mechanism has extensively been studied. Other similar mechanisms are used for building up new models, like hyperbolic geometry.

Hyperbolic Geometry Models based on this second concept tries to see the emergence of important features such as scale-freeness, efficient navigability in the network and strong clustering coefficient. Models using only preferential attachment as defined in the original article of Barabási and Albert [5] do not exhibit these two features that seems to be present in real-world networks according to [35]. Recent works like [28] suggest that a hidden hyperbolic metric space may be found for every complex network. The curvature of a given hyperbolic space influences the value of the power-law exponent of the degree distribution. The authors also explain that hyperbolic spaces can be used to efficiently build complex networks, as many of the common properties exhibited by complex networks emerge from the use of such spaces.

Some representation of hyperbolic geometry represents the space using the Poincaré disk model [28]. It is a model generalizing the flat euclidean geometry. Such spaces show a negative curvature $-\zeta^2 < 0$. Such a model consists in a 2D disk with a radius R in which the distance between every pair of points $(u, v) = ((r_u, \theta_u), (r_v, \theta_v))$, referred to as d_{uv} , is computed using the hyperbolic law of cosines:

$$\cosh(\zeta d_{uv}) = \cosh(\zeta r_u) \cosh(\zeta r_v) - \sinh(\zeta r_u) \sinh(\zeta r_v) \cos \theta_{uv} \quad (2.1)$$

In the above equation, $\theta_{uv} = \pi - |\pi - |\theta_u - \theta_v||$ is a mathematical way to compute the shortest angle between u and v , that is an angle ranging from 0 to π . The main difference between euclidean and hyperbolic geometry lies in the importance some points have compared to others.

2.2.1.3 Strong Clustering Coefficient

This sub section focuses on the clustering coefficient metric. This metric aims at quantifying the cliquishness (i.e., the capability for a network to embed cliques). This metric is useful to describe the small-world phenomenon as well as to quantify the transitivity of a network. From a network point of view, this metric is computed by comparing the number of triangles in the graph with the maximum number of triangles that could exist. From definition provided in Section 2.1, clustering coefficient involves the average clustering coefficient. It has been first introduced by Watts and Strogatz to compute the clustering coefficient of their model [43]. This quantity along side with the average shortest path allows to define precisely the small-world phenomenon: in real-world networks, the clustering coefficient is high and does not depend on the size of the graph. For the model proposed by Watts and Strogatz this value is for instance close to $\frac{3}{4}$.

2.2.1.4 Triadic Closure

Clustering coefficients are powerful tools for measuring the transitivity of a node and, by extension, verifying whether a network adheres to the principle of triadic closure. In general, triadic closure is a principle, introduced by Georg Simmel in 1908 in his book *Soziologie* [39] (See a translation in [40]), which states that in a network, if a node is connected to two other nodes, it is highly likely that these two nodes are also directly connected. In social networks, this property can be more informally summarized as "the friends of my friends are my friends". From a network generation perspective, this formulation helps in the design of mechanisms for creating links between nodes: if a node u is connected to a node $v \neq u$, it is possible to choose a neighbor of v to create a connection between this neighbor and u . Among all existing network generation models, we can cite [6, 37]. The model proposed by [6] for instance uses this principle, in addition to the principle of preferential attachment, to construct networks that exhibit both scale-free properties and a high clustering coefficient. In their article, the authors do not give any specific value or expression for the clustering coefficient to ensure triadic closure. However, as in the case of the Watts-Strogatz model [43], both mention an average clustering coefficient greater than 0.7. This model is discussed in more detail in Section 2.2.2.3.

2.2.1.5 Navigability

Navigability is a notion introduced to address the problem of finding a path between any pair of vertices that is close to the shortest path, in a network that is not fully known, relying only on local information. This question has been investigated in the study of the small world phenomenon by Stanley Milgram in [42]. Navigability is also close to the idea of the six degrees of separation. This theory states that, on Earth, the social distance between any two people is on average 6. Later, physicists such as Boguña et al. in [8] explained that navigability in complex network may be explained by the existence of an underlying hidden metric space. Such a space is called a hyperbolic space. This concept has already been presented in Section 2.2.1.2. Short paths between vertices have to be found without a global knowledge of the network, using only local information. An algorithm may be designed to find this shortest path using information of the hidden metric space after embedding the network in the space.

These properties: small-world, scale-freeness, navigability and triadic closure, are present in many real-world networks. A key question is: how were such properties acquired during the evolution of these networks? What mechanisms can explain their emergence? The following section reviews several well-known generative network models that attempt to provide insights into these questions.

2.2.2 Generation of Complex Networks

This section focuses on existing models found in the literature used to build artificial networks satisfying properties defined in Section 2.2.1. The aim of this generation is to find mechanisms that can explain the origin of certain properties in complex networks. The first model studied here is a model designed to produce model exhibiting the small-world phenomenon. The next models are introduced to produce scale-free networks. Several models are presented derived from a popular one called the Barabási-Albert model.

2.2.2.1 The Watts-Strogatz Model

The Watts-Strogatz model is a fundamental network model that was developed by Duncan J. Watts and Steven H. Strogatz in 1998 [43]. It was designed to address an intriguing problem in observed real networks: the small world phenomenon. This phenomenon refers to the idea that, in many real-world networks, any two individuals are connected by a surprisingly short chain of hops and, in the same time, the network shows a highly clustering topology. This model was motivated by the need to understand how such small-world properties emerge in social networks, as well as in various other domains, including the brain's neural connections and the structure of the internet.

The Watts-Strogatz model starts with a regular lattice network, which is a structured network where each node is connected to its nearest neighbors. The model then introduces randomness by rewiring some edges, which leads to the emergence of small-world characteristics. The Watts-Strogatz model can be formally defined using three parameters as follows

Definition 4 Let $n \in \mathbb{N}$, $k \in \mathbb{N}$ such that $k < \frac{n}{2}$ and $p \in [0, 1]$ a probability. Let n represent the number of nodes, k the number of nearest neighbors each node is initially connected to, and p the probability of edge rewiring. The model can be expressed mathematically as follows:

1. Start with an ordered ring of n nodes, where each node i is connected to k nearest neighbors on either side. This creates a regular lattice network, which can be represented as a graph $G(n, k)$.
2. For each edge (i, j) in $G(n, k)$, with a probability p , rewire the edge to connect to a different node m selected uniformly at random, such that $m \neq i$ and $m \neq j$. This process is done independently for each edge. The resulting network is denoted as $G(n, k, p)$.

In this formal definition, $G(n, k, p)$ represents the Watts-Strogatz network with n nodes, each initially connected to k nearest neighbors, and edges rewired with probability p . This mathematical representation captures the key steps of the Watts-Strogatz model and allows for a precise description of the network structure and randomness. The clustering coefficient of this structure is $\frac{3(k-2)}{4(k-1)} \simeq \frac{3}{4}$ while the average shortest path is close to $\frac{n}{2k}$ for the initial graph. It is therefore not a small world due to typical path length (in the following of this section, $C(n, k, p)$ will refer to as the clustering coefficient while $L(n, k, p)$ will refer to as the typical shortest path). In general, for small values of p , the network retains its regular lattice structure. This means that nodes are still connected to their nearest neighbors, and the network exhibits high clustering. However, the introduction of a few random edges means that, in practice, there are shorter paths between most pairs of nodes, which is characteristic of the small-world phenomenon. Watts and Strogatz have shown this phenomenon comes quickly as p increases. On the opposite side, namely as p tends towards 1, networks become increasingly random, with most edges being rewired. In this limit, the small-world properties disappear, and networks start resembling random networks with short average path lengths ($L(n, k, p) \sim \frac{\ln(n)}{\ln(k)}$) but low clustering ($C(n, k, p) \sim \frac{k}{n}$).

Finally, the in-between range of p values reveals a special mix of features that define the small-world behavior in the Watts-Strogatz model. In this range, the network exhibits both small-world properties and the characteristics of a regular lattice. The small-world phenomenon is most pronounced when the value of p falls within this transition range. This suggests that to observe the small-world phenomenon, the value of p should be considered small. In their article, Watts and Strogatz have exhibit the trend of the clustering coefficient and the length as function of parameter p , showing the quick drop in the average shortest path and the later drop of clustering coefficient.

The networks obtained through this generative process are single static graphs, achieved after a finite number of rewiring steps. Properties are assessed on the final graph obtained after all the rewiring. Thus, time is absent from the process, and dynamics play no role in the results. There is no evolution in the network; all the rewiring could be applied at once, and the same result would be achieved.

2.2.2.2 The Barabási and Albert Model

In this section, we present the Barabási-Albert (BA) model. This model, introduced in the article [5] in 1999, consists in a generative process for building scale-free networks. The generated graphs fall into the category of "growing networks". The model is based on the preferential attachment mechanism, which manages the connections of new vertices added to the network, feeding its growth. This principle can be summarized as follows: a new vertex is more likely to join a vertex with a high degree in the network than one with a low degree. The Barabási-Albert model can be formally defined as follows, using 3 parameters:

Definition 5 *Let $n_0 \in \mathbb{N}$, $m_0 \in \mathbb{N}$ and $m \leq m_0$. The generative process is described by the following algorithm:*

1. *Start with a seed graph G_0 with n_0 vertices and m_0 edges.*
2. *To build G_{t+1} from G_t for any $t \in \mathbb{N}$, add a new vertex u to V_t and connect it with m distinct vertices. These vertices are chosen so that the probability u connects to them depends on their degree :*

$$\forall v \in V_t, \Pr[u \sim v] = \frac{\deg(v)}{\sum_{w \in V_t} \deg(w)}$$

With this model, it is possible to build networks with a power-law degree distribution. The exponent γ of the power-law equals 3, which is, as we showed in Section 2.2.1.2, a common exponent value one may find in real-world networks. An illustration of this phenomenon is provided figure 2.2. This figure represents the degree distribution of a network produced using the Barabási-Albert model. The seed graph G_0 contains $m_0 + 1$ vertices and m_0 edges and is connected. The final graph of the sequence contains 100000 vertices. The degree distribution represented on Figure 2.2 exhibits a power law degree distribution with $\gamma = 3$. The tail of this power law is said to be fat as it does not fit exactly the theoretical equation represented by a red dashed line. This comes from the stochastic behavior of the BA process.

This model highlights one mechanism that could explain how scale-freeness emerges in complex networks. However, the emergence of the property is observed when the graph is large, for a large number of iterations of the process. Therefore, even though the model builds dynamic graphs with growing size, the purpose of this model is to build graphs that become scale-free only for large orders. Note also that the measure of this property is made on a static graph rather than on the sequence of graphs. This point will be further investigated in Chapter 3 in which metrics are applied to temporal networks (called dynamic graphs).

2.2.2.3 BA-based models

This section introduces new models based on the BA mechanism and highlights what properties emerge within the networks generated by these models.

Degree distribution of a BA graph

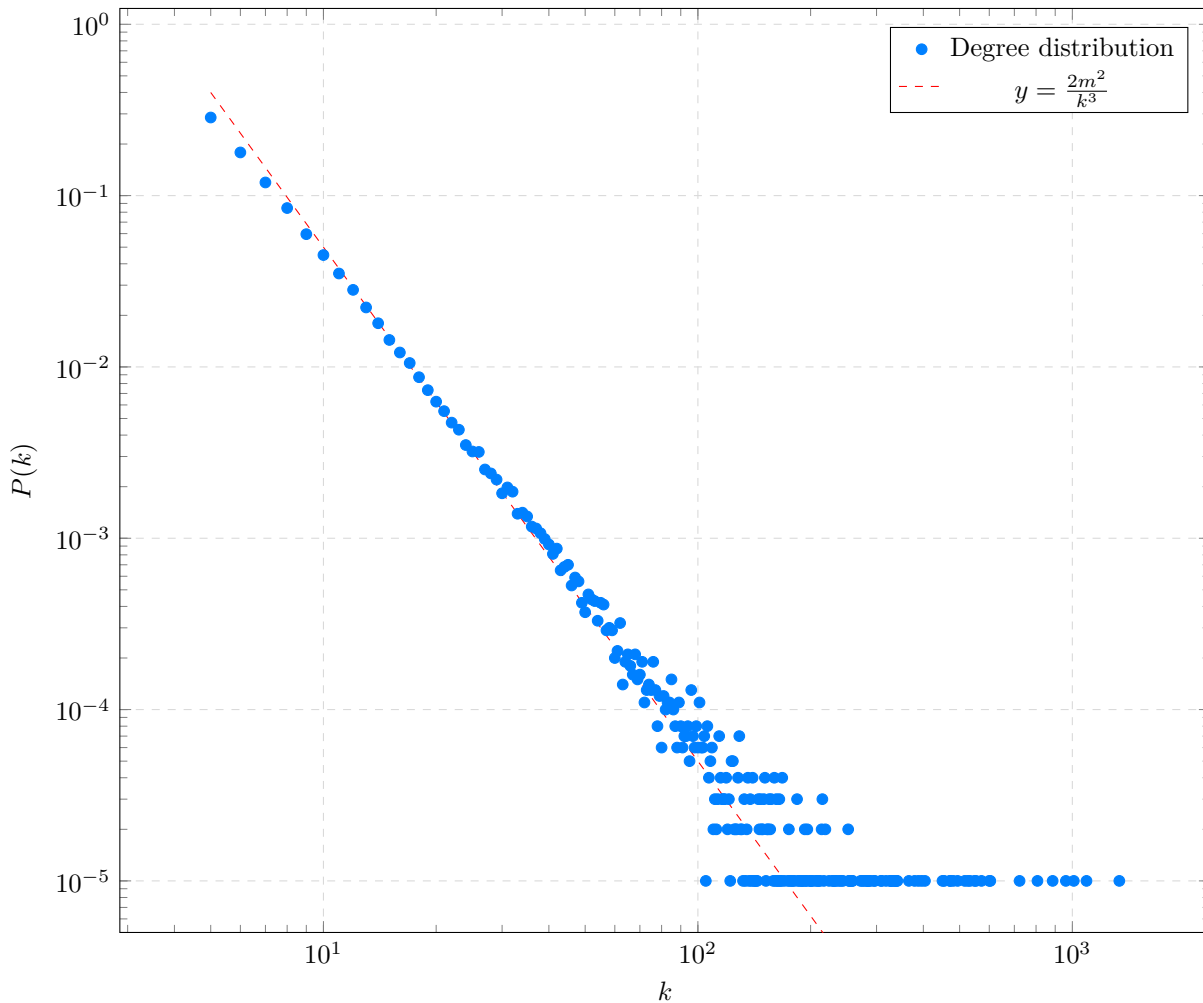


Figure 2.2: The degree distribution of a BA graph obtained with $n = 100000$ final vertices, $m_0 = 6$ initial vertices and $m = 5$ new connections every step.

Triadic Closure The Triadic Closure model aims at generating networks exhibiting both a strong clustering coefficient alongside the scale-free property. This is not the case of BA model where the clustering coefficient depends on time and shows a decreasing trend. Triadic Closure is a new principle for connecting new nodes to others. An example of model using this mechanism may be found in [6]. It is worth mentioning the number of connections a node makes when inserted is fixed and is a parameter of the model. It will be referred to as k and must be greater than or equal to 2. The principle behind this model can be divided in two steps. When a new node is added to the network, a first connection, using preferential attachment, with one already existing node, is created. Then, the $k - 1$ remaining connections are performed, either using the PA mechanism or such as to build a triangle. Bianconi and her colleagues have shown networks generated using such a mechanism exhibit the scale-free property as well as strong clustering coefficient.

Popularity vs Similarity This model has been designed to introduce a new way for nodes to connect with others. Instead of only considering popularity, which is a synonym for a node of having a high

degree, new nodes will also consider homophily. Homophily comes from the intuitive idea that new nodes may prefer to connect with nodes similar to them if possible. An implementation of this idea in complex network construction may be found in [35, 47, 33]. The main contribution for these models was to represent this notion of homophily using a mathematical framework called hyperbolic geometry. Each time a new vertex is added to the network, polar coordinates are attached to the node. Distribution of node angles enable the computation of the similarity. The main difference between euclidean and hyperbolic geometry has already been explained in Section 2.2.1.2.

Conclusion The study of large structures such as the World Wide Web or social networks, is important for understanding the emerging properties of complex networks. In this domain, observed properties of networks have been first studied. Properties of networks are features of the corresponding graph models. These feature may concern the degree distribution, distance between vertices or average clustering coefficient for instance. The causes that produce these properties have been then investigated to determine what may lead to them. The study conducted here involves generative processes allowing scientists to build networks having the same structure as those found in real world networks. One can, for example, cite the rewiring method used by Watts and Strogatz to transform a regular graph into a so-called small-world graph. Similarly, the preferential attachment method used by Barabási and Albert is employed to produce scale-free networks. In both cases, the generation mechanism is used to produce a final static graph exhibiting the desired property or properties. Although these two models start from an initial configuration and result in a graph through the application of rules, the intermediate sequence of graphs does not seem to be explored in a the same way for both of them. The model of Barabási and Albert and its derivatives is based on an iterative process that produces a sequence of graphs, the limit of which is used only for its properties. This sequence of graphs may be seen as a time-stamped graph and preferential attachment as a generator of such graphs.

These generative models, while successful for building networks that exhibit some a priori selected properties, do not take into account the time dimension and the evolution of the network. In the following sections, we will study how time can be introduced in graphs and what it entails on some fundamental measures and notions, like connectivity, distance, and shortest path.

2.3 Time in Graphs

Adding the notion of time in graphs is a way to generalize the concept of graph. Instead of considering vertices and edges as constant sets, they are now time-dependent. This induces the idea that everything may change over time, such as edge and node properties. The time considered to deal with dynamic graph is most of the time discrete. Time might be continuous or discrete. For instance, if two nodes are connected by an edge which value equals the euclidean distance between them, and if the nodes are moving, then this value changes continuously. In the context of this work we will not consider such a situation. We propose to consider discrete time. In Chapter 5, we will continue the discussion about possible time discretization.

Moreover, in this paradigm, vertices and edges are allowed to be removed and/or added. Different names for such graphs can be found in the literature such that *temporal graphs*, *temporal networks*, *time-varying graphs*, *evolving graphs* or *dynamic graphs* for instance. There exist different ways to introduce time in a graph, or to model it.

In the following sections dedicated to time-varying graphs, we first introduce the formalism used to incorporate time into graphs. Then, we present fundamental properties of these graphs, which are analogous to properties found in static graphs. The first property is the concept of a journey, which extends the notion of a path. The second is time-connectivity, an extension of the traditional notion of connectivity. This notion is further explored to propose a classification of time-varying graphs based on their features. Next, we delve into generative models to propose mechanisms for producing time-varying graphs.

2.3.1 Formalism

It is possible to model time-dependence by introducing a labeling function that associates, to each edge (resp. vertex), a set of dates corresponding to the edge (resp. vertex) presence in the graph. From a formal point of view, this can be written as follows:

Definition 6 *Graph with Time Labeling Functions:*

Let $G = (V, E)$ be a static graph (also referred to as underlying graph).

- \mathcal{T} a set of non-negative integers,
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ a time presence function associating to each edge e and each date t .
 $\rho(e, t) = 1$ if e is present in the graph at date t and $\rho(e, t) = 0$ otherwise,
- $\psi : V \times \mathcal{T} \rightarrow \{0, 1\}$ a time presence function for the vertices,
- $\zeta : E \times \mathcal{T} \rightarrow \mathbb{N}$ a traversal time function associating to each edge e and each date t the time to cross e starting from date t .

Then $(G, \mathcal{T}, \rho, \zeta, \psi)$ may be qualified as a time-varying graph.

This way to model dynamic graphs may be found in [21, 9]. It is a very complete model where presence of vertices/edges and edge traversal times are considered. Not all the works, found in the literature, model graphs with all these parameters. For instance ψ is often omitted and vertices are therefore considered being part of the graph during the whole time the graph exists. In addition, in some paradigms, instead of deleting nodes, all their links are removed, making them isolated. For this specific model, isolated vertices are considered as absent.

Some even use simpler model when studied graphs are constrained. For instance if every edge is present at most one time step, then graphs may be modeled as $(G, \mathcal{T}, \lambda)$ where $\lambda : E \rightarrow \mathbb{N}$ is labeling function associating to each edge its single time presence (see [26] as an example of such a paradigm). In this paradigm, the set of vertices is most of the time constant (except when some failure communication

protocol are studied) and only edges are allowed to appear or disappear. If the time is finite, then it is possible to model \mathcal{T} as an interval $[0, T]$, where T is the greatest time in \mathcal{T} .

The second way to model time in graphs is to use a sequence of static graphs. In this formalism, the whole sequence refers to as a time-varying graph. :

Definition 7 *Sequence of Static Graphs:*

Let \mathcal{T} be a set, $(G_t)_{t \in \mathcal{T}} = (V_t, E_t)_{t \in \mathcal{T}}$ be a sequence of static graphs. Then $\mathcal{G} = (G_t)_{t \in \mathcal{T}}$ is a time-varying graph.

This definition may be found in [46]. Here, instead of having a function labeling edges, the dynamic graph consists in the sequence of all the static graphs representing its evolution through time. It is worth mentioning that the order of the graphs in the sequence matters.

2.3.2 Problems and Properties in Graphs with Time

Introducing time in graphs induces a new paradigm in graph theory. A new concept emerges from this paradigm : the notion of *journey*. This notion, which generalizes the notion of path in the graph theory, is a key concept in the study of time-dependent graphs. It is a fundamental notion leading to the study of new problems and property focusing on the effect of time in graphs. The concept of journey is relevant for instance to properly define and study question of reachability in time-dependent graphs. It is also helpful to define concepts such as time connectivity which is a very studied question.

This Section is organized as follows. First, we provide a definition of what a journey is. Then, the notion of reachability is defined as well in the same section. The notion of time connectivity of a whole graph and related problems are defined and studied in Section 2.3.2.2. To illustrate every question and notion defined in this section, an instance of a temporal graph with four vertices and three time steps is provided Figure 2.3.

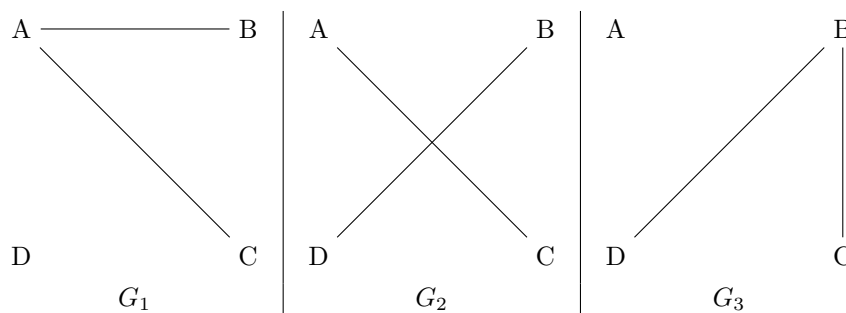


Figure 2.3: Representation of a dynamic graph $\mathcal{G} = (G_1, G_2, G_3)$.

2.3.2.1 Journey

The notion of a journey is closely related to that of a *route*. A route in a time-varying graph is an ordered sequence of edges from a vertex u to a vertex v (u and v may be the same) that respects the time order of the edges. For instance a route between vertex A and vertex D on graph represented Figure 2.3 would

be $(\{A, B\}, \{B, D\})$. A journey between two vertices u and v is then defined as a route in which the time at which each edge is crossed is specified. Such a notion may be found in [23, 46] for instance. Using the example given above, a journey between vertex A and vertex D might be $((\{A, B\}, 1), (\{B, D\}, 2))$. A simple illustration of this journey is given Figure 2.4. Let us defined more formally this notion:

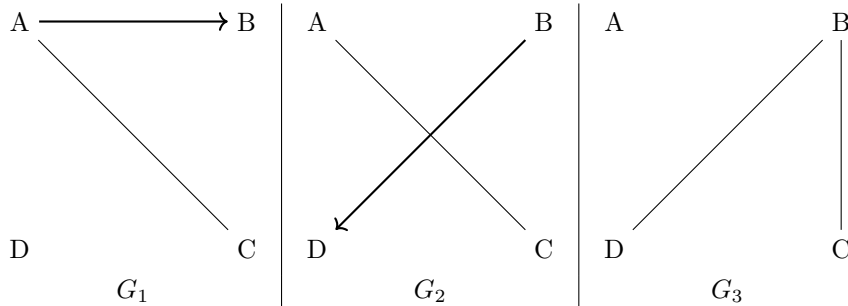


Figure 2.4: Representation of a journey between vertices A and B in the dynamic graph $\mathcal{G} = (G_1, G_2, G_3)$.

Definition 8 *Journey:*

Let $\mathcal{G} = (G_t)_{t \in \mathcal{T}}$. Let $V = \bigcup_{t \in \mathcal{T}} V_t$ and $E = \bigcup_{t \in \mathcal{T}} E_t$. Let u and v be vertices in V . Then a journey $\mathcal{J}(u, v)$ between u and v is a sequence $((e_i, t_i))_{i \in [0, \tau]}$, such that:

- τ is the length of the sequence.
- $(e_i)_{i \in [0, \tau]}$ is a path from u to v in the underlying graph $G = (V, E)$.
- $t_i \leq t_{i+1}$ for all $i \in [0, \tau - 1]$.

There are several points worth mentioning about the notion of journey. First, more than one journey may exist between two vertices. For instance $((\{B, A\}, 1), (\{A, C\}, 2))$ is a journey from B to C as well as $((\{B, C\}, 3))$. Second, the notion of journey is not a symmetric relationship: if there exists a journey from u to v , it is possible that there is no journey from v to u . On Figure 2.3, it is possible to find a journey from vertex A to vertex D (see above), while there is no journey from D to A . The question of finding journeys between vertices in the time-varying graph is referred as the *reachability* problem. This problem may be defined as a decision problem. It is formally defined in this document by the following definition:

Definition 9 *Reachable* (\mathcal{G}, u, v)

Require: $\mathcal{G} = (G_t)_{t \in \mathcal{T}}$ a dynamic graph, u and v two vertices from \mathcal{G} .

Ensure: 1 if there exists $\mathcal{J}(u, v)$, 0 otherwise.

All the points addressed above may be rewritten using this problem. For instance, using the example Figure 2.3, $\text{Reachable}(\mathcal{G}, A, D) = 1$ while $\text{Reachable}(\mathcal{G}, D, A) = 0$.

Each edge may also be assigned a traversal time for specifying the time spent crossing it. On the example provided Figure 2.4, traversal times were not specified. It is possible to fix this time to 1 for instance. This means, for a journey $((e_i, t_i))_{i \in \mathcal{T}}$, that $t_i < t_{i+1}$. If the traversal time is set to 0, then it

is possible to traverse more than one edge at a given date.

This leads us to another remark about the central notion of shortest path in static graphs. The analogous, based on the journey notion, is not straightforward translated in time-varying graphs. Three variants of shortest paths may be derived from the notion of journey.

For instance one may define the shortest journey that minimizes the number of edges in the path, or the foremost journey which minimizes the arrival date of the path, or the fastest journey, that minimizes the difference between the starting date and the arrival date of the path. These variants were introduced in [46]. With this consideration, $((\{B, A\}, 1), (\{A, C\}, 1))$ is a journey from B to C using two edges at step 1. Note that this journey is a foremost journey from B to C , but not a shortest journey as it requires two edges while $((\{B, C\}, 3))$ is a shortest journey. In the case where all edges have the same traversal time, this shortest journey is also the fastest one. If the time-varying graph consider different traversal times on the edges, then shortest and fastest journeys are generally different. For instance on 2.5, a fastest journey between A and B is $((\{A, C\}, 1), (\{C, B\}, 3))$ reaching B after 4 steps, while the shortest journey $((\{A, B\}, 1))$ reaches B after 7 steps.

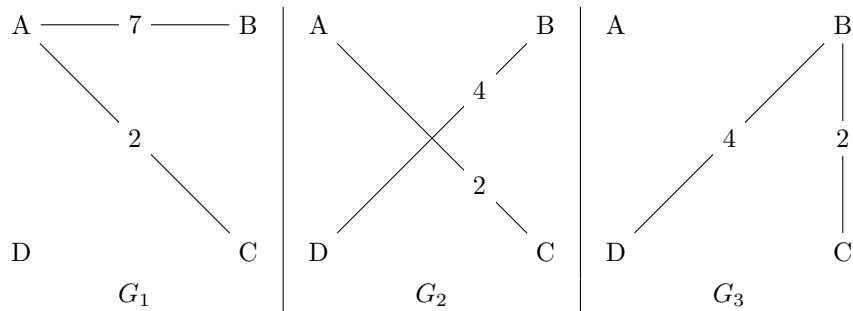


Figure 2.5: Representation of a dynamic graph $\mathcal{G} = (G_1, G_2, G_3)$ with traversal times represented by labels on edges.

2.3.2.2 Connectivity

An important question treated when time is introduced in graphs is whether any given vertex in the graph may reach all the vertices in the graph. If every vertex u of a dynamic graph G may reach every other vertex v , then the dynamic graph is said to be time connected. For instance, the example provided Figure 2.3 is not a time connected graph as D cannot reach A . An example of time connected graph is provided Figure 2.6. Indeed, a journey exists between any two vertices (note that edges are non-oriented):

- from A to B : $((\{A, B\}, 1))$;
- from A to C : $((\{A, B\}, 1), (\{B, C\}, 2))$;
- from A to D : $((\{A, D\}, 2))$;
- from B to A : $((\{B, A\}, 1))$;
- from B to C : $((\{B, C\}, 2))$;
- from B to D : $((\{B, A\}, 1), (\{A, D\}, 2))$;
- from C to A : $((\{C, D\}, 1), (\{D, A\}, 2))$;
- from C to B : $((\{C, B\}, 2))$;
- from C to D : $((\{C, D\}, 1))$;
- from D to A : $((\{D, A\}, 2))$;

- from D to B : $((\{D, C\}, 1), (\{C, B\}, 2))$;
- from D to C : $((\{D, C\}, 1))$.

One important thing to notice is that a graph may be time connected while every static graph of the sequence may be disconnected. The previous example illustrates this situation.

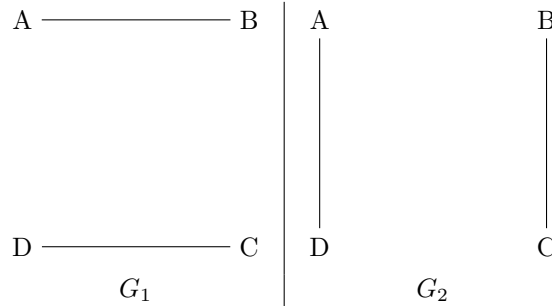


Figure 2.6: Representation of a time connected dynamic graph $\mathcal{G} = (G_1, G_2)$.

Classification The concept of connectivity, alongside with the one of journey, has led to the study of various notions. Several properties are derived from the simple notion of time-connected graph. These notions and their relationships have motivated researcher to classify time-varying graphs according to their properties. This is for example the case of the article [38]. In this paper the authors designed different classes of time-varying graphs according to the connectivity/reachability of graphs. For example, there is a class of graphs where a vertex can reach all other vertices. Such a vertex may be referred to as a *temporal source*. Another example is the class of graphs where there is a journey between every pair of vertices in both directions: for all vertices u and v , there exists a journey $(e_i, t_i)_{0 \leq i \leq k}$ from u to v and a return journey $(e'_i, t'_i)_{0 \leq i \leq k'}$ from v to u such that $t_k < t'_0$. This means it is possible to send a message from any vertex u to any vertex v , and v can reply to u .

This class may be referred to as the class of round connected graphs. The example provided on Figure 2.6 is not round connected. An example of round connected graph may be found on Figure 2.7. Such a

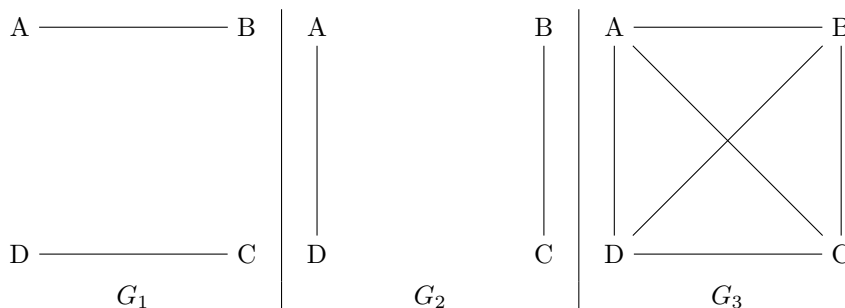


Figure 2.7: Representation of a round connected dynamic graph $\mathcal{G} = (G_1, G_2, G_3)$.

graph is indeed round connected as edges at step 3 may be used to build journeys to go back to every source vertex. This is a consequence of the the last graph being a complete graph and (G_1, G_2) is already time connected.

2.3.3 Generation of Time-Varying Graphs

The generation of time-varying graphs generally aims to create graphs with some specific properties. The rest of this section is dedicated to some time-varying graphs generators. For each of them, a brief description of the mechanism producing graphs is studied.

2.3.3.1 Generator of Edge-Markovian Graphs

This section presents the Edge-Markovian Graphs Generator (EMGG), its formal definition and some of its fundamental properties. This model has been introduced in [11] and is a generalization of markovian graphs. The Edge-Markovian Graphs Generator (EMGG) is a stochastic process that produces an infinite sequence of static graphs. Let G_t refer to as the graph produced at step t . $G_t = (V_t, E_t)$ where V_t (resp. E_t) represents the set of vertices (resp. edges) at step t . The EMGG is parameterized by two probabilities, denoted as p and q , along with an initial condition or seed graph, denoted as G_0 . The set of vertices of the graph does not change during the evolution process, so, for all $t > 0$, $V_t = V_0 = n$. Given two vertices u and v , if at step t the edge $(u, v) \in E_t$, the edge is said *present* and *absent* otherwise. The EMGG operates as follows: at each step, all possible edges (present or absent) are examined¹. The generator determines for each edge if it will remain in the same state (present/absent) in the next snapshot graph or if it will change. The decision is based on two probability parameters: $0 \leq p \leq 1$ and $0 \leq q \leq 1$. The role of p is to define the probability that an edge present at a given step remains present during the next step, while the role of q is to define the probability that an edge absent at a given step remains absent during the next step. Note that in the original paper the, p is defined as the death rate and q as the birth rate. To have an equivalent definition, it is necessary to replace p by $1 - p$ and q by $1 - q$. The behavior of this process is summarized in the diagram of Figure 2.8.

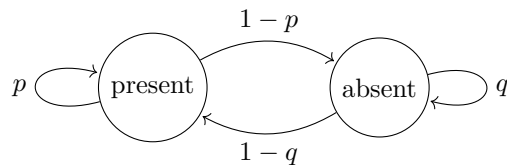


Figure 2.8: Description of the states in the Edge-Markovian Graphs Generator.

A few specific values of p and q lead to distinct behaviors. For instance, when both p and q are set to 0, the generated graphs exhibit a blinking behavior, with edges alternating between present and absent at each step. Conversely, when p and q are both set to 1, the graphs remain static throughout, with G_t equal to the initial graph G_0 at all time steps. In the case where $q = 1 - p$, the process becomes time-homogeneous, meaning that the generation of each new graph is independent of the previous step. Other cases result in different behaviors, which are explored in the next chapter.

¹there are $n(n - 1)/2$ such edges

Definition 10 *Edge-Markovian Graph Generator (EMGG):* An EMGG is parameterized through 4 parameters $n \in \mathbb{N}^*$, p and $q \in [0, 1]$ and an initial configuration G_0 . Instances produced by such a generator are such that:

- for all step t , $|V_t| = n$;
- for all pair of vertices $e = (u, v) \in V_t^2$:
 - if $e \in E_t$, then $e \in E_{t+1}$ (remain present) with probability p and becomes absent with probability $1 - p$;
 - if $e \notin E_t$, then $e \in E_{t+1}$ (becomes present) with probability $1 - q$ and remain absent of E_{t+1} with probability q .

The maximum number of edges that may be contained at a given step t is $\binom{n}{2}$. The set of edges is evolving through time and computing Edge-DynamicScore gives an information about its dynamics. In the following sub sections we establish a solid foundation for understanding its dynamics and its relationship with probabilities p and q .

2.3.3.2 Simple Time-Varying Graphs Generation

The model discussed here has been designed to study time-varying graph connectivity for graphs satisfying two conditions. First, produced graphs are *simple*, which means every existing edge must appear exactly once in a graph. Second, graphs are *proper*, which means two edges sharing one common vertex cannot exist at the same time. The model considered has been introduced in [10]. The purpose was to study questions of connectivity on temporal graphs. The proposed model is defined as follows:

Definition 11 Let $n \in \mathbb{N}$, let $p \in [0, 1]$. The generator produces graph using the following steps: First, the generator create a complete graph with n vertices. For every edge, a number is drawn uniformly at random in $[0, 1]$. This number is considered as the label of the edge. Then, the generator filters edges and keeps only those having a label lower than or equal to p . The generator next changes the label as integers ranging from 1 to the $|E|$ (the number of remaining edges) following the order induced by the labels. This represents the final graph : $G_t = ([1, n], \{e | \lambda(e) = t\})$ where $\lambda(e)$ is the label of edge e .

This construction ensures produced graphs simple and proper. Indeed, every snapshot graph G_t has only one edge and this edge is different at any step. Such graphs are completely changing from one step to the next one. Another point to study here is the mechanism lurking underneath the creation of such graphs.

2.3.3.3 Generating Graph with a Given Feature

The model discussed here is designed to generate time-varying graphs satisfying a given property. This idea has been developed in [19]. In this article the authors present the notion of Microcanonical Randomized Reference Model (MRRM). One key concept necessary to present MRRM is the concept of

feature. A feature is a function that takes a time-varying graph as an input. Thus, the function returning the underlying graph of a time-varying graph is a feature as well as the function returning 1 if the a time-varying graph is connected and 0 otherwise. A MRRM is therefore a generative model defined as follows:

Definition 12 *Let x be a feature and \mathcal{G}^* be a time-varying graph. A MRRM is a model that takes \mathcal{G}^* as input and which returns a time-varying graph \mathcal{G} such that $x(\mathcal{G}^*) = x(\mathcal{G})$. The output graph \mathcal{G} is drawn uniformly at random among all the time-varying graphs \mathcal{G}' satisfying $x(\mathcal{G}') = x(\mathcal{G}^*)$.*

This definition may be constrained introducing a state space. Such a space is a set of time-varying graphs satisfying some conditions. For instance the state space may only contain graphs with a fixed number of vertices and bounded in time.

