



MASTER THESIS

2019/2020

STUDENT NAME: **MATTEO BOIDI – S254312**

RESEARCH TOPIC

*Network Science applied to Project Management:
Line Graph and Z configurations*

Abstract:

The traditional analysis and practices of PM embody structural limits. A new approach through network science can fill some of these limits. Representing a project with evolving networks and measuring its structural measures can offer a useful tool for risk management.

This research is born thanks to a collaboration with ENEL Green Power. This thesis recapitulates the beginning of the collaboration with ENEL. They shared two projects, i.e., a solar and a wind farm, with which it has been built the network.

This approach will be very innovative and will be a starting point for future studies in which it will be possible to exploit these interconnections to more accurately estimate the riskiness of the individual node and project as a whole, and, with the adoption of machine learning, make predictions.

“If we consider the humankind journey to discovering the secrets of nature, we can say that it has been punctuated by two revolutionary intellectual events. The first one was the birth of philosophy in the ancient Greek colonies, during the 6th century BC. The second one was the formulation of the scientific method, proposed by Galilei and Newton in the 17th century AD, which is based on the experiments as tools for inquiring nature. Now, we are expecting a third revolutionary event that will allow us to untangle Complex Systems.”
(Prof. Pier Luigi Gentili)

ACKNOWLEDGMENT

I would like to thank Paolo Eugenio Demagistris Fabrizio Monge and Enrico Fenoaltea sincerely. Without their support, contribution, commitment, and availability, it would not be possible to pursue such research.

INDEX

1. LITERATURE REVIEW
 - 1.1. Network science
 - 1.2. Economy and Projects as complex systems
 2. “ANT” PROJECT
 - 2.1. Overview
 - 2.2. Introduction to traditional practices in PM
 - 2.3. The Algorithm
 - 2.4. The Code
 3. APPLICATION TO STOCK MARKET
 - 3.1. Database
 - 3.2. Network building, link and correlation
 - 3.3. Stock market’s structural measures
 - 3.3.1. Modularity
 - 3.3.2. Density
 - 3.3.3. Average clustering
 - 3.3.4. Average shortest path
 - 3.3.5. Diameter
 - 3.3.6. Network Transitivity
 - 3.4. Results
 4. DISCUSSIONS AND FUTURE RESEARCH
 5. BIBLIOGRAPHY
- APPENDIX

1. LITERATURE REVIEW

1.1 Network Science

Reductionism is scientifically applied methodologically in the pursuance of attempting to explain a whole system by its constituents, particular parts, and their interactions.

In order to understand what is a complex system is useful to start from the definition of *reductionism*: the explanation of complex life-science phenomena and processes by the laws of chemistry and physics. Alternatively: a theory or procedure with the aim of reducing complex phenomena to simple terms.

The system of interest within a reductionist approach is unpacked into small and isolated portions of the world, and the parameters involved are under full control [1].

A peculiar feature of reductionism is the capability of modelling phenomena that are nonlinear through models that are linear by approximating the variables linearly. Nevertheless, reductionist approaches are limited; they can treat limited classes of real-world systems. The complexity intrinsic in naturally real-world arising phenomena cannot be entrenched within the theoretical analysis.

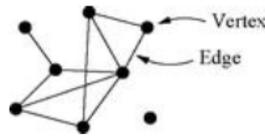
Reductionism is in contrast with *Holism* (system's proprieties cannot be inferred and determined by its constituent parts. Summarized: the ensemble is more than the sum of its parts alone). The holistic thinking, that dates back to Aristotle (the whole is greater than the sum of its parts), has been institutionalised during the 50s; indeed, attempts to find a mathematical equation¹ able to express a general systems theory (GST) were pursued².

One of the most popular extensions of the scientific method, in order to deal with nonlinear phenomena explicitly and become more integrationist, is up to the field of Complex Networks [1].

From an academic standpoint, networks are made up of points connected in pairs by links. In the literature, points are labelled as vertices or nodes while links edges [2].

¹ Checkland, P. B. 1999. *Systems Thinking, Systems Practice*. Chichester, UK: John Wiley & Sons Ltd.

² Peter Checkland, *Soft systems methodology: a thirty year retrospective*.



The simplified representation achievable through networks reduces interacting systems to abstract structures seizing the connection patterns.

The main topics inward the network landscape that are being deepened are three:

- i) the nature of the individual components;
- ii) the nature of the connections or interactions;
- iii) the architecture of connections between units.