There are several ways to generate time-varying graphs using this model. In their article [19], Gauvin et al. present three methods. The first method involves shuffling, either by randomizing the time labels of the edges or the snapshot graphs while preserving a given feature. The second method uses the feature to partition the state space, where the feature is used to build equivalence classes. The third method views a MRRM as a transition matrix, treating it as a linear stochastic operator that transforms a time-varying graph \mathcal{G}^* into another time-varying graph \mathcal{G} .

A key strength of such networks lies in the concept of features. A feature is only constrained by the input, which must be a time-varying graph, but is flexible in terms of output. For example, a feature could be a decision problem, such as determining the time connectivity of a time-varying graph, or a graph, such as extracting the underlying static graph from a time-varying one. However, for this model to function, both a feature and a time-varying graph must be defined. As a result, this approach is not suitable for discovering time-varying graphs that satisfy certain properties, as a graph exhibiting the property must already be provided as input.

2.4 Discussion on Time in Graphs

Graphs are widely used to represent and study complex networks. However, static graphs are often insufficient to explain dynamical phenomena in real-world networks. To address this, the concept of time-varying graphs has been introduced, allowing changes in the graph over time. In this framework, both edges and vertices can evolve, with elements appearing or disappearing throughout time.

Some models utilize this idea to generate time-varying graphs. For example, the Barabási-Albert model generates a scale-free graph through an iterative process. Other models focus directly on the time-varying graph itself, exploring properties introduced by the addition of time and how these properties differ from those of static graphs. Even simple concepts like paths become richer in time-varying graphs, raising new questions that do not arise in static graph contexts.

One specific aspect that captured our interest in the study of these graphs is the generation of time-varying graphs. There are several approaches to generating such graphs. Some methods focus on the structure of the graph itself, whether static or time-varying. Examples include the Watts-Strogatz model, Microcanonical Randomized Reference Models (MRRM), and the model presented in Section 2.3.3.2. Other methods consider a time-varying graph as a flow of static graphs, where each graph is generated based on the previous one. The Barabási-Albert model, its derivatives, and the Generator of Edge-Markovian Graphs (EMGG) are examples of this approach. For instance, the Barabási-Albert model aims to construct a scale-free graph by continually adding vertices, while the EMGG keeps the vertex set constant and its dynamics quickly converges to a stationary distribution.

Our work concentrates on the dynamics of graph generation, particularly the processes that generate sequences of graphs based on previously generated ones. We observed several limitations in existing models. First, the graph's order (i.e., the number of vertices) is typically fixed or grows steadily, determined by the model's parameters rather than emerging naturally from the process. Second, while some metrics exist for time-varying graphs, they tend to describe the graph itself, not the generative process. For example, time-connectivity is a property of the graph, but it doesn't quantify the changes the process imposes on the graph.

We aim to study the dynamics of graphs obtained by generative processes, where features like graph order and edge count evolve without being driven by fixed parameters. We aim to study the dynamics of graphs obtained by generative processes where features like graph order and edge count evolve without being driven by fixed parameters. For that purpose, two new metrics are introduced to quantify and qualify the generative processes themselves. These ideas lead us to define the concept of *dynamic graphs* produced by generative processes.

2.5 Formalizing the Concept of Dynamic Graphs

The concept of a *dynamic graph* is introduced to account for the existence of an underlying process that generates the graph. Unlike a time-varying graph, where two consecutive graphs in the sequence (see Definition 7) may be independent from an evolutionary perspective, in a dynamic graph, two consecutive static graphs are linked by an underlying generative process, making the study of its dynamics relevant.

This paradigm is inspired by the study of dynamical systems, which describe the evolution of a system's state (represented as a state vector $x(t)$) through differential equations $\frac{dx(t)}{dt}$. A similar approach, inspired by statistical physics, is found in Strogatz's work on such systems [41].

The key idea here is to focus on the underlying "story" driving the evolution of the graph. In our study, *dynamic* refers to the rules governing the transition from one static graph to the next. Repeated application of these rules generates a sequence of static graphs, forming what we call a dynamic graph. The number of times these rules are applied may be finite or infinite, depending on the application. This

section concentrates specifically on the transition function and aims to provide a formal representation of *dynamic* and use this formalism to define the notion of a dynamic graph.

The process is represented as a function f_d , analogous to how systems of differential equations are studied. Applying f_d to a graph G_t generates a new graph G_{t+1} , such that $G_{t+1} = f_d(G_t)$. There are no constraints on the nature of this function; it can be either deterministic or stochastic. The formal definition of a *dynamic* is given as follows:

Definition 13 *Dynamic:*

A *dynamic* is a function $f_d : G = (V, E) \mapsto G' = (V', E')$, where G and G' are graphs.

This framework offers a flexible perspective for exploring iterative generative processes, allowing for various scenarios without restricting features like graph order. Under this definition, it is possible for $|V'| \neq |V|$ or $|V'| = |V|$ with $V \neq V'$. While this definition uses a single graph as input, it could be extended to functions that take multiple graphs as inputs, without affecting the explanations provided here.

A *dynamic graph* in the context of this work, is a sequence of graphs starting from an initial condition G_0 , and using a dynamic function f_d to generate new graphs from previously produced ones. The number of graphs in the sequence may be a parameter. Dynamic graphs are formally defined as follows:

Definition 14 *Dynamic Graph:*

Let $\mathcal{T} = [0, T]$ or \mathbb{N} be a set of non-negative integers, G_0 a static graph, and f_d a dynamic. Then, the time-varying graph $\mathcal{G} = (G_t)_{t \in \mathcal{T}}$, where $G_{t+1} = f_d(G_t)$ for all $t \in \mathbb{N}$ if $\mathcal{T} = \mathbb{N}$, or $G_{t+1} = f_d(G_t)$ for all $t \in [0, T - 1]$ otherwise, is a **dynamic graph**.

This definition emphasizes the importance of a mechanism to produce graphs. Instead of generating a time-varying graph in a single step, the generative process is iterative, building new graphs from already generated ones. Examples of generative processes producing dynamic graphs include the Barabási-Albert model and its derivatives, as well as the Generator of Edge-Markovian Graphs, both of which provide iterative rules for generating sequences of graphs.

Conclusion

The concept of graphs is a powerful tool for representing and studying complex networks, helping to understand phenomena in real-world systems. Some models, like the Watts-Strogatz model, utilize static graphs, while others, such as the Barabási-Albert model, incorporate time to transform an initial seed graph G_0 into a sequence of graphs. However, these models typically focus on a single limiting graph to assess properties such as small-world phenomena or scale-freeness.

In contrast, recent studies emphasize time-varying graphs, which examine entire sequences rather than individual static instances. For example, time-connectivity explores whether every vertex can find a temporal path to all others, introducing challenges not present in static graph theory.

A fundamental aspect of time-varying graphs is their generation through processes that create a flow of graphs, termed dynamic graphs. These are defined by functions that transform one graph into another using specific rules. This dynamic framework, inspired by statistical physics and differential equations, allows graph characteristics to evolve as a result of the generative process.

The following chapters will explore the consequences and properties of these generative processes, focusing on metrics that assess graph dynamics rather than static features. Additionally, we will examine a specific example of a generative process in Chapter 4.

Qualifying and Quantifying Graph Dynamics

This chapter focuses on the study of generative processes for dynamic graphs, a concept defined in the previous chapter in Section 2.5. As mentioned earlier, several questions arise when these generative processes are allowed to modify the set of vertices without any fixed constraints. For example, the set of vertices may change over time, and its size may depend on the rules applied. This opens up several possible scenarios. For instance, the sequence of graphs might converge, toward an empty graph, toward a constant static graph ($G_{t+1} = G_t$), or might change for ever. This raises questions about the evolution of the vertex and edge sets: is it sufficient to study the number of vertices and edges, or do we need more sophisticated metrics to fully capture the dynamics? This question leads to an investigation of changes in the composition of vertices and edges and the methods for measuring them.

To address these challenges, this chapter introduces several key concepts. The first, called **sustainability**, assesses the overall progression of a sequence of graphs. This concept forms a foundation for analyzing dynamic graph generators. A graph is considered **sustainable** if its dynamics do not eventually loop or stagnate. A formal definition of sustainability is provided in Section 3.1.

The second concept introduced is a metric designed to quantify the dynamic changes in a graph. This metric, called **DynamicScore**, captures the global changes between consecutive graphs. Together with the analysis of vertex and edge count evolution, DynamicScore offers a deeper understanding of time-varying graph dynamics. This metric is further discussed in Section 3.2.

Lastly, an additional question, expanding on the idea of sustainability and generalizing the concept of **journeys** in time-varying graphs, is explored in Section 3.3. This section examines communication in dynamic graphs where the set of vertices evolves over time. Two problems are introduced in this context, focusing on whether information can persist despite the loss of vertices over time.

3.1 Sustainability

The primary aim of this research is to define the concept of "sustainability" within the context of dynamic graph generators. This study focuses on generators that evolve the entire set of vertices and those based on dynamic mechanisms. This concept is one of the contributions of this thesis. Indeed, to the best of our knowledge, it has not been previously explored, which necessitates the development of an appropriate definition.

Our investigation centers on three specific scenarios related to dynamic graph generators, discussed in Section 3.1.1. These scenarios outline the conditions that a dynamic graph or its generator must satisfy to be considered sustainable. Additionally, we reflect on this new metric, recognizing that it may not be inherently intuitive.

Further, in Section 3.1.2, we study sustainability on two well-known models: the Barabási-Albert model and the Generator of Edge-Markovian Graphs.

3.1.1 Definition

The sustainability metric is a novel tool introduced to study generative processes. This section outlines the development of the sustainability concept and the underlying choices that shaped its definition. Three predicates were formulated to define the sustainability of dynamic graph generators.

First, the generators must ensure persistence by avoiding the production of graphs that eventually become empty. Second, the generators should exhibit non-periodicity; initially, this condition aimed to prevent graphs from becoming static over time, but it was later generalized to include deterministic generators. The intent here is to avoid the creation of graphs that lack "dynamics" or that become repetitively predictable. Third, we sought to control growth by preventing the generation of graphs that expand exponentially, instead promoting a balanced growth that allows for elements of decay or reduction.

However, in our contributions, only the first two predicates were retained. The third predicate, concerning growth control, proved difficult to quantify and was deemed more dependent on specific applications. The two predicates used to define the sustainability notion are summarized in the following definition:

Definition 15 *Sustainability Predicates:*

Let $\mathcal{G} = (G_t)_{t \in \mathcal{T}}$. \mathcal{G} is said to be sustainable if it satisfies the following two predicates:

$$P_0 : \nexists T \in \mathcal{T}, \forall t \geq T, G_t = (\emptyset, \emptyset)$$

$$P_1 : \nexists T \in \mathcal{T}, \exists k \in \mathbb{Z}^+, \forall t \geq T, G_{t+k} = G_t$$

3.1.2 Comparison to Existing Generative Processes

Examples of existing generative models can be found in Chapter 2. In this section, we apply the definition of sustainability to evaluate whether these generative models ensure the sustainability of the graphs they produce.

3.1.2.1 Barabási-Albert Model

Some well-known generative processes, such as the Barabási-Albert model, produce sustainable dynamic graphs. In these models, both the graph order and the number of edges are controlled: for all $t \in \mathbb{N}$, $|V_{t+1}| > |V_t|$, and $G_t \neq (\emptyset, \emptyset)$. Therefore, these graphs satisfy the two criteria required to be classified as sustainable.

For these generators, graph sustainability is evident and is inherently defined by the model itself, thus requiring no further analysis of sustainability. However, some may argue that since the process grows by adding one vertex and a fixed number of edges, the changes in composition become insignificant as the size of the graphs increases. In Section 3.2.3.3, it is even demonstrated that a metric quantifying the dynamics of time-varying graphs for both vertices and edges approaches zero over time.

Thus, even though the generated graphs are sustainable, one might contend that they become non-sustainable as time approaches infinity. This observation suggests that the notion of sustainability could be further refined by using other metrics quantifying the dynamics of time-varying graphs.

3.1.2.2 Edge Markovian Graphs

The question of sustainability may also be studied in the case of Edge-Markovian Graphs (EMG). Indeed, here the model is parameterized with two probabilities p and q . The question of sustainability may be treated to see whether or not there exist parameters such that produced graphs are always sustainable or not. To this end, we first expose some well-known result about EMG and then provide an analysis of the density evolution. This analysis is then used to state about sustainability. In the following, the number of edges in a generated graph at step t will be referred to as m_t and the graph density will be referred to as \hat{m}_t .

Known Properties of EMGG In order to ease the understanding of the dynamics of EMGG instances, some results about EMGG are presented. First note that the state of each edge is independent of the state of the other edges of the graph, thus, studying the probability of presence/absence of each edge independently from the others is correct. As presented in [11] the transition matrix P for a single edge satisfies:

$$P = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix} \quad (3.1)$$

The analysis of Markovian processes and more especially the study of two-states markovian processes has shown that for each single edge, the distribution of presence, in the context of EMGG, converges toward a stationary distribution π as long as $|p+q-1| \neq 1$ (see [15] for a proof). The situation $|p+q-1| = 1$ is



Figure 3.1: The two different configurations where $|p + q - 1| = 1$. On the left, $p = q = 1$. This means whatever the initial configuration, the edge will stay in the same state. On the right, $p = q = 0$. This means the edge state changes at every step.

discussed below. As a stationary distribution of a Markov chain, π satisfies $\pi = \pi P$. The value of vector π is stated in the following theorem:

Theorem 1 *Stationary distribution:*

For p, q probabilities such that $|p + q - 1| \neq 1$, the stationary distribution π is $\left(\frac{1-q}{2-p-q} \quad \frac{1-p}{2-p-q} \right)$.

Proof: It is sufficient to notice that $\pi = \left(\frac{1-q}{2-p-q} \quad \frac{1-p}{2-p-q} \right)$ is a distribution and that $\pi = \pi P$. ■

Thus, the presence of an edge has a Bernoulli distribution of parameter $\pi^* = \frac{1-p}{2-p-q}$ as a stationary distribution. As every edge is independent one from the other, the number of edges has a binomial distribution of parameters $\binom{n}{2}$ and π^* . The situation for which $|p + q - 1| = 1$ can be declined in two sub-cases: either $p = q = 1$ or $p = q = 0$. On the one hand if $p = q = 1$, then graphs produced by EMGG remains unchanged forever. This means $G_t = G_0$ for all t . In this configuration, the density is constant and equal to the density of the very first graph of the sequence of snapshot. On the other hand if $p = q = 0$, then produced graphs are 2-periodic and more precisely, E_{t+1} is the complementary of E_t for all t . In this configuration, the density switches at every time step. More precisely, $\hat{m}_t = \hat{m}_0$ for t even and $\hat{m}_t = 1 - \hat{m}_0$ for t odd. Thus, in both cases the density of a produced graph does not converge to a stationary distribution. These configurations are gathered in the figure 3.1. For all other cases, the eigenvalue $p + q - 1$ provides the speed of convergence toward the stationary distribution. This convergence rate is an exponential in $|p + q - 1|$. A convenient way to observe this is to consider the matrix P in its diagonal form:

$$P = U \begin{pmatrix} 1 & 0 \\ 0 & p + q - 1 \end{pmatrix} U^{-1}$$

Then, it is sufficient to notice that for any initial distribution $\pi^{(0)}$, the distribution at step t $\pi^{(t)}$ is given by

$$\pi^{(t)} = \pi^{(0)} P^t$$

Density Evolution of Edge-Markovian Graphs In order to better understand the relationships between EMGG dynamics and the Edge-DynamicScore metric, that will be presented in the next section, we first show that the number of edges is on average close to a quantity depending only on p and q . To that end, we prove the following lemma on the evolution of the density:

Lemma 1 *Evolution of the Density*

Let consider EMGG parameterized by (n, p, q) . Let (G_0, \dots, G_t) be a sequence of graphs produced by

EMGG. Then, the expected normalized density for the graph G_{t+1} satisfies the following equation:

$$m_{t+1} \simeq f_{p,q}(\hat{m}_t) = \hat{m}_t p + (1 - q)(1 - \hat{m}_t) = (p + q - 1)\hat{m}_t + (1 - q) \quad (3.2)$$

Proof: as the process is a Markov chain, m_{t+1} depends only on \hat{m}_t . Second, it is worth mentioning that every edge is independent from the others. The expected number of edges that remain present is $p\hat{m}_t$ while the expected number of edges changing their state from absent to present is $(1 - q)(1 - \hat{m}_t)$. The expected number of edges present at step $t + 1$ is thus the sum of these two quantities. ■

This lemma provides a valuable interpretation of the density expectation, which allows us to further investigate the existence of a fixed density. By analyzing the expectation, we can identify a specific value that represents a fixed point within the computation process. In the context of the function $f_{p,q}$, a fixed point refers to a value m^* for which $f(m^*) = m^*$ holds true. The computation of this fixed point value is carried out according to the procedure outlined in the subsequent lemma.

Lemma 2 Expected Number of Edges:

Let G be a graph produced by EMG(n, p, q) Let \hat{m}_t be the density of graph at step t . Then, as long as $|p + q - 1| < 1$ an expectation value for \hat{m}_t , referred to as m^* , satisfies $f_{p,q}(m^*) = m^*$:

$$m^* = \frac{1 - q}{2 - p - q} \quad (3.3)$$

Proof: this result comes from finding a fixed point to the function $f_{p,q}$ ■

This fixed point value matches with the probability of presence of an active edge in the stationary regime. It is not surprising as it gives, in both case, the average and expected value of the graph density. These findings enable us to gain deeper insights into the dynamics of the system and the properties associated with the EMGG, paving the way for a more comprehensive understanding of its behavior.

Sustainability As a consequence of the above result, it may be shown that, if p and q are both different from 0 and 1, then graphs are likely to be sustainable. This may be proved using two arguments. First, if p and q are both different from 0 and 1, then there always a chance for one edge to transit from one state to another. Moreover, the estimation provided in Lemma 2, shows that, if p and q are both different from 0 and 1, the expected density of the graph is neither 0 nor 1, which means the graph is neither empty nor complete. Now the question is, what is happening if p or q equals 0 or 1. Let us discuss each case. First, if $p = q = 1$, then every edge stay in its state. Therefore graphs remains steady and are therefore not sustainable. If, $p = q = 0$, then every edge switch state at every step. Produced graphs are periodic with a period 2 and are therefore not sustainable. If $p = 0$ and $q = 1$, then after one step, produced graphs are empty and therefore not sustainable. If $p = 1$ and $q = 0$, then after one step, produced graphs are complete graphs and therefore not sustainable. If p or q equals 1 while the other is different from 0 and 1, then if $p = 1$ then graphs will become empty with high probability as every edge stay present from the first moment they are present. The same thing occurs when $q = 1$, but graphs will rather become empty

for similar reasons. However, if p or q equals 0 while the other is different from 0 and 1, then graphs are likely to be sustainable. Indeed, if $p = 0$ for instance, then once an edge appear, it disappear at the very next step but if an edge is absent, then it may stay absent with probability q and become present with probability $1 - q$. It is therefore unlikely for a graph to become periodic. The same reason may be applied for the case $q = 0$. All these results may be gathered in the following theorem:

Theorem 2 *Sustainability Conditions:*

The Edge-Markovian Graph Generative process produces sustainable graphs if and only if p and $q \in]0, 1[$ or p or q equals 0 while the other is different from 0 and 1.

As can be seen, the sustainability conditions are simple to state, but verifying them for a given generator is not straightforward.

Conclusion on the Sustainability

Unlike these cases, some generators are based on mechanisms making the evolution of the vertices (and edges) more difficult to predict, and the dynamics is worth studying. This is the case of the generative models proposed in Chapter 4. The sustainability metric has been designed to study such parameterized generative models producing dynamic graphs. Indeed, before studying complex properties about dynamic graphs, it is worth making sure these graphs do not become empty for instance. This is why, the sustainability metric is a first milestone in the study of dynamic graphs when their set of vertices/edges changes over time. Once the sustainability is ensured or almost certain for some families of dynamic graphs, the next step in the study of sustainability is to look at the reasons a dynamic graph is sustainable. In the context of this thesis, we have found some families of graphs seeming to exhibit a structure maintaining the whole graph sustainable while others just produce graphs without any visible structure. Example of graphs exhibiting a structure maintaining their sustainability are investigated in Section 4.3.4.

3.2 DynamicScore

In this section, we introduce a complementary metric for studying the evolution of graph order. This metric is called DynamicScore. It is a quantitative metric that provides a global perspective on the changes that can occur within the graph over a time step. It derives from an existing metric known as the Jaccard distance, named after the author who first used it in his work [25]. Initially utilized in the field of biology, it also plays a key role in set theory, as it corresponds to a distance in the topological sense. Furthermore, it provides a measure of dissimilarity between two subsets of a larger set. This means that this metric is a distance, taking values between 0 and 1, where a value of 1 indicates that the two input sets do not share any common elements.

3.2.1 Definition

The DynamicScore is a novel metric aiming at quantifying the dynamic of a graph. The motivation lurking underneath this metric is to answer the question "how dynamic a graph is?". It is worth mentioning this question is not straightforward. When we first attempted to define a measure of the dynamics of a

graph we were tempted to say a graph is slowly dynamic when its size remains nearly the same during a short number of steps. However, a graph can have very little change in size and yet the entire set may have evolved. Indeed, maintaining a constant size does not prevent various scenarios from occurring. For instance, the set of vertices may remain steady, or it may have completely changed. In the first case, it seems relevant to qualify the dynamics as low. In the second case, the graph may be qualified as a highly dynamic graph. The metric should emphasize this situation by providing a low (resp high) score indicating the graph has not changed too much (resp has changed a lot). Therefore, the objective was to introduce a metric that accounts for all these scenarios and enables the quantification of graph dynamics, and in some cases, characterizes certain phenomena.

The definition of the DynamicScore introduced in this section encompasses every aspect we want a metric quantifying the dynamics of graphs to embed. The DynamicScore is an answer to this question. It effectively measures the similarity between two consecutive graphs in a graph sequence. The metric is a distance based on the Jaccard distance d_J of two sets. Notably, the DynamicScore emphasizes changes in composition, both at a local level over time between two consecutive steps, and at a global level spanning the entire graph. It is formally defined as follows:

Definition 16 *Vertices-DynamicScore:*

Given a dynamic graph G , such that at time t $G_t = (V_t, E_t)$. We call **Vertices-DynamicScore** (also referred to as *V-DynamicScore* or *Vertex DynamicScore* in the following) at time t and denoted by \mathcal{D}_t^V , the ratio:

$$\mathcal{D}_t^V = \frac{|V_{t+1} \Delta V_t|}{|V_{t+1} \cup V_t|}$$

where $|A|$ denotes the size of set A . The Δ operator for any two sets A and B referred to as $A \Delta B$ and is defined as $A \Delta B = (A \cup B) - (A \cap B)$.

Similarly, for a given dynamic graph the definition of its edges DynamicScore is defined as follow:

Definition 17 *Edges-DynamicScore:*

Given a dynamic graph G , such that at time t $G_t = (V_t, E_t)$. We call **Edges-DynamicScore** (also referred to as *E-DynamicScore* or *Edge DynamicScore* in the following) at time t and denoted by \mathcal{D}_t^E , the ratio:

$$\mathcal{D}_t^E = \frac{|E_{t+1} \Delta E_t|}{|E_{t+1} \cup E_t|}$$

One interesting thing with these metrics is that they do not require anything but the set of vertices and the set of edges to be defined. The DynamicScores serve as dissimilarity measures, enabling comparisons between two consecutive snapshot graphs. A score of 0 indicates that the two sets are identical, while a score of 1 means that they do not share any common elements. In general, a value close to 0 suggests minimal changes in the graph between two consecutive steps, whereas a value close to 1 implies significant modifications have occurred. It should be noted that graph order and the V-DynamicScore measure two different quantities. For instance, between two consecutive time steps, t and $t + 1$, the value \mathcal{D}_t^V can range from 0 to 1 while the order of the graph remains the same. The following section investigate further

these observations. **Remark:** to avoid the problem of the division by zero and as the DynamicScore is a distance, \mathcal{D}_t^V will be set to zero if both V_t and V_{t+1} are empty.

3.2.2 Specific Values of DynamicScore

For illustrating these definitions, we will consider different cases for a dynamic graph, from t to $t + 1$. In a first time we will see how V-DynamicScore may be used as a complementary metric to graph order. We will then provide few results when dynamic graphs grow and see different scenarios.

3.2.2.1 Constant Graph Order

Let assume that between t and $t + 1$ the order remains the same, thus $|V_{t+1}| = |V_t| = n$. Here the focus will be on Vertices-DynamicScore, but observations are equivalent in the case of edges. To illustrate the expressiveness of this metric, we analyze three different cases.

Case 1: if all vertices are replaced, then, on the one hand, graphs do not share any common vertex. The composition of the dynamic graph has thus completely changed in one step. On the other hand, $\mathcal{D}_t^V = 1$. Thus, the DynamicScore, by providing a value of 1, perfectly highlights the dramatic change of the vertex set, during one step.

Case 2: if the set of vertices remains the same, the graph composition does not change in one step ($V_t = V_{t+1}$). Moreover, $\mathcal{D}_t^V = 0$. As in the previous case, the absence of changes is encompassed in DynamicScore values. This may be extended to graphs with small changes. In that case, DynamicScore values are close to 0.

Case 3: The last case is when half of the vertices are replaced. In that case, the intersection of the two consecutive sets $V_t \cap V_{t+1}$, the set of created vertices $V_{t+1} - V_t$ and the set of removed vertices $V_t - V_{t+1}$ contains $\frac{n}{2}$ elements. Then $\mathcal{D}_t^V = \frac{2}{3}$.

This is summarized in the diagram 3.2.

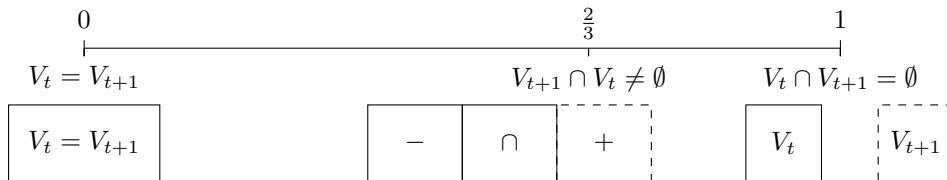


Figure 3.2: Illustration of the dynamic score for vertices. Here n_{t+1} is assumed to equal n_t . For the case $\mathcal{D}_t^V = \frac{2}{3}$, $-$ refers to as $V_t - V_{t+1}$, the set of non-conserved vertices. \cap refers to as $V_t \cap V_{t+1}$, the set of conserved vertices. Finally, $+$ refers to as $V_{t+1} - V_t$, the set of created vertices.

This shows that DynamicScore is a complementary metric with the evolution of graph order. Indeed, when the graph order does not change, the set of vertices as well as the set of edges may change. The DynamicScore takes into account such changes as it provides information about changes in the

composition of a dynamic graph between consecutive steps. It is even possible to go further and to find a connection between the number of conserved vertices ($|V_t \cap V_{t+1}|$) and the V-DynamicScore:

Theorem 3 V-DynamicScore and Conserved Vertices:

Let $\mathcal{G} = (G_t)_{t \in \mathbb{N}}$ be a dynamic graph. Assume that $|V_t| = |V_{t+1}| = n$ for some $t \in \mathbb{N}$. Define $\alpha = \frac{|V_t \cap V_{t+1}|}{n}$ as the proportion of conserved vertices from step t to step $t + 1$. Then, the V-DynamicScore and α are connected as follows:

$$\mathcal{D}_t^V = 2 \frac{1 - \alpha}{2 - \alpha}$$

Proof: Let $\alpha = \frac{|V_t \cap V_{t+1}|}{n}$. Then, using the definition of the V-DynamicScore and the assumption that $n = |V_t| = |V_{t+1}|$, the following holds:

$$\mathcal{D}_t^V = \frac{|V_t \Delta V_{t+1}|}{|V_t \cup V_{t+1}|} = \frac{|V_t| + |V_{t+1}| - 2\alpha n}{|V_t| + |V_{t+1}| - \alpha n} = 2 \frac{1 - \alpha}{2 - \alpha}$$

■

One thing to notice with this relationship is that \mathcal{D}_t^V ranges from 0 to 1 as α does. The second thing to notice is that the V-DynamicScore is a decreasing function of α . With all these arguments, it is clear that DynamicScores are complementary metrics of the graph order or the number of edges as when they do not change, DynamicScores spotlights composition changes with values ranging from 0 (no changes) to 1 (every thing has changed).

3.2.2.2 Growing Network

Let us now focus on a growing dynamic graph, meaning at each step t , $V_t \subseteq V_{t+1}$ and $E_t \subseteq E_{t+1}$. Here we will focus on three different situations. The first one is when n_t and e_t grows linearly. In that case, it is possible to show that both V-DynamicScore and E-DynamicScore tends toward 0 as t tends to infinity. The second situation is when n_t grows as an exponential ($n_t = n_0 \times a^t$). For that case, it is possible to find a relationship between \mathcal{D}_t^V and a . The third situation does not concern a specific grow but when e_t may be expressed as a polynomial of n_t . In that case, it is possible to exhibit a relationship between \mathcal{D}_t^E and \mathcal{D}_t^V . Before delving into these cases, let us proof the following lemma that will be helpful for proofs:

Lemma 3 General V-DynamicScore of Growing Graph Let assume $V_t \subseteq V_{t+1}$ for all $t \in \mathbb{N}$, then:

$$\mathcal{D}_t^V = 1 - \frac{n_t}{n_{t+1}}$$

Proof: From the assumption $V_t \subseteq V_{t+1}$, it comes $|V_t \Delta V_{t+1}| = n_{t+1} - n_t$ and $|V_t \cup V_{t+1}| = n_{t+1}$. Thus, the following holds:

$$\mathcal{D}_t^V = \frac{n_{t+1} - n_t}{n_{t+1}} = 1 - \frac{n_t}{n_{t+1}}$$

■

Let us now delve into the different cases.

Linear Grow Let us assume that $n_t = a \times t + n_0$ and $e_t = b \times t + e_0$. It is possible to prove that in this configuration, V-DynamicScore and E-DynamicScore converges toward 0.

Theorem 4 *Linear Growth DynamicScores:*

Let us assume that $n_t = a \times t + n_0$ and $e_t = b \times t + e_0$. Let assume $\mathcal{T} = \mathbb{N}$. Then, $\lim_{t \rightarrow \infty} \mathcal{D}_t^V = \lim_{t \rightarrow \infty} \mathcal{D}_t^E = 0$.

Proof: The result come from $\lim_{t \rightarrow \infty} \frac{n_t}{n_{t+1}} = \lim_{t \rightarrow \infty} \frac{e_t}{e_{t+1}} = 1$ and from Lemma 3. ■

This theorem, alongside with both V-DynamicScore and E-DynamicScore are decreasing function of t , proves that the dynamics of linear growing graphs tends to zero and therefore graphs are less dynamic as time grows. This result may be extended to polynomial growth as, for any polynomial function $P(t)$, the ratio $\frac{P(t)}{P(t+1)}$ tends to 1 as t tends to infinity.

Exponential Growth For this paragraph, let us assume the size of the graph grows as an exponential of the form $n_t = n_0 \times a^t$ and $e_t = e_0 \times b^t$ with $a \geq 1$ and $b \geq 1$. In that case, the V-DynamicScore does not tend toward zero but remains constant over time. The value of this constant depends on the value of a and the relationship between them is stated in the following theorem.

Theorem 5 *Exponential Growth DynamicScore:*

Let us assume that $n_t = n_0 \times a^t$ and $e_t = e_0 \times b^t$. Then, for all $t \in \mathcal{T}$, then $\mathcal{D}_t^V = 1 - \frac{1}{a}$ and $\mathcal{D}_t^E = 1 - \frac{1}{b}$.

Proof: This comes from Lemma 3 and $\frac{n_t}{n_{t+1}} = \frac{1}{a}$. ■

As opposed to graph size increasing as a polynomial of time, here the DynamicScore remains constant over time. Moreover, the bigger a and b , the closer \mathcal{D}_t^V and \mathcal{D}_t^E are to 1.

Graph Densification Let us now delve into the case where graphs become denser. In this context, it is still assumed that graphs are growing (i.e., $V_t \subset V_{t+1}$), but here a polynomial relation exists between the graph order and the number of edge : $e_t = c \times (n_t)^a$ with $a \in [1, 2]$. It is therefore possible to find a relationship between V-DynamicScore and E-DynamicScore as follows:

Theorem 6 *Graph Densification DynamicScore:* Let assume there exist $c > 0$ and $a \in [1, 2]$ such that $e_t = c(n_t)^a$ for all t . Then the following holds:

$$\mathcal{D}_t^E = 1 - (1 - \mathcal{D}_t^V)^a$$

Proof: Using Lemma 3 and the densification relationship between e_t and n_t , \mathcal{D}_t^E may be written as follows:

$$\mathcal{D}_t^E = 1 - \frac{e_t}{e_{t+1}} = 1 - \left(\frac{n_t}{n_{t+1}} \right)^a = 1 - \left(1 - \left(1 - \frac{n_t}{n_{t+1}} \right) \right)^a = 1 - (1 - \mathcal{D}_t^V)^a$$

■

From this it is possible to check whether a growing graph gets denser over time using this relationship. However, this relationship only holds if the graph is growing. If this is not the case, another method should be used to see if the graph gets denser over time.

3.2.3 Comparison to Generator Models

From now we will use this metric in order to describe the dynamic of graphs produced by generative processes. In a first time, the focus will be on network with neither removal nor addition of vertices, so that the focus will be only on edges dynamics. To illustrate this, a model has been designed based on Erdős-Rényi random graph model. It will be proven that considering this model it is possible to analytically estimate the DynamicScore distribution and that, assuming a few hypotheses, the DynamicScore is roughly constant. Then, the EMGG, presented in Section 2.3.3.1, is studied. For this model, we will prove that the average value of the E-DynamicScore is depending only on one parameter. In a third time, the focus will be on growing networks. They are such that a new vertex is added at every step of the generative process using preferential attachment. Such a mechanism has been described more formally above so this section will only deal with properties of the DynamicScore. It will be shown that Vertices-DynamicScore, as well as Edges-DynamicScore, converge toward 0, meaning graph dynamics becomes small.

3.2.3.1 Sequence of Erdos-Renyi Graphs

As a first model, we here investigate a model based on the Erdős-Rényi (ER) [16] random graph model. More precisely, the generator produces a sequence of graphs such that every snapshot graph is an ER graph with n vertices as a probability of connection p . The motivation behind this choice is that the mechanism involved in the generative process is simple to describe. Moreover, the analysis of the distribution of the DynamicScore is enabled in this paradigm, which is not always the case. Such an analysis will be a first milestone in the study of DynamicScore, and comparison with other models will enlighten features and properties of other models as we will see in other sections. Defining formally such a generative process leads to the following definition:

Definition 18 *Erdős-Rényi Dynamic Graph Generator (ERDGG):*

Let $n \in \mathbb{Z}^+$ and $p \in [0, 1]$. A graph produced by ERDGG(n, p) is a sequence of independent random graphs generating using the mechanism introduced in the ER model [16].

The sequence of every graph generated using this generation mechanism is said to be a "time-independent" Markov chain, meaning that it is a Markov chain with no dependencies between two consecutive steps. Various properties may be found about static graphs obtained from the initial ER model. Among others, the distribution of the number of edges and is known and is such that, for every graph $G = (V, E)$ obtained from ER(n, p) generation paradigm, $|E|$ has a binomial distribution $|E| \sim \mathcal{B}\left(\binom{n}{2}, p\right)$. This result is obtained reasoning on the independence between every single edge of which the distribution of presence is of course a Bernoulli scheme of parameter p . This is sufficient to prove the following result, providing an approximation for the distribution of the DynamicScore in the general case. The approximation just mentioned concerns one argument in the proof. Indeed, the proof relies on the independence on the number of removed or created edges $|E_t \Delta E_{t+1}|$ and the number of edges present at least once $|E_t \cup E_{t+1}|$, which does not hold as the intersection is included in the union, but will provide results obtained from simulation and help the understanding of the dynamics of the generator.

Theorem 7 *Let n and p be as defined above. Let $t \geq 0$, and $G_t = (V, E_t)$ and $G_{t+1} = (V, E_{t+1})$ two consecutive generated graphs. Let \mathcal{D}_t^E denote the E -DynamicScore as defined above. Then, \mathcal{D}_t^E can be approximated by a lognormal distribution with parameters μ and σ^2 satisfying:*

$$\begin{aligned}\mu &= \ln \left(2 \frac{1-p}{2-p} \right) \\ \sigma^2 &= \frac{4 - 11p + 12p^2 - 4p^3}{2 \binom{n}{2} p(1-p)(2-p)}\end{aligned}$$

Proof: First, let consider the distribution of $|E_t \Delta E_{t+1}|$ and the distribution of $|E_t \cup E_{t+1}|$. To establish the distribution of $|E_t \Delta E_{t+1}|$, the following reasoning on a single edge $e \in V^2$ is enough:

$$\begin{aligned}\Pr(e \in E_t \Delta E_{t+1}) &= \Pr((e \in E_t \cap e \notin E_{t+1})) + \Pr[(e \in E_{t+1} \cap e \notin E_t)] \\ &= p(1-p) + (1-p)p \\ &= 2p(1-p)\end{aligned}$$

Thus, as every edges are independent one another, $|E_t \Delta E_{t+1}| \sim \mathcal{B}(\binom{n}{2}, 2p(1-p))$. Following a similar way, the distribution of $|E_t \cup E_{t+1}| \sim \mathcal{B}(\binom{n}{2}, p(2-p))$. Using the assumption both $|E_t \Delta E_{t+1}|$ and $|E_t \cup E_{t+1}|$ are independent provides that $\ln(\mathcal{D}_t^E) \sim \mathcal{N}(\mu, \sigma^2)$, with $\mu = \ln\left(2 \frac{1-p}{2-p}\right)$ and $\sigma^2 = \frac{1}{\binom{n}{2}} \left(\frac{1}{2p(1-p)} + \frac{1}{p(2-p)} - 2 \right)$ (σ^2 lies on the value in the statement). Therefore $\mathcal{D}_t^E = \exp(\ln(\mathcal{D}_t^E)) \sim \text{Lognormal}(\mu, \sigma^2)$. ■

Two main observations may be derived from this statement. The expectation of a lognormal distribution parameterized using μ and σ^2 is $\exp\left(\mu + \frac{\sigma^2}{2}\right)$ (it is worth mentioning that the parameters of X having a lognormal distribution are the same parameters for $\exp(X)$ having a normal distribution). However knowing only the mean value of a distribution is not enough. One would also require the coefficient of variation CV, which measures the dispersion of the values a random variable may have around its expected value. For a lognormal distribution, this value is $\sqrt{e^{\sigma^2} - 1}$. Providing σ^2 is small, Taylor expansion of this expression leads to the approximation $\sqrt{e^{\sigma^2} - 1} \simeq \sigma$. As σ is small for huge values of n , the dispersion of DynamicScore values around μ will be very small. Therefore, the mean value of the DynamicScore can be considered equal to the value that the DynamicScore takes without too much variability. In addition, the expected value may be considered close to μ as $\frac{\sigma^2}{2}$ is close to zero.

The application of DynamicScore to this generative model gives us more insights about the dynamics of produced graphs. Indeed, the definition of the generator implies that two consecutive snapshot graphs are completely independent one another. The generative process is therefore a "Time-Homogeneous" Markov Chain. Thus, there is no reason to think there is any common information between two consecutive steps. However, it has just been shown that DynamicScore is almost constant. As this metric represents the renewal rate of edges between consecutive steps, it is possible to use it to compute the conservation rate, which will also be almost constant. Moreover, the average DynamicScore exhibits a decreasing trend on p . It is sufficient to notice that $\frac{1-p}{2-p} = 1 - \frac{1}{2-p}$ is a decreasing function of p . Possible values for the

DynamicScore ranges from 0 to 1, indicating its ability to capture all sort of changes in the graph. In addition to this, the function $p \mapsto 2\frac{1-p}{2-p}$ is an involution (it is its own reverse), therefore, it is possible to produce generated graphs with a chosen DynamicScore: for a given DynamicScore D , parameter p must equal $2\frac{1-D}{2-D}$ for the average DynamicScore to equal D . As we see, it is possible to design the generator to produce graphs with a fixed dynamics.

3.2.3.2 Edge-Markovian Graphs

The results presented in this section have been published and presented during the 12th International Conference on Complex Networks and their Applications [1].

This section explores the relationship between the Edge-Markovian Graphs Generator (EMGG) and the DynamicScore, focusing on the computation of an expectation value regardless of the graph's density. This model has already be presented and defined in Section 2.3.3.1. This is an extension of the sequence of Erdos-Rényi graph generator which can be seen as a special case where $p = 1 - q$. The main result of this work is that the average DynamicScore does not depend on the parameter q but only on p . Additionally, similar to the previous dynamic graph generator, we prove that the average DynamicScore can be equal to any possible values between 0 and 1. These findings can be explained by the generative process being a Markov chain. A last important point to emphasize is that in this section, only the DynamicScore of edges will be studied, since the set of vertices remains constant.

The following theorem provides a precise value of this expectation, elucidating the crucial role played by the parameters p and q in this context:

Theorem 8 Average General DynamicScore

Let G be a graph produced by $EMG(n, p, q)$ Let \hat{m}_t be the density of graph at step t . Then, in average:

$$\mathcal{D}_t^E = 1 - \frac{p\hat{m}_t}{1 + q(\hat{m}_t - 1)} \quad (3.4)$$

Proof: The proof consists in finding the average number of edges in $E_t \Delta E_{t+1}$ and in $E_t \cup E_{t+1}$. For the first one, it consists in computing, on average, the number of edges which state is changing. Assuming the density of edges at t is \hat{m}_t , then the density of edges that change from present to absent is on average $(1 - p)\hat{m}_t$ and the density of newly present edges is on average $(1 - q)(1 - \hat{m}_t)$. Therefore, the size of $E_t \Delta E_{t+1}$ is on average:

$$|E_t \Delta E_{t+1}| = (1 - p)\hat{m}_t + (1 - q)(1 - \hat{m}_t)$$

For computing the union size, it is sufficient to notice that it contains all the present edges at step t plus appearing edges $(1 - q)(1 - \hat{m}_t)$. Thus, the size of the union is in average:

$$|E_t \cup E_{t+1}| = \hat{m}_t + (1 - q)(1 - \hat{m}_t)$$

It is therefore possible to estimate the average DynamicScore:

$$\mathcal{D}_t^E = \frac{(1-p)\hat{m}_t + (1-q)(1-\hat{m}_t)}{\hat{m}_t + (1-q)(1-\hat{m}_t)} = 1 - \frac{p\hat{m}_t}{1 + q(\hat{m}_t - 1)}$$

■

This result must be evaluated for densities close to m^* . The reason for this is that, as the generation process is Markovian and $|p + q - 1| < 1$, the sequence of produced graphs quickly converges to a limit, which happens to be an ER graph with parameters $\binom{n}{2}$ and π^* . Since these graphs have an edge count following a binomial distribution with parameters $\binom{n}{2}$ and π^* , we conclude that the typical density is also a highly representative value of the density of the produced graphs. This implies density of these graphs are close to $\pi^* = m^*$. Combining this theorem with the fixed point density of generated graphs provides DynamicScore at the fixed point density:

Theorem 9 E-DynamicScore in Average around m^* :

For all p, q such that $|p + q - 1| < 1$

$$\mathcal{D}_t^E(m^*) = 2 \frac{1-p}{2-p}$$

Moreover, $\mathcal{D}_t^E(m^*)$ may take any value from 0 to 1.

Proof: it results from the combination of both theorem 8 and lemma 2. ■

Notably, the average E-DynamicScore is independent of the value of q and exhibits a decreasing trend as p increases. The range of possible values for the E-DynamicScore ranges from 0 to 1, indicating its ability to capture the extent of changes in the graph. These findings are illustrated on Figure 3.3, obtained through simulations, considering various values of p and q , both ranging from 0 to 1, while excluding the endpoints. These visual representations offer a good understanding of the relationship between EMGG instances, their E-DynamicScore on average, and the parameters p and q . By examining these figures, we obtain experimental confirmation and deeper understanding of the behavior and characteristics of the EMGG, corroborating the insights provided by the above-stated theorem, particularly in relation to the E-DynamicScore. The impact of the parameter q on the average E-DynamicScore is found to be negligible, whereas parameter p appears to be more influential in determining its value. Notably, it is observed that the E-DynamicScore can encompass the entire range from 0 to 1 as p varies from 1 to 0.

3.2.3.3 Barabasi-Albert Generating Graphs

In [5], the generative process is clearly described. For this model, the focus is both on the evolution of the vertices and on the edges. Using our notations the generation of the graph starts with a seed graph $G_0 = (V_0, E_0)$ such that $|V_0| = n_0$ and $0 \leq |E_0| = m_0 \leq \frac{1}{2}n_0(n_0 - 1)$. Note that in the original research article, no information is given about the initial number of edges. At every time step $t + 1$ a new vertex u_{t+1} is added and this new vertex is linked to $m(\leq n_0)$ vertices already in V_t . Thus $V_{t+1} = V_t \cup \{u_{t+1}\}$ and $|E_{t+1}| = |E_t| + m$. From this it is possible to compute both Vertex and Edge-DynamicScore. As the number of node inserted in the graph at each step is one, $\mathcal{D}_t^V = \frac{1}{n_0+t+1}$. Moreover, the number

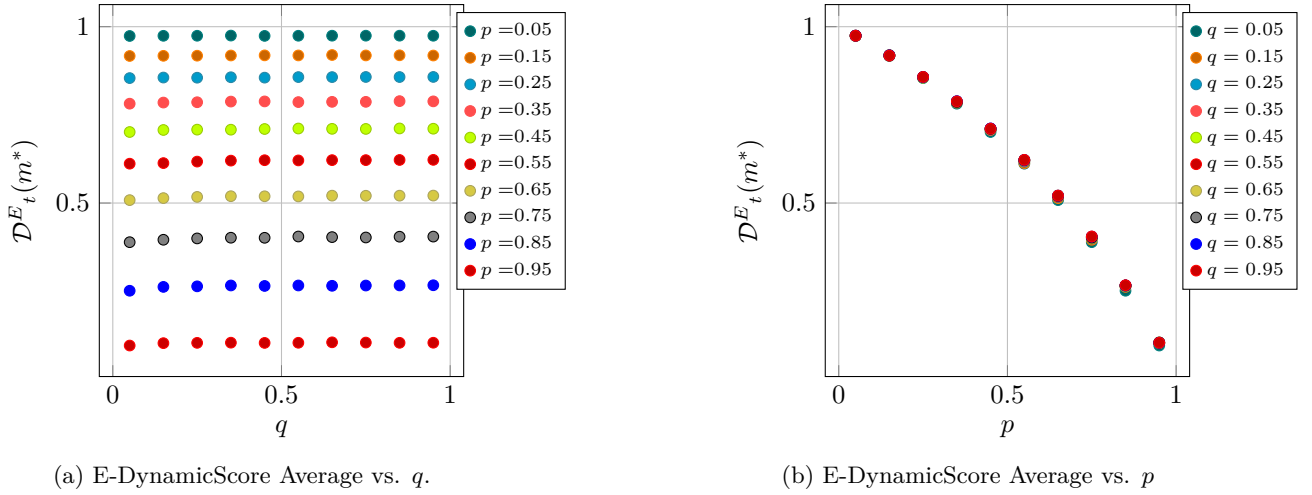


Figure 3.3: Average dynamic score as a function of the parameters p and q . On the left, the parameter p is set and the parameter q ranges from 0.05 to 0.95. One may notice that for a fixed value of parameter p , the average dynamicScore does not depend on q . On the right, the parameter q is set and the parameter p ranges from 0.05 to 0.95. As observed with the picture on the left, the average dynamicScore does not depend on q so all the marks are mingled.

of new connections being m and no connection being removed leads to $\mathcal{D}_t^E = \frac{m}{m_0 + (t+1)m}$. Thus, both the Vertex and the Edge-DynamicScore are decreasing and tend toward 0 as t tends to infinity. The two plots in table 3.1 illustrate this decreasing trend. Graphs produced using such a mechanism are

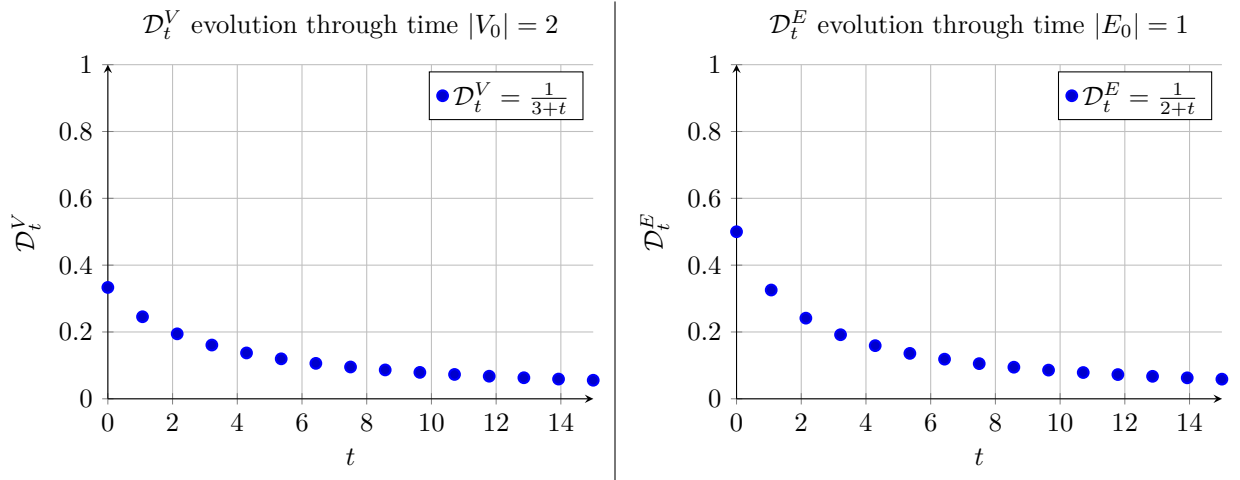


Table 3.1: Dynamic Score of both the edges and vertices for the Barabási-Albert model. The considered parameters are $|V_0| = 2$, $m_0 = |E_0| = 1$ and $m = 1$.

such that their dynamics is decreasing and becomes very close to 0 as the number of step increases. Even though the original purpose of such a model is not to produce very changing graphs but rather to produce networks fitting a certain property one can find in real networks, the study of the DynamicScore on such a model faces an observation raised on the original paper written by Barabasi et. al. One main condition stated in the paper says that to observe an emerging scale-free trend in network, it is necessary that preferential attachment (the way new vertices connect to already existing vertices by selecting with a higher probability those having a high degree) comes with a growing network. On top of that, the reasoning involved to prove that the generating model produces scale-free networks is an asymptotic reasoning and to consider that the graph grows in size to infinity. Such a condition implies that the size

of the network is always increasing, meaning the network must be considered an infinite process that continuously evolves. With this approach, we can say that preferential attachment, as designed in the BA paradigm, leads to graphs with low dynamics.

Conclusion on the DynamicScore Metric

The DynamicScore (both of the vertices and the edges) metric defined in this section offers a new vision on the study of dynamic graphs. It provides a way to compute composition changes in a dynamic graph between two consecutive steps by measuring dissimilarities observed in snapshot graphs between these two steps. This metric may be used to quantify the dynamics of a time-varying graph. As the metric can only take values ranging from 0 to 1, it is possible to use the DynamicScore value to say whether a dynamic graph is rather dynamic (value close to 1) or rather not dynamic (value close to 0) with the special case of a null DynamicScore (constant graph). The importance of the DynamicScore goes further as for some model (EMGG) as its expected value may be proved to be independent from one or many of the parameters of the model. Moreover, specific values of V-DynamicScore and E-DynamicScore may be obtained under some circumstances such as polynomial growth, exponential growth or graph densification. It is also a complementary metric of the graph order. When the number of vertices does not change from one step to the next one, V-DynamicScore helps providing information on what have changed in the graph.

As a further study of the DynamicScore, this metric will be further study in Chapter 4 where it will be used to have a better understanding of the dynamics of a generative process. Moreover, application of the metric to real world networks is presented on Chapter 5. In this chapter, different transformations of real world data will be studied and their dynamics explored through the use of the DynamicScore metric.

3.3 Information Persistence Problem

The work presented in this section has been published and preseted during the 2024 edition of the French Regional Conference on Complex Systems [2].

This section introduces a new problem, generalizing concepts of reachability in dynamic graphs (see Section 2.3.2.1). The *information-persistence* problem addressed here focuses on how information remains present in a dynamic graph when the set of vertices changes over time. To the best of our knowledge this is the first time this problem is formally defined and addressed. While they may appear similar in appearance, the broadcasting and epidemic spreading problems [20, 29, 34] are, to a large extent, different from the problem described in this work.

In the context of a dynamic graph wherein the set of vertices evolves over time, along with a defined communication strategy, the aim is to check whether information persists within the graph at a specified step T , and potentially quantify the proportion of vertices retaining this information.

The objective of this study is to introduce two novel problems relating to this subject. Firstly, in Section 3.3.1, we delve into the formulation of the problem. Concurrently, we provide clear definitions of dynamic graphs and broadcasting strategies to mitigate any potential ambiguity. Additionally, we introduce concepts relevant to dynamic graphs, such as dynamic graph generation, to broaden the scope of information persistence study. This section culminates in the formal definition of the *information persistence problem* and the *information covering problem*. Section 3.3.3 is dedicated to the presentation of some preliminary results for a restricted set of instances. In particular we provide algorithms solving the defined problems and show that their time complexity are polynomial. After dealing with these points, a last section (Section 3.3.4) is dedicated to open problems this new paradigm offers.

3.3.1 Problem Formulation

The *information-persistence* problem is defined using two parameters: a dynamic graph $\mathcal{G} = (G_t)_{t \in \mathcal{T}}$, with $\mathcal{T} = [0, T]$, and a communication strategy A specifying how information spread. Knowing these two information, the question is whether or not there exists a couple of vertices (u, v) such that $u \in V_0$, $v \in V_T$ and u can reach v through a journey according to the communication strategy.

3.3.1.1 Communication Strategy

The parameter A concerns the communication strategy. In this work, we restrict the communication policies to local broadcasting strategies only. Thus, when communicating at time t , a vertex sends the information to all its neighbors connected to it at time t . The broadcasting strategy describes the way information spread between vertices. Many works have been devoted to this problem, especially in the domain of mobile ad hoc networking [45]. But in most studies, the set of vertices remains the same all the time. However, whatever the case, changing or unchanging vertices set, the strategy must specify the conditions for sending a message to connected neighbors. For instance, a minimum delay of one time step might be mandatory on a vertex between the reception of the information and its transmission to its neighbors. Some strategies may select vertices to which transmit the information within the neighborhood, or may allow only a restricted number of transmissions. All these points have to be clearly defined. In this work we restrict our study to two algorithms and remarks are made to highlight relevant questions.

Constant Flooding The first algorithm to be discussed is a variant of the flooding algorithm. We call it constant flooding. The principle is that once a vertex receives the information, it keeps transmit it to its neighbors as long as it is present in the graph. The constant flooding algorithm is defined as follows:

Definition 19 *Constant Flooding Algorithm (CF):*

Let G be a dynamic graph. Let u be a vertex in this graph. Let t be the date at which u receives the information. Then as long as u remains in the graph, u transmits the information to its neighbors at every step $t' > t$.

It is important to notice that if a vertex receives information at step t it starts its transmission from the next time step, at $t + 1$.

Simple Flooding The strategy discussed here is known as simple flooding algorithm. The principle is the following: once a vertex receives the information, it is allowed to send it to its neighbors only once. A delay of one time step has to be observed between the reception and the emission. In the current work this algorithm only waits one time step before the transmission of the information to its neighbors, however it could be possible to consider other variants of this algorithm for which the transmission could be done later. The algorithm is defined as follows:

Definition 20 *Simple Flooding Algorithm (SF):*

Let G be a dynamic graph. Let u be a vertex in this graph, meaning there exists t such that $u \in V_t$. Let t' be the first date such that u receives the information. Then u sends the information to its neighbors at $t' + 1$ only.

Note that the transmission might be done even if the vertex has no neighbors connected.

3.3.2 Studied Problems

In this part, we define notions and set formalism aiming at studying and defining the information persistence and the information coverage problems. For the rest of this part, we assume that a broadcasting strategy A and a dynamic \mathcal{G} are defined. The first step is to introduce the notion of *reachability* between two vertices $u \in V_0$ and $v \in V_T$ as it plays a key role in the definition of the two problems.

Definition 21 *Reachability:*

Let $u \in V_0$ and $v \in V_T$. We say that u can reach v using the broadcasting strategy A , and we note $u \xrightarrow{A} v$, if v can receive information from u according to the broadcasting strategy A .

Although reachability has already been defined in Section 2.3.2.1, here the definition is made in a different paradigm. Indeed, vertices are allowed to disappear, and therefore vertices present at step $t = 0$ may be absent at step T . Therefore the time at which the journey starts and when it ends now matters. As the goal is to define a problem dealing with persistence of information, the date at which the journey starts and when it ends are set to the 0 and T respectively. Note that the target vertex may be reached before date T , but it still must exist in the dynamic graph at this date. However, if the information is not introduced at a step later than 0, it may be relevant to define the reachability according to the date at which u sends the information and the date at which v receives it. Note also that this definition of reachability also implies that there exists a journey from u to v , satisfying the condition implied by the broadcasting strategy A .

From the definition of reachability it is possible to define two decision problems. The first one is defined as the capability for a dynamic graph to exhibit a vertex $u \in V_0$ and another vertex $v \in V_T$ such that $u \xrightarrow{A} v$. This problem will be referred to as the *information persistence problem* in this document. Formally, this problem can be defined as follows:

Definition 22 *Information Persistence Problem:*

Let $\mathcal{G} = (G_t)_{0 \leq t \leq T}$ be a dynamic graph and let A be a broadcasting strategy. Then, \mathcal{G} is said to be A -persistent if it satisfies:

$$\exists u \in V_0, \exists v \in V_T, u \xrightarrow{A} v \quad (3.5)$$

The second problem defined in this work is called the *information coverage problem*. It is the capability for a dynamic graph to exhibit for every vertex $v \in V_T$, at least one vertex $u \in V_0$ such that $u \xrightarrow{A} v$. This means there exists a subset $S \subset V_0$ such that for every vertex $v \in V_T$, there exists $u \in S$ such that u can reach v using the strategy A . Formally, this problem is defined as follows:

Definition 23 *Information Coverage Problem:*

Let $\mathcal{G} = (G_t)_{0 \leq t \leq T}$ be a dynamic graph and let A be a broadcasting strategy. Then, \mathcal{G} is said to be A -coverable if it satisfies:

$$\forall v \in V_T, \exists u \in V_0, u \xrightarrow{A} v \quad (3.6)$$

Now that both problems have been defined, the remainder of this section is dedicated to their study. As a first step, an analysis of the two problems will be presented assuming the dynamic graph is known. An algorithmic study will show that the problems can be solved in polynomial time by conducting a simple simulation of information spreading within the graph. Subsequently, a link will be established between the persistence of information and the processes of generating dynamic graphs. We demonstrate in this case that sustainability alone is not sufficient for information to persist within a graph and attempt to identify conditions under which the generated graphs exhibit information persistence in the case of the D3G3 model presented in the next Chapter.

3.3.3 Remarks and First Results

The main result is an algorithm to solve both the information persistence problem and the information coverage problem when the broadcasting strategy is CF (constant flooding). Its time complexity is studied and is shown to be polynomial. This algorithm takes as an input a dynamic graph \mathcal{G} and a set of vertices $I_0 \subset V_0$ having information at date 0. This algorithm depends on the broadcasting strategy studied and simulates the spread of information.

With this algorithm, it is possible to answer the question of the persistence problem. Indeed, it is sufficient to apply this algorithm with the whole set of initial vertices $I_0 = V_0$. If the algorithm ends returning a non-empty set, then, \mathcal{G} is A -persistent. With the same idea, it is possible to answer the information coverage problem. If the result of the spreading algorithm with $I_0 = V_0$ is V_T , then it means every vertex in V_T can be reached by at least one vertex in V_0 .

3.3.3.1 Time Complexity of the Spreading Algorithm

Let us now study the time complexity of algorithm 1. The goal is to prove that the time complexity of this algorithm is $O\left(\sum_{t=1}^T n_t m_t\right)$, with $m_t = |E_t|$. This comes from the complexity of the most nested *for* loop. In the worst case, the time complexity of set operations are bounded by the size of the set. Here,

Algorithm 1 *Spreading*(\mathcal{G}, I_0)

Require: $\mathcal{G} = (G_t)_{0 \leq t \leq T}$ a dynamic graph, $I_0 \subset V_0$ set of vertices having the information at date $t = 0$.

Ensure: $I_T \subset V_T$ set of vertices receiving information from vertices in I_0 or \emptyset if no such vertices exist.

```

1:  $I \leftarrow I_0$ 
2: for  $t \leftarrow 0$  to  $T$  do
3:    $I \leftarrow I \cap V_t$ 
4:   if  $I = \emptyset$  then
5:     return  $\emptyset$ 
6:   end if
7:   for  $(x, y) \in E_t$  do
8:     if  $x \in I$  and  $y \notin I$  then
9:        $I \leftarrow I \cup \{y\}$ 
10:    else if  $x \notin I$  and  $y \in I$  then
11:       $I \leftarrow I \cup \{x\}$ 
12:    end if
13:  end for
14: end for
15: return  $I$ 

```

the considered set is $I \subset V_t$ for any given $t \in [1, T]$. Therefore, the time complexity of the lines 7–13 is $O(m_t \times n_t)$. As it is the biggest time complexity of the first *for* loop (lines 2–14), the time complexity of the whole algorithm is thus $O\left(\sum_{t=1}^T n_t m_t\right)$.

As mentioned above, this algorithm solves both the information persistence problem and the information coverage problem. We can therefore deduce that the complexity of these two algorithms is $O\left(n_0 + \sum_{t=1}^T n_t m_t\right)$, where the term n_0 comes from the construction of I_0 , a copy of V_0 . Thus, we have established the existence of algorithms solving the information persistence and the information coverage problems in polynomial time.

3.3.4 Questions and Open Problems

This section aim at discussing open problems and remarks about the two defined problems. Questions about the communication algorithms are first addressed. Then connections with the sustainability property are investigated.

3.3.4.1 Questions Related to the Simple Flooding Algorithm

The reader may have noticed that the previously defined algorithm is not convenient if the node has no neighbor at the moment of transmission. It would be interesting to postpone the transmission at another date in order to improve the performances of the process from an information-persistence point of view. However, considering such a possibility raises many questions.

First, it could be possible to remove the delay between the reception of the information and its transmission to neighbors. This situation is similar to the notion of non-strict path in temporal graphs. In that case, given a connected component, as soon as one vertex receives the information, then all the vertices of that connected component also receive and send the information. As a consequence, all these nodes will never send the information at later dates. This means that only the vertices present at step

0, if the information is introduced at date 0, will have the information. This means that some vertices must exist both at step 0 and T to ensure the persistence of information.

In an opposed direction there are some ways that take into account the possible future neighborhood of the node to estimate the moment when the information could be sent. Indeed, another way to define the moment to send the information would be to wait until the neighborhood is not empty. This question is not treated in this document, however it may offer interesting wondering. For instance, it is possible to study questions such as:

- When should a given node send the information it owns so that information persistence is guaranteed?

A last question concerns the possibility for a node to transmit the information several times. The defined algorithm does not allow multiple transmissions: once the information has been spread by one vertex, this vertex does not transmit it again. It is possible to imagine some applications where the vertex can receive the information several times. Every time the information is received, the vertex will transmit it again to its neighbors. This defines a new algorithm and the questions, previously defined, may be addressed for this new strategy.

3.3.4.2 Connection between Simple Flooding and Constant Flooding

One final aspect to address here is the connection between simple flooding and constant flooding strategies. Specifically, it is observed that if a dynamic graph is SF-persistent, then it is also CF-persistent. To illustrate this, it suffices to note that if a dynamic graph \mathcal{G} is SF-persistent, then there exist vertices $u \in V_0$ and $v \in V_T$ such that information is transmitted from u to v via a temporal path using the simple flooding communication strategy. However, this path is also observable using the constant flooding communication strategy. Indeed, for propagation with a simple flooding strategy, information can only be transmitted once it has been received, whereas in the case of a constant flooding strategy, information is continuously transmitted after being received. Thus, in the scenario where the communication strategy is constant flooding, the information can indeed follow the same path between u and v as when the communication strategy is simple flooding. Note also that there are some cases in which SF cannot achieve information persistence, while CF can.

3.3.4.3 Sustainability and Information Persistence Problem

This section focuses on studying the problem of information persistence from the perspective of dynamic graph generators. Here, we assume that graphs are the product of a generation mechanism as defined earlier. One of the initial observations is that if a generator produces A -persistent graphs for any diffusion strategy A , then the graphs do not become empty, which is a characteristic of sustainability. However, the converse is not necessarily true. There are cases where the produced graphs are sustainable without being information persistent.

For instance, considering the generative model defined in Chapter 4, it is possible to find parameter values such that the produced graphs are sustainable with high probability without being information persistent. An example may be found in Section 4.3.1.2.

Conclusion on the Information-persistent Problem

This section introduced the Information Persistent Problem and redefined several basic concepts related to the study of dynamic graphs, now considering the possibility for vertices to disappear over time. An example of this is the Reachability problem, which now requires that the source vertex is present at the initial date and the target vertex at the final date.

Additionally, two new problems related to communication were introduced. The first one, referred to as the *information-persistent problem*, addresses the problem of the existence of a journey from a vertex present at the first step to a vertex present at the last date. The second problem, referred to as the *information-covering problem*, aims to quantify the minimum number of vertices required to transmit information from step 0 to all the vertices present at the last step. A communication algorithm is used to describe how information traverses edges and how vertices are allowed to communicate to their neighbors. In this document, we proved that when a whole dynamic graph is known, the problem may be solved using a polynomial time algorithm. This algorithm simply simulates the spread of the information through the dynamic graph. The problems are then analyzed under the condition where only the generative process is known to see what result can be inferred from this limited knowledge.

The two problems defined in this section shows a natural continuation of the sustainability property. The objective was to see whether a dynamic graph generator that produces sustainable graphs also generates information-persistent instances. We showed that this is not always the case (see Section 4.3.1.2). The study of these problems is therefore motivated by the need to identify, for dynamic graphs, conditions of information-persistence as they are different from those ensuring sustainability. Future research should investigate these conditions further.

Moreover, additional perspectives could include exploring other notions defined in the context of time-varying graphs, such as time-connectivity. For example, the basic notion of time-connectivity could be extended to account for the addition and removal of vertices over time.

Conclusion

This chapter focused on the study of dynamic graph generators as defined in Section 2.5. Dynamic graphs are produced by an iterative generative process that starts from an initial condition and results in a sequence of static graphs. Understanding the dynamics of a graph requires studying the underlying generative process that governs its evolution over time. The chapter explores different questions related to various models of dynamic graph generators. For instance, some generative models do not constrain the size of the graph, making essential the analysis of its evolution to predict whether it will become

empty or will produce loops. Additionally, understanding changes in the composition of the graph is crucial, especially when the size of the graph remains constant.

To this end, two metrics and two problems have been designed to study these scenarios. The first metric, called sustainability, characterizes a dynamic graph generator as sustainable if the generated graphs neither become empty nor periodic. While this property is often obvious in well-known models such as the Barabási-Albert model, which produces infinitely growing and hence sustainable graphs, it becomes less clear for models like the Generator of Edge-Markovian Graphs. For these, specific conditions need to be identified to ensure sustainable instances. The sustainability metric is particularly useful for studying generators, such as the one presented in the next chapter, where the network size and state result from the application of predefined rules.

The second metric introduced is the DynamicScore, which highlights compositional changes between two consecutive graphs generated by a process. Preliminary results demonstrate how the DynamicScore reflects the dynamics of time-varying graphs. Applying this metric to existing generator models showed, for example, that the Barabási-Albert model produces dynamic graph instances with both the V-DynamicScore and E-DynamicScore decreasing toward zero. A notable property was observed with the Edge-Markovian Graph Generator, where the average value of the E-DynamicScore was shown to be independent of one of the parameters of the model. Although this chapter did not apply the DynamicScore to real-world data, this is a goal for Chapter 5.

In addition to these metrics, two problems were defined, extending beyond the sustainability metric and generalizing concepts related to time-varying graphs. These problems can be explored once a dynamic graph generator is proven to produce sustainable graphs. One example of such redefinition is the modified Reachability problem, which accounts for disappearing vertices over time. Two new communication-related problems were introduced: the information-persistent problem, which explores the existence of a journey between a vertex present at the first step and another at the last step, and the information-covering problem, which seeks to determine if there exists a subset of the initial vertices that are able to transmit information from the initial step to all vertices present at the last step. A polynomial-time algorithm was proposed to solve these problems when the entire dynamic graph is known, but the solution differs when the graph is unknown. The study highlighted that conditions for information persistence differ from those for graph sustainability, necessitating further research to identify these conditions.

Future work should explore other properties of time-varying graphs, such as extending time-connectivity to account for the addition and removal of vertices. For example, a possible extension could involve determining whether there exists a subset of initial vertices that can reach all existing vertices, not just those present at the last step.

Illustrative Case Study of Dynamic Graph Generators Analysis

The main sections of this chapter have been published and presented during the 2nd Symposium on Algorithmic Foundations of Dynamic Networks [3]. The results presented in Section 4.4 have been published in the Applied Network Science [4].

The model presented in this chapter serves as an illustrative case study of a dynamic graph generator, where sustainability and DynamicScores, as introduced in the previous chapter, are not straightforward. Inspired by Conway’s Game of Life [12], our model diverges significantly: it uses random vertex positioning rather than a grid, resulting in nondeterministic behavior. This divergence necessitates new tools and metrics beyond those used in cellular automata. Our initial studies examined parameter settings that affect the sustainability of the generated graphs. We identified specific parameter families that influence whether a graph remains sustainable or collapses over time. For example, one set of parameters led us to define and explore sustainability zones, while another revealed structural patterns that support sustained graph growth. Finally, a particular parameter set showcases sustainable dynamic graphs which are not information-persistent (see Section 3.3).

4.1 Definitions and Generative Model and Definitions

In this section we define a parameterized model generating families of dynamic graphs : the Degree-Driven Dynamic Geometric Graph Generator (D3G3). D3G3 is a parameterized generator and according to the parameters, it can produce a wide variety of dynamics. It will be used as a case study. A first global analysis of the generated graph families is performed in Section 4.2. Section 4.3 focuses on specific values of the parameters and present a rigorous analysis of the evolution of the dynamics of the graph and of the likelihood of its sustainability. Graphs produced by D3G3 are sequence geometric graphs ordered by a timestamp. A geometric graph is defined by an euclidean space and a threshold d . For this study, without loss of generality we consider a 2D-unit-torus (i.e., a square $[0; 1]^2$ where the two opposite sides

are connected). Each vertex is characterized by a set of coordinates, such that given two vertices u and v it is possible to compute their euclidean distance: $dist(u, v)$. Given V the set of vertices, the set of edges E is defined in the following way: $E = \{(u, v) \in V^2 \mid dist(u, v) \leq d\}$

The choice to use a unit torus in two dimensions to position the points of a random geometric graph offers several significant advantages. Firstly, by adopting this topological structure, border effects are eliminated, thus ensuring a more precise representation of connectivity between graph points. Furthermore, by basing the probability of connection between vertices on the surface area each vertex occupies on the torus, we achieve a method that considers the spatial geometry. This approach is bolstered by existing analyses such as [13, 36] Drawing upon these previous works allows for further analysis, focusing on specific aspects of the problem, such as calculating the probability of a vertex having a certain degree.

Finally, the choice of the unit torus and more generally, the use of geometric random graphs, stems from their natural way of connecting vertices. In these models, the connection between two vertices is only determined by the distance separating them: for two vertices to be connected, it is necessary and sufficient for their distance to be less than a connection threshold. There are, of course, alternative methods for connecting vertices. For instance, one could mention the model proposed by B.M. Waxman in [44], where the connection between two vertices is not guaranteed by their distance being strictly less than a threshold, but rather by a probability density function parameterized by this threshold. This last model was not considered in our work.

Graphs generated by D3G3 are produced thanks to an evolution process. This mechanism is parameterized by an initial graph (the seed graph) and by two transition rules driving the evolution of the graph between two consecutive time steps. Apart from a random generator, no external decision or additional information is used by this mechanism. Rules are based on vertex degrees only and rely on a random generator for positioning new vertices in the 2D euclidean space. This leads to the name of the generator: *Degree-Driven Dynamic Geometric Graphs Generator* or D3G3.

Definition 24 Degree Driven Dynamic Geometric Graph Generator

An instance of D3G3 is defined by an initial graph, a set of parameters and two rules:

- $G_0 \neq (\emptyset, \emptyset)$ the seed graph,
- parameters:
 - $d \in]0, \frac{\sqrt{2}}{2}[$
 - S_s a set of non-negative integers
 - S_c a set of non-negative integers
- rules applied on G_t leading to G_{t+1} :
 - if $v \in V_t$, then $v \in V_{t+1}$ if and only if $\deg(v) \in S_S$ (conservation rule)
 - if $v \in V_t$ and if $\deg(v) \in S_C$ then add a new vertex to V_{t+1} with a random position in the unit-torus (creation rule)

Thus, evolution of the graph between two consecutive time steps t and $t + 1$, is driven by two rules applied to each vertex $v \in V_t$ simultaneously. The first rule determines for a vertex $v \in V_t$ whether it is kept at step $t+1$ while the second rule concerns the possibility for a vertex $v \in V_t$ to create a new vertex in V_{t+1} according to its degree. The position of conserved vertices does not change from one step to the next one. Conversely, position of new vertices is drawn uniformly at random in the torus. This choice has been made to study the behavior of dynamic graph in a first time without considering every possible way to place new vertices on the torus. As we show in this document, this paradigm allows interesting dynamics to emerge without requiring too much parameters. Indeed, introducing a function to set position of new vertices implies the analysis of this function and therefore an analysis of its parameters. For the moment, this study focuses only on the case where new vertices have random position. Before going further in the analysis of this generative model, let us defined some vocabulary:

Definition 25 Conserved/Create/Removed/Duplicated nodes

Let $G = (G_t)$ a graph produced by D3G3. Let $t \geq 0$. Let $u \in V_t$ and $v \in V_{t+1}$, then

- u is said to be a conserved vertex if and only if $u \in V_t \cap V_{t+1}$.
- u is said to be removed if and only if $u \in V_t - V_{t+1}$.
- u is said to be a creator/creating vertex if and only if $\deg(u) \in S_C$.
- v is said to be a created vertex if and only if $v \in V_{t+1} - V_t$.
- u is said to duplicate if and only if it is both a conserved and a creator node.

4.2 Theoretical Analysis

While the model is very simple, it presents a wide variety of dynamics and long-term evolution. According to S_S and S_C composition, several classes of dynamic behaviors have been identified. These classes have been defined by computing two measures: the evolution of the order of the graph, and the evolution of the Vertex DynamicScore. Results are reported at the end of this section in Tables 4.1 and 4.2.

4.2.1 First Results on Sustainability

The first presented results concern sustainability of graphs generated by D3G3. In order to state the results, a first lemma concerning the evolution of the vertices and edges of D3G3 instances is proved:

Lemma 4 *Let $G = (G_t)_{t \in \mathbb{N}}$ dynamic graph generated by D3G3. Then, for all $t \in \mathbb{N}$, if $V_t = V_{t+1}$, then $E_t = E_{t+1}$.*

Proof: This comes from property of conserved vertices. Indeed, conserved vertices of any instances of D3G3 have a fixed position. Therefore, for any $t \in \mathbb{N}$ if all the vertices of V_t are conserved, then no edges are removed nor added as edges depends only on the distance between vertices. This ends the proof. ■

From this lemma, it comes that the study of sustainability of D3G3 instances relies mainly on the evolution of the set of vertices. Indeed, if the set of vertices does not change, then the set of edges does not change neither. It is therefore sufficient to restrain on the evolution of the set of vertices. From this remark, we can prove the main result of this section :

Theorem 10 *Let $G = (G_t)$ a dynamic graph generated by D3G3, if its order and its V-DynamicScore are never equal to 0 then the graph is sustainable.*

Proof: The sustainability defined in Section 3.1 involves two properties. A dynamic graph is said to be sustainable if it never becomes empty nor periodic. Therefore, to prove this theorem, it suffices to focus on the periodic aspect as dynamic graphs are assumed not to become empty. Moreover, the V-DynamicScore of considered dynamic graphs is positive, meaning they never become static. It comes from these remarks that these dynamic graphs are sustainable if they do not become periodic for any period $k > 1$. To demonstrate that this never occurs, let us employ a proof by contradiction. Let assume there exists $k > 1$ and $T \in \mathbb{N}$ such that for all $t \geq T$, $V_t = V_{t+k}$. Let $t \geq T$. As dynamic graphs are assumed not to become static, then $V_t \neq V_{t+1}$. Two cases may lead to such a situation. The first case is at least one vertex has been removed from the graph. In that case, this vertex never appear in the graph later, which leads to a contradiction. The second case occurs when at least one vertex is created. In such a case, to satisfy the periodic assumption, this vertex must disappear to ensure $V_t = V_{t+k}$. However, this implies $V_{t+1} \neq V_{t+k+1}$ as a vertex that is removed never appears again. This is also a contradiction. Thus, the periodic assumption cannot be true which implies the dynamic graph is sustainable. ■

4.2.2 Limit Case Analysis

Regarding the generative process, the obtained graphs depend on the threshold d , on the seed graph G_0 and on the two sets S_C and S_S . Each combination of d , S_C and S_S does not necessarily lead to

sustainable graphs. However, some simple cases may lead to a positive or negative answer to the question "are the produced graphs sustainable?". These cases are called *limit cases*. They occur when at least one of the two sets S_S and/or S_C is either \mathbb{N} or the empty set. The following paragraphs address these situations.