The latter, the particular pattern of interactions, plays a pivotal role in understanding systems' behaviour. A network can be composed of a system's pattern of connections where vertices are the system's components while edges the connections.

A large set of tools are already in place – statistical, computational, and mathematical – for analysing, understanding, and modelling networks. Networks are a powerful means for patterns depiction of the system's parts interactions or connections [3].

A lot of aspects of the reality can be represented through a network: a system whose architecture is made of individual parts or components linked together in many ways; i.e., that interact with each other.

The application of *Network Science* has recent origins, but the theoretical and abstract study of these objects was borne in more remote times with *Graph Theory*.

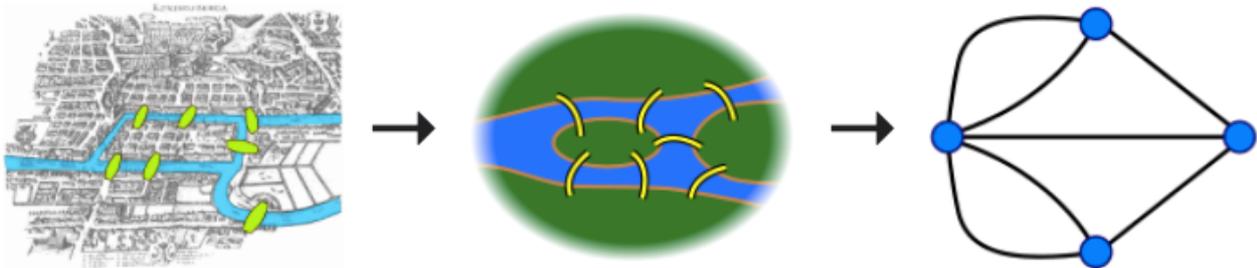
Historically, the study of networks dates back to 1736 when a mathematician called Leonard Euler came up with the negative resolution of the Seven Bridges Problem of Königsberg laying the ground for *Graph Theory* and so prefiguring the *topology* [3].

Königsberg (current Kaliningrad, Russia) was located on the banks of the river Pregel, and comprehended two masses of land linked to each other through seven bridges. The riddle consisted of finding a path crossing all the seven bridges exactly once each [4].

Euler determined that the problem has no solution [3]. He proved this making use of a graph, thanks to this mean he was able to abstract away all the details of the original problem except for its connectivity: the graph has four nodes, the lands, and seven links joining the vertices with the same layout of Königsberg's bridges.

He demonstrated that there is no Eulerian path - a path in which each edge is traversed exactly once - since any such path, except for the endpoints, must at the same time enter and leave every node it passes through; so there can be two nodes with an odd number of edges attached - odd *degree* - within

the network. In the Königsberg graph, all the four nodes have an odd degree; it follows that there is no solution [3].



Euler's proof is considered to be the first theorem in the field of discrete mathematics called *Graph theory*, which is the principal mathematical language for describing the properties of networks. A network is, in a simplistic form, a collection of discrete elements (nodes) and connections (edges) that link the components.

The abovementioned example is a *static* graph, a graph whose structure is fixed.

The topology of a complex system was firstly modelled by random graphs. The forerunners of dynamic networks are Erdős and Rényi with the so-called *random networks* [5]: $G(n,p)$ the graph is built linking vertices randomly. In the graph, each link is included within the graph probabilistically (with a probability p) and independently from every other link. Consequently, graphs with M links and v vertices have equal probability of: $p^M(1-p)^{\binom{v}{2}-M}$

Despite the formalism and comprehensiveness of the theoretical results obtained by Erdős and co-workers, random networks ultimately are not a good fit for modelling natural structures and phenomena. Indeed, heterogeneous structuring, not the relative uniformity and simplicity of *ER* networks, is the rule in Nature [1].

During the end of the 90s and the beginning of the 00s and to the credit of sociologists, networks started to be systemically applied to represent and model natural phenomena, with an obvious focus on social relations [6]. The concept of *small-world* took hold and its ubiquity, as the following papers [7], [8], and [9].

The characteristics of these networks are small average shortest path lengths between pairs of nodes and relatively high clustering coefficients.

The Watts and Strogatz, *WS*, model' mechanism:

1. Built a ring over n nodes.
2. Every node in the ring is linked with its k nearest neighbours ($k-1$ neighbours if k is odd).

3. For every link in the “ n -ring with k nearest neighbours,” shortcuts are created by replacing the original edges $u-v$ with a new edge $u-w$ with a uniformly random choice of existing node w at rewiring probability p .³

The WS model is used to build small-world networks.