4.2.2.1 Case: $S_S = \mathbb{N}$ and $S_C = \mathbb{N}$

At each time step each vertex of the graph is both a conserved and a creator vertex, it is therefore duplicated. Then, as soon as $G_0 \neq (\emptyset, \emptyset)$, the order of the graph increases exponentially: $n_t = 2^t n_0$ and thus $\lim_{t \rightarrow \infty} n_t = +\infty$.

Consider an instance of dynamic graph produced by D3G3 with these parameters sets, is this graph sustainable? The analysis of the DynamicScore leads to the conclusion. Indeed, every vertex is duplicated, thus, $|V_{t+1}| = 2n_t$. From that, it comes $\mathcal{D}_t^V = \frac{1}{2}$. In conclusion, for all t , the graph is never null and $\mathcal{D}_t^V > 0$, thus from Theorem 10 the graph is sustainable.

4.2.2.2 Case: $S_S = \mathbb{N}$ and $S_C = \emptyset$

If $S_S = \mathbb{N}$ and $S_C = \emptyset$, then for all t , whatever the degree of any vertex, it is not a creator, hence no new vertices are added to the graph. In addition, all vertices have their degree in S_S , thus every vertex is conserved between t and $t+1$. Thus, $G_t = G_0$ for all $t \in \mathbb{N}$ which means the graph is static and therefore not sustainable.

4.2.2.3 Case: $S_S = \emptyset$ and $S_C = \mathbb{N}$

This case is the opposite situation of the previous one. All the vertices are creators, but none are conserved. Hence, for all $t \in \mathbb{N}$, if $v \in V_t$, $v \notin V_{t+1}$, but, as $\deg(v) \in S_C$, the creation rule generates a new node that replaces v . So, for all $t \in \mathbb{N}$, $n_t = n_0$. The graph is only always changing as all nodes are renewed. From the DynamicScore point of view, this implies $V_t \cap V_{t+1} = \emptyset$ and $\mathcal{D}_t^V = 1$ and the graph is sustainable.

4.2.2.4 Case: S_S is a non-empty finite set and $S_C = \emptyset$

In such a case, no vertex will be created in this situation since $S_C = \emptyset$. Thus for all t , $n_t \leq n_0$. If G_0 is not null, then, several cases may occur:

1. first case: all vertices in G_t have their degree in S_S : for all $v \in V_t$, $\deg(v) \in S_S$. Then, all vertices are conserved, the order remains unchanged and the graph is static, $G_{t+1} = G_t$, hence the graph is not sustainable,
2. second case: opposite to the first one, for all $v \in V_t$, $\deg(v) \notin S_S$, hence G_{t+1} is null and the graph is not sustainable,
3. last case: some vertices have their degree not belonging to S_S , they are removed and as a consequence $n_{t+1} < n_t$. At each occurrence of this case, the order of the graph is strictly decreased by

one, then this can happen at most n_0 times before the graph becomes empty unless it becomes static before.

As a consequence, for the limit case, the graph is not sustainable.

4.2.2.5 Case: $S_S = \emptyset$ and S_C is a non-empty finite set

Note first that for all t , $V_{t+1} \cap V_t = \emptyset$, no vertex is conserved from t to $t + 1$. In addition, for all t , $n_{t+1} < n_t$, unless if for all $v \in V_t$, $\deg(v) \in S_C$, then $n_{t+1} = n_t$. Thus the evolution of the graph order is non increasing. In addition, as $S_S = \emptyset$, $\mathcal{D}_t^V = 1$, so the graph can never become static. The only way for the graph to be non sustainable is to become null. It is then not possible to conclude about the sustainability. Indeed, consider the case for which at a given date t , $0 < n_t \leq k + 1$ and $S_C = [0, k]$ then for all $t' > t$, $n_{t'} = n_t$. Such a dynamic graph is sustainable. Conversely, if $S_C = \{k\}$, as soon as less than $k + 1$ nodes have their degree equal to k the graph becomes empty after two time steps.

4.2.2.6 Graph Order Increase

For some specific parameters sets, the increase of the order of the graph asymptotically tends to zero, which has some effect on graph sustainability. This is the object of the following theorem:

Theorem 11 *Let $\mathcal{G} = (G_t)_{t \geq 0}$, S_S and S_C such that S_S or S_C equals \mathbb{N} while the other one is finite and not empty, then the probability that graph order increases tends toward 0 as the graph order tends to infinity: $P(n_{t+1} > n_t) \xrightarrow{n_t \rightarrow +\infty} 0$.*

Proof Let $P(n_{t+1} > n_t)$ denote the probability that graph order increases between step t and step $t + 1$ and $D = S_S \cap S_C$. For a node $u \in V_t$, u duplicates if and only if $\deg(u) \in D$.

$$\begin{aligned} P(n_{t+1} > n_t) &= P(\exists u \in V_t, \deg(u) \in D) \\ &= 1 - P(\forall u \in V_t, \deg(u) \notin D) \end{aligned}$$

As every node is independent from another, the probability $P(\deg(u) \notin D)$ is the same for all $u \in V_t$, the following holds:

$$P(n_{t+1} > n_t) = 1 - (P(\neg D))^{n_t}$$

where $P(\neg D)$ represents the probability a node does not satisfy the duplicating condition. As positions of vertices are independent of each other:

$$P(\neg D) = 1 - \sum_{k \in D} \binom{n_t - 1}{k} p^k (1 - p)^{n_t - k - 1}$$

Let $M = \max D$. For large values of n_t and as $p \in]0, 1[$:

$$\forall k \leq M, 0 < \binom{n_t - 1}{k} p^k (1 - p)^{n_t - 1 - k} \leq \binom{n_t - 1}{M} (1 - p)^{n_t - 1 - M}$$

As $0 < |D| \leq M + 1$:

$$P(\neg D) \geq 1 - \left((M + 1) \binom{n_t - 1}{M} (1 - p)^{n_t - 1 - M} \right)^{n_t} \quad (4.1)$$

Knowing that

$$\forall N, \forall k \leq N, \binom{N}{k} = \frac{N!}{k!(N - k)!} = \frac{1}{k!} \prod_{i=N-k}^N (N - i)$$

and that M is a constant not depending on n_t :

$$\binom{n_t - 1}{M} \underset{n_t \rightarrow +\infty}{\sim} \frac{n_t^M}{M!}$$

Rewriting the right side of 4.1 leads to:

$$A \binom{n_t - 1}{M} (1 - p)^{n_t} \underset{n_t \rightarrow +\infty}{\sim} \frac{A}{M!} n_t^M (1 - p)^{n_t} \quad \text{where } A = \frac{M + 1}{(1 - p)^{M+1}}$$

As $1 - p \in]0; 1[$ is not depending on n_t , $n_t^M (1 - p)^{n_t}$ tends toward 0 as n_t tends to infinity. Thus, $A \binom{n_t - 1}{M} (1 - p)^{n_t} \rightarrow 0$, thereby:

$$\begin{aligned} (P(\neg D))^{n_t} &\underset{n_t \rightarrow +\infty}{\sim} \exp \left(n_t \ln \left(1 - \frac{A}{M!} n_t^M (1 - p)^{n_t} \right) \right) \\ &\underset{n_t \rightarrow +\infty}{\sim} \exp \left(\frac{A}{M!} n_t^{M+1} (1 - p)^{n_t} \right) \end{aligned}$$

From this, $(P(\neg D))^{n_t} \rightarrow 1$, and hence we deduce the wanted theorem. ■

4.2.2.7 Case: $S_S = \mathbb{N}$ and S_C is a non-empty finite set

As $S_S = \mathbb{N}$, between two time steps, all nodes are conserved, thus, for all t , $V_t \subseteq V_{t+1}$. The first consequence is that if G_0 is not a null graph, for all t , G_t is not a null graph. Let us analyze the evolution of the order of the graph. Given t such that, there exists a vertex v such that $\deg(v) \in S_C$, then $n_{t+1} > n_t$. But, if for some t' , all $v \in V_{t'}$, $\deg(v) \notin S_C$, then for all $t > t'$, $n_t = n_{t'}$ and the graph will be static. From Theorem 11 $P(n_{t+1} > n_t) \underset{n_t \rightarrow +\infty}{\rightarrow} 0$ which implies $P(n_{t+1} - n_t = 0) \underset{n_t \rightarrow +\infty}{\rightarrow} 1$. As all vertices are conserved between two time steps:

$$P(|V_{t+1} - V_t| = 0) \underset{n_t \rightarrow +\infty}{\rightarrow} 1$$

This leads to this limit on the probability the graph becomes static:

$$P(G_t \text{ is static}) \underset{n_t \rightarrow +\infty}{\rightarrow} 1$$

Hence, for this limit case, we can conclude that the graph is asymptotically non sustainable.

4.2.2.8 Case: S_S is a non-empty finite set and $S_C = \mathbb{N}$

For all $t \in \mathbb{N}$, for all $v \in V_t$, either v duplicates (if $\deg(v) \in S_S$) or v is not conserved but is a creator. In both cases, between two consecutive time steps, each vertex is at the origin of a new one. Thus, $\mathcal{D}_t^V > 0$ and $n_t > 0$, hence, from Theorem 10, the graph is sustainable.

4.2.2.9 Summary of results

$S_S \backslash S_C$	\mathbb{N}	finite set	\emptyset
\mathbb{N}	$\forall t, n_t = 2^t n_0$ $\forall t, \mathcal{D}_t^V = 0.5$	$\forall t, n_{t+1} \geq n_t$ $\forall t, 0 \leq \mathcal{D}_t^V \leq 0.5$ $\lim_{n_t \rightarrow \infty} P(n_{t+1} > n_t) = 0$	$\forall t, G_t = G_0$ $\forall t, (\mathcal{D}_t^V, \mathcal{D}_t^E) = (0, 0)$
finite set	$\forall t, n_{t+1} \geq n_t$ $\forall t, 0.5 \leq \mathcal{D}_t^V \leq 1$ $\lim_{n_t \rightarrow \infty} P(n_{t+1} > n_t) = 0$	General cases (see Section 4.2.3)	$\forall t, n_{t+1} \leq n_t$ $\lim_{t \rightarrow \infty} n = \text{constant}$ $\lim_{t \rightarrow \infty} \mathcal{D}_t^E = (0, 0)$
\emptyset	$\forall t, n_{t+1} = n_t$ $\forall t, \mathcal{D}_t^E = (1, 1)$	$\forall t, n_{t+1} \leq n_t$ $\forall t, V_t \neq \emptyset \implies (\mathcal{D}_t^V, \mathcal{D}_t^E) = (1, 1)$	$\forall G_0, G_1 = (\emptyset, \emptyset)$

Table 4.1: Order and DynamicScore evolution for the different cases. n_t denotes the order of graph G_t , \mathcal{D}_t^V its vertices DynamicScore and \mathcal{D}_t^E the edges DynamicScore.

Stemmed from these results, sustainability property of dynamic graphs can be established. The results are reported on Table 4.2.

$S_S \backslash S_C$	\mathbb{N}	finite set	\emptyset
\mathbb{N}	Sustainable	Asymptotically non sustainable	Non sustainable
finite set	Sustainable	General cases (see Section 4.2.3)	Non sustainable
\emptyset	Sustainable	Depends on the parameters	Non sustainable

Table 4.2: Sustainability of dynamic graphs according to parameters sets of D3G3.

4.2.3 General Cases

General cases correspond to all cases for which both S_C and S_S are non empty sets and none of both sets are equal to \mathbb{N} . We classify all possible cases according to the tree represented on Figure 4.1.

The case $S_C = S_S$ composed of consecutive integers will be considered in section 4.3. In the present section we consider the cases for which $S_C \neq S_S$.

- if $S_C \cap S_S = \emptyset$ and $S_C \cup S_S \subset \mathbb{N}$ then the order of the graph is non-increasing.
- if $S_C \cap S_S \neq \emptyset$ and $S_C \cup S_S = \mathbb{N}$ then the order of the graph is non-decreasing.
- If $S_C \cap S_S = \emptyset$ and $S_S \cup S_C = \mathbb{N}$, then $|V_t| = |V_0|$, the order of the graph is constant.

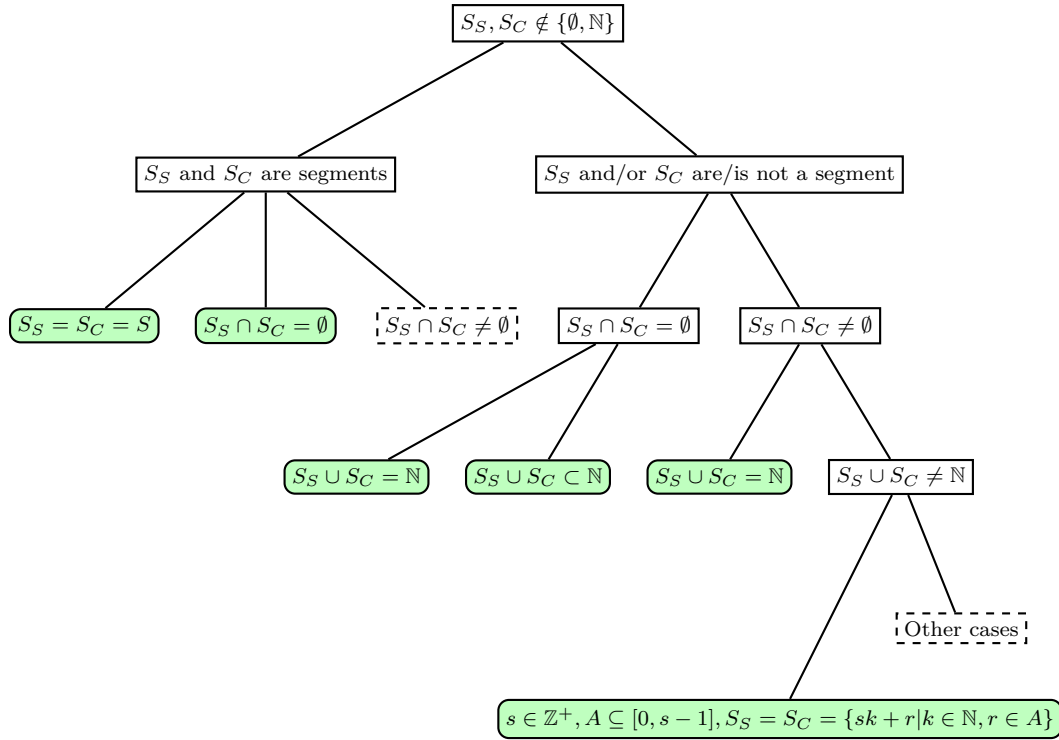


Figure 4.1: Leaves of the tree represent the general cases. Rounded corners green boxes corresponds to cases for which results are presented in this Section, in Section 4.3 and in Section 4.4. Dashed boxes are cases not covered within this report.

These three cases are carefully analyzed below. In a first time the two sets are assumed to be disjoint. On the second time they are assumed to cover the whole set \mathbb{N} . Finally a consequence of the two first results is given assuming the two sets make a partition of \mathbb{N} , which means they are disjoint and their union is \mathbb{N} .

Two different evolutions raise from the others. There is one where the order of the graph is always decreasing. This case occurs when both sets S_S and S_C does not share common elements. There is another case where the order of the graph is always increasing. This case occurs when both sets S_S and S_C cover the whole set \mathbb{N} . These results are proved in the two following theorems:

Theorem 12 *Disjoint sets*

Let $t \geq 0$ and $G_t = (V_t, E_t)$ a graph and S_S and S_C two sets of positive integers. If $S_S \cap S_C = \emptyset$, then the series $(|V_t|)_{t \geq 0}$ is decreasing.

Proof: Let consider $(G_t)_{t \geq 0}$ a generated graph. Let $t \geq 0$ and u be a vertex in V_t . Then, as $S_S \cap S_C = \emptyset$, the degree of node u can't belong to both sets. It follows that vertex u can't be both conserved and a creator. As this holds for every vertex in V_t , the order of generated snapshot graph is not increasing between two consecutive steps. ■

Theorem 13 *Union set*

Let S_S and S_C subsets of \mathbb{N} . If $S_S \cup S_C = \mathbb{N}$, then the series $(n_t)_{t \geq 0}$ is increasing.

Proof: The main argument here is the same used in the proof of theorem 12, except that the degree of every node in V_t belongs to at least one of the two sets S_S and S_C . Therefore, the order of generated

snapshot graphs is not decreasing between two consecutive steps. ■

These two theorems shows that there exists configurations of S_S and S_C such that instances of D3G3 are either non-decreasing or non-increasing. The next section deals with configuration of both sets such that the instances have a constant number of vertices.

4.2.3.1 Partition sets

In this section, S_S and S_C are considered to be a partition of \mathbb{N} . This means $S_S \cap S_C = \emptyset$ and $S_S \cup S_C = \mathbb{N}$. From Theorems 12 and 13, every graph $G_t = (V_t, E_t)$ has a constant order. Many dynamics emerge from that case. Two of them have already been treated above when dealing with limit cases (see Section 4.2.2): $S_S = \mathbb{N}$ and $S_C = \emptyset$ (non-sustainable) and $S_S = \emptyset$ and $S_C = \mathbb{N}$ (sustainable).

Other dynamics exist such as when $S_S = \mathbb{Z}^+$ and $S_C = \{0\}$. In that case, D3G3 produces graphs evolving until no isolated vertices exist. This may be an alternative method for building static geometric graphs with no isolated vertices. The only case where graphs are always sustainable is when the seed graph contains exactly one vertex. More generally, for any $m \in \mathbb{N}$, if $S_C = [0, m]$ and $S_S = [m + 1, +\infty)$, then generated graphs are always sustainable if $|V_0| \leq m + 1$. Indeed, the degree of any vertex in a graph with less than $m + 2$ vertices is always lower than $m + 1$, so between two consecutive steps it will not be conserved, but will create a new node. Therefore generated graphs always have a positive V-DynamicScore and are never empty. From application of Theorem 10, such graphs are always sustainable.

4.2.4 The Redistributed Model

Before delving into the study of general cases a simpler model is introduced. The goal of this model is to provide a tool aiming at stating, for given parameters m, M and d , whether the graph is likely to be sustainable or not. This model is studied as it helps understanding the evolution of graph order. This work has been presented during the 2023 edition of the *French Regional Conference on Complex Systems* that was held in Le Havre. In this model, between two time steps, conserved nodes are randomly repositioned on the torus. This ensures that, at each time step, the new obtained graph is a random geometric graph. The new position of the vertices are redistributed over the torus with a uniform random position, independent from the previous one. When applying this change on the conservation rule, the resulting graph G_{t+1} is a random geometric graph whose order depends on the order of G_t . This reduces dramatically the difficulty of estimating the n_{t+1} as a function of n_t . We call this model "the redistributed model" or the Redistributed Degree Driven Dynamic Geometric Graph Generator (RD3G3). It is formally defined as follows:

Definition 26 *Redistributed Degree Driven Dynamic Geometric Graph Generator (RD3G3):*

An instance of the redistributed model is defined by an initial graph, a set of integer and a rule:

- parameters:
 - $G_0 \neq (\emptyset, \emptyset)$ the initial graph,
 - $d \in]0, \frac{\sqrt{2}}{2}[$
 - S a set of non-negative integers
- rules applied on G_t leading to G_{t+1} :
 - for all $v \in V_t$ such that $\deg(v) \in S$, $v \in V_{t+1}$ (conservation rule) with a new position and a new vertex is added to V_{t+1} with a random position in the unit-torus (creation rule)

With this model it is now possible to analyze different families of parameters in order to find conditions ensuring produced graphs are sustainable. The aim of this model is to provide a first milestone in the study of generative processes and the sustainability of the dynamic graphs they produce. Indeed, estimating the evolution of the number of nodes is easier with this model than with the original D3G3 model. This difficulty lies in the structure induced by conserved vertices and is further explained in Section 4.3.4. This section highlight the difference between the RD3G3 and D3G3 through the study of the seed graph parameter G_0 .

4.3 Segments

In this section, parameters S_S and S_C are limited to equal sets of consecutive integers. Both sets are such that $S_S = S_C = [m, M]$ (called segments), with $m, M \in \mathbb{N}^2$, and referred to as S in the following. The evolution of graph order for different values of parameters m and M is investigated. The study of these parameters has led to various conjecture through the analysis of RD3G3 presented in Section 4.2.4. The purpose of this model is to provide ideas to treat D3G3 for which we do not have any theoretical tools expressing conditions for the generator to produce sustainable graphs. In order to find such conditions, the work has been separated in different parts. First, some statements and properties are theoretically and experimentally prove for a special case: $S = \{0\}$ in Section 4.3.1. For this case, a relationship connecting expected graph order at a step $t + 1$ and graph order at step t is found under some conditions. We also provide an upper bound for n_t ($t > 0$) showing generated graphs cannot exceed a finite value. Then, a theoretical analysis of RD3G3 in the general case ($S = [m, M]$) is provided. The study of the function estimating n_{t+1} knowing n_t leads to study three scenarios. These scenarios are identified according to the number of fixed-points the function has: one, two or three. The sustainability for each case is investigated and tools are provided when the answer is not clear. In the next part of this section, V-DynamicScore of graphs is studied through experimentation. It is shown to be equal in average to $\frac{2}{3}$. The reason behind this particular value is also explained in this part. Finally, open conjectures and the difference between D3G3 and RD3G3 are explained in the two last sections. They concern the sustainability of graphs and provide a glance at the results we are now working on.

4.3.1 $S = \{0\}$

The case $S_S = S_C = S = \{0\}$ is considered in this section. The seed graph, G_0 , is supposed to be a random geometric graph whose order is arbitrarily chosen. We first derive a bound on the graph order. Then the transition from step 0 to step 1 is studied. Finally, the mean value of graph order is estimated. An approximation for small values of the distance threshold d is provided. These results are then used to state about generated graphs sustainability. Then the information-persistent problem as defined in Section 3.3 is studied on instances of graphs.

4.3.1.1 Sustainability

Let us first delve into the question of the sustainability of generated graphs. The first result provided here concerns bounds in the order of generated graphs.

Theorem 14 *Bounded Graph Order:*

Let $S = \{0\}$, $d \in]0; \frac{1}{2}[$ and $G_0 = (V_0, E_0)$ such that there exists at least one node $u \in V_0$ being isolated (i.e., $\deg(u) = 0$), then, for all $t > 0$, $n_t \leq \frac{8}{\pi d^2}$

Proof

The unit torus and $0 < d < \frac{1}{2}$ are considered. Let $S = \{0\}$ and $G_0 = (V_0, E_0)$. Let $t > 0$ such that G_t is not empty. The size of V_t is maximized as soon as there is not enough free space on the torus for adding a new isolated vertex. If we consider an empty torus, a vertex can be put anywhere. The area covered by this node is equal to πd^2 . For the rest of the proof, let consider, for each vertex u on the torus, the disk of radius $d/2$ and center u and referred it as $D(u)$. From this, the condition for two nodes u and v to be non connected is: $D(u) \cap D(v) = \emptyset$. After the addition of the first vertex u_1 , a second vertex u_2 can be added to the torus if it satisfies $D(u_1) \cap D(u_2) = \emptyset$. The area occupied by the two disks is then $2 \times \pi(d/2)^2$, and the remaining free area is thus $1 - 2 \times \pi(d/2)^2$. Assuming N vertices with non intersecting disks are already present in the torus, the free area is then $1 - N\pi\frac{d^2}{4}$. This quantity has to be positive so N must be lower than or equal to $\frac{4}{\pi d^2}$. This last quantity is an upper bound for the number of isolated vertices. However, the rules says that isolated vertices duplicate. Thus, the number of vertices at one step t can not exceed twice the upper bound of isolated vertices. Hence, for all $t > 0$, $n_t \leq \frac{8}{\pi d^2}$. ■

This theorem provides an upper bound for graph order for $S = \{0\}$, ensuring graph order cannot exceed a certain value. However, this does not provide any information about graph order evolution, which is the purpose of the two following theorems.

Theorem 15 *Expected graph order at step 1:*

Let $S = \{0\}$, $d > 0$ and $G_0 = (V_0, E_0)$ be a random geometric graph of order n_0 , then $\frac{n_1}{2} \sim B(n_0, (1 - p(d))^{n_0-1})$, where $p(d)$ is the area of a circle of radius d on the torus. An expectation value for n_1 is therefore $2n_0 \cdot (1 - p(d))^{n_0-1}$.

Proof

For sake of clarity, in the remaining part of this proof, $p(d)$ will be referred to as p . At the very first step $t = 0$, $G_t = G_0$ is a random geometric graph and its nodes are uniformly distributed over the unit torus. Let $(u, v) \in V_0^2$. For a fixed threshold d , let consider the probability that u and v are connected. v is connected to u if and only if $\text{dist}(u, v) \leq d$. It means v is in the disk of center u and radius d . If we denote by $X(u, v)$ the event “ u and v are connected”, the wanted probability is the ratio of the area of the surface of a disk of radius d over the area of the unit torus. For all $(u, v) \in V_0^2$, $X(u, v) \sim B(p)$.

Let's study the degree distribution of a node $u \in V_0$. As the position of every point is independent one from the others, variables $X(u, x)$ are independent for all $x \neq u$. More over the degree of u is the number of connections u has to other nodes :

$$\text{deg}(u) = \sum_{u \neq v} X(u, v)$$

All variables $X(u, v)$ being independent for all $v \neq u$, $\text{deg}(u) \sim B(|V_0| - 1, p)$ as a sum of $|V_0| - 1 = n_0 - 1$ independent Bernoulli variables of same parameter p . Let's consider $Y_0(u)$ the event “ u is conserved at step 1” ($Y_0(u) = 1$ if and only if u is conserved and 0 otherwise), then, as $S = \{0\}$:

$$P(Y_0(u) = 1) = P(\text{deg}(u) = 0) = (1 - p)^{n_0 - 1}$$

This means $Y_0(u) \sim B((1 - p)^{n_0 - 1})$. Thus the number of conserved vertices at step 1 is:

$$Y_0 = \sum_{u \in V_0} Y_0(u)$$

As all points have an independent position, $\frac{n_1}{2} = Y_0 \sim B(n_0, (1 - p)^{n_0 - 1})$. ■

Theorem 16 (*Expected value of graph order*)

Let $S = \{0\}$, $d > 0$ and $G_0 = (V_0, E_0)$ such that there exists at least one node $u \in V_0$ being isolated (i.e., $\text{deg}(u) = 0$), then either the graph becomes empty, or the average number of conserved nodes is $l(d) = 1 - \frac{\log(\frac{\sqrt{1+4\alpha}-1}{2})}{\log \alpha}$ with $\alpha = \frac{1}{1-p}$ and $p = p(d)$.

Proof

Let $t \geq 1$. Two cases are to be discussed: the case of conserved vertices from step $t - 1$ to step t ($V_t \cap V_{t-1}$) and the case of created nodes at step t ($V_t - V_{t-1}$). As the number of created nodes is the same as the number of conserved nodes from $t - 1$ to t , we set $c_t = |V_t \cap V_{t-1}| = |V_t - V_{t-1}|$.

First let's study the number of conserved vertices from step t to step $t + 1$ among those conserved from step $t - 1$ to step t . $c_{t+1}^{\text{conserved}}$ denotes this number. Let $u \in V_t \cap V_{t-1}$. The probability for u to be conserved is the probability that its degree to created nodes remains equal to 0.

$$\text{deg}(u) = \sum_{v \in V_t - V_{t-1}} X(u, v)$$

Let $v \in V_t - V_{t-1}$. As in the previous section, $X(u, v) \sim B(p)$ and $\text{deg}(u) \sim B(c_t, p)$ as a sum of independent Bernoulli variables of same parameter p . $Y_t(u)$ denotes the event “ u is conserved at step $t + 1$ ”. The probability that u survives is $P(Y_t(u) = 1) = P(\text{deg}(u) = 0) = (1 - p)^{c_t}$, thus: $Y_t(u) \sim B((1 - p)^{c_t})$.

Therefore, the number of conserved vertices at step $t + 1$ among those conserved at step t is:

$$c_{t+1}^{\text{conserved}} = \sum_{u \in V_t \cap V_{t-1}} Y_t(u)$$

As the position of created nodes are independent from themselves and from conserved vertices, $Y_t(u)$ are independent for all $u \in V_t \cap V_{t-1}$, $c_{t+1}^{\text{conserved}} \sim B(c_t, (1-p)^{c_t})$.

Let's study the number of conserved vertices among created nodes. c_{t+1}^{created} denotes this number. Let $u \in V_t - V_{t-1}$. To study the degree of u , two cases must be studied. The first one is the number of connections between u and all other created nodes (denoted as $\deg^C(u)$). The second one is the number of connections to already present nodes (denoted as $\deg^S(u)$). $\deg^C(u)$ and $\deg^S(u)$ can be obtained using the following formulas:

$$\begin{aligned} \deg^C(u) &= \sum_{v \in V_t - V_{t-1}, u \neq v} X(u, v) \\ \deg^S(u) &= \sum_{v \in V_t \cap V_{t-1}} X(u, v) \end{aligned}$$

As the position of created points on the torus are independent one from the others, $\deg^C(u)$ is a sum of independent Bernoulli variables and therefore, $\deg^C(u) \sim B(c_t - 1, p)$. For $\deg^S(u)$, connections between a created node and an already present node are not independent from each other: knowing u is connected to an already present node means it is close to it and as other conserved nodes are farther than d , it implies that $\deg^S(u)$ is not a sum of independent Bernoulli variables. However, as a first approximation, this quantity will be considered as a sum of independent Bernoulli variables.

Thus, the computation of the expectation of $c_{t+1} = c_{t+1}^{\text{conserved}} + c_{t+1}^{\text{created}}$ gives:

$$c_{t+1} = c_t(1-p)^{c_t} + c_t(1-p)^{2c_t-1}$$

By looking for a limit to this series gives $l \geq 0$ satisfying:

$$l = l(1-p)^l + l(1-p)^{2l-1}$$

Solving this equation gives $l = 0$ or :

$$l = 1 - \frac{\log\left(\frac{\sqrt{1+4\alpha}-1}{2}\right)}{\log \alpha} \quad \text{with } \alpha = \frac{1}{1-p}$$

■

Experiments have been run to see if this relationship is accurate. The results are summarized on figure 4.2. This figure shows the accuracy of the expectation of graph order given in theorem 16. Indeed, the blue curve, close to the dashed line, highlights that theoretical expectation and experimental results are close to be equal. This result being proved leads to an approximation for small values of threshold d .

Corollary 1 *Let $d > 0$ and $l(d)$ as defined in the previous theorem (16). Then for small values of d :*

$$l(d) \sim -\frac{\log\left(\frac{\sqrt{5}-1}{2}\right)}{\pi d^2} = \frac{\log \phi}{\pi d^2}$$

where ϕ is the golden ratio $\left(\frac{1+\sqrt{5}}{2}\right)$.

Proof Let $d > 0$ be small. Thus, applying Taylor expansion gives $\frac{1}{1-\pi d^2} \sim 1+\pi d^2$ and $\log\left(\frac{1}{1-\pi d^2}\right) \sim \pi d^2$. The numerator comes from $4 \cdot \frac{1}{1-\pi d^2} \simeq 4$. The golden ratio is obtained using operations on log and by noticing that $\frac{2}{\sqrt{5}-1} = \frac{2(\sqrt{5}+1)}{4} = \phi$, the golden ratio. Combining these results leads to the statement of the corollary. ■

It is therefore possible to state that, in the case where $S = \{0\}$, it is possible to theoretically get an expectation of graph order as well as to get an upper bound for graph order depending on parameter d . Moreover, this expectation is close to the observed value meaning the generated dynamic graphs are likely to be sustainable if the time is finite. Another question is to check whether or not generated graphs are information-persistent. This is the purpose of the next paragraph.

4.3.1.2 Relationship with the Information Persistence Problem

This section focuses on studying the problem of information persistence to see if produced graphs with D3G3 and $S = \{0\}$ may lead to information-persistent graphs if they already are sustainable. It is not possible to produce information-persistent graph with such a consideration. Indeed, if we expect the condition $V_0 \cap V_T = \emptyset$ to be satisfied (no initial vertices are still present at a given time T), then the information will have vanished from the dynamic graph.

To understand this, it is essential to understand that $S_S = S_C = 0$ implies that only isolated vertices are retained and can generate new vertices in the graph. If a new vertex u connects to an isolated vertex v , then both u and v will disappear in the subsequent time step. Furthermore, only isolated vertices can retain information in the graph, as any other vertex disappears along with the information it carries. Hence, we deduce that the produced graphs when $S_S = S_C = 0$ are not A -persistent for any considered strategy A . The case $S_S = S_C = \{0\}$ therefore exhibits an example of a family of graphs, proved to be sustainable but not information-persistent. It is worth mentioning it as it might be counter intuitive.

4.3.1.3 Study of Graph Evolution

In this Section we aim at estimating the evolution of the graph order during graph dynamics. However, in the D3G3 model, between two time steps, non-conserved nodes are removed from the graph and conserved nodes are located at the same position, which entails a remanent graph. This remanent graph induces a structure influencing the computation of graph order. More precisely, nodes that are about to be removed connected to conserved ones interfere in the probability that conserved nodes at time t are still conserved at time $t+1$. This is linked to computing the degree of the neighbors of a node u knowing the degree of node u . To our knowledge, this is a difficult question. For that purpose, a relaxed version of the D3G3

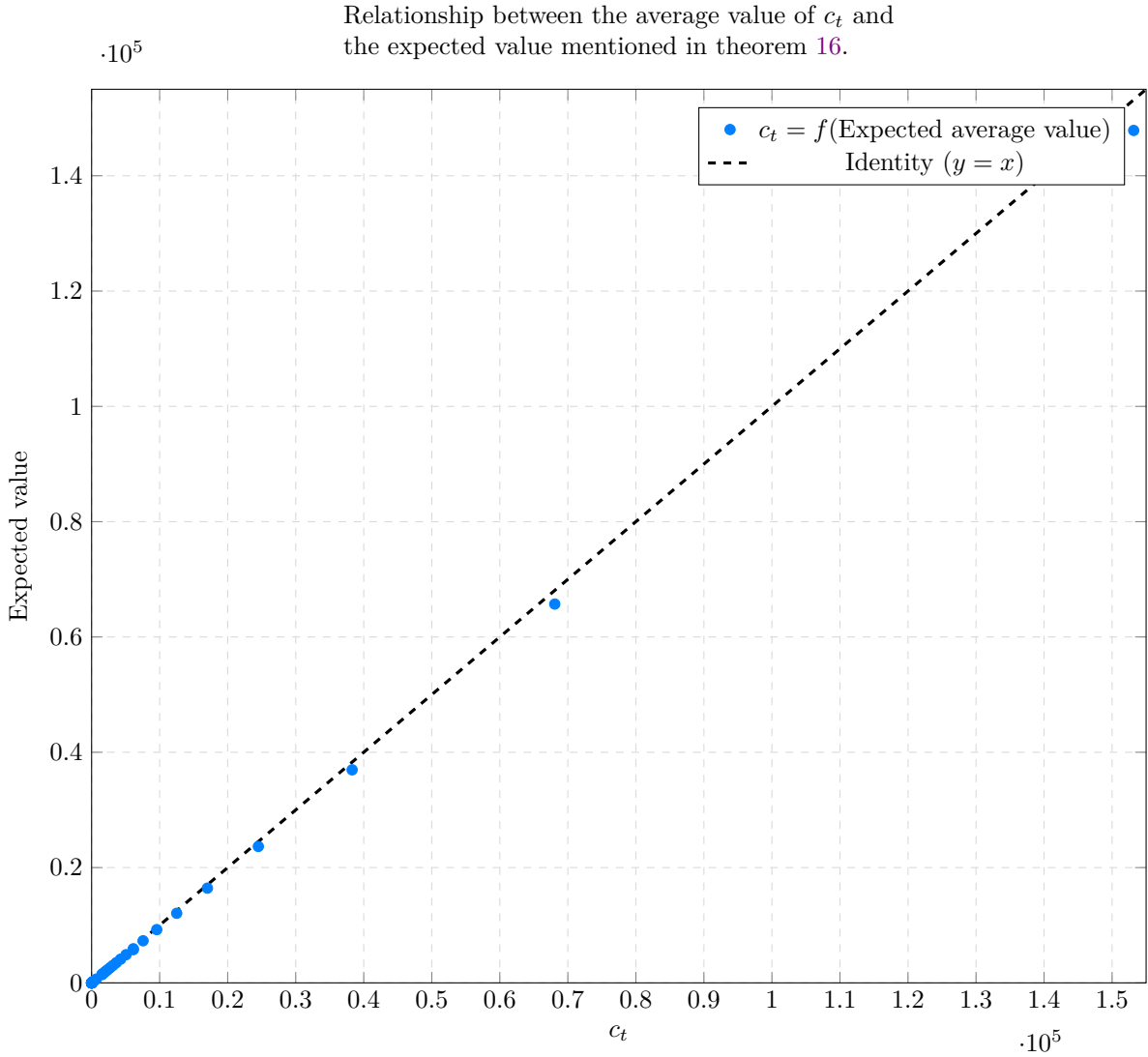


Figure 4.2: Relationship between the average value of c_t and the expected value. Each point correspond to a single threshold d . d is ranging from 0.001 to 0.01 with a step of 0.0005 and from 0.01 to 0.2 with a step of 0.005

model is considered enabling analytical study of this evolution. In this model, conserved nodes are moved (i.e., their position are changed) such that obtained graph is a new random geometric graph at each step. We call this model "the redistributed model". This will help us proving the following theorem:

Theorem 17 Let $G = (G_t)_{t \in \mathbb{N}}$ be a dynamic graph obtained with the redistributed model, then at every step t , $\frac{n_{t+1}}{2} \sim B(n_t, p(S, d, n_t))$, where $p(S, d, n_t)$ is the probability that a node is conserved between step t and $t + 1$:

$$p(S, d, n_t) = \sum_{k=m}^M \binom{n_t - 1}{k} p^k (1 - p)^{n_t - 1 - k}$$

Here, $p(d)$ refers to the probability for two different nodes to be connected (i.e., the probability that the distance between them is lower than or equal to d), which is, for $d \leq \frac{1}{2}$, πd^2 .

Proof

In the redistributed model, at time step t a RGG (G_t) is built. If the graph order at time t is equal to

n_t , the graph order at $t + 1$ is equal to twice the number of surviving nodes at time t . As every node has an independent position in the torus, this probability is the same for all nodes. Let's denote it $p(S, d, n_t)$. Let $u \in V_t$. Then:

$$p(S, d, n_t) = P(\deg(u) \in S) = \sum_{k=m}^M P(\deg(u) = k) \quad (4.2)$$

$$p(S, d, n_t) = \sum_{k=m}^M \binom{n_t - 1}{k} p^k (1 - p)^{n_t - 1 - k} \quad (4.3)$$

Assuming one node is a conserved node does not affect the probability of conservation for other nodes. The number of conserved nodes can be computed summing independent Bernoulli's events of parameter $p(S, d, n_t)$. This gives $\frac{n_{t+1}}{2}$ follows a binomial distribution of parameter n_t and $p(S, d, n_t)$. ■

Computing expectation for a binomial distribution leads to an expectation for n_{t+1} knowing n_t . Indeed, this expectation is $2n_t p(S, d, n_t)$. For a fixed set S , this provides a relationship between n_t and n_{t+1} :

Definition 27 *Expectation of graph order:*

Let m, M and d be parameters for the redistributed model. Let $G = (G_t)$ be an obtained graph with such parameters. Then, the expectation of graph order at step $t + 1$ (n_{t+1}) knowing graph order at step t (n_t) is $f_{S,d}(n_t)$ and satisfies $n_{t+1} = f_{S,d}(n_t) = 2n_t p(S, d, n_t)$, and then:

$$\forall n \in \mathbb{N}, f_{S,d}(n) = 2np(S, d, n) \quad (4.4)$$

This quantity is referred to as the relationship in the sequel. Studying the relation for every value of m, M and d turns out to be a difficult problem. However some results may be conjectured. A first conjecture concerns the variations of the relationship:

Conjecture 1 *Let m, M and d be parameters of the model. Let $S = [m, M]$ and $f_{S,d}$ the relationship as defined above. Then there exists $n_* \in \mathbb{N}$ such that $f_{S,d}$ is increasing on $[0, n_*]$ and decreasing on $[n_* + 1, +\infty[$.*

This conjecture is difficult to prove due to the sum involved in the computation of $f_{S,d}$. However, it is not necessary to study the relationship for all integers. It is possible to perform the study on a limited interval. This is the purpose of theorem 18 (below). But before proving this theorem, it is necessary to provide another formulae computing variations of $f_{S,d}$:

Lemma 5 *Let m, M and d be parameters of the model. Let $\Delta f_{S,d}$ defined as the variation of $f_{S,d}$: for $n \in \mathbb{N}$, $\Delta f_{S,d}(n) = f_{S,d}(n + 1) - f_{S,d}(n)$. Then:*

$$\forall n \in \mathbb{N}, \Delta f_{S,d}(n) = 2 \sum_{k=m}^M (k + 1) \binom{n}{k} p^k (1 - p)^{n - 1 - k} \left(1 - \frac{n + 1}{k + 1} p \right)$$

Proof: Let m, M and d be parameters of the model. Let n a be non-negative integer. This proof only

focuses on the terms of the sum of $\Delta f_{S,d}$:

$$\begin{aligned}\Delta f_{S,d}(n) &= 2 \left(\sum_{k=m}^M (n+1) \binom{n}{k} p^k (1-p)^{n-k} - n \binom{n-1}{k} p^k (1-p)^{n-1-k} \right) \\ &= 2 \sum_{k=m}^M p^k (1-p)^{n-1-k} \left((n+1) \binom{n}{k} (1-p) - \binom{n-1}{k} \right)\end{aligned}$$

Let $k \in \mathbb{N}$ such that $m \leq k \leq M$. Every term of the sum of $\Delta f_{S,d}$ can be expressed as follow only using results on binomial coefficients:

$$\begin{aligned}(n+1) \binom{n}{k} (1-p) - n \binom{n-1}{k} &= (k+1) \binom{n+1}{k+1} (1-p) - (k+1) \binom{n}{k+1} \\ &= (k+1) \left(\binom{n+1}{k+1} - p \binom{n+1}{k+1} - \binom{n}{k+1} \right) \\ &= (k+1) \left(\binom{n}{k} - p \binom{n+1}{k+1} \right) \\ &= (k+1) \left(\binom{n}{k} - p \frac{n+1}{k+1} \binom{n}{k} \right) \\ &= (k+1) \binom{n}{k} \left(1 - p \frac{n+1}{k+1} \right)\end{aligned}$$

This last equality leads to the following form of $\Delta f_{S,d}$:

$$\Delta f_{S,d}(n) = 2 \sum_{k=m}^M (k+1) \binom{n}{k} p^k (1-p)^{n-1-k} \left(1 - \frac{n+1}{k+1} p \right)$$

■

It is now possible to state the following theorem about variations of $f_{S,d}$:

Theorem 18 *Let m, M and d be the parameters of the model. Let $S = [m, M]$ and $f_{S,d}$ the relationship as defined above. Let $p = p(d)$ be the probability for two different nodes to be connected. Then, $f_{S,d}$ is increasing between 0 and $\frac{m+1}{p} - 1$ and decreasing from $\frac{M+1}{p} - 1$ to infinity.*

Proof: The goal is to prove that $\Delta f_{S,d}(n)$ is positive for $n < \frac{m+1}{p} - 1$ and negative for $n > \frac{M+1}{p} - 1$. To understand this, $\Delta f_{S,d}(n)$ can be rewritten as follow (lemma 5):

$$\forall n \in \mathbb{N}, \Delta f_{S,d}(n) = 2 \sum_{k=m}^M (k+1) \binom{n}{k} p^k (1-p)^{n-1-k} \left(1 - \frac{n+1}{k+1} p \right)$$

It is sufficient to notice that, for all $k \in S$, the sign of every single term of the sum is the sign of $\left(1 - \frac{n+1}{k+1} p \right)$. For fixed k , the term is positive if and only if n is lower than $\frac{k+1}{p} - 1$. As this last term is an increasing function of k , all terms of the sum are therefore positive if n is lower than $\frac{m+1}{p} - 1$ and negative if n is greater than $\frac{M+1}{p} - 1$. Hence, the relationship is increasing from 0 to $\frac{m+1}{p} - 1$ and decreasing from $\frac{M+1}{p} - 1$ to infinity. ■

Thanks to theorem 18, conjecture 1 is proved for intervals $[0, x_m]$ and $[x_M, \infty[$ $x_m = \frac{m+1}{p} - 1$ and

$x_M = \frac{M+1}{p} - 1$. At this stage, quantifying more precisely the evolution of the graph order is not achievable. However, a study of the fixed points of $f_{S,d}$ enables to draw some conclusion about generated graphs sustainability.

4.3.1.4 Graph Evolution and Sustainability

First note that knowing the variations of $f_{S,d}$ is not enough to deal with graphs sustainability. Indeed, as claimed by the following theorem, big graphs are not sustainable.

Theorem 19 *Non-sustainability of big graphs:*

Let m , M and d be parameters of the model. Let $f_{S,d}$ be the relationship. Then, there exists $N > 0$ such that for all $n > N$, $f_{S,d}(n) < 1$.

Proof: For this proof, it is sufficient to prove that $f_{S,d}(n) \rightarrow 0$ when $n \rightarrow +\infty$. Let n such that $n \geq 2M + 1$. In this situation, for all $k \leq M$, binomial coefficient $\binom{n-1}{k} \leq \binom{n-1}{M}$. Moreover, as $(1-p) < 1$, $x \mapsto (1-p)^x$ is decreasing. Therefore, for all $k \leq M$, $(1-p)^{n-1-k} \leq (1-p)^{n-1-M}$. It is thus possible to get the following inequality for all $k \leq M$:

$$\binom{n-1}{k} (1-p)^{n-1-k} p^k \leq \binom{n-1}{M} (1-p)^{n-1-M} p^k$$

Noticing $p < 1$ and $f_{S,d}$ is a sum of $M - m + 1$ elements, $f_{S,d}(n)$ can be bounded as follow

$$f_{S,d}(n) \leq 2n(M - m + 1) \left(\binom{n-1}{M} (1-p)^{n-1-M} \right)$$

As M is fixed, the binomial coefficient $\binom{n-1}{M}$ is equivalent to a polynomial of degree M as n grows to infinity:

$$\binom{n-1}{M} \sim \frac{n^M}{M!}$$

Therefore, $f_{S,d}(n)$ is equivalent to the product of a polynomial and an exponential function converging towards 0. This implies $f_{S,d}(n)$ converges towards 0 as n tends to infinity. ■

This theorem says that there always exists a graph order limit such that graphs whose order are greater than this limit are likely to become empty. Therefore, it is not possible to obtain sustainable graphs with a large amount of nodes.

A new mathematical concept is introduced with the aim of studying graph sustainability. This concept is referred to as a fixed point. The analysis of generated dynamic graphs has led to the introduction of this concept for a specific reason: it aims to classify parameters into distinct categories according to functions $f_{S,d}$. These categories, detailed above, provide a framework within which the sustainability of graphs can be explained based on the category to which a dynamic graph belongs. Fixed points are defined as follows:

Definition 28 *Fixed Point:*

Let m , M and d be parameters of the model. A fixed point for the relationship $f_{S,d}$ is a non-negative integer n such that:

$$\begin{cases} f_{S,d}(n) \leq n \text{ and } f_{S,d}(n+1) > n+1 \\ \text{or } f_{S,d}(n) \geq n \text{ and } f_{S,d}(n+1) < n+1 \end{cases}$$

Such fixed points characterize variation of graph order. Indeed, graph of order n for n taken between two consecutive fixed points is either always decreasing or increasing. From experiments performed on the redistributed model as well as on D3G3, three different cases appear and are conjectured as follow:

Conjecture 2 For all m , M and d being parameters of the model, the relationship $f_{S,d}$ has either one, two or three fixed points.

This conjecture is the main tool aiming at studying sustainability in the segment case. Indeed, in the three different cases, it is possible to answer whether a given set of parameters is sustainable or not. However, there is no characterization about parameter values that may help founding which case parameters lead to. The only one claim that can be made is that d does have an influence on this case.

The conjecture 2 is assumed in this subsection. This section aims at stating about sustainability in the three different cases. This is illustrated by a description of the behavior of the relationship $f_{S,d}$ in every case.

One fixed point First let's consider the case where the relationship has only one fixed point. When it has only one fixed point, this point is 0. This comes from $f_{S,d}(0) = 0$. Moreover, for all n , $f_{S,d}(n) < n$. As for a snapshot graph of order n_t at step t , $f_{S,d}(n_t)$ gives the expectation value of n_{t+1} at step $t+1$. Graph orders of generated graphs are decreasing in average. Graphs obtained in this case are therefore not sustainable. This is illustrated by Figure 4.3.

Two fixed points For the two fixed points case, 0 is also a fixed point. This also comes from $f_{S,d}(0) = 0$. The other fixed point is greater than zero. The case where $f_{S,d}$ have two fixed points were conjectured in our work to be a consequence for m , the lower bound of S , to equal to 0. It was also conjectured that $m = 0$ implies $f_{S,d}$ has two fixed points. In the following theorem, we finally give a proof of that result which lies in the value of $f_{S,d}(1)$:

Theorem 20 (*Characterization of the two fixed points*) Assuming the hypothesis that $f_{S,d}$ has either one, two or three fixed points, then the relationship $f_{S,d}$ has two fixed points if and only if $m = 0$.

Proof: To prove this statement, let us reason through double implications. First, we prove that $m = 0$ implies $f_{S,d}$ has exactly two fixed points. Let assume $S = [0, M]$ for any $M \in \mathbb{N}$. Then, for all $n \leq M+1$, $f_{S,d}(n) = 2n$. This comes from $\binom{n-1}{k} = 0$ for any $n \in \mathbb{N}$ and $k \geq n$. Thus, for all $n \in \mathbb{N}$ and $k \geq n$, the following holds:

$$f_{S,d}(n) = 2n \left(\sum_{k=0}^{n-1} \binom{n-1}{k} p^k (1-p)^{n-1-k} \right) = 2n$$

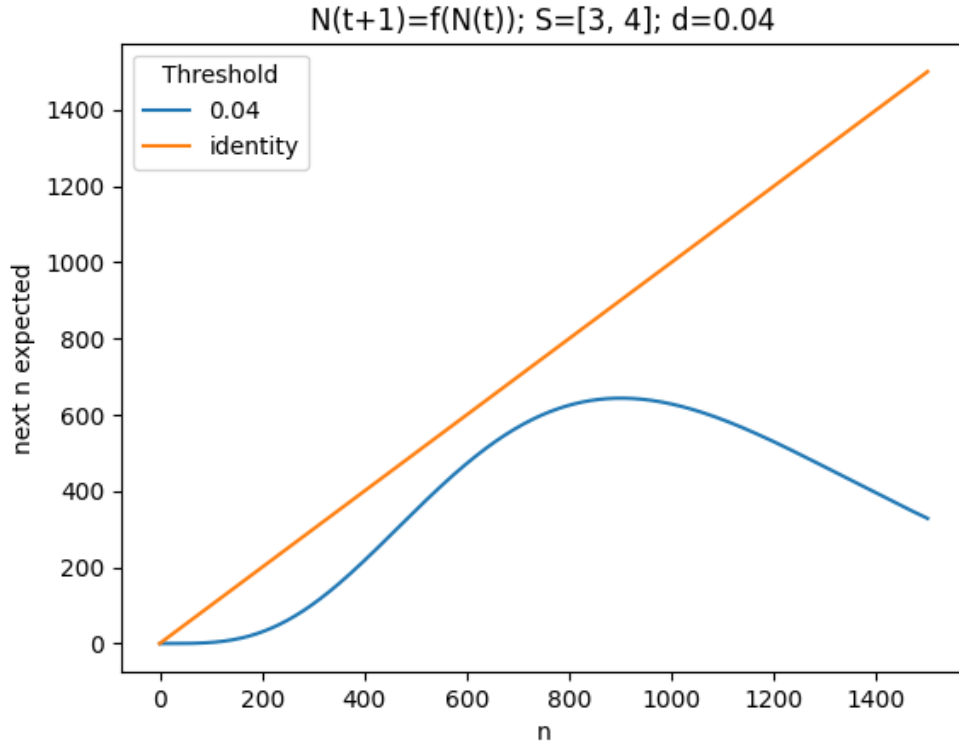


Figure 4.3: One fixed point

This implies at least $f_{S,d}(1) = 2 > 1$. However, from theorem 19, there exists $n_1 > M + 1$ such that $f_{S,d}(n_1) < 1 < n_1$. As $f_{S,d}(M + 1) = 2(M + 1)$ and $f_{S,d}(n_1) < 1$, there exists $n_2 \in [M + 2, n_1]$ satisfying $f_{S,d}(n_2) \geq n_2$ and $f_{S,d}(n_2 + 1) < n_2 + 1$ (for the rest of the proof, let assume n_2 is the smallest positive integer satisfying this property). Therefore, $f_{S,d}$ has at least two fixed points. The last step is to prove $f_{S,d}$ does not have three fixed points. Let assume $f_{S,d}$ does have three fixed points. There exists $n_3 > n_2$ such that $f_{S,d}(n_3) \leq n_3$ and $f_{S,d}(n_3 + 1) > n_3 + 1$. Thus, for the same reason n_2 exists, there also exists $n_4 \in [n_3 + 2, n']$ satisfying $f_{S,d}(n_4) \geq n_4$ and $f_{S,d}(n_4 + 1) < n_4 + 1$, which implies $f_{S,d}$ has four fixed point, which is a contradiction with the hypothesis of the theorem. This means $f_{S,d}$ has exactly two fixed points.

Let us now proof $f_{S,d}$ has two fixed points implies $m = 0$. Let assume first, $m \geq 1$. In that case, the binomial coefficient $\binom{n-1}{k} = 0$ for $n = 1$, implying $f_{S,d}(1) = 0$. Two cases occur, either no fixed point exists or at least one. On the one hand, only 0 is a fixed point, so $f_{S,d}$ has exactly one fixed point. On the other hand, the smallest positive fixed point n must satisfy $f_{S,d}(n) \leq n$ and $f_{S,d}(n + 1) > n + 1$ because $f_{S,d}(1) = 0$. Thus, there exists $n' \geq n + 1$ such that $f_{S,d}(n') \geq n'$ and $f_{S,d}(n' + 1) < n' + 1$. It follows that $f_{S,d}$ has three fixed points and $m \geq 1$ cannot lead to $f_{S,d}$ with two fixed point. By contraposition, we deduce the wanted implication. ■

A consequence for this theorem from Theorem 19 is that generated graphs when $m = 0$ are sustainable as long as their graph order does not exceed a limit. Such a limit has not been computed theoretically. Its existence is a consequence of Theorem 19. Graphs whose order exceeds this limit are likely to become

empty. An example of parameters leading to $f_{S,d}$ having two fixed points is illustrated by figure 4.4.

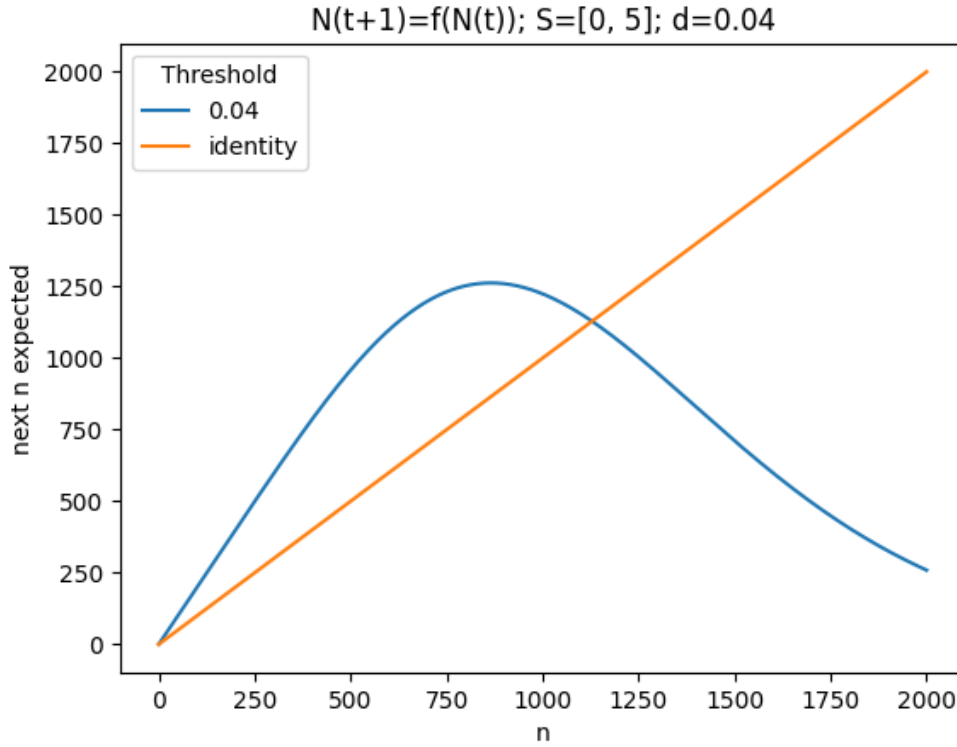


Figure 4.4: Two fixed points

Three fixed points For the last case, the goal is to show that graph order is likely to remain bounded. Deeply looking at this case raises the question of values of graph order for which the size is not too large and not too small so that it does not collapse. For that purpose we define an interval, called *sustainable interval*, such that, if the graph order remains within that interval, this ensures the persistence of the graph. This sustainable interval is considered as a tool to study graph sustainability. It concerns expectation of graph order evolution through time. It says that if the image of the function $f_{S,d}$ for all integers within the interval does not exceed the upper bound, then the graph is likely not to collapse. Let's define more precisely this concept:

Definition 29 *Sustainable Interval:*

Let m , M and d be parameters of the model. Let consider $f_{S,d}$ such that it has three fixed points. Let N_m be the first positive fixed point and N'_m the smallest integer greater than N_m such that $f_{S,d}(N'_m) \geq N_m$ and $f_{S,d}(N'_m + 1) < N_m$ (N'_m exists as a consequence of Theorem 19). The sustainable interval associated to m , M and d is defined as the interval $[N_m, N'_m]$.

This definition is illustrated through figure 4.5. Such an interval satisfies a property about the values $f_{S,d}$ takes when it is restricted to it:

Theorem 21 *Sustainability in the sustainable interval:*

Let m , M and d be parameters of the model. Let assume the relationship $f_{S,d}$ has three fixed points and

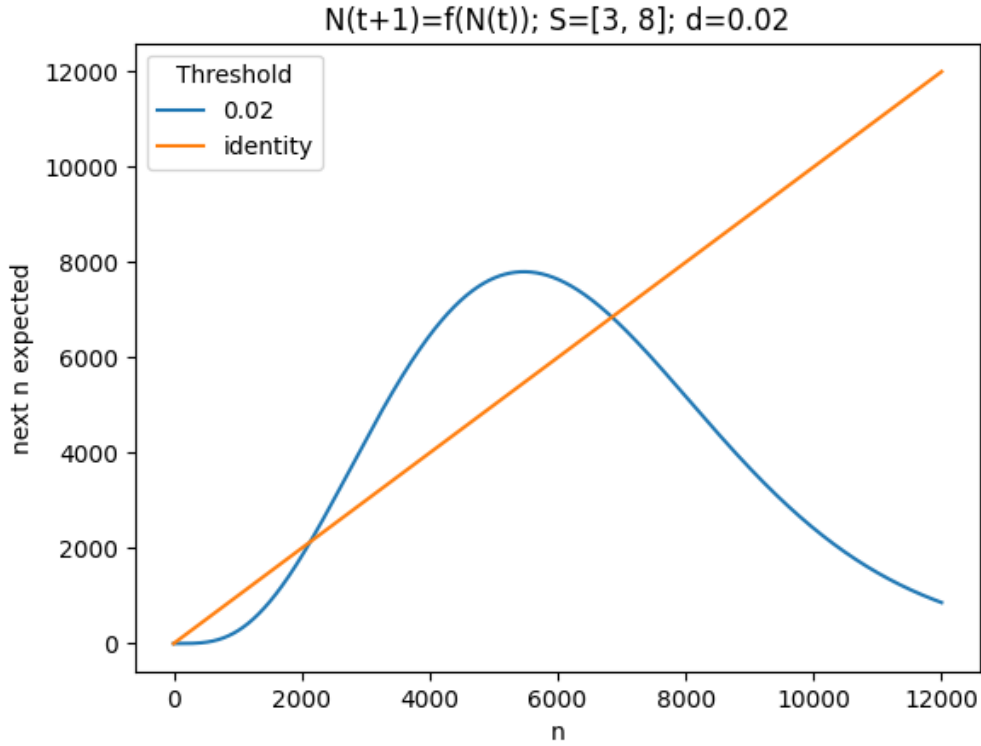


Figure 4.5: Three fixed points

that $[N_m, N'_m]$ is its associated sustainable interval. Then, the relationship satisfies:

$$\forall n \in \mathbb{N}, n \in [N_m, N'_m] \Leftrightarrow f_{S,d}(n) \geq N_m$$

Moreover, if the $f_{S,d}$ does not exceed N'_m , then it satisfies:

$$\forall n \in [N_m, N'_m], f_{S,d}(n) \in [N_m, N'_m]$$

If such a condition is observed, then, graphs produced with a seed graph with an order within $[N_m, N'_m]$ are likely to be sustainable.

Proof: Let assume $f_{S,d}$ first increases and then decreases (see Conjecture 1) and it has 3 fixed-points. As a consequence of Theorem 20, $m \geq 1$, meaning $f_{S,d}(1) = 0 < 1$. Thus, as N_m is the first positive fixed-point and as $f_{S,d}$ increases, every $n \in [0, N_m]$ satisfies $f_{S,d}(n) \in [0, N_m]$. As $f_{S,d}(n)$ tends toward 0 as n tends to infinity (see Theorem 19), then there exists N'_m such that for every $n \geq N'_m$, $f_{S,d}(n) \leq N_m$. For the value of $n \in [N_m, N'_m]$, they satisfy $f_{S,d}(n) \geq N_m$ as a consequence of Conjecture N_m . The second part of the theorem is just a remark. ■

Main interpretation of that theorem is that graphs are sustainable with high probability in the sustainable interval if and only if there are no values of $f_{S,d}$ exceeding the upper bound of the sustainable interval.

Building the Sustainable Interval The following paragraphs provide arguments aiming at obtaining the sustainable interval. They also provide arguments to check whether the relationship exceeds the upper bound of the interval. The theorem 18 clearly gives bounds to find out the maximum of the relationship $f_{S,d}$. Three algorithms are sufficient to answer both questions: an algorithm to compute the argument of the maximum of the relationship $f_{S,d}$, an algorithm to find its fixed point between 0 and the argument of the maximum and an algorithm to solve $f_{S,d}(n) = y$ for n greater than the argument of the maximum and $y > 0$ lower than or equal to the maximum. In the following, these algorithms are first implemented. It is then explained how to use them to answer questions about the sustainable interval.

The argument maximum: To compute the argument maximum of the relationship, it is sufficient to study $f_{S,d}$ on the interval $[x_m, x_M]$ for x_m and x_M as defined above. This is a consequence of theorem 18. Let's denote it N_* .

The first positive fixed point: To find the fixed point of $f_{S,d}$ mentioned in the definition of the sustainable interval, it is sufficient to compute the argument maximum of it. The previous algorithm answers this question. Then, as the relationship is increasing from 0 to N_* , it is sufficient to iterate and find an integer n such that $f_{S,d}(n) \leq n$ and $f_{S,d}(n+1) > n+1$.

The solution of the equation: For the last algorithm, the goal is to find an integer n such that n is greater than N_* of $f_{S,d}$, $f_{S,d}(n) \geq y$ and $f_{S,d}(n) < y$, for a fixed y which is assumed to be positive and lower than the maximum of $f_{S,d}$.

From these algorithms it is possible to implement algorithms stating the existence of the sustainable interval and its bounds. For the existence or not of the sustainable interval, it is sufficient to check whether the maximum of the relationship is greater than its argument. This comes from that sustainable interval exists if and only if there are values of the relationship that exceed their argument. As the relationship is increasing from 0 to $f_{S,d}(N_*)$, then sustainable interval exists if and only if $f_{S,d}(N_*) > N_*$. For computing the sustainable interval boundaries, it is sufficient to know the value of the first fixed point N_m (as it provides the lower bound) and to solve the equation $f_{S,d}(x) = N_m$ as finding the corresponding x to this equation provides the upper bound (N'_m). The existence of N'_m is ensured by theorem 19.

4.3.2 Vertex DynamicScore

The goal is to highlight a characterization aspect of the segment family using the V-DynamicScore metric. As edge DynamicScore will not be studied for that case, vertex DynamicScore will be referred to as DynamicScore in this section. As in this particular configuration, conserved vertices are the same as created nodes, it is possible to state particular results about the value of DynamicScore:

Theorem 22 *Let S be a segment set of non-negative integers and $d \in]0, \frac{1}{2}[$. Let G be a generated graph*

of order n_t at step t and number of conserved nodes from step t to step $t + 1$ referred to as s_t . Then:

$$\mathcal{D}_t^V = \frac{n_t}{n_t + s_t}$$

Proof To prove this result, it is sufficient to notice that $n_{t+1} = 2s_t$, as $S_S = S_C$, which means the number of conserved nodes is the same as the number of created nodes. Thus, applying some basic result about set sizes and noticing that $s_t = |V_t \cap V_{t+1}|$, leads to:

$$\begin{aligned} |V_t \cup V_{t+1}| &= n_t + n_{t+1} - s_t = n_t + s_t \\ |V_t \Delta V_{t+1}| &= n_t + n_{t+1} - 2|V_t \cap V_{t+1}| = n_t \end{aligned}$$

It follows that the vertex DynamicScore is equal to $\frac{n_t}{n_t + s_t}$. ■

Result about the DynamicScore observed in generated graphs parameterized with a segment set S is stated in the following conjecture:

Conjecture 3 *Let $m, M \in \mathbb{N}$. Let $S = [m, M]$ and $d > 0$ be parameters of RD3G3. Then the DynamicScore of generated graphs is in average equal to $\frac{2}{3}$.*

Although this conjecture has not been proved theoretically, experimentation have been performed. They are in line with this conjecture telling that the average DynamicScore of generated graphs is roughly equal to $\frac{2}{3}$. Results of this experimentation are gathered on picture 4.6. A possible interpretation of this conjecture and performed experimentation relies on the result stated in theorem 22 and on results from last part. Indeed, if vertex DynamicScore is close to $\frac{2}{3}$, it means $s_t \simeq \frac{n_t}{2}$. Then, as $n_{t+1} = 2s_t$, it comes $n_{t+1} \simeq n_t$, which means that the graph order is close to a fixed point of the relationship $f_{S,d}$ mentioned in the previous section.

4.3.3 Conjecture on the Sustainability

When studying the generator process D3G3, a conjecture quickly emerged. It suggested that, for any chosen value of m , there exists, for a sufficiently small value of d , an integer M such that for the D3G3 generator with parameters $S = [m, M]$ and d , most of the dynamic graphs produced by the process would be sustainable. To make this conjecture more formal and easier for the reader to follow, the following statement is provided:

Conjecture 4 *Let m be a non-negative integer. There exists $d \in [0; \frac{\sqrt{2}}{2}]$ and $M \geq m$ such that, the probability a dynamic graph produced using D3G3 is sustainable is close to 1.*

This conjecture has been elaborated after testing several configurations of parameters m , M and d .

For d greater than 0.1, simulations exhibited graphs that quickly become empty, and therefore non-sustainable. Conversely, the smaller the tested value of d , the higher the proportion of graphs that reach the end of the simulations, approaching or even reaching 100%. Regarding the study around parameter M , simulations revealed that when m is fixed and M increases, a threshold effect occurs on the rate of

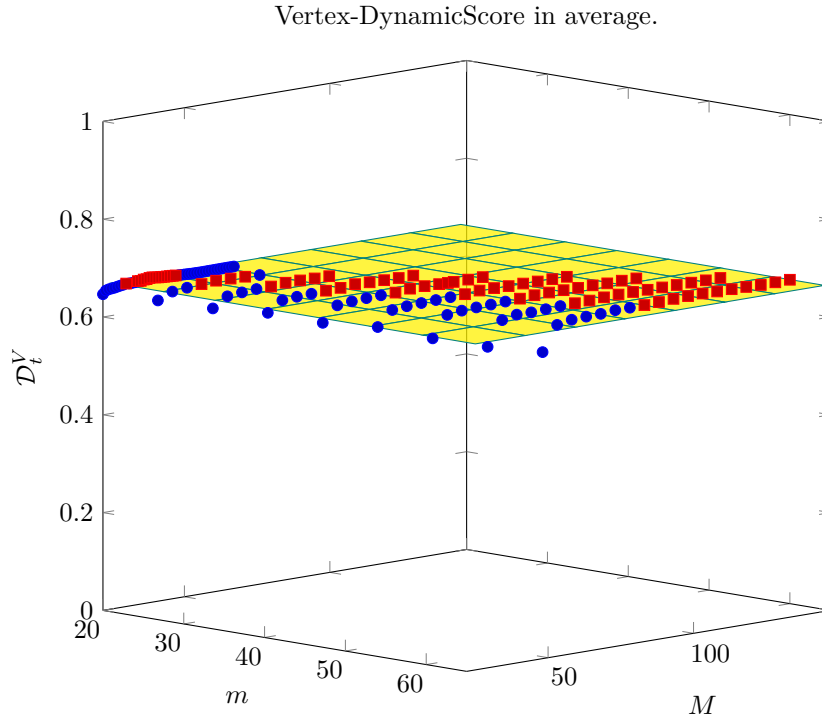


Figure 4.6: Mean value of vertex DynamicScore got from experimentation. Points represent the average over 20 run and 30000 time steps for a single m and M . The yellow surface is the plan of equation $z = \frac{2}{3}$. For all these parameters, d is set to 0.05. Red points represent DynamicScore greater than $\frac{2}{3}$. Blue points represent DynamicScore lower than $\frac{2}{3}$.

sustainable graphs. This threshold effect indicates that once M exceeds a certain value, the proportion of graphs reaching the maximum number of iterations jumps from less than 10% to nearly 100%. In the initial tests, values of m ranged between 1 and 8. To conduct these tests, values of M were chosen based on the values of m : from $m + 1$ to $4m$ (more values were tested for small values of m to clearly observe this threshold effect). For each pair of tested m and M , 100 tests were conducted, each consisting of 1,000 iterations. The graphs were considered sustainable if they reached the last iteration and were not empty. Although the tested values of m and M were small, this allowed us to form initial hypotheses to be tested. One initial hypothesis was to determine conditions on M for the conjecture to be true. More precisely, the goal was to find a function of f such for $M \geq f(m)$, produced graphs are sustainable. The success rate of all simulations conducted for each (m, M) pair is represented in Figure 4.7. This Figure shows the percentage of sustainable simulations. The value of parameter d is set to 0.05. Other values of d lower than 0.1 also shows this threshold effect, and the choice to represent this value is arbitrary. This graph highlights the emergence of a threshold effect for the value of M relative to m . This initial observation supports the conjecture, even suggesting the existence of a threshold value M' such that for $M \geq M'$, the graphs produced by D3G3 would be sustainable with a high success rate.

However, a deeper study showed us that this observation was not correct, and even worse, that for larger values of m , there was no M that would result in sustainable graphs. An initial observation of this phenomenon occurred when we tested parameters like $m = 13$ and $M = 31$. For this case $M \geq 2m$, yet the simulations performed with these parameter values showed a very low success rate. The result of these

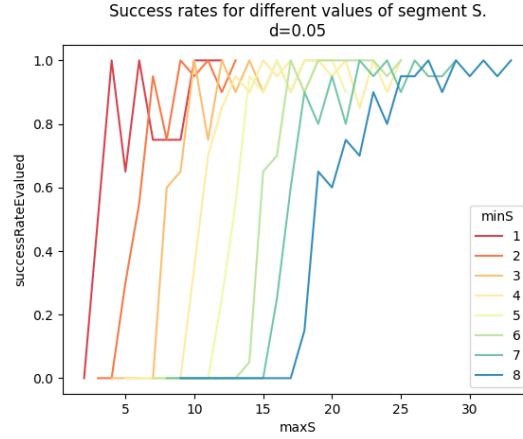


Figure 4.7: Success rate of simulations for m ranging from 1 to 8. Each curve represent a different value of m . The x-axis is a value of M and the y-axis is the success rate.

simulations are detailed in Section 4.3.4. It turned out that the graph produced often tended to grow significantly before collapsing. This did not happen for smaller values of M . We then assumed that there was not a threshold effect but rather a step effect. This means that for a given m , the values of M that result in sustainable graphs, with high probability, are contained within an interval $[M_1, M_2]$. We then tested much larger values of m to examine this hypothesis. Tested values of parameter m ranged were 200 and 300. For these values of m , we did not find any M such that graphs obtained were sustainable. We then tried to understand how this could be possible. One argument, which is unfortunately not a proof, is when we studied the sustainability interval. For each values of M , only one produced a sustainable interval with no value exceeding the upper bound. However, this area is too small to ensure the graph to be sustainable. For all other case, either there is no sustainable interval or the function $f_{S,d}$ exceed the interval. Thus, for such value of m there is no M such that produced graphs are sustainable.

4.3.4 The Purpose of the Non-redistributed Model

The reader may wonder why did we study the non-redistributed model while the redistributed one already provide a wide possibility for which sustainability is an interesting question. The reason is that stating about sustainability requires different answers according to the studied model. Indeed, sustainability conditions are different between the redistributed and the non-redistributed model. On the one hand, conditions on the redistributed model seems to depend only on the graph order. It is therefore sufficient to study $f_{S,d}$ to already have an answer about the sustainability of produced graphs. On the other hand, the sustainability conditions for the non-redistributed does not depend only on graph order. Indeed, geometric structures of small graphs influence their sustainability. This is a consequence of conserved vertices that does not move from one step to the next one. Therefore conserved vertices are more likely to still remain present when the graph is small as the number of interactions with other vertices are unlikely. This may be observed on graphs such as those generated with parameters $S = [13, 31], G_0$ complete geometric graph composed of 14 vertices and d ranging from 0.02 to 0.1 for instance. For these configurations graphs start from a complete graph and grow until a threshold value depending on parameter d . An illustration of this evolution is given Figure 4.8 representing the evolution of the graph

order and the V-DynamicScore for one obtained graph. On this graph, it is possible to see that up to step 400 the graph is in a growing phase. During this phase the value of the V-DynamicScore is stable and close to $\frac{2}{3}$. As the graph order does not increase too much between two consecutive steps, this means around half the number of vertices are conserved while other are new. Then after this step, the graph features start oscillating meaning the graph enter into a different phase. This change of phase is an evidence of the existence of a growing structure maintaining the graph sustainable. However, we do not have proof of the existence of this structure. Moreover, the conditions that ensure such a structure makes the graph grow up to a certain size are also unknown. The only thing we can say is that such a scenario does not occur in the case of graphs generated with RD3G3 using the same parameters. For this setup, all becomes empty after two steps: the graph at the second iteration step is a random graph with $2n_0$ vertices which implies too few vertices satisfy the duplication condition.