Later on, another important characteristic came up called *power-law* [10]:

“A common feature of real-world networks is the presence of hubs or a few nodes that are highly connected to other nodes in the network. The presence of hubs will give the degree distribution a long tail, indicating the presence of nodes with a much higher degree than most other nodes. Scale-free networks are a type of network characterized by the presence of large hubs. A scale-free network is one with a power-law degree distribution. For an undirected network, we can just write the degree distribution as: $P(k) \sim k^{-\gamma}$

where γ is some exponent. This form of $P(k)$ decays slowly as the degree k increases, increasing the likelihood of finding a node with a very large degree. Networks with power-law distributions are called scale-free because power laws have the same functional form at all scales”⁴.

Further investigation brought the *scale-free* paradigm [11], [12], and, more importantly, exhibited the presence of the scale-free organization in man-made networks [13], [14].

The scale-free topology within a net is characterized if the following three sufficient conditions hold:

- Growth: older vertices have a higher number of connections
- Preferential attachment: new vertices tend to be attached to vertices with many connections (probability is proportional to the number of links)
- Model variants also considering vertex fitness

During this period, we have attended the birth of *Complex Networks*, i.e., networks with a structure that is complex, irregular, and dynamically evolving in time [15]. Such movement caught on also thanks to the increased computing powers [16].

The following citation sums up the reason why this field has become a powerful tool:

“Because of its generality for representing connectivity in diverse real systems in an integrative way, complex networks are promising for integration and unification of several aspects of modern science, including the inter-relationships between structure and dynamics.”⁵

³ Lunhan Luo and Jianan Fang, A Study of How the Watts-Strogatz Model Relates to an Economic System’s Utility.

⁴ https://mathinsight.org/scale_free_network

⁵ Luciano da Fontoura Costa et al. (2011) Analyzing and modeling real-world phenomena with complex networks: a survey of applications, *Advances in Physics*, 60:3, 329-412.

The pivotal aspect of graph theory is its ability to abstract away the details of a problem and so being able to extrapolate the topological features of a system. It is a language for describing abstract models, and it is a useful mean for the analysis of empirical data [3].

Behavioural characteristics can be deduced by the structural properties of networks like the connectedness, and so it is possible to size, for example, an epidemic or the transmission of global information [3].

Graph properties can be viewed in terms of probability distributions, given that graphs are considered stochastic objects and not purely deterministic ones [5].

The great watershed between Graph Theory and Network Science is that networks are not static; they evolve in time according to different dynamical rules [1]. Indeed, graph theory is not so relevant to systems arising in the real world.

Dynamical processes are embodied in real-world networks, such as the removal or addition of vertices and/or edges over time. Therefore, interacting agents' networks have not only topological features and properties but also dynamical ones [17].

1.2 Economy and Projects as complex systems

In order to understand the economy in a better and different way, it seems appropriate to treat it as a *complex system*.

Characterization of a complex system is needed before proceeding to treat the economy and so financial markets as it.

In 1977 Prigogine, Allen, & Herman defined the characterization of the modern complexity theory [18]. This theory, an ontological science, came up during the first half of the 20th century on the enthusiasm of evolution theory of Darwin. The science of complexity's seeds are a combination of Newtonian and thermodynamic universal views applied to open systems [37].

Complex systems consist of profuse unique elements interacting in multiple form. These elements can adapt, change, and learn; the links and connections loosen, reform, and change over time shifting the system's boundaries. The ways these elements interplay with each other are *reflexive*.

This field studies the way patterns, forms, and institutions emerge and become established and then constantly disturbed, challenged and influenced by particular events occurring in particular places at particular times. Events, usually unpredictable, invade the patterns and so changing their behaviours.

Indeed, a characterization of the *future* can be expressed as it follows: a complex combination dependant on (i) the effect of specific variations or events at particular places and times, and (ii) the effect present patterns. These variables and their interplay shape what happen next, a *path-dependency* is inferable. The unfolding future relies on patterns chartered in the past that interplay with current events.

The literature defined a set of core features associated with the concept of complex systems.

The scientific field is trying to postulate a unifying framework in the pursuance of modelling and understanding systems whose behaviour is difficult to control and predict, i.e., human brain and world economy.

A taxonomy is proposed for clarification purposes.