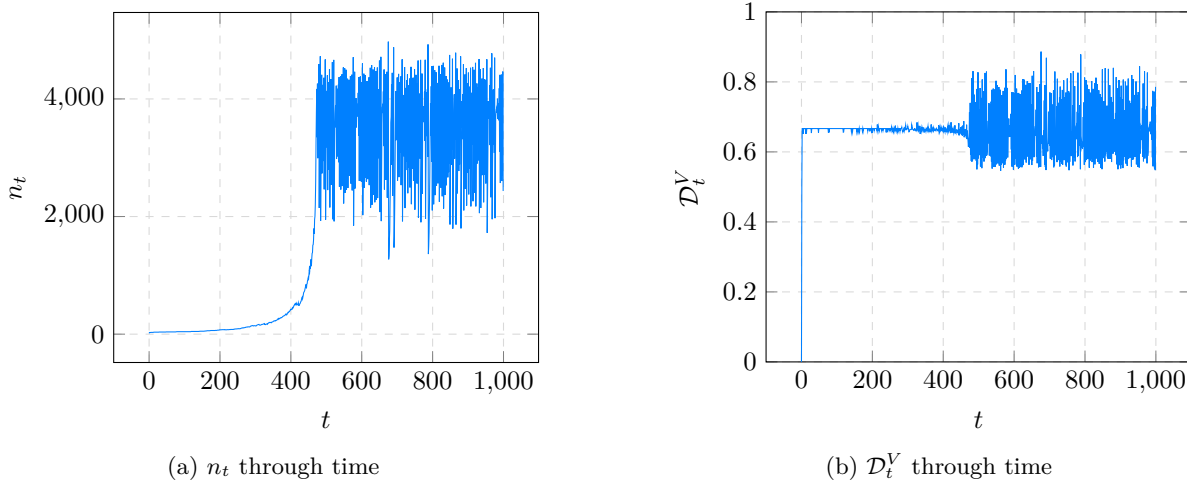


Figure 4.8: Example of the evolution of a dynamic graph produced by D3G3. Parameters are $S = [13, 31]$ and $d = 0.05$.

Simulations have been run to seek for graphs reaching a fixed number of iterations. Graphs reaching this condition are considered sustainable up to this fixed date. For every value of d tested, 20 simulations were run. The ratio of sustainable graphs obtained from this experiment is represented Table 4.3. On this table it is possible to see that the ratio of sustainable graphs is non null though decreasing. For small values of d , sustainability is observed on a large amount of run showing the main difference between the redistributed model and the original one. In addition, these results also support Conjecture 4, at least on finite time.

d	Sustainability rate
0.02	0.9
0.03	0.8
0.04	0.8
0.05	0.7
0.06	0.65
0.07	0.95
0.08	0.55
0.09	0.5
0.1	0.25

Table 4.3: Ratio of sustainability graphs for different thresholds d . Each run has a limit of 3500 iterations.

Conclusion on Segments

The focus of this section is to identify conditions that ensure the sustainability of dynamic graphs for a specific family of parameters: both sets S_S and S_C are equal and contain consecutive non-negative integers (i.e., $S = S_S = S_C = [m, M]$, for $m, M \in \mathbb{N}$). In this setup, a new model, called the redistributed model (RD3G3), is introduced as a simplified version of the D3G3 model for theoretical analysis. This model modifies the conservation rule such that the positions of conserved vertices are adjusted in the same way that newly created vertices are placed on the torus.

This section investigates the sustainability of both models, highlighting their fundamental differences. In the D3G3 configuration, we first proved that when both sets contain only zero, the produced graphs are likely to be sustainable when d is close to zero. Additionally, we demonstrated that when the seed graph is a complete geometric graph with $n_0 = m + 1$ vertices, the graphs are likely to be sustainable for small values of parameter d . The exact cause behind the phenomenon ensuring sustainability in these graphs remains unknown; however, the presence of a growing structure might provide an explanation.

For the RD3G3 model, sustainability is analyzed through a function estimating the evolution of the number of vertices over time. This analysis led to a conjecture that there are only three different scenarios for analyzing sustainability. According to this conjecture, the estimating function has either one, two, or three fixed points. If the function has one fixed point, the graphs are almost certainly non-sustainable. If it has two fixed points, the graphs are almost surely sustainable, except when they grow too large. In the case where the function has three fixed points, sustainability becomes less straightforward. For this scenario, we have shown that it is relevant to define an interval such that, when the seed graph's order remains within this interval, sustainable graphs are produced.

Additionally, the V-DynamicScore was studied for this model in the context of sustainable graphs. The results indicate that, on average, this metric equals $\frac{2}{3}$, meaning that roughly half of the vertices are replaced at each step. This suggests that sustainable graphs tend to reach a form of equilibrium over time.

Future work should extend the analyses performed for this family of parameters. For the RD3G3 model, analyzing the variations of the estimated function could refine the conditions under which graphs are sustainable. This could draw from the study of complex systems, where the behavior of the derivative near fixed points determines whether they are attractive or repulsive. For the D3G3 model, further investigation is needed into the structural properties that maintain graph sustainability. One potential approach is to adjust the parameters such that both sets are segments, with S_C containing few lower values than S_S . While this idea has been tested on a few values and has not produced convincing results, it is worth mentioning for future exploration.

4.4 Infinite Sets and Asymptotic Graph Order Evolution

The work presented in this section has been published in the Applied Network Science journal [4].

This section introduces a new family of parameters to work with the RD3G3 model introduced in the Section 4.3 to simplify the analysis of the D3G3 model. The main result we prove here is that sustainability of large dynamic graphs may be theoretically explored leading to three different possible behaviors. This analysis has been possible using results on infinite sums and properties of complex numbers. These results are provided in the subsequent Section. Then, using these properties, we are able to obtain an equivalent of $f_{S,a}(n)$ for large values of n . This equivalent highlights an exponential growth of produced graphs (i.e., there exists $a > 0$ such that $f_{S,a}(n) \sim an$).

This section aims at presenting our work on the RD3G3 model for restrained values on the parameter S . Indeed, this work focuses on sets of the form $S = \{sk + r \mid r \in A, k \in \mathbb{N}\}$ for fixed $s \in \mathbb{Z}^+$ and $A \subset [0, s - 1]$. This equivalent will also help understanding the behavior of generated graphs with high orders. It will also provide an answer to whether generated graphs are sustainable or not. Indeed, depending on the value of s and A , three different scenarios are possible. If the base of the exponential is less than 1, then the graphs are, on average, decreasing in size over time. If the base equals 1, then the graphs are, on average, constant in size from one step to the next. If the base is greater than 1, then the graphs are increasing in size and generally sustainable. Although sustainability is ensured in the last scenario, further consideration is required regarding the size of the graph in the other cases. This is proved in Section 4.4.3.4, where we show that for small-sized graphs, a different approach is necessary. Indeed, the choice of values for S influences the sustainability of produced graphs.

4.4.1 Intermediate Result

The computation of the equivalent relies on the properties satisfied by roots of unity. As a reminder, a n th root of unity for some positive integer n is defined as follow:

Definition 30 *Let n be a positive integer. Then a n th root of unity is a complex number ω such that $\omega^n = 1$.*

Even though such numbers have been extensively studied, we have not found any article nor courses proving the following statements. This is why the proof are provided here with some properties complex numbers satisfy which may be found in [22] at section 2.5. Most important ones are gathered in the following lemma:

Lemma 6 *Let n be a positive integer. Then the following holds:*

- $\omega_n = \exp\left(\frac{2i\pi}{n}\right)$ is a n th root of unity;
- a complex number ω is a n th root of unity if and only if there exist k such that $\omega = \omega_n^k$;
- if a complex number ω is a n th root of unity, then its modulus satisfies $|\omega| = 1$;
- sum of j th powers of n th root of unity, for any non-negative integer j , are such that:

$$\sum_{k=0}^{n-1} (\omega_n^k)^j = \begin{cases} n & \text{If } n \text{ divides } j \\ 0 & \text{Otherwise} \end{cases}$$

Such properties are useful to prove the following lemma concerning infinite sums:

Lemma 7 *Let $s \in \mathbb{Z}^+$, $n \in \mathbb{N}$ and $x \in \mathbb{R}$. Let $r \in [0, s - 1]$, then we get the following equality:*

$$\sum_{k=0}^{+\infty} \binom{n}{sk+r} x^{sk+r} = \frac{1}{s} \sum_{j=0}^{s-1} \omega_s^{-jr} (1 + \omega_s^j x)^n$$

where $\omega_s = \exp\left(\frac{2i\pi}{s}\right)$ is an s th root of unity.

Proof: Let $s \in \mathbb{Z}^+$, $n \in \mathbb{N}$ and $x \in \mathbb{R}$. Let $r \in [0, s - 1]$. Let $\omega_s = \exp\left(\frac{2i\pi}{s}\right)$. The first thing to notice is that the infinite sum on the left side of the equality converges. For any values of k such that $sk + r > n$, the binomial $\binom{n}{sk+r} = 0$. Thus, the infinite sum contains only finitely many non-zero terms. Then, it is sufficient to notice that, according to properties roots of unity satisfy, the following holds:

$$\forall m, \frac{1}{s} \binom{n}{m} x^m \sum_{j=0}^{s-1} \omega_s^{j(m-r)} = \begin{cases} \binom{n}{m} x^m & \text{If there exists } k \text{ such that } m = sk + r \\ 0 & \text{Otherwise} \end{cases}$$

From this the following equations hold:

$$\begin{aligned}
\sum_{k=0}^{+\infty} \binom{n}{sk+r} x^{sk+r} &= \frac{1}{s} \sum_{m=0}^{+\infty} \binom{n}{m} x^m \sum_{j=0}^{s-1} (\omega_s^j)^{m-r} \\
&= \frac{1}{s} \sum_{m=0}^{+\infty} \sum_{j=0}^{s-1} \omega_s^{-jr} \binom{n}{m} (\omega_s^j x)^m \\
&= \frac{1}{s} \sum_{j=0}^{s-1} \omega_s^{-jr} \left(\sum_{m=0}^{+\infty} \binom{n}{m} (\omega_s^j x)^m \right) \\
&= \frac{1}{s} \sum_{j=0}^{s-1} \omega_s^{-jr} \left(\sum_{m=0}^n \binom{n}{m} (\omega_s^j x)^m \right) \\
&= \frac{1}{s} \sum_{j=0}^{s-1} \omega_s^{-jr} (1 + \omega_s^j x)^n
\end{aligned}$$

This ends the proof. ■

This lemma on roots of unity helps getting another expression of the function $f_{S,d}$:

Theorem 23 Let $s \in \mathbb{Z}^+$, $n \in \mathbb{N}$ and $A \subset [0, s-1]$. Let S as defined above, then:

$$f_{S,d}(n) = \frac{2}{s} n \left(\sum_{r \in A} \left(\sum_{j=0}^{s-1} \omega_s^{-jr} (1 - p + \omega_s^j p)^{n-1} \right) \right)$$

Proof: Let $s \in \mathbb{Z}^+$, $n \in \mathbb{N}$ and $A \subset [0, s-1]$. Rewriting $f_{S,d}(n)$ lead to the following expression

$$\begin{aligned}
f_{S,d}(n) &= 2n \sum_{r \in A} \left(\sum_{k=0}^{+\infty} \binom{n-1}{sk+r} p^{sk+r} (1-p)^{n-1-(sk+r)} \right) \\
&= 2n(1-p)^{n-1} \sum_{r \in A} \left(\sum_{k=0}^{+\infty} \binom{n-1}{sk+r} \left(\frac{p}{1-p} \right)^{sk+r} \right)
\end{aligned}$$

Thus, applying the result of lemma 7 provides:

$$\begin{aligned}
f_{S,d}(n) &= \frac{2}{s} n (1-p)^{n-1} \sum_{r \in A} \left(\sum_{j=0}^{s-1} \omega_s^{-jr} \left(1 + \omega_s^j \frac{p}{1-p} \right)^{n-1} \right) \\
&= \frac{2}{s} n \sum_{r \in A} \left(\sum_{j=0}^{s-1} \omega_s^{-jr} (1 - p + \omega_s^j p)^{n-1} \right)
\end{aligned}$$

This ends the proof. ■

This theorem provides an exact formulae for the estimation function $f_{S,d}$. It is important to notice that this re-written formulae involves only finite sums. It is therefore easier to deal with its analysis which is the purpose of the following sub-section.

4.4.2 The Equivalent and First Interpretation

From result obtained in the last subsection, it is possible to get an equivalent for great values of n for $f_{S,d}$:

Theorem 24 *Let $s \in \mathbb{Z}^+$, $n \in \mathbb{N}$ and $A \subset [0, s - 1]$. Let S as defined above, then for large values of n :*

$$f_{S,d}(n) \sim \frac{2|A|}{s}n$$

Proof: This comes from theorem 23 and from properties on complex numbers. More precisely, for each value of $r \in A$ there is exactly one value of $j \in [0, s - 1]$ such that $\omega_s^{-jr}(1 - p + \omega_s^j p) = 1$ (for $j = 0$). For all other values of j , $(1 - p + \omega_s^j p) \neq 1$ and have a modulus lower than 1. Therefore, raised to the $n - 1$ -th power, if $j \neq 0$ $(1 - p + \omega_s^j p)^{n-1} \rightarrow 0$ as n grows to infinity and $(1 - p + \omega_s^j p)^{n-1} = 1$ if $j = 0$. The rest is computation of limits. ■

This result has an interpretation for graphs generated with the model. Indeed, for a given snapshot graph at step t of order n_t , the application of the rule will produce a graph with an expected order $\left(\frac{2|A|}{s}\right)n_t$ at step $t + 1$. The evolution of graph order exhibits three different trends depending on whether $\left(\frac{2|A|}{s}\right)$ is lower than, equal to or greater than 1. The next section goes further in the analysis of these three cases. It also highlights the differences between graph order evolution of big and small graphs: interpretation depends on the smallest values of parameter S .

4.4.3 Generated Graphs Interpretation

This section aims at going further in the interpretation of previously stated results. More precisely, this section highlight three different asymptotic graph order evolution that occur from stated equivalent in 4.4.2. Moreover, interpretation for small graph order is given. This will help knowing whether generated graphs are likely to remain steady or not depending on the smallest values of the parameter S . Before dealing with each case, it is important to understand the meaning of theorem 24. This theorem states that for any given generated graph having n_t nodes at a step t and assuming n_t is big enough, then, at the next step, n_{t+1} is expected to be close to $\left(\frac{2|A|}{s}\right)n_t$. Therefore, starting with a seed graph of order N big enough would lead, after t steps, to a graph of order

$$n_t \simeq \left(\frac{2|A|}{s}\right)^t N$$

This is why graph order is said to grow exponentially. From this, three cases have to be observed:

- The first case is $\frac{2|A|}{s} < 1$. This means generated graphs order are likely to decrease when their order is large.
- The second case is $\frac{2|A|}{s} > 1$. This means generated graphs order are likely to increase when their order is large.
- Finally, the third case is $\frac{2|A|}{s} = 1$.

4.4.3.1 Exponential Increasing

The first studied case is when s and A both satisfy $\frac{2|A|}{s} > 1$. For this case, as S is not bounded the order of generated graphs is likely to tend to infinity. Generated graphs are therefore likely to be sustainable. An instance illustrating this case is given Figure 4.9.

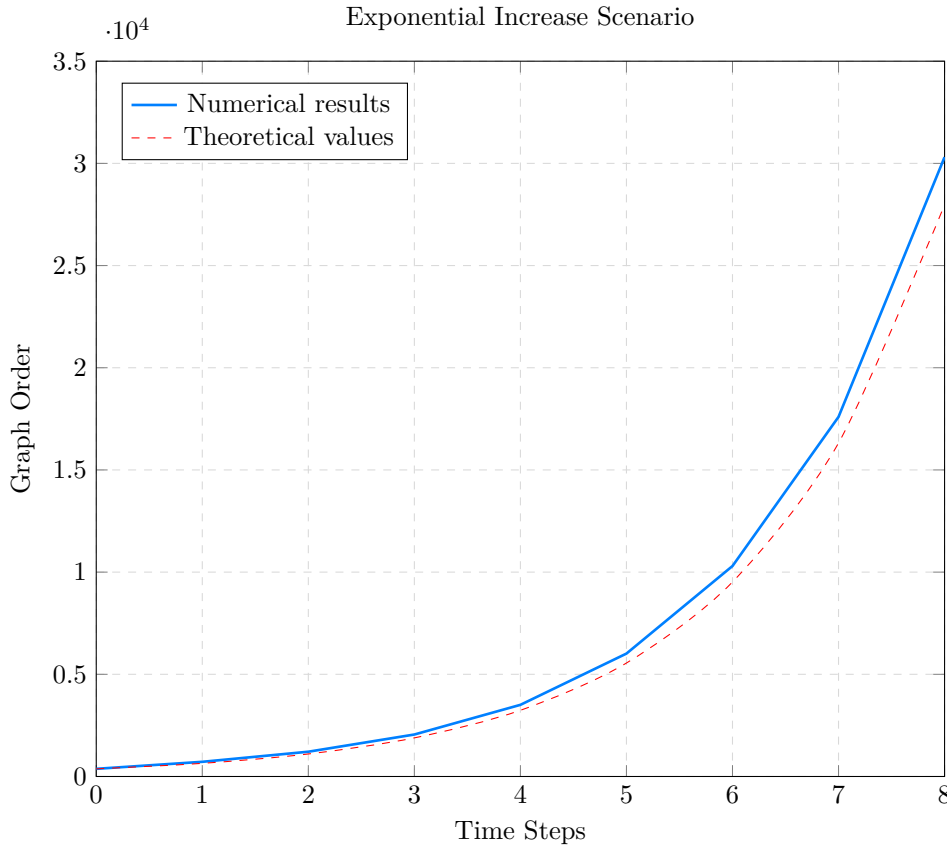


Figure 4.9: Scenario of exponential increase. $A = [0, 5]$, $s = 7$, $n_0 = 375$. The theoretical value is given by the formula $n_t \approx \left(\frac{2|A|}{s}\right)^t n_0$

4.4.3.2 Exponential Decreasing

The second studied case is when s and A both satisfy $\frac{2|A|}{s} < 1$. For this case, graph order of generated graphs is likely to decrease exponentially. An instance illustrating this case is given Figure 4.10. However, it is not enough to conclude on the sustainability of generated graphs. Indeed, when graphs become small enough (close to 0), one may consider to take into account the smallest values of set S . This last case is further studied in section 4.4.3.4.

4.4.3.3 Quasi Constant Evolution

The results presented in this section have been published in the Applied Network Science journal [4].

Two points must be noticed for the last case. First, this case happens if and only if s is even. Indeed, if s is odd, whatever the set A one may choose, the numerator will be even. Second, for a given time step t , application of the rule on a graph which order is n_t will produce a graph which order is expected to be $n_{t+1} = n_t$. It is however necessary to go further as $f_{S,d}$ only provides an expectation. The graph order

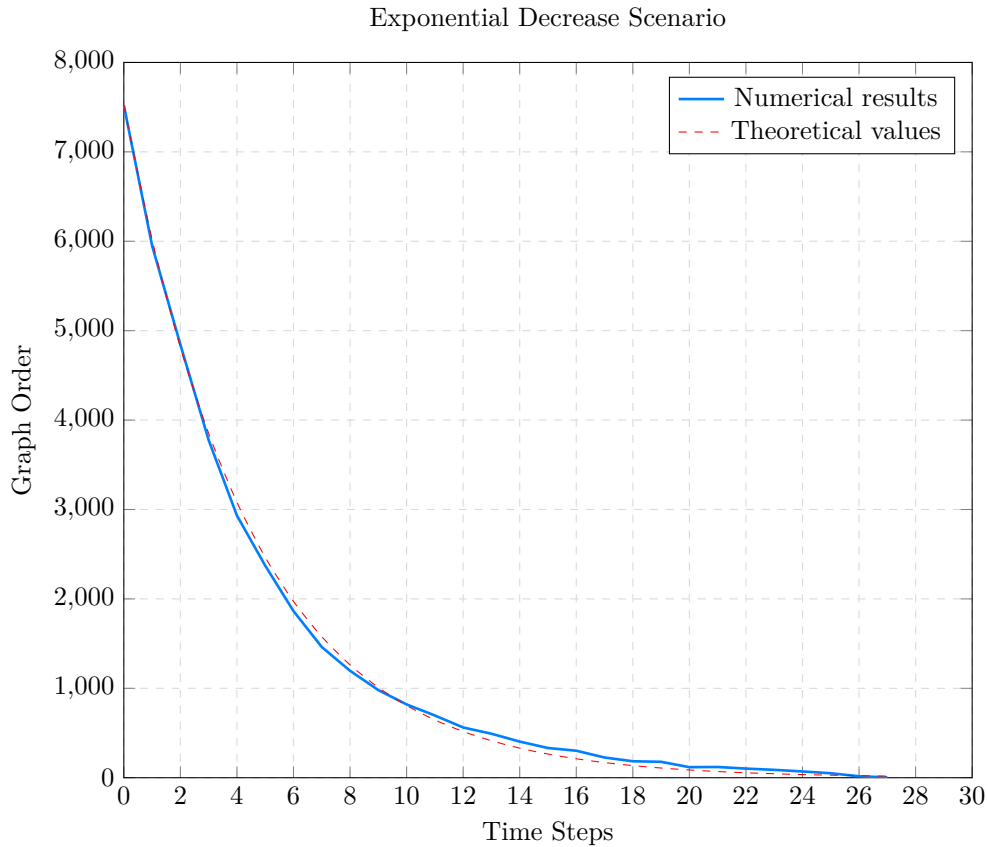


Figure 4.10: Scenario of exponential decrease. $A = [4, 5]$, $s = 5$, $n_0 = 7523$. Theoretical values are given by the formula $n_t \approx \left(\frac{2|A|}{s}\right)^t n_0$.

will indeed change a little. An estimation for this change can be obtained with the standard deviation of a binomial law. Despite all these considerations, simulations have been performed. They all show that graph order changes through time with little variations. These simulations are represented in figure 4.11. It is worth noticing graph order is not constant all along the simulation, but rather increasing or decreasing a little bit every time.

A further step to this study is to take into account the standard deviation $\sigma_{S,d}$ associated with graph order evolution. For a given $n_t \in \mathbb{Z}^+$ order of a graph at step t , $\sigma_{S,d}(n_t)$ tells how far away from n_{t+1} is $f_{S,d}(n_t)$, which in this case is roughly n_t . Thus, applying Chebishev's inequality [18], for instance, states that for any given real number $k > 0$:

$$\Pr[n_{t+1} \notin [n_t - k\sigma_{S,d}(n_t), n_t + k\sigma_{S,d}(n_t)]] \leq \frac{1}{k^2}$$

The computation of $\sigma_{S,d}(n)$ for large enough values of n lead to an equivalent which is the purpose of the following theorem:

Theorem 25 *Let $s \in \mathbb{Z}^+$, $n \in \mathbb{N}$ and $A \subset [0, s - 1]$. Let S as defined above, then for large values of n :*

$$\sigma_{S,d}(n) \sim \frac{1}{s} \sqrt{n|A|(s - |A|)}$$

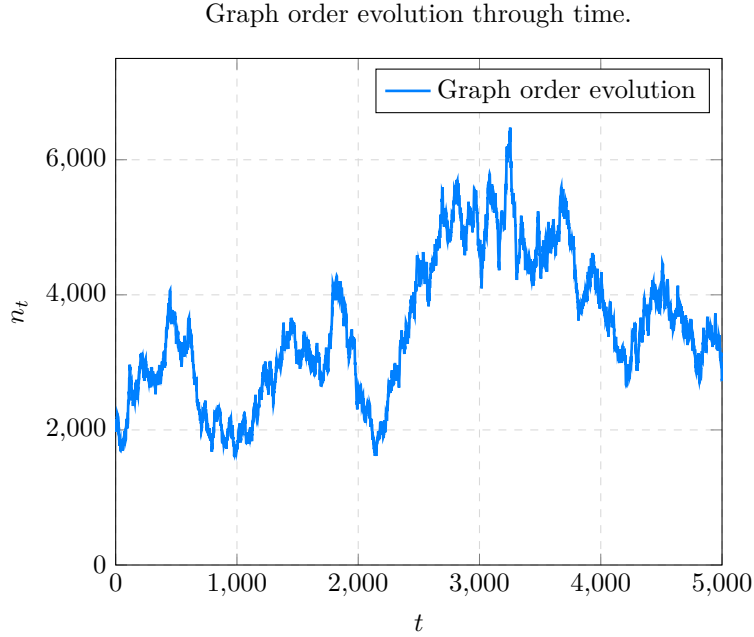


Figure 4.11: Simulation performed considering $s = 4$, $A = [0, 1]$ and $d = 0.05$. The number of steps is 5000 and the initial seed graph is a random geometric graph of order 2000.

Proof: The proof of this theorem relies on the same argument as for theorem 24 and on the definition of the standard deviation of binomial distributions. ■

This theorem states that the standard deviation $\sigma_{S,d}(n)$ is proportional to \sqrt{n} for large values of n . This provides better information about the possible values n_{t+1} may have depending on n_t . Indeed, now above stated inequality can be rewritten as follow:

$$P\left(n_{t+1} \notin \left[n_t - \frac{k}{2}\sqrt{n_t}, n_t + \frac{k}{2}\sqrt{n_t}\right]\right) \leq \frac{1}{k^2}$$

4.4.3.4 Sustainability of Small Generated Graphs

The question of whether a small generated graph is sustainable or not does not depend on the asymptotic variation of the graph order. The answer to this question relies on the smallest values that the parameter S contains. Indeed, on the one hand, whatever the values of s one may consider, if $A \subset [k, s-1]$ for any $k \geq \frac{s}{2}$, then graphs whose order does not exceed k do not have nodes with a degree greater than or equal to k . Therefore such graphs become empty because they do not have any node satisfying the duplication rule. A further step is to consider small values of parameter S . For instance, for $d = 0.05$, $s = 16$ and $A = [8, 15]$, the full-lined curve of $f_{S,d}$ represented in figure 4.12 shows that for small values of n , $f_{S,d}(n) < n$. This means that graph order of small graph is expected to decrease between two consecutive steps and graphs are likely to become empty. Therefore generated graphs, for this configuration are likely not be sustainable.

On the other hand, whatever the values of s one may consider, if $A \subset [0, k+1]$ for any $k < \frac{s}{2}$, then graphs whose order does not exceed k have nodes with a degree lower than or equal to k . Therefore such graphs do not become empty because they have all their nodes satisfying the duplication rule. As

for the first case, a further step is to consider small values of parameter S . For instance, for $d = 0.05$, $s = 16$ and $A = [0, 7]$, the dotted curve of $f_{S,d}$ represented in figure 4.12 shows that for small values of n , $f_{S,d}(n) \geq n$. This means that graph order of small graphs is expected to increase between two consecutive steps. Therefore, as soon as graph order does not exceed a certain quantity, generated graphs are likely to conserve few nodes and therefore are likely to be sustainable.

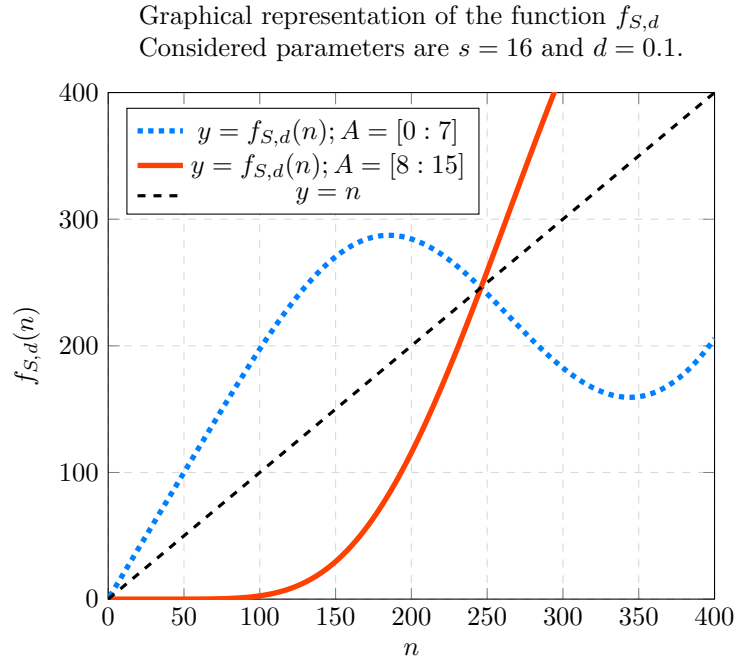


Figure 4.12: Theoretical graphical representation of $f_{S,d}$ for value of n from 0 to 400. The blue curve correspond to $A = [0, 7]$ and the red one correspond to $A = [8, 15]$

Conclusion on Infinite Sets

This section has delved into the study of the RD3G3 model presented in Section 4.2.4. This section aims at providing a new family of parameters for which sustainability may be investigated through the analysis of an asymptotic equivalent of $f_{S,d}$, the function estimating the evolution of the number of vertices through time. The study of this equivalent as proved that when the size of a snapshot graph n_t is big enough, then $f_{S,d}(n_t) \sim a \times n_t$ with $a \in [0, 2]$. The sustainability is ensured when $a > 1$ as produced graphs are growing exponentially. This not always the case when $a \leq 1$. For this case, the study of $f_{S,d}$ for small values is required to see whether produced graphs are sustainable.

Conclusion

This chapter was dedicated to the computation of sustainability and DynamicScores on dynamic geometric graphs generated by a specific parameterized generator. This simple generator is parameterized through three variables: a connection threshold d aiming at connecting all points closer than a distance d and two sets S_S and S_C containing non-negative integers. The first one aims at deciding whether a vertex is kept between two consecutive steps and the second one whether a vertex is at the origin of a new vertex

at the very next step. Several properties are shown about the whole family of graphs the generator provides for a single configuration. All these properties shown try to answer a single question. This question concerns graph *sustainability*. It is defined as the property, for a given graph obtained with a given seed graph and evolving rules, that the graph does not become empty after a finite number of steps. Defining this concept for this model is not simple since the evolving rules are not deterministic. It involves probabilistic computations and therefore questions about a possible threshold for which the graph is said to be sustainable if the probability of the emptiness of the graph is greater than this threshold. Here the focus has been made on two different metrics, graph order evolution and V-DynamicScore defined in Chapter 3. Different values of the parameters have been studied, but it has not been possible to try them all as the amount of possible cases is far too big. Cases for which properties have been shown are limit cases, the general case and two very specific cases. The first one may be referred to as "segments" and the second one concerns regular sets. Limit cases have led to a first classification when at least one of the two parameter sets is either empty or contains all non-negative integers. General cases highlights some properties for specific conditions on both sets. The case of segment sets has revealed theoretical difficulties, especially when computing graph order between consecutive steps. This has led to the creation of a new tool named the "sustainable interval". This tool aims at estimating bounds that frames graph order even though it is not always reliable as probabilities are involved. The case of regular infinite sets spotlighted an interesting and surprisingly simple equivalent of the graph order when this graph order is big enough. This equivalent eases the study of graphs as it provides three different scenarios. For one of them sustainability is possible to state. For the others, however it is mandatory to investigate the case of small graphs and how their order evolves through time. This model shows interesting aspects and we have proved that answering the sustainability question is not straightforward. Further investigations also show that it is also possible to find configuration of this model leading to sustainable graphs which are not information-persistent. This situation occurs when isolated vertices only are both conserved and at the origin of new vertices.

Analysis of Real World Networks

In this chapter, we establish a link between the theoretical contributions and the study of real networks. The objective is to compute the metrics created and presented in Chapter 3 using real world networks data, with a particular emphasis on the DynamicScore (see Section 3.2). Our focus will be on networks that emerge through interactions.

This study addresses two types of interactions commonly found in the literature: instantaneous and permanent contacts. Readers can refer to the survey by Holmes and Saramaki [24] for more details. Instantaneous contacts are modeling interactions between entities that occur over short time periods. Example of such interactions are handshake between two people, e-mails or communications between routers on the Internet for instance. Permanent contacts are modeling interactions such that once initiated, persist indefinitely. These types of contacts can be observed in biological systems, such as in the mycelium of filamentous fungi.

In both cases, the data considered in this section is represented as sequences of events in the form u, v, t . For instantaneous contacts, this indicates that an interaction between u and v occurs at time t (directional if contacts are oriented). In the case of permanent contacts, this means that vertex u starts interacting with vertex v from time t until the end of the experiment. The analysis provided in this chapter serves as an initial connection between the DynamicScore and real-world networks modeled as dynamic graphs. The conclusions drawn here are preliminary and further investigation is required to fully harness the potential of the DynamicScore for studying real-world networks.

In this section, we propose various approaches to transform a sequence of events into a dynamic graph. We describe the different characteristics of these transformations and provide a framework for their interpretation. For instantaneous contacts, we demonstrate that these transformations allow for multiple

perspectives at various time scales for the same network. For permanent contacts, we show that the DynamicScore can be used on filtered graphs to characterize different phases in the evolution of the network.

This chapter is divided into two parts. The first part is dedicated to the presentation of various transformations that can be applied to convert a sequence of events into a dynamic graph. The second part focuses on their application to real data, offering insights into network dynamics through the analysis of both the DynamicScore and the evolution of the graph's order.

5.1 Transforming Events into Graphs

This section is dedicated to exploring the different transformations of real-world data into a time-varying graph in the case of Person-to-Person networks. Real-world data studied here comes in the form of sequences of events, where each event is defined as a temporal edge (u, v, t) , indicating that entity u interacted with entity v at time t . In the case of directed networks, we will say that u is the source and v is the destination. We denote by T the largest time in the event sequence. The goal is to transform this sequence of events into a time-varying graph $\mathcal{G} = (G_i)_{0 \leq i \leq I}$ with I the index of the last graph in the sequence of graphs. The choice of the transformation depends on the nature of the event. For instance, events may model instantaneous contacts between two vertices. This case is investigated in Section 5.1.1. Otherwise, events may model the arrival or the creation of new vertices and edges in the system. This case is investigated in Section 5.1.2.

5.1.1 Events as Instantaneous Contacts

This section focuses on events as instantaneous contacts. To transform a sequence of timestamped events \mathcal{E} into a time-varying graph \mathcal{G} , methods consist in grouping temporal edges into a finite sequence of static graphs (G_0, \dots, G_I) . Other choices are possible, such as modeling the resulting graph as a static graph $G = (V, E)$ equipped with a function $\lambda : E \rightarrow \mathcal{P}(\llbracket 1, T \rrbracket)$ associating to each edge its presence times. For our purpose, the model of sequence of snapshot graphs is well-suited. Note also that the sequence of graphs produced must satisfy the constraint of preserving the order of timestamps of edges in the sequence. The size of \mathcal{G} (and thus the value of I) is not necessarily equal to T : it can be smaller or greater. Indeed, each graph in the sequence can group events from several consecutive timestamps. For instance, we might have a sequence of events where the time unit is one second while these events are grouped into graphs representing exchanges happening on a daily scale (see Section 5.2.1.1). Several transformations are possible depending on the regrouping policy. Four of these transformations are presented and detailed in the remainder of this section. To illustrate the transformations discussed, we will use as an example the temporal event sequence shown in Figure 5.1. This sequence will serve as an example to illustrate the function of each transformation and to understand the resulting graphs.

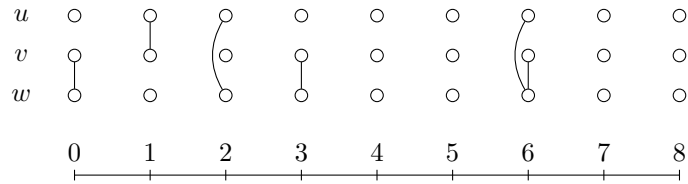


Figure 5.1: Illustration of a sequence of event. Here, $T = 8$ and only three vertices are interacting in the whole sequence

5.1.1.1 Time Interval

The Time Interval Transformation involves gathering events by time intervals to obtain a single time-varying graph. The length of an interval, also called the period, is denoted by τ . In this method, each graph G_i represents all interactions that occurred between time step $i\tau$ and $(i+1)\tau - 1$, inclusive. Thus, there are $\frac{T}{\tau}$ static graphs in the time-varying graph, where T is the date of the last event in the sequence of events.

In this construction, if a vertex interacts at time t_1 and at a later time t_2 , but does not interact at an intermediate time, it is considered absent throughout the interval (t_1, t_2) and thus absent from the graphs corresponding to those dates. Therefore, vertices that do not interact with any others during an interval $[i\tau, (i+1)\tau - 1]$ will be considered absent from graph G_i , even if they have interacted in the past and will interact again later.

An illustration of what is obtained once the transformation is applied is given in Figure 5.2. In this figure, it is possible to see that each graph covers three consecutive dates, and they differ from one graph to another. Additionally, it is also possible to see that vertex u does not interact with any other vertex between dates 3 and 5, inclusive, and thus, it is not part of the set of nodes in graph G_1 . The graphs shown in the illustration are: $G_0 = (\{u, v, w\}, \{\{u, v\}, \{u, w\}, \{v, w\}\})$, $G_1 = (\{v, w\}, \{\{v, w\}\})$, and $G_2 = (\{u, v, w\}, \{\{u, w\}, \{v, w\}\})$.

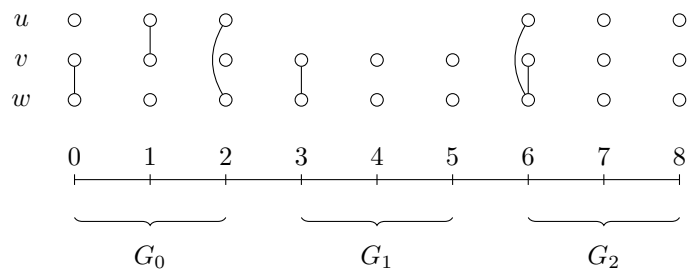


Figure 5.2: Illustration of a transformation from a sequence of event to a time-varying graph. Here, $T = 8$ and $\tau = 3$. As u does not interact with another vertex, it is not present in G_1 .

This construction is studied in this document as it is the most intuitive one and it is possible to get results on DynamicScores as it will be presented in Section 5.2.1.2.

5.1.1.2 Sliding Window

The Sliding Window Transformation is based on the sliding window principle. As a beginning, a window size W is set as well as a shift τ . As for all examples presented for illustrating transformations in this Chapter a sequence of events of size T is also considered. In this method, each graph G_i represents all interactions that occurred between times i (inclusive) and $i + W$ (exclusive). An illustration of such a transformation is provided on Figure 5.3. On this example, the time-varying graph produced is composed of 7 graphs. The sequence is the following: $G_0 = G_1 = (\{u, v, w\}, \{\{u, v\}, \{v, w\}, \{u, w\}\})$, $G_2 = (\{u, v, w\}, \{\{u, v\}, \{v, w\}\})$, $G_3 = (\{v, w\}, \{\{v, w\}\})$ and $G_4 = G_5 = G_6 = (\{u, v, w\}, \{\{u, v\}, \{v, w\}\})$.

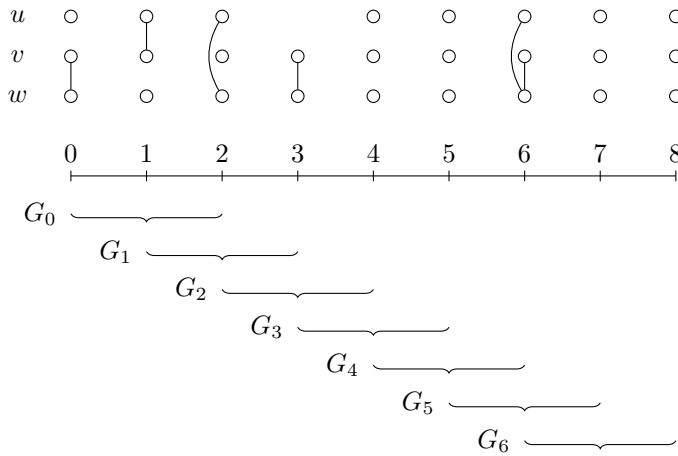


Figure 5.3: Illustration of a transformation from a sequence of event to a time-varying graph. Here, $T = 8$, $W = 3$ and $\tau = 1$.

This transformation produces $\frac{T-W}{\tau} + 1$ static graphs. The transformation is smooth in the sense that it allows for capturing small changes in the structure of the time-varying graph. This is due to the ability to adjust the shift size τ . It should be noted that the window size is not related to the time unit separating two consecutive events. For example, events may be divided by seconds while the window size may be days, weeks, or even months. Furthermore, it is advisable to choose τ allowing for an integer division to ensure that no graph (specifically the last one in the sequence) has a smaller size than the others.

5.1.1.3 First-to-last Vertices

The First-to-last transformation is based on the following principle: if a vertex u is present both at time t_1 and t_2 such that $t_1 < t_2$, then u is considered present at every time $t \in [t_1, t_2]$, even if it does not interact with any other vertex. Referring back to the example in Figure 5.2, this means that u is also present in graph G_1 . This can be expressed more formally as:

$$\forall u \in \bigcup_{i=0}^T V_i, [\exists (t_1, t_2) \in [0, T]^2, t_1 < t_2 \wedge u \in V_{t_1} \wedge u \in V_{t_2} \Rightarrow \forall t \in [t_1, t_2], u \in V_t]$$

Regarding the sequence of graphs produced, it is similar to the one given for the time intervals. They cover the dates in the same way, but graphs obtained for the two transformations are different due to the just given rule. Such a construction allows a vertex to be considered present in the graphs from the first

date it interacts with another vertex in the network and disappears as soon as it no longer interacts with any other vertex. Referring back to the example in Figure 5.1, the graphs produced are as follows: $G_0 = (\{u, v, w\}, \{\{u, v\}, \{u, w\}, \{v, w\}\})$, $G_1 = (\{u, v, w\}, \{\{v, w\}\})$, and $G_2 = (\{u, v, w\}, \{\{u, w\}, \{v, w\}\})$. We note that $u \in V_1$ because even if it does not interact with another vertex between times 3 and 5 inclusive, it is present in graphs G_0 and G_2 .

This consideration on the vertices of the graphs may also be considered for the edges: an edge is present in graph G_i if it belongs to $G_{i'}$ and $G_{i''}$ where $i' \leq i \leq i''$. This second transformation will be referred to as First-to-last Vertices and Edges (F2LVE) in the rest of the document. This transformation is illustrated on Figure 5.4.

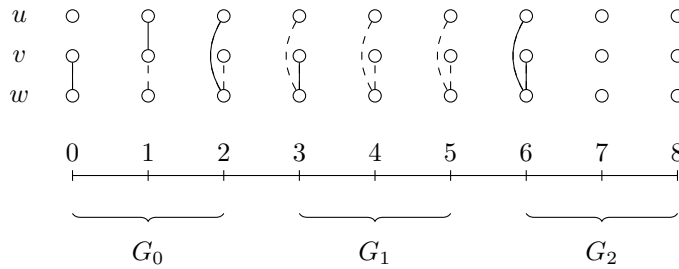


Figure 5.4: Illustration of a transformation from a sequence of events to a time-varying graph using the F2LVE transformation. Here, $T = 8$ and $\tau = 3$. Dashed lines indicate edges part of the resulting graph but which are not occurring at that date.

5.1.1.4 Growing Model

This model is based on the principle of growing networks: for a given $\tau > 1$ integer, for each graph G_i in the produced sequence of static graphs, G_i contains all the events occurring before $(i + 1)\tau$. Thus, the sequence of graph is increasing : $V_i \subset V_{i+1}$ and $E_i \subset E_{i+1}$ for all $i \in [0, T/\tau]$. An illustration of such a transformation is provided Figure 5.5. The sequence of events provided for this example is different from the one used on Figure 5.1 to have a better illustration of the principle. The time-varying graph obtained is $\mathcal{G} = (G_0, G_1, G_2)$ with $G_0 = (\{v, w\}, \{\{v, w\}\})$, $G_1 = (\{u, v, w\}, \{\{u, v\}, \{v, w\}\})$, $G_2 = (\{u, v, w\}, \{\{u, v\}, \{v, w\}, \{u, w\}\})$.

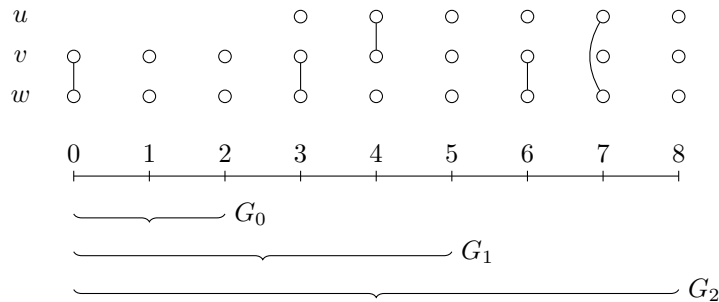


Figure 5.5: Illustration of a transformation from a sequence of event to a dynamic graph. Here, $T = 8$ and $\tau = 3$.

Such a transformation helps studying a network at a global scale. Indeed, it gathers every event into

graphs and no edges as well as no vertices are removed from graphs. It is a relevant transformation to see how many different contacts are new on every time interval of length τ for instance.

5.1.1.5 Remarks

Each model presented above has different uses. The choice of one transformation rather than another one depends on the specific application considered, as well as the meaning of the sequence of graphs that form a time-varying graph.

For example, if the Sliding Window transformation is used instead of the Time Interval transformation, the first one will produce more graphs, but the analysis of the dynamics produced by the two transformations will lead to different interpretations. The dynamics produced by the Sliding Window transformation will provide a finer granularity than those produced by the Time Interval transformation, as well as smoother dynamics, without a straight cut between groups of events. Thus, the choice of a given transformation might be driven by the semantics of the data itself. If there exists an obvious reason for grouping events, for instance if every day one parameter changes, then using the Time Interval transformation would probably be more relevant than the Sliding Window transformation.

Note that the Sliding Window transformation considers much smaller time gaps between two consecutive graphs (the size of the window) than in the case of the Time Interval transformation, where the time gap between two graphs is the length of the time interval. Concerning Time Interval construction, one of its advantage is that the resulting graph sequence is smaller, which helps ensure better memory complexity. This advantage, however, should be put into perspective, as the small size of the sequence also means that the number of data points that can be obtained is limited. A small number of data points may not lead to significant results and could bias the study being conducted. This phenomenon is observed and described when studying a dataset of e-mails sent between core members of an European research center (see Section 5.2.1.1).

Another example is the comparison between the Time Interval and the First-to-Last Vertices transformations. It can be noted that what differentiates them in terms of DynamicScore is a difference in the DynamicScore of the vertices only. Indeed, the sequence of edge sets is the same between the two transformations. Moreover, the Dynamic Score of the vertices for the Time Interval transformation will always be greater than that of the First-to-Last transformation due to the conservation of vertices in the latter transformation.

It is worth mentioning that all these transformations are meaningful due to the appearance as well as the disappearance of vertices over time. Indeed, if the set of vertices remains steady, then all these constructions provide graphs with the same DynamicScore (constant and equal to 0).

In this document, the focus is on transforming sequences of events into time-varying graphs. We examine what happens when events are grouped in batches within a time interval. Other models are also

possible but have not been covered. For instance, one could imagine associating an edge with a given event. This edge persists in the time-varying graph for a certain duration depending on the application considered. This duration symbolizes the time of an interaction. In the case of a conversation, for example, two people might start talking at time t . Since we do not know when their conversation ends, we can set an arbitrary duration to model the length of a typical conversation. This approach allows for obtaining a second sequence of events that models appearances and disappearances. From this second sequence, it is possible to derive a time-varying graph that is more detailed than just a sequence of graphs. This type of time-varying graph can be modeled as a temporal graph in the form $G = (V, E, \lambda, \zeta)$ (see Section 2.3.1). This approach has some limitations. For instance, the length of a conversation is set arbitrary. Such a technical detail is nonetheless requires as no information is provided on the end of an interaction. This information is necessary to build time-varying graph.

5.1.2 Events as Permanent Contacts

In this section, events are considered as the introduction of permanent edges. This means once the edge is introduced it never disappears. It is also the case of the vertices at the each end of these edges. Examples are biological systems such as bacteria, filamentous fungi or molecules. Therefore, transformations look at different aspects of the system. Instead of providing multiple scenarios as it is done in the section dealing with instantaneous contacts, a general method is presented here. This method consists in extracting sub time-varying graphs from a bigger one using filter functions. First, a time-varying graph is produced from the sequence of events. The way this graph is produced is similar to the Growing Transformation (see Section 5.1.1.4). Then one or many filters are applied on the time-varying graph to obtain new graphs, subsets of the whole graph.

5.1.2.1 Filtering

Filtering a time-varying graph \mathcal{G} is a process extracting a sub time-varying graph \mathcal{G}' using a filter function. A filtering function is a function $f : G = (V, E) \mapsto G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. \mathcal{G}' is obtained after the application of the filter to every snapshot graph of \mathcal{G} . The filtering function f may only filter vertices depending on their features. In that case, E' is built such as $E' = E \cap (V' \times V')$ (i.e., the set of edges present in G having their vertices in V'). The feature of the vertices depends on the nature of the studied network. For instance, if vertices have spatial coordinates, then a filter may select vertices close to a given position. This is what have been done when studying the time-varying graph of a filamentous fungus called *Podospora anserina* (see Section 5.2.2). Note that this process produces two time-varying graphs. In addition to the filtered graph, there is also the complementary of this graph. This graph may be seen as the filtered graph obtained using the complementary filter \bar{f} of f defined as follows : $\bar{f} : G = (V, E) \mapsto G'' = (V'', E'')$ such that $V'' = V - V'$ and $E'' = E - E'$. It is something worth to bear in mind as metrics may be investigated on both sub time-varying graphs.

5.2 Illustrative Case Study

This section is dedicated to the application of transformations defined in Section 5.1. These transformations aim at transforming event sequences into time-varying graphs.

5.2.1 Instantaneous Contacts

In order to illustrate how DynamicScore values may be influenced by transformations, two contact networks have been studied. These two networks are available on the Stanford Network Analysis Project (SNAP) web site [32]. SNAP is a research platform developed by the Stanford University (USA). It is dedicated to the analysis of large social and information networks. SNAP provides tools and data to ease the study of complex networks, covering various domains such as social networks, communication networks, citation networks, and more. SNAP provides temporal networks in which data are given as sequences of events as described above. For this document, the focus has been made on two networks. The first network is composed of e-mails sent between institution members of a large European research institution. The second network is a dataset that consists of routers that are part of the Internet.

For this study, we present the dataset and information about each network. We then present the different transformations used in our work and the obtained results. For this section, every transformation is applied on sequences of events to build time-varying graphs. For each transformation, for any time t , four quantities are studied: the order of the graph, n_t , the number of edges, e_t , the Vertex-DynamicScore, \mathcal{D}_t^V and the Edge-DynamicScore, \mathcal{D}_t^E .

The objective is to highlight the role of each transformation. All the results obtained from the application of each transformation are then put together to draw a global picture of the dynamics of the network. Let us delve in the study of these datasets.

5.2.1.1 Email Network

The network studied represent the e-mail network extracted from a large European research institution dataset. These data are available on the Web site on SNAP ([32]). They only represent e-mails between members of the institution as e-mails coming from outside of the institution were not collected. The data are gathered in a CSV file in which each entry represents a temporal edge (u, v, t) meaning that person u sent an e-mail to person v at time t . Every person in the dataset corresponds to one email address and is represented by a positive integer and the time t , given in seconds, starts from 0 at the beginning of the study. The period covered by the dataset ranges from October 2003 to May 2005 (18 months) and precisely 803 days. The network is composed of 986 vertices, 24,929 static edges and 332,334 temporal edges. The difference between static and temporal edges is that static edges represent edges in the footprint graph, with no time information, while temporal edges are edges in the time-varying graph.

An extended version of this network has already been studied by J. Leskovec et al. in the paper [31].

In this paper, authors show a relationship between the number of edges e_t and the number of nodes n_t at a given step t . This relation is proved to be as follows : $e_t = n_t^a$ with $a \simeq 1.12$. The purpose of the authors was to prove that temporal networks tend to get denser rather than to grow linearly. The network built from the data is a growing network in which, each snapshot graph gathers all the interactions occurring up to a given month. The first graph is therefore all the emails sent during October 2003, and the last one is the graph of all the interactions up to May 2005.

In order to highlight the potential and the differences between each transformation, the core version of this network, available at [32], is studied. Every transformation is applied to the sequence of events and both Vertices and Edges DynamicScores are computed. The goal is to see whether the DynamicScore metrics may lead to a better understanding of the behavior of users. First, it is necessary to set a time interval which may symbolize the time length of an interaction. In their study [31], authors set this length to one month. However, such a length represents only 18 samples, which is not relevant to conduct a robust study of the network. To solve this problem, we choose to set the time length to one week for every transformation. The Time Interval transformation gathers events occurring during one week. The Sliding Widow transformation has a window length of one week and a shift of one day. The Growing Network transformation also gathers data week after week. Finally, in the First-to-last transformations for vertices and for both vertices and edges, each snapshot will group interactions week by week.

An important point to notice is that the sequence of events provided gives events as follows: u, v, t where u is the sender, v is the receiver and t is the time at which the e-mail is sent. It is relevant to have a discussion on how to consider a connection between two individuals in the sequence of events. The method proposed by Kossinets et al. [27] states that two vertices A and B are connected if and only if A sends an e-mail to B and vice versa. However, this biases the meaning associated with the connections between two individuals. Moreover, the sequence of events does not provide further information than the sending of an e-mail between two people at a given date. Therefore, we have decided, for this network, to consider the links oriented. It does not change the way the DynamicScore is computed.

Remarks about the study of Leskivec .

We have performed the same analysis as the analysis provided by Leskivec et al. in their paper [31]. In this paper, they study the densification phenomenon in growing networks. This phenomenon highlights that instead of growing linearly, the number of edges at a given step t ($e(t)$) grows as a power of the number of vertices in the network ($n(t)$). In their study, the authors assume that there should exist $a \in [1, 2]$ such that $e(t) = n(t)^a$. Moreover, the authors present several studies of networks that appear to exhibit this property. More specifically, the network they studied represents e-mail exchanges in an European research center, which is a bigger version of the one treated here. In their article, the authors demonstrate that for this network, the constant a is approximately 1.12. They show that this relationship is plausible by providing an R-squared value of 1. However, the study includes too few data points (18), each corresponding to one month within the eighteen-month study period. Moreover, the data points are

close to each other and of the same order of magnitude. Therefore, it is challenging to clearly establish a relationship between the two quantities.

Growing Network Transformation Let us first deal with the growing-network transformation. This transformation results in a sequence of 75 graphs G_0, \dots, G_{74} , where each G_i contains a subset of all interactions that occurred before or during the i -th week of the study. Concerning the dynamics of the graph, Figure 5.6a shows that both DynamicScores of vertices and edges decrease and tend toward zero. This indicates that the graph orders do not increase exponentially. The peak value at the beginning is a consequence of the construction of the graph sequence. This is not surprising, since the time-varying graph is obtained by applying the Growing-Network transformation to the sequence of events, starting from a null graph. Moreover, the graph order does not increase too much between two consecutive weeks. This last point is illustrated Figure 5.6b. These two arguments put together imply that the dynamics of the graph is collapsing and therefore the graph becomes more and more static.

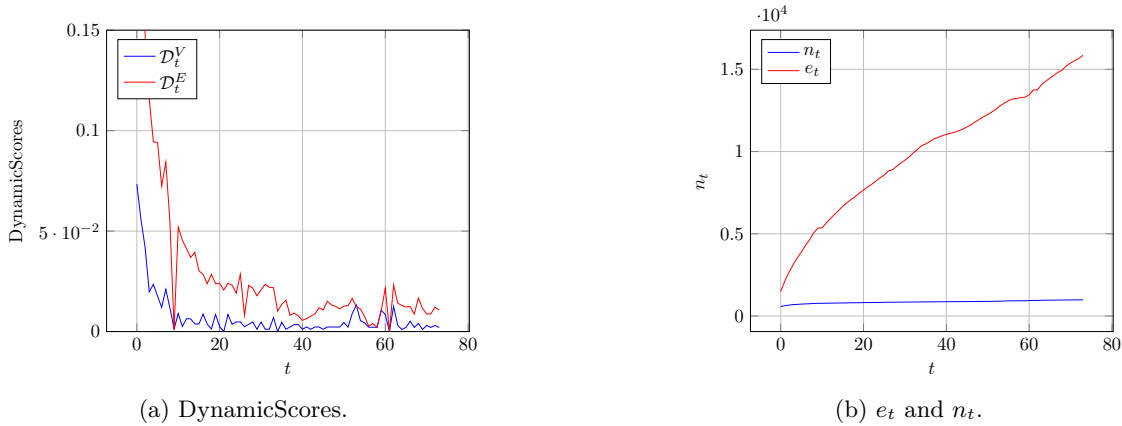


Figure 5.6: DynamicScores, n_t and e_t graph order evolution for the Growing Network transformation for the E-mail network.

Time Interval Transformation Let us now consider the Time Interval transformation. The length of the interval considered here is one week, meaning that every graph gathers information occurring during one week. As for the Growing Network transformation, the obtained graph sequence is composed of 75 graphs. One important thing to bear in mind concerning the transformation is that between two consecutive steps, the two graphs do not cover a common time period. Having a DynamicScore lower than 1 means some vertices and edges are conserved. Low values of the Vertices-DynamicScore are a sign that large amounts of vertices are conserved between two consecutive weeks. On the opposite, high values of the Edges-DynamicScore highlight significant changes in the composition of the vertices set. It is possible to notice some peak values of the DynamicScores, n_t and e_t (Figure 5.7a). These peaks correspond to holidays such as summer and winter breaks. For other value, values of e_t and n_t are slowly increasing while the DynamicScores of these two values remain almost steady through time. However, the Edges-DynamicScore is in average around 0.7, indicating that around 46% of the edges are conserved, while the Vertices-DynamicScore is in average 0.2, which indicates that around 89% of the vertices are

conserved. These observations allow us to say that in average, user of the network does not change too much while a non-negligible part of the edges are conserved in average every week.

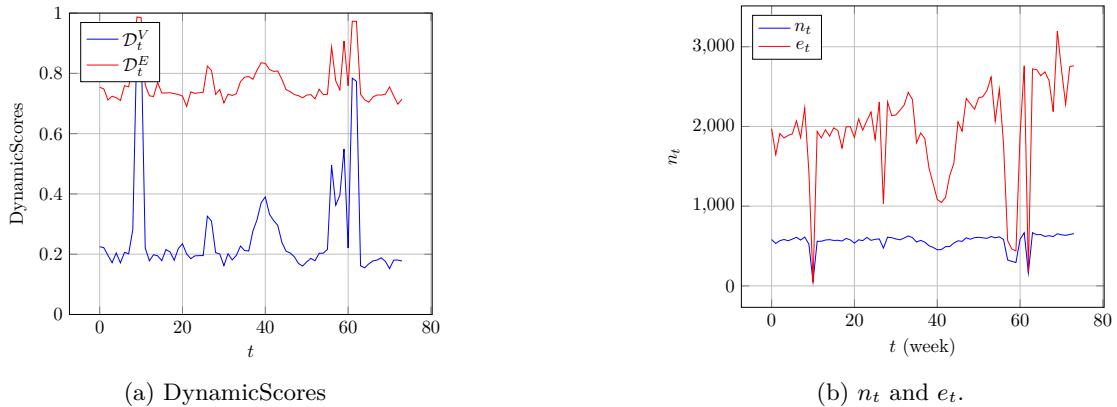


Figure 5.7: DynamicScores and graph order evolution for the Time Interval transformation.

Sliding Window Transformation Let us now consider the Sliding Window transformation. The length of the time window is one week and the time shift is one day. Results obtained for this transformation are very different from the ones of the Time Interval transformation (the last paragraph). The first difference is about the values of the DynamicScore. Indeed, both DynamicScores are clearly less than half the values obtained for the Time Interval transformation. The second difference concerns the time evolution of both DynamicScores which seems more sensitive than in the case of the Time Interval transformation with regular peaks. Those regular peaks with small values indicate there is one day during the week (Sunday) such that users and user exchanges do not change. This is represented on Figure 5.8b. While e_t and n_t are roughly the same for both transformation, it is therefore possible to say that using the Sliding Window transformation helps represent smoother dynamics compared to the Time Interval transformation.

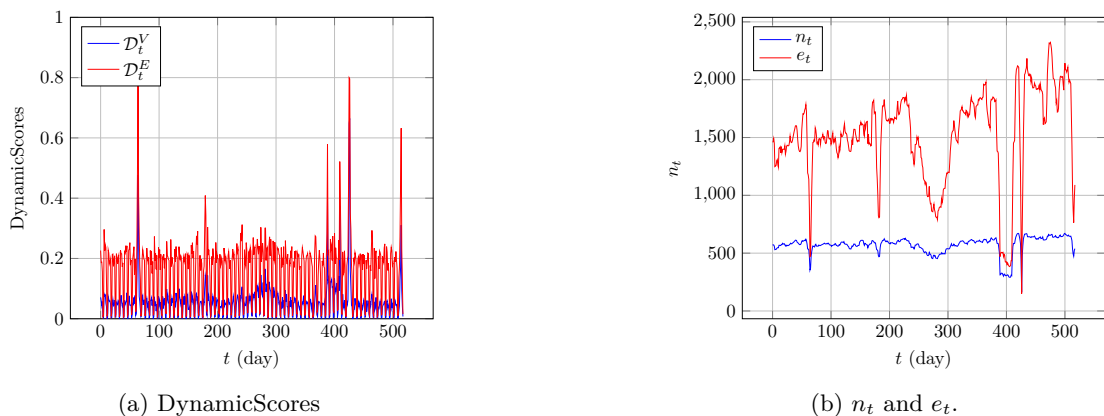


Figure 5.8: DynamicScores and graph order evolution for the Sliding Window transformation.

First-to-last Vertices and Edges Transformation The First-to-last transformation applied to the email network chosen here is the one applied to both the vertices and the edges. It gathers information week by week. For this transformation once a user sends or receives an e-mail, it is present in the network

and will be removed in the graph representing the week just after it sends or receives its last e-mail. This consideration is the same for the edges: once a user A sends at least one e-mail to B at week i , an arc is introduced in G_i ; this arc is removed at week i'' if A sends at least one e-mail to B at week $i'' - 1$ and A never sends e-mails again to B . The evolution of DynamicScores, e_t and n_t for this transformation are gathered on Figure 5.9. The vertices DynamicScore is very low, indicating the set of vertices remains steady through time. This is however not the case of the set of the edges, for which changes seem more important. This might be interpreted as the fact that, even if the staff remains almost unchanged week after week, pairs of people exchanging emails are mostly different each week. This hypothesis is strengthened by the observation that the Edges-DynamicScore remains almost constant while the number of arcs is increasing during the first period, decreasing during the last period and almost constant between weeks 30 and 50. Thus, the high value of the Edges DynamicScore cannot be attributed to the change of the number of edges, but to its composition.

One drawback is highlighted by this study. This concerns the last graphs produced by the generation. Indeed, a side effect starting is observed from week 50 to the end of the experiment, with a decrease similar to the increase in the number of vertices and edges related to the construction of the network. This decrease does not indicate that users will never interact again, but at least by the time for the simulation to end.

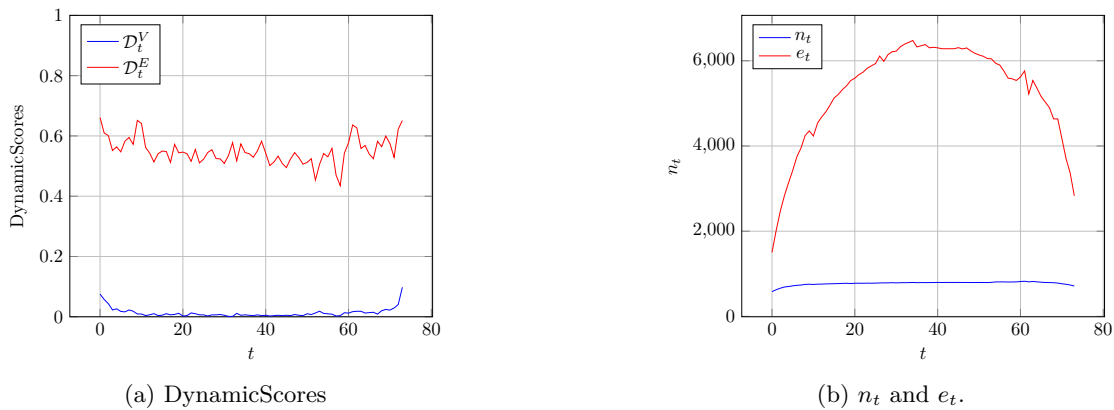


Figure 5.9: DynamicScores and graph order evolution for the First-to-last transformation.

Conclusion: as a conclusion to the analysis of this dataset, every transformation provides different information on the analysis of the network. All these analyses are complementary one another. On the one hand, the Growing Network transformation indicates a global network with a decreasing dynamic and very few new vertices introduced in the network. On the other hand, the Time Interval and the Sliding Window transformations indicates more impacting changes week after week, more important for the Time Interval transformation. The First-to-last transformation highlights some potential behavior of email communication practices between people. In addition the study of the DynamicScore indicates that two different users do not keep exchanging e-mails until the end of the experiment.

5.2.1.2 Autonomous System

The dataset considered in this Section deals with the graph of routers comprising the Internet. It is composed of sub-graphs called Autonomous Systems (AS). Each AS exchanges traffic flows with some neighbors (peers). It covers 785 days of exchange between each AS ranging from November 8th, 1997 to January 2nd, 2000. The temporal network is composed of 3015 vertices and 10695 oriented edges for the first graph of the sequence (November 8 1997) and goes up to 6474 vertices and 26467 oriented edges for the last graph of the sequence (January 2nd, 2000). This network allows addition and deletion of vertices, which means that some AS may disappear over time.

Leskovec et al. conducted a study in [31] to highlight temporal graph densification phenomenon as explained in Section 5.2.1.1. In this paper, authors show a connection between the number of edges e_t and the number of nodes n_t at a given step t . This relation is mathematically represented as follows: $e_t = bn_t^a$. This relationship is illustrated by Figure 5.10. The model was validated using the Ordinary Least Squares method, yielding $a \simeq 1.18$ and $b = 0.87$ with a R^2 value of 0.997. The theoretical values closely align with the observed data, and the exponent being significantly greater than 1. Moreover the value of b is close to 1 and the dataset is composed of 785 entries, which underscores the robustness of the analysis. This modeling does not highlight the change in the composition (addition and deletion) of vertices and edges. The DynamicScore captures these changes and the purpose of the following paragraphs is to illustrate how this metric allows such an understanding.

The study conducted here focuses on the different transformation on the sequence of events. The goal is to analyze the dynamics of the network at different scales. In order to have enough samples and a time length ensuring relevant analyses, the time length of almost all intervals is around one week. Thus, the Time Interval transformation gathers events occurring during one week. The Sliding Widow transformation has a window length of one week and a shift of one day. The Growing Network transformation also gathers data week after week. Finally, in the First-to-last transformation for vertices and for both vertices and edges, each snapshot will group interactions week by week. One important thing to mention is that graphs presented here are non-oriented. Indeed, each data file gathering exchanges between routers contains both u, v, t and v, u, t . Let us now delve into the analysis of the network.

Growing Network Transformation The Growing Network transformation applied to the sequence of events provides a time-varying graph with 104 snapshot graphs. Each graph G_i of the sequence gathers events occurring during or before week i . Thus, this time-varying graph focuses only on addition of new vertices in the network. The evolution of n_t and e_t over time is represented on Figure 5.11b and the evolution of \mathcal{D}_t^E and \mathcal{D}_t^V is represented on Figure 5.11a. Both the evolution of n_t and \mathcal{D}_t^V indicates a very low dynamics with few changes. This means that addition of new vertices in the global graph is negligible between two consecutive weeks. It is nonetheless possible to notice that the value of the DynamicScore is roughly constant and non-zero. As the graph considered here is a growing graph, which means no vertices are considered to disappear, the evolution of the size of the graph may be considered

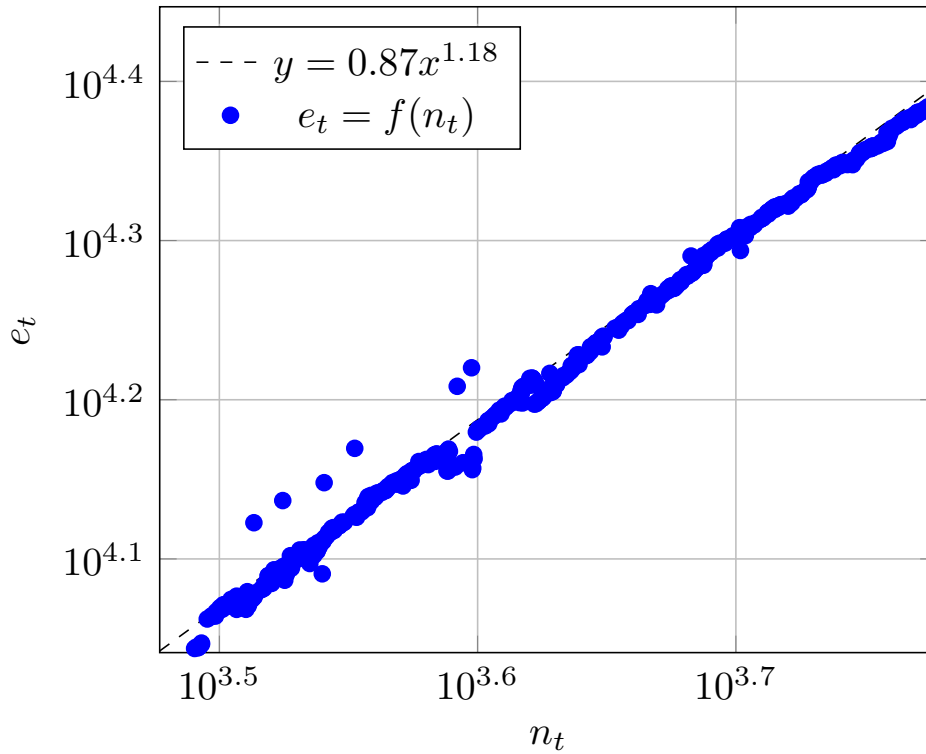


Figure 5.10: Comparison between the number of vertices and the number of edges of each graph. Each blue point represents a measured number of vertices and number of edges computed from the data analysis. The dashed black line represents the theoretical expectation.

as an exponential. The exponent computed with the mean of the Vertices DynamicScore and Theorem 6, shows that the number of vertices in the global network (the graph of all routers that have existed) could grow as $n_t = n_0 (a)^t$ with $a = 1.013$. However, the number of samples is small and the exponent too close to 1, to confirm this trend in the evolution of n_t . However, it is interesting to notice it.

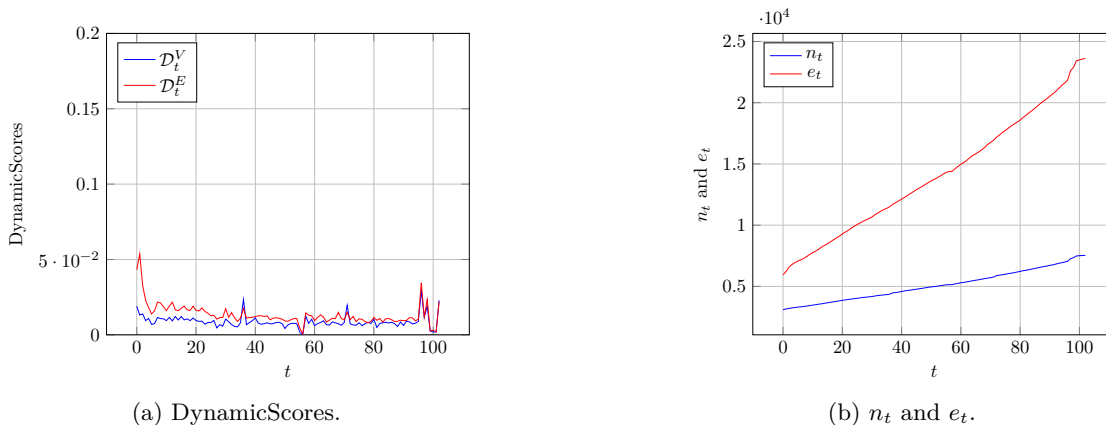


Figure 5.11: DynamicScores, n_t and e_t evolution for the Growing Network transformation for the Autonomous System network.

Time Interval Transformation Let us study the Time Interval transformation. The length of the interval considered here is one week, meaning all the graphs gathers information occurring during one week. The evolution of D_t^V and D_t^E is represented on Figure 5.12a. The evolution of n_t and e_t is represented on Figure 5.12b. DynamicScores are slightly greater than DynamicScores measured on the

growing graph. Moreover, the evolution of the size of the graph is not too important. This indicates that deletion and addition of vertices and edges are balanced with more addition than deletion in average. Two peaks are observed which may be associated to different sorts of event occurring in the network. It is either a lack of data occurring at some moments or a breakdown in the system. We do not have access to this information. It is therefore impossible to state about the nature of these events and we have chosen to ignore them in the interpretation of the analysis.

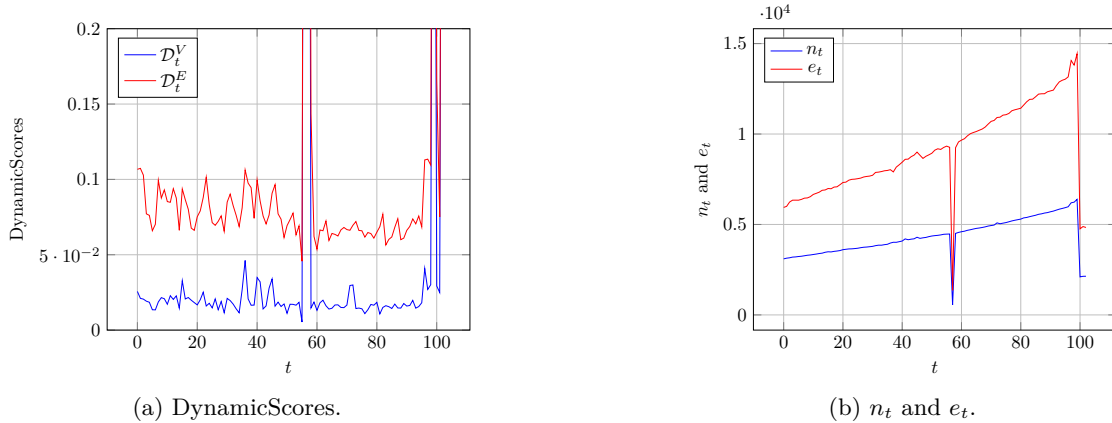


Figure 5.12: DynamicScores, n_t and e_t evolution for the Time Interval transformation for the Autonomous System network.

Sliding Window Transformation Let us study the Time Interval transformation. The length of the interval considered here is one week and the shift is one day, meaning all the graphs gathers information occurring during one week and two consecutive graphs are covering six common days. The evolution of the DynamicScores is represented on Figure 5.13a and the evolution of n_t and e_t is represented on Figure 5.13b. DynamicScores are lower than the one obtained with the Time Interval transformation. This shows that the network composition does not change too much. The Vertices-DynamicScore even indicates that very few changes occur and thus that routers remain almost the same from one day to the very next one. As compared to the results obtained with the e-mail network, it seems there is no regular pattern such as null values at periodic intervals. It is worth noting that for some network the DynamicScore may highlight regular changes in the composition which is not observed in the evolution of the graph order or in the evolution of the number of edges.

First-to-last Transformation Let us now study the FLTVE transformation. As for the other transformation, the length of the time interval is one week, meaning interactions occurring during one week between routers are gathered in the same graph. The evolution of DynamicScores are represented of Figure 5.14a and the evolution of n_t and e_t are represented on Figure 5.14b. As for the previous transformations, the DynamicScore is low indicating that connections between routers and the presence of routers in the network are robust: once a connection is established or a new router is introduced, it remains present in the network for a long time.

Conclusion The conclusion of this study on the Autonomous System (AS) network reveals several key insights into the dynamic evolution of the network. By analyzing various temporal transformations

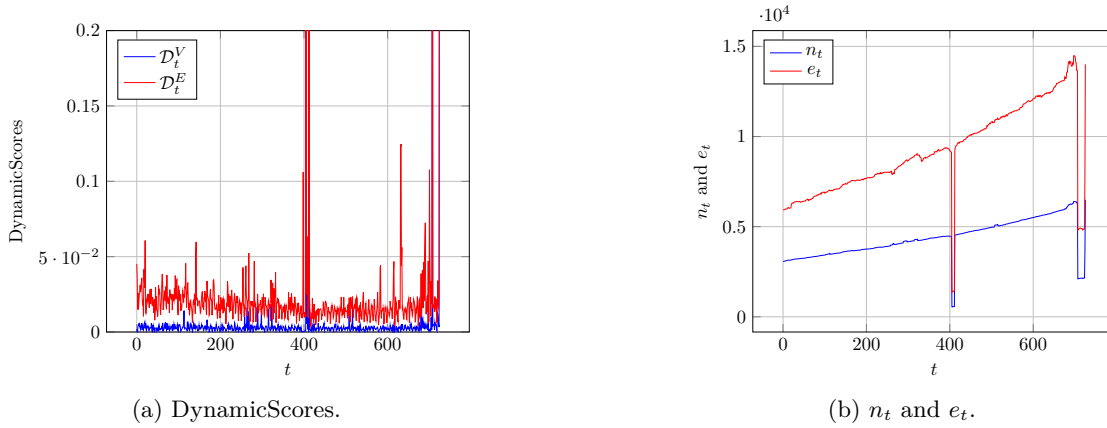


Figure 5.13: DynamicScores, n_t and e_t evolution for the Sliding Window transformation for the Autonomous System network.