It is useful to report some quotations from “*Science* 2 April 1999” [19]:

- “To us, complexity means that we have structure with variations.” (18, p. 87)
- “In one characterization, a complex system is one whose evolution is very sensitive to initial conditions or to small perturbations, one in which the number of independent interacting components is large, or one in which there are multiple pathways by which the system can evolve. Analytical descriptions of such systems typically require nonlinear differential equations. A second characterization is more informal; that is, the system is “complicated” by some subjective judgment and is not amenable to exact description, analytical or otherwise.” (52, p. 89)
- “In a general sense, the adjective “complex” describes a system or component that by design or function or both is difficult to understand and verify. [...] complexity is determined by such factors as the number of components and the intricacy of the interfaces between them, the number and intricacy of conditional branches, the degree of nesting, and the types of data structures.” (50, p. 92)
- “Complexity theory indicates that large populations of units can self-organize into aggregations that generate pattern, store information, and engage in collective decision-making.” (39, p. 99)
- “Complexity in natural landform patterns is a manifestation of two key characteristics. Natural patterns form from processes that are nonlinear, those that modify the properties of the environment in which they operate or that are strongly coupled; and natural patterns form in systems that are open, driven from equilibrium by the exchange of energy, momentum, material, or information across their boundaries.” (51, p. 102)
- “A complex system is literally one in which there are multiple interactions between many different components.” (40, p. 105)

- “Common to all studies on complexity are systems with multiple elements adapting or reacting to the pattern these elements create.” (2, p. 107)
- “In recent years the scientific community has coined the rubric ‘complex system’ to describe phenomena, structure, aggregates, organisms, or problems that share some common theme: (i) They are inherently complicated or intricate [...]; (ii) they are rarely completely deterministic; (iii) mathematical models of the system are usually complex and involve nonlinear, ill-posed, or chaotic behaviour; (iv) the systems are predisposed to unexpected outcomes (so-called emergent behaviour).” (14, p. 410)

The last quotation, from the physicist Nigel Goldenfeld, that is beneficial to state is the following: “Complexity starts when causality breaks down” [20].

Indeed, the proprieties abstracting from the above-mentioned lists, associated with complex systems are [18]:

- i) **Nonlinearity**: output changes are not proportional to the input ones; there are no straight-line or direct relationships between independent and dependent variables. Equations of motion’s nonlinearity have a tremendous impact on different microstates achievable given different values of the initial conditions [21]. Nonlinear interactions are also defined as *synergistic* [18].
- ii) **Feedback**: the theory of causal graphs provides the prevalence of feedback in a complex system [22].
- iii) **Robustness and decentralization**: given the abovementioned characteristic, complex systems are decentralized and distributed, so they are stable, antifragile, under perturbations. In a computational language is the ability to correct errors [23]. Charles Bennet’s statement: “Spontaneous order; a huge number of uncoordinated interactions gives rise to an order.”
- iv) **Spontaneous order, Emergence**: a huge number of uncoordinated interactions give rise to a specific order. Downward causation [24]. A property is defined as emergent if it is impossible to predict it albeit all system’s characteristics are known. The elements interplay within the network create properties that belong to the whole.
- v) **Hierarchical organization**: within complex systems there are often hierarchical and sub systems [25].
- vi) **Large size systems**: a system with numerous parts [26] and [27].

Complex systems’ features, listed above, describe an entity organized within several structure levels and proprieties interacting with each other and showing lawlike and casual regularities, order, a kind of symmetries and periodic behaviour.

The physicist Gentili Pier Luigi description is pedagogical: “A Complex System is composed of many constituents, which often are diverse, if not unique. These constituents are strongly interconnected. Therefore, a Complex System can be described as a network. Such natural networks are maintained far from the condition of thermodynamic equilibrium by external and/or internal gradients of intensive variables, such as temperature, pressure, concentrations of chemicals, et cetera. Complex Systems that involve only inanimate matter are driven by force fields. On the other hand, the behaviour of a Complex System that includes living beings is information-based. In fact, a peculiarity of the living beings is that of exploiting matter and energy to encode, process, store, and communicate information.”⁶

Statistical complexity ([28], [29] and [30]) of a system, is the appropriate measure able to capture the qualitative notion of complex order; indeed, is statistical and not deterministic [31].

Nowadays, complex system science has been considered a new field of science that studies how the components of a system give rise to system’s collective behaviours, and how the system reaches out with its environment.

The study of those systems breaks down into three main fields:

1. The study of how interactions give birth to behaviour’s patterns.
2. The study of how to characterize and describe complex systems.
3. Understanding, through the evolution and formation of patterns, the creation of complex.

Given the above overview of complex systems, it is time to present why the economy can be treated as a complex system.

A natural tendency for patterns of relationships to emerge when things can be connected in many different ways. Economists and marketers study these patterns. The nature of these patterns is fluctuating; indeed, they are composed of a collection of individual and varied actions and behaviours. Furthermore, these patterns are exposed to force majeure events like tsunamis, external shocks, pandemics, wars, migration, climate change; this reinforces the abovementioned fluctuating nature, events are mainly responsible for patterns shifts. As Varoufakis expressed in its book [32], change is rarely spontaneous; it is dependent on exogenous factors.

Keynes stated that classical economy theorists resemble Euclidean geometers in a non-Euclidean world [33]. The abundance of evidence indicates that markets are nonlinear, non-equilibrium, and economic behaviour is collective in nature [34].

Léon Walras was the first one to employ differential calculus in the pursuance to put economics on a scientific foundation, namely, with a physics approach. Walras pioneered the *general equilibrium*

⁶ Gentili Pier Luigi, An Interdisciplinary Investigation into Complex Systems

theory [35]; indeed, economics, before his masterpiece, was not considered a mathematical field. Economists were regarded as primarily philosophers using very little statistical data and mathematics [36]. Mathematical representation plays a pivotal role in empirical analysis and economic theory. Naturally, mathematical representation goes hand in and with social changes and technology [37]; nowadays, computing power has reached a terrific scale; indeed, the before mentioned representation starts becoming more and more accurate.

Physics and chemistry deal with complex systems that are usually artificially simplified, repeated in labs, and controlled. Economics is a non-adaptive system. Social and biological systems are much more complex and often ruled by human irrationality and so they are highly unpredictable. An economic system is an open system that exchanges ‘energy’ with the external environment, often in an unexpected, that is nonlinear and evolutionary way. This dynamical evolution constantly changes relations, which makes mathematical modelling hard [38].

The tremendous interconnection that are at the base of the present world have been shifting this paradigm making economic thinking a broad and wide-ranging bucket [34].

Economic thought history can be split into two eras: classical and neo-classical. The upheaval occurred during the end the nineteenth century. The main difference between this two schools of thought is their view: the first one used to address economic issues in a holistic view (Malthus, Smith, Ricardo, Marx, and Mill) bringing to light the idea of ‘political economy’ whilst the second one in a relative and particular way (Walras, Marshall, and Jevons) soughing to bring into the economic domain the elegance and precision of differential calculus [37]. The latter one brought mathematical rigor but focusing in a too narrow aspect: this field shifted from trying to assess the causes and effects behind the wealth of nation to explaining why an egg costs more than a cup of tea [38]. The increased level of abstraction and the so restricted scope of analysis drifted away from the real-world issues. With the aim to be as objective, analytical, and predictive as possible, economics has hunted to eliminate social, welfare, and political issues form its realm, making it ‘scientific’ and value-free.

Neo-classical economics is primarily the product of Anglo-Saxon culture based on individualism. Criticisms the limitations of neo-classical thinking started to rise from Veblen onwards: [39], [40], [41], [42], and [43]. One of these limitations is that neo-classical thinking adopted, generally speaking, the thermodynamic math, assuming, with the entropy’s maximization, that economy moves toward equilibrium [37]: “Stiglitz’s theoretical work ...tells him that markets cannot be characterised by equilibrium of competitive supply and demand and hence ...that policy responses may have effects that are very different from those they are commonly believed to have”⁷. A traditional economic theory can be structured as follow: the postulation of beliefs and preferences and then the derivation

⁷ Backhouse (2004:266)

of an equilibrium; indeed, individuals maximize their preferences. This is a top-down approach⁸. Furthermore, the *Economic Man* does not stand; indeed, men are complex and path-dependent: “economic, aesthetic, sexual, humanitarian, devotional—organic man, with his complex of habits and thoughts, the expression of each is affected by the habits of life formed under the guidance of each... The base of action – the point of departure – as any step in the process – is the entire organic complex of habits of thought that have been shaped by the past process”⁹. Men do not act in an independent way from each other; they, on the contrary, influence one to another, a typical evidence of this behaviour is during economic bubbles that are self-organizing processes.

In the *History of Economic Thought*¹⁰ it is expressed the general shared approach to treat economy separated from ethics and values but as Allen and Varga¹¹ showed in their work, actions and values are interrelated and self-reinforcing; they are not additive. In conclusion it is impossible to separate ethics and values from economic choices. Economic embodies political, social, environmental issues, and the long term.

Schumpeter (1954) pioneered the discontinuous path that lead the economy from one equilibrium to another, principally due to new technologies. This process of creation and destruction is aligned with the evolutionary thinking. The development of patterns is episodic¹².

In conclusion the global economic system is complex and characterized by actors that are not consistent and rational with a lack of information access. Behavioural economics, brought in eye in 2002 by Daniel Kahneman and Vernon Smith, shows the assumption of rational agent is violated.

The assumptions that the past could be a good predictor of the future, the stability, and simple cause-effect relationship do not stand. Uncertainty, defined by Knight as “unknown unknown”, is considered by the complexity and evolutionary thinking as necessary for development. George Lennox Sharman Shackle stated: “What does not yet exist cannot now be known. The future is imagined by each man for himself and this process of the imagination is a vital part of the process of decision. But it does not make the future known. The absolute and eternal difference between the recorded past and the unformed future, despite its overwhelming significance for the very stuff of human existence, has been often overlooked in our economic theories”. Nassim Nicholas Taleb must be mentioned when dealing with uncertainty, he advocates that in economy the units that survive are those “antifragile” thriving for uncertainty (*Antifragile: Things That Gain from Disorder*).

⁸ C. A. Sims, “Macroeconomics and reality,” *Econometrica*, vol. 48, no. 1, pp. 1–48, 1980.

⁹ Veblen (1898)

¹⁰ Hunt (2002)

¹¹ Allen and Varga (2007)

¹² Stephan Jay Gould (2007)

Furthermore, he asserts, in *Fooled by Randomness* and *The Black Swan*, that future change's characteristic is his impossibility of being predicted.

The application of an evolutionary and complex approach is not new: [44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62]. Furthermore, recently a network approach has started to be considered [63].

It is time to explore the possibilities that treating the economy from a complexity perspective would unfold. In order to do so, Doyne Farmer statement is a quite comprehensive introduction: "The complex systems approach is intermediate between traditional economic theory and econometrics. Traditional economic theory is top-down, modeling decision making from first principles, and then testing against data later. By 'first principles' I mean the requirement that theories have "economic content", i.e. that they derive behaviors from preferences. Econometrics, in contrast, takes a bottom up, data-driven, but fundamentally ad hoc approach. The complex systems approach sits in the middle, taking a bottom up data-driven approach that differs from traditional econometrics by explicitly representing agents and institutions and modeling their interactions, without the requirement that everything be derived from fundamental principles. It has the potential to free behavioralism from the straightjacket of equilibrium modeling, and to bring the computer revolution fully into economics. This path will at times involve abandoning economic content in favor of economic realism." [64].

The ensemble is just the sum of the parts, if the interactions are linear. Complex systems are systems in which an *emergent phenomenon* arises from the interactions of low-level building blocks.

A useful example, for clarification purposes, of an emergent phenomenon are phase transitions, studied and deepened by physicists, in which the individual characteristics of a huge number of particles make the system evolves into a state in which the particles' behaviour is collective. One of the most interesting and studied phenomena is the *superconductivity*, in which particles' quantic characteristics in a conductor make the system, beneath a critical temperature, is subject to a transition phase thanks to which the *resistivity* is null [65].

The Italian physicist Gentili gives a useful clarification: "An example of an emergent property is the power of Complex Systems to self-organize in time and originate periodic processes... We can find periodic processes also in economy. The business cycles are the spontaneous periodic variations of the GDP of a nation, which, after the phase of growth and the peak, always shows the phases of recession, depression, and recovery, in a cyclic manner [Samuelson, P. A.; Nordhaus, W. D. "Economics" 19th ed. McGraw-Hill, New York, 2004]."¹³

¹³ Gentili Pier Luigi, *An Interdisciplinary Investigation into Complex Systems*

Complex systems are studied in order to understand the synthetic proprieties of the interactions, and the goal is to characterize and foresee to the type of phenomena the different interactions lead.

Natural, physical, and social phenomena are dependent on nonlinearity. Interestingly, the pivotal aspect of nonlinear interactions is even more present in social science and biology than physics, but the underlying models are less mathematically intense and developed.

Pareto, with his studies of the distribution of income, described a *power-law* distribution, a critical concept for complex systems and nonequilibrium phenomena.

Adam Smith is regarded as the first one to articulate the abovementioned concept of emergent phenomenon clearly: the invisible hand [66]; “How do the collective wealth and shared