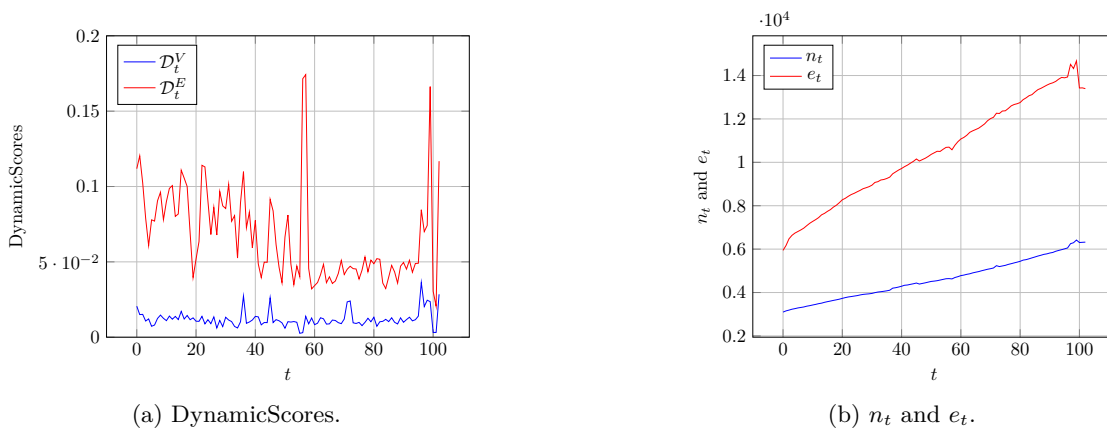


Figure 5.14: DynamicScores, n_t and e_t evolution for the FTLVE transformation for the Autonomous System network.

(Growing Network, Time Interval, Sliding Window, and First-to-Last), it is possible to understand the changes within the network at different scales. The analysis of the Growing Network shows that the addition of new vertices is relatively steady. Although the weekly growth is small, it follows a slight exponential trend, suggesting that the total number of routers increases over time. However, this trend is too weak to be definitively confirmed. The Time Interval transformation highlights a balance between addition and removal of vertices and edges. The DynamicScores, which are slightly higher than those of the Growing Network, indicate regular changes in the network structure. However, two unexplained peaks are observed, which may point to specific events or data gaps. The Sliding Window transformation reveals even fewer changes in the composition of the network, particularly from one day to the next. This suggests notable stability, with routers remaining largely unchanged from one day to the following. The First-to-Last (vertices and edges) transformation confirms the robustness of the network, where routers and their connections, once established, tend to persist over time. This observation supports the idea that Autonomous Systems form a relatively stable network within the broader Internet network. In conclusion, this analysis demonstrates that the Autonomous System network exhibits a certain level of temporal stability, with gradual additions of routers and occasional changes in connections. The low dynamics changes, as measured by DynamicScores, indicate that the network evolves slowly but steadily, remaining largely robust despite fluctuations.

5.2.2 Permanent Contacts

This section focuses on transformations where events are new permanent connections. This type of connections are common in many growing structures of biological systems. Once a connection has been established between two entities, this connection is never removed unless one of the two entities disappears. The illustrative case considered in this section is the mycelium, or thallus, of the filamentous fungus named *Podospora anserina*.

5.2.2.1 Data and Model

For building the dynamic graph, the data used have been provided by the CNRS, UMR 8236 –LIED, Université Paris Cité, France, in the context of a joint work. For the purpose of their article [30], the authors have studied the dynamics of the fungus in a constrained environment. The interested reader can find the details about the experimental setup in [14].

Starting from a germinating ascospore, a type of seed, the mycelium continuously grows until it reaches the limits of the culture area (a Petri dish). At that moment, the observation of the thallus stops. The ascospore can be considered as the root of the network from which the mycelium grows. The network is made of branches called hyphae. Over time, new hyphae are created at the occasion of what is called branching events, when a new hypha emerges from an existing one. Two hyphae can also meet and merge, an event called anastomosis. The graph model extracted from this mycelium is composed of three types of nodes. Nodes with degree one correspond to the extremity of each hypha and are called apexes. Branching and anastomosis events both create degree-three nodes. Finally, based on structural hyphae properties, degree-two nodes are distributed along the hyphae. The image 5.15 illustrates part of such a network.

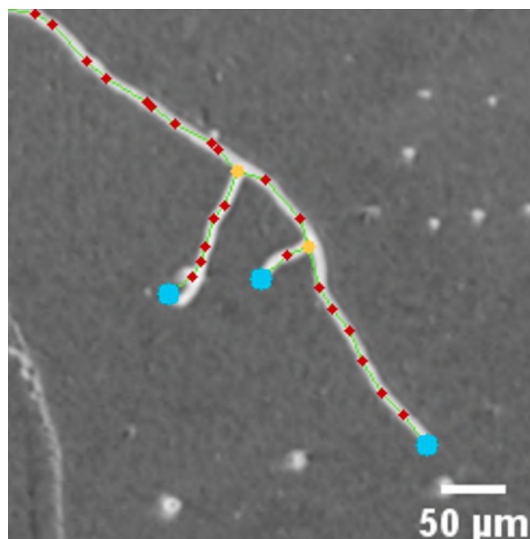


Figure 5.15: A short portion of the mycelium of *Podospora anserina*, and its graph model. Blue, degree-one, nodes are the apexes, yellow, degree-three, nodes are branching nodes and red, degree-two, nodes are structural nodes, distributed along the hyphae. (Courtesy of Thibault Chassereau)

The dynamics of the network is captured through series of images, called panoramas, taken every 18 minutes. Each panorama can be associated with a static graph. The biological experiment produces 70 panoramas, resulting in 70 time-stamped static graphs. As hyphae are not moving but only growing during the experiment, nodes, once identified, never change. This allows the construction of the dynamic graph. Figure 5.19 illustrates the geometric shape of the network.

Due to the nature of the connections between vertices, the only transformation considered here to transform the sequence of events is the Growing Network transformation defined in the previous section. However, according to biologists, it turns out that the dynamics of the fungus may be split in two phases. In the first phase, the exploration of the environment is favored, while in the second phase, a densification (from a spatial point of view) of the network can be observed, interpreted as the exploitation of resources. As illustrated in Figure 5.19, the thallus spans an almost circular surface. At each time step, this surface can be divided into two parts: an inner disk surrounded by a wide ring. According to biologists, it seems that the exploration phase is more present in the ring, while the exploitation characterizes the inner disk. Therefore, transformations studied here will try to find a way to characterize these two phases. The way transformations are considered here is to cut the whole graph according to the position of nodes within the mycelium. All nodes and incident edges belonging to the inner disk define a first dynamic graph. The second dynamic graph is obtained by considering nodes and edges belonging to the surrounding ring. Once this process is done, an analysis of the constructed time-varying graphs is performed. The focus is on studying the Dynamic Scores (of vertices and edges) to determine the significance of this metric.

To build our time-varying graphs, we use data provided by our colleagues. This data includes temporal information. Indeed, panoramas obtained every 18 minutes allow for the time-stamping of nodes. All data were then gathered in Comma Separated Values (CSV) files listing sequences of timed events. Each row in the file contains two vertex identifiers, u and v , their spatial coordinates, and the date at which each of them appears in the network, t_u and t_v respectively. As the considered mycelium is a growing network, the transformation of the sequence of events used is based on the filtering method developed in Section 5.1.2. Thus, the time-varying graph is obtained from the aggregation of every timed event following the same principle used to transform a sequence of instantaneous contacts into a growing time-varying graph (see Section 5.1.1.4).

5.2.2.2 Objectives

In addition to measuring the DynamicScore on this dynamic graph, a discussion with biologists led us to consider an interesting question underlying the dynamics of the fungus as a two-phase dynamic. Indeed, as previously mentioned, the biologists believe that two phases can be identified during the growth of the mycelium. Our goal would be to identify these two phases by dividing, at each time step and according to geometric arguments, the whole graph in two graphs.

To better understand the concept, let us go into more detail. Branching events can be of two types.

Apical branching, which occurs near the apex and is associated with the exploration phase. Lateral branching, which occurs at any point along the hypha, but not near the apex, and is associated with the exploitation phase. The densification, mentioned in the previous section, is mainly due to lateral branching. One approach to distinguish these two phases is to divide the area occupied by the thallus into two surfaces. Representing the area occupied as a disk of radius r_t growing through time, the idea is to split that disk into a disk with a smaller radius and an outer ring. Several cut-off radius have been studied ranging from 0% of r_t to 90% of r_t . Different time-varying graphs are built for every cut-off representing only the outer ring. The DynamicScore is then computed on outer ring graphs. Results of the analysis are presented in the following section.

5.2.2.3 Analysis

For the moment, only two datasets are available representing two different experiments. The authors of [30] have already shown that the evolution of the number of vertices grows as an exponential.

The first performed analysis consists in the computation of the Vertex-DynamicScore and Edge-DynamicScore on the whole dynamic graph. The results, reported on Figure 5.16, bring important information about the mycelium and its evolution. After a starting phase (the 15 first dates), both \mathcal{D}_t^V and \mathcal{D}_t^E seem to converge to a constant value (approx. 0.1). In the case of a strictly growing network, this indicates exponential growth. Moreover, as can be observed in the image, both measures evolve in exactly the same way, which means that the dynamics of the edges are similar to the dynamics of the vertices. This indicates that the network, while growing, remains largely close to a tree structure. It is worth mentioning, however, that due to anastomosis events, the network is not exactly a tree.

We also perform direct measurements on the evolution of the graph order and on the comparative evolution of nodes and edges. Both are represented in Figure 5.17a for one of these experiments. As highlighted by the results on \mathcal{D}_t^V and \mathcal{D}_t^E , these measures confirm that the evolution of the number of edges is proportional to the number of vertices, a sign of a tree-like topology.

The second analysis is performed on the portion of the dynamic graph located in the ring. The evolution of the Vertices-DynamicScore for all the cut-offs is illustrated on Figure 5.18. when the cut-off is equal to 0% this means that the ring represents the whole graph, and the inner disk is null. As shown in the graph, the curve for this case is exactly the one of Figure 5.16. When the value of the cut-off increases, between two consecutive dates, some new nodes corresponding to the growth, appear in the dynamic graph, while some nodes are removed from the dynamic graph. Indeed, as the radius of the whole network increases between two consecutive dates, some nodes present in the ring at t , belong to the inner disk at $t + 1$. This, of course, entails an increase in the value of \mathcal{D}_t^V . But, the general trend over time, for cut-off values up to 70%, is a slow decrease toward a constant value. For larger values, after 50 steps, the value of \mathcal{D}_t^V seems to increase. The stress on the 70% cut-off of the radius r_t may be explained as follows: to split a disk into two equivalent regions, it is sufficient to divide the radius by $\sqrt{2}$. The area of a circle of radius $\frac{R}{\sqrt{2}}$ is indeed half the area of a circle of radius R and $\frac{1}{\sqrt{2}} \simeq 0.707$. This would confirm some hypotheses expressed by some members of the team working on the fungus,

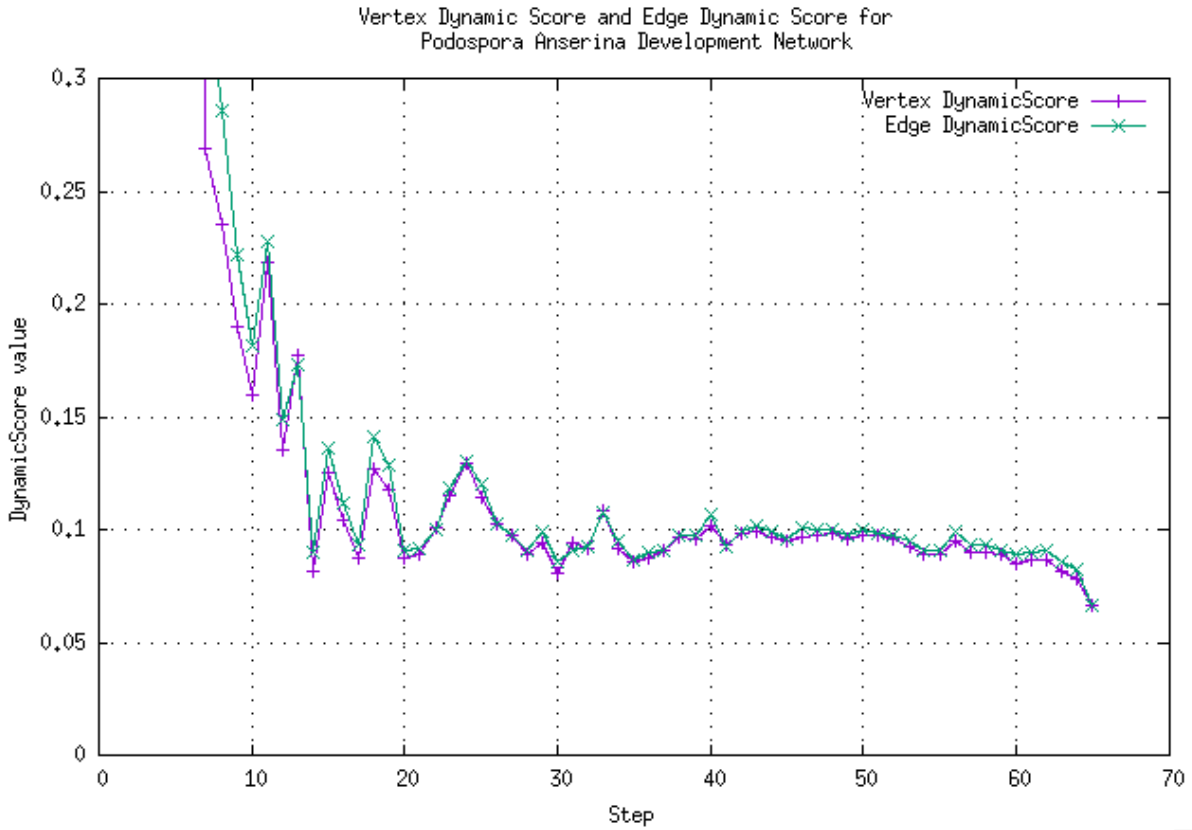


Figure 5.16: Evolution of both \mathcal{D}_t^V and \mathcal{D}_t^E for the mycelium until the end of the experiment.

who emit the idea that a balance between exploration and exploitation may be found by splitting the area occupied by the fungus into two equivalent surfaces. However, some further investigations should be performed by also computing the evolution of \mathcal{D}_t^V for the inner disk. Indeed, the growth of the mycelium is the addition of the growth occurring both in the inner disk and in the ring areas. The question of the existence of a constant cut-off value leading to an equivalent contribution of both areas is still open.

It is worth mentioning that the model of the circle used to cut the growing structure is maybe not accurate enough to observe a difference in the evolution of the DynamicScores. Discussions with the team working on the fungus explained us that the area might rather be an ellipse than a perfect disk. An illustration of the last graph in the time-varying graph is represented for one experiment is provided on Figure 5.19. This Figure highlights that the structure of the filamentous fungus may be modeled as a growing ellipse.

For the moment, the data analysis of this network has not been further investigated as this work has been started recently and we lack of perspectives to carry this work further. However, we can still make some assumptions and discuss different perspectives such as new ways to filter the whole time-varying graph. Remarks about this consideration are for instance discussed in the following section where different ways to filter the time-varying graphs are exposed.

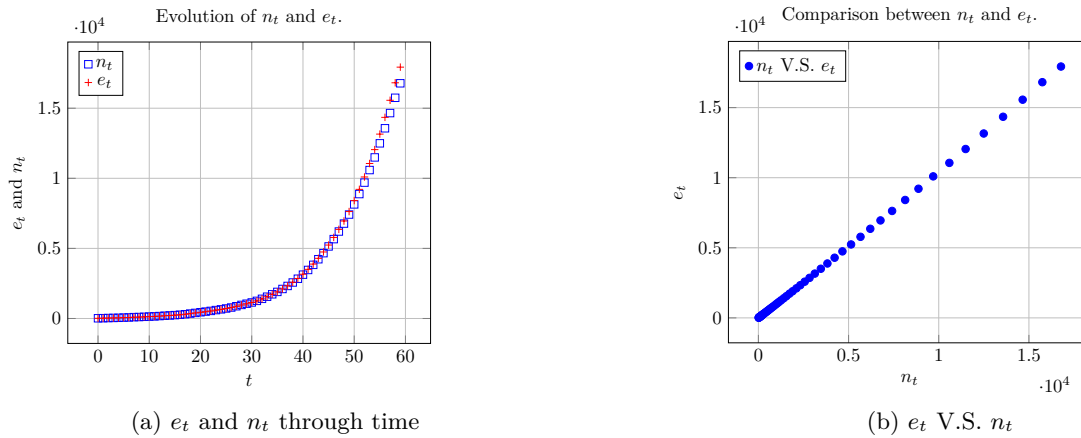


Figure 5.17: Comparison between the evolution of n_t and e_t for the whole growing dynamic graph of PA.

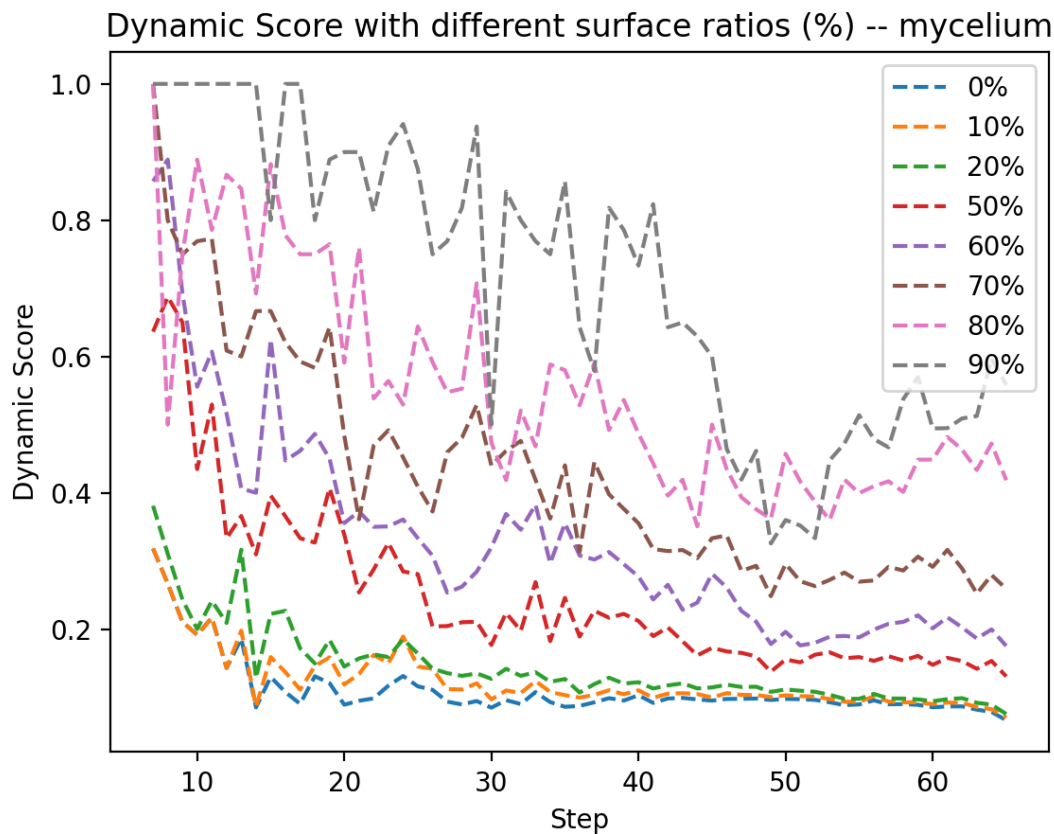


Figure 5.18: Evolution of \mathcal{D}_t^Y through time for different cut-off values of r . A cut-off value of 0% means that network is entirely contained in the ring. The inner disk is null.

5.2.2.4 Improving Filtering Methods

This section deals with several remarks aiming at improving filtering methods to obtain new analyses of this network.

The Ellipse Model One way to improve filtering methods is to consider the area occupied by the fungus as an ellipse instead of a disk. Indeed as explained in a previous paragraph, the area occupied by the fungus may be modeled as an evolving ellipse as it better fits the evolution of the fungus in its

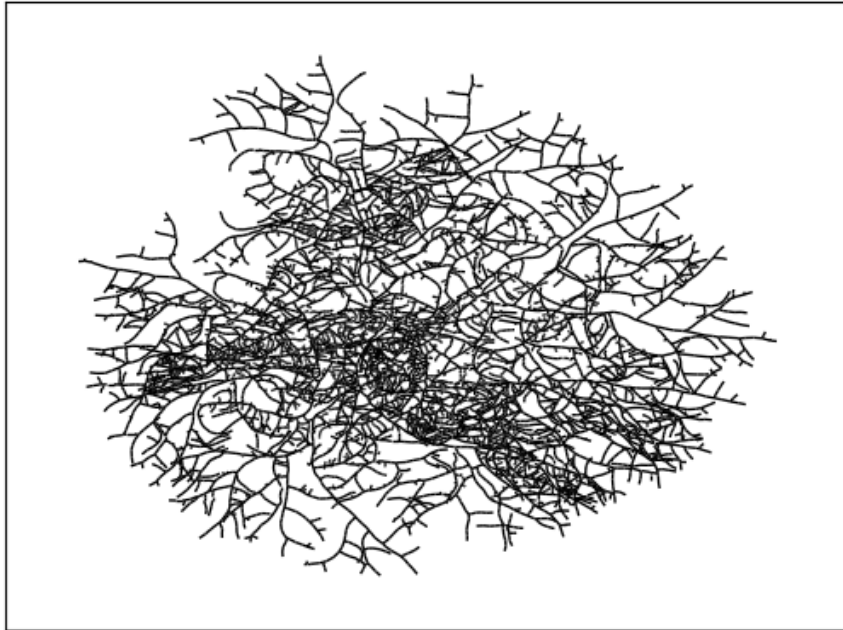


Figure 5.19: The complete network after adding all the vertices and the edges. This picture represents only the edges.

environment. The center of this ellipse may be set using the spore as it is the case for the circle model. The division of the ellipse follows the same principle as in the case of disk but they now concerns the width and the height of the ellipse. To find the values of the axes, it is possible to use inertia matrices. The eigenvalues of the matrices may be used to then compute the width and the height of the ellipse as well as its axes.

Filtering Branches One goal of the study about this mycelium is to observe a difference between the exploration phase and the exploitation one. To this end, the filtering function may focus on selecting only apical branching or lateral ones. Indeed apical branches are observed as branches used in the exploration phase while lateral branches are used in the exploitation phase. However, obtaining information about the nature of branches requires to set this information in the sequence of event, or at least to develop algorithms aiming at deciding whether a branch is apical or lateral. Some recent works of our colleagues, based on the angle between the mother hypha and the emerging one, as well as the position of the emerging hypha with respect to the apex, allow them to identify apical branching from lateral ones. These results were too recent to be exploited in our work.

General Observation

Transformation introduced in Section 5.1 have been applied to event sequences to obtain time-varying graphs. In the context of instantaneous contacts, every transformation have been applied. Each transformation offers a vision of a network at a different scale. Gathering interpretations of these transformations

provide a global vision of a network dynamic. For instance, applied on an autonomous system network, it has been shown that the network remains stable over time with a low dynamic at the week scale and a global slow increasing trend. A similar analysis performed on the network of core members of a large European research institution reveals a different evolution with a global decreasing dynamic and impacting changes week after week.

In the context of sequence of permanent contacts, transformations have a different role. Here a growing time-varying graph is first obtained using the sequence of event. Then filters applied to the network allow the extraction of a sub time-varying graph. This method have been applied to the network of a filamentous fungus called *Podospora Anserina*. For this network different filters have be considered and metrics have been applied to one category of filter: those cutting the area occupied by the fungus in two distinct regions. Metrics have been applied to these transformations but for the moment, we do not have enough data to precisely describe the dynamic of the network. Ideas have been developed to provide future analyses on the network. Most of them focus on the way sub time-varying graphs are extracted. Some simply change the geometry of the occupied area from a circle to an ellipse. Some go deeper and focus on vertex nature. It is for instance possible to filter vertices considered as apical branches or lateral ones.

Whatever the application, transformations studied here helps having a better understanding of the dynamic of networks seen as time-varying graphs. The application of multiple transformation offer different analysis at different scale of a single network. The union of all these transformations provide a broader view on the evolution of this network. This work have focused on very specific network in which edges model either instantaneous or permanent contacts. As a future perspective, it would be interesting to apply these transformations to other networks to see their impact on the interpretation of network dynamic. Example of candidates time-varying networks may be found in a survey of Peter Holme and Jari Saramäki [24].

Conclusion

In conclusion, this data analysis section focused on applying the metrics defined in Chapter 3 to real-world networks modeled as sequence of time-stamped events. The primary goal was to transform events into time-varying graphs, taking into account the nature of the events. Events in the context can represent either instantaneous or permanent contacts. For instantaneous interactions, various techniques were explored, including aggregation, time intervals, sliding windows, or first-to-last (where vertices and edges exist from their first to their last appearance). In the case of permanent contacts, a cumulative graph is first constructed and then filtered based on vertex or edge features.

These transformations provide comprehensive understanding of network dynamic through multiple perspectives. the analysis spans different time scales, allowing us to identify networks that exhibit low dynamics at the global level but high dynamics locally. It also enables the identification of networks

with both global and local trends showing low incremental changes. For permanent contacts, filtering methods were implemented and designed to study a filamentous fungus. Since the network is embedded in a metric space, filters have been used to select vertices close to a fixed coordinate. However, this work is not yet complete enough to provide a definitive analysis of network dynamics.

For future research, it would be also beneficial to extend this methodology to other types of contact networks and to study different transformation techniques to gain a deeper understanding of the underlying dynamics of these networks. Further perspectives and ideas have been discussed to advance the study of permanent contact networks.

The early sections of this dissertation focus on the fundamental object of this study: graphs as explored through both complex networks and time-varying graphs. In the analysis of complex networks, attention is initially directed toward properties observed in real-world systems, such as the small-world phenomenon, scale-free distributions, strong clustering, and navigability. From these observations, researchers have designed mechanisms capable of producing graphs with these properties such as the Barabási-Albert and Watts-Strogatz models.

The study of time-varying graphs builds on static graph theory by introducing the concept of time. Unlike static graphs, where properties are analyzed at a single moment, time-varying graphs examine properties over an entire sequence of graphs, with key metrics such as time-connectivity and reachability coming into play. A few models, such as the Microcanonical Randomized Reference Model and the Edge-Markovian Graph Generator, have been developed to create temporal graphs for the purpose of studying these dynamic properties.

A critical intersection of these two fields lies in the emergence of dynamics, understood as iterative processes that generate sequences of graphs from an initial condition. This dissertation centers on these dynamics, treating them as the main object of study and using them to bridge the gap between complex network theory and time-varying graph theory, thereby providing new insights into the evolving nature of dynamic graphs.

The main contribution of this thesis lies in the study of dynamic graphs, particularly through the investigation of iterative processes that generate them. A key question we explored is what occurs when constraints are not imposed on the graph generator. Specifically, what happens when the number of vertices in the graph is allowed to vary over time? This raises several important questions: First, is it

possible for a graph to eventually become empty? Second, how can we quantify the changes that occur between successive steps in an iterative process?

To address these questions, we introduced two key metrics. The first, *sustainability*, is defined as a condition in which a graph generator produces instances that remain both non-empty and non-periodic over time. This metric helps us determine whether the generated graphs maintain structural complexity, avoiding collapse or repetitive patterns. The second metric, the *DynamicScore*, quantifies the changes that occur between two consecutive time steps. This score is defined at both the vertex level (V-DynamicScore) and the edge level (E-DynamicScore).

The initial observations from applying these metrics were insightful. When applied to well-known iterative processes, such as the Barabási-Albert (BA) model, we found that the dynamic graphs produced by this model are sustainable. Additionally, we observed that both the V-DynamicScore and E-DynamicScore decrease towards zero over time, indicating a stabilization in the network's dynamics. For the Edge-Markovian Graph Generator (EMGG), we demonstrated that sustainability is parameter-dependent and that the E-DynamicScore is influenced primarily by a single parameter on average.

Following this, we explored the importance of having sustainable graph generators in addressing broader issues. For example, if an iterative process produces sustainable graphs, is it possible for information introduced at an initial time step to persist over time? Additionally, how many vertices are necessary to ensure that this information reaches all vertices present at a given future time? These questions correspond to the problems of persistent information and information covering, respectively. However, for both problems, we proved that knowing whether the graphs produced are sustainable is not sufficient to provide definitive answers.

A more detailed study of a graph generator, even with simple parameters, revealed that determining sustainability is not straightforward. This requires a systematic classification of the parameters into distinct families. The Degree Driven Dynamic Geometric Graph Generator (D3G3) served as a rich example for the study of sustainability. We observed that certain parameter families result in the preservation of a stable graph over time, ensuring sustainability. Additionally, a variant of this model, the Redistributed Degree Driven Dynamic Graph Generator (RD3G3), was used to demonstrate that even when the evolution of the order of the graph can be estimated, further discussion and additional tools are still required to definitively resolve the question of sustainability.

Another important finding emerged from the application of DynamicScores to the study of real-world networks. When real networks are modeled as temporal sequences of instantaneous events, transforming these events into temporal graphs and analyzing them using DynamicScores allows us to profile the dynamics of the network. For example, a network may exhibit decreasing dynamics as its size grows, while still displaying high dynamics week-to-week. Conversely, a network could experience strong growth over time, yet demonstrate low dynamics from one week to the next. This dynamics profiling provides

valuable insights into how the structure and activity of real-world networks evolve over time.

An initial exploration into applying these metrics to networks where events model the creation of permanent contacts has also been undertaken. By applying filters to the graphs obtained through successive aggregation of interactions, coupled with the use of DynamicScores, we aim to extract temporal subgraphs whose dynamics differ from those of the global graph and other subgraphs. However, the work conducted so far has only laid the foundation for this analysis, and further research will be necessary to draw more definitive conclusions.

Several perspectives for future research can be considered. First, regarding sustainability, new questions could be introduced, similar to those posed in communication problems. For instance, one could investigate whether there exists a set of vertices that allows reaching all the vertices that have existed throughout the evolution of a graph. It would also be interesting to explore other information diffusion algorithms. This would broaden the study of information persistence and information coverage problems.

In terms of the study of the D3G3 generator, several open questions remain. For example, it would be worth investigating the causes that allow a graph to maintain a growing structure. Further exploration of different parameter families could also be undertaken. In fact, ongoing research on a variant of the integer segment family aims to explain what enable the maintenance of a growing geometric structure in the above mentioned paragraphs.

Regarding the DynamicScore, several lines of work are possible. Some of them concern the study of the D3G3 and RD3G3 models, for which we observe, but cannot yet prove, similar average DynamicScore values across different parameter families. For example, in the case of segments, the average V-DynamicScore is approximately $\frac{2}{3}$, which is not an insignificant value. Other potential directions include the work already initiated on real networks. For instance, one could use the vertices or edges DynamicScore to split graphs in cases where the networks involve permanent contacts.

My Publications

- [1] Vincent Bridonneau, Frédéric Guinand, and Yoann Pigné. DynamicScore: A novel metric for quantifying graph dynamics. In Hocine Cherifi, Luis M. Rocha, Chantal Cherifi, and Murat Donduran, editors, *Complex Networks & Their Applications XII*, pages 435–444. Springer Nature Switzerland.
- [2] Vincent Bridonneau, Frédéric Guinand, and Yoann Pigné. Persistence of information in dynamic graphs. In *Proceedings of the French Regional Conference on Complex Systems, May 29-31, 2024, Montpellier, France*, pages 205–213. Zenodo.
- [3] Vincent Bridonneau, Frédéric Guinand, and Yoann Pigné. Dynamic Graphs Generators Analysis: An Illustrative Case Study. In David Doty and Paul Spirakis, editors, *2nd Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2023)*, volume 257 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:19, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [4] Vincent Bridonneau, Frédéric Guinand, and Yoann Pigné. Asymptotic dynamic graph order evolution analysis. *Applied Network Science*, 9(1):21, 2024.

Bibliography

- [5] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.
- [6] Ginestra Bianconi, Richard K. Darst, Jacopo Iacovacci, and Santo Fortunato. Triadic closure as a basic generating mechanism of communities in complex networks. *Phys. Rev. E*, 90:042806, Oct 2014.
- [7] S Boccaletti, V Latora, Y Moreno, M Chavez, and D Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, 2006.
- [8] Marián Boguñá, Dmitri Krioukov, and K. C. Claffy. Navigability of complex networks. *Nature Physics*, 5(1):74–80, 2009. Number: 1 Publisher: Nature Publishing Group.
- [9] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.
- [10] Arnaud Casteigts, Michael Raskin, Malte Renken, and Viktor Zamaraev. Sharp thresholds in random simple temporal graphs. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–326, 2022. ISSN: 2575-8454.
- [11] Andrea E. F. Clementi, Claudio Macci, Angelo Monti, Francesco Pasquale, and Riccardo Silvestri. Flooding Time of Edge-Markovian Evolving Graphs. *SIAM Journal on Discrete Mathematics*, 24(4):1694–1712, January 2010.
- [12] John Conway et al. The game of life. *Scientific American*, 223(4):4, 1970.
- [13] Josep Díaz, Dieter Mitsche, and Xavier Perez. Dynamic random geometric graphs. *arXiv preprint cs/0702074*, 2007.
- [14] J. Dikec, A. Olivier, C. Bobée, Y. D’Angelo, R. Catellier, P. David, F. Filaine, S. Herbert, Ch. Lalanne, H. Lalucque, L. Monasse, M. Rieu, G. Ruprich-Robert, A. Véber, F. Chapeland-Leclerc,

- and E. Herbert. Hyphal network whole field imaging allows for accurate estimation of anastomosis rates and branching dynamics of the filamentous fungus *podospora anserina*. *Scientific Reports*, 10(1):3131, Feb 2020.
- [15] Anthony H. Dooley. Markov odometers. In Sergey Bezuglyi and Sergiy Kolyada, editors, *Topics in Dynamics and Ergodic Theory*, London Mathematical Society Lecture Note Series, pages 60–80. Cambridge University Press, 2003.
- [16] Paul Erdos and Alfred Renyi. On random graphs i. math. *Debrecen*, 6:290–297, 1959.
- [17] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.
- [18] William Feller. *An introduction to probability theory and its applications*. John Wiley & Sons, New Jersey, USA, 1991.
- [19] Laetitia Gauvin, Mathieu Génois, Márton Karsai, Mikko Kivelä, Taro Takaguchi, Eugenio Valdano, and Christian L. Vestergaard. Randomized reference models for temporal networks, 2020.
- [20] C. Gkantsidis, M. Mihail, and A. Saberi. Hybrid search schemes for unstructured peer-to-peer networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1526–1537, Miami, FL, USA, 2005. IEEE.
- [21] Fabíola Greve, Luciana Arantes, and Pierre Sens. What model and what conditions to implement unreliable failure detectors in dynamic networks? In *Proceedings of the 3rd International Workshop on Theoretical Aspects of Dynamic Distributed Systems*, pages 13–17. ACM, 2011.
- [22] Charles Robert Hadlock. *Field theory and its classical problems*. American Mathematical Society, Rhodes Island, USA, 2000.
- [23] Petter Holme. Network reachability of real-world contact sequences. *Physical Review E*, 71(4), 2005.
- [24] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012. Temporal Networks.
- [25] Paul Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50, 1912.
- [26] David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002.
- [27] Gueorgi Kossinets and Duncan J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.
- [28] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, September 2010. Publisher: American Physical Society.
- [29] Marcelo Kuperman and Guillermo Abramson. Small world effect in an epidemiological model. *Physical Review Letters*, 86(13):2909–2912, March 2001. arXiv:nlin/0010012.

- [30] Clara Ledoux, Cécilia Bobée, Éva Cabet, Pascal David, Frédéric Filaine, Sabrina Hachimi, Christophe Lalanne, Gwenaél Ruprich-Robert, Éric Herbert, and Florence Chapeland-Leclerc. Characterization of spatio-temporal dynamics of the constrained network of the filamentous fungus *podospira anserina* using a geomatics-based approach. *PLOS ONE*, 19(2):e0297816, 2024. Publisher: Public Library of Science.
- [31] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1):2, 2007.
- [32] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- [33] Alessandro Muscoloni and Carlo Vittorio Cannistraci. A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities. *New Journal of Physics*, 20(5):052002, May 2018.
- [34] M. E. J. Newman. Spread of epidemic disease on networks. *Physical Review E*, 66(1):016128, July 2002.
- [35] Fragkiskos Papadopoulos, Maksim Kitsak, M. Angeles Serrano, Marian Boguna, and Dmitri Krioukov. Popularity versus similarity in growing networks. *Nature*, 489(7417):537–540, September 2012.
- [36] Mathew Penrose. *Random Geometric Graphs*. Oxford University Press.
- [37] Garry Robins, Pip Pattison, and Peng Wang. Closure, connectivity and degrees: new specifications for exponential random graph (p^*) models for directed social networks. *Unpublished manuscript. University of Melbourne*, 2006.
- [38] Nicola Santoro, Walter Quattrociocchi, Paola Flocchini, Arnaud Casteigts, and Frederic Amblard. Time-Varying Graphs and Social Network Analysis: Temporal Indicators and Metrics, February 2011. arXiv:1102.0629 [physics].
- [39] Georg Simmel. *Soziologie*. Duncker & Humblot Leipzig, 1908.
- [40] Georg Simmel. *Sociology: Inquiries into the Construction of Social Forms*. BRILL, 2009. Google-Books-ID: hD6xCQAAQBAJ.
- [41] Steven H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, March 2001.
- [42] JEFFREY TRAVERS and STANLEY MILGRAM. An experimental study of the small world problem**the study was carried out while both authors were at harvard university, and was financed by grants from the milton fund and from the harvard laboratory of social relations. mr. joseph gerver provided invaluable assistance in summarizing and criticizing the mathematical work discussed in this paper. In Samuel Leinhardt, editor, *Social Networks*, pages 179–197. Academic Press, 1977.
- [43] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, June 1998.

-
- [44] Bernard M Waxman. Routing of multipoint connections. *IEEE journal on selected areas in communications*, 6(9):1617–1622, 1988.
- [45] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 194–205, 2002.
- [46] B. Bui Xuan, A. Ferreira, and A. Jarry. COMPUTING SHORTEST, FASTEST, AND FOREMOST JOURNEYS IN DYNAMIC NETWORKS. *International Journal of Foundations of Computer Science*, 14(2):267–285, 2003.
- [47] Konstantin Zuev, Marián Boguñá, Ginestra Bianconi, and Dmitri Krioukov. Emergence of Soft Communities from Geometric Preferential Attachment. *Scientific Reports*, 5(1):9421, August 2015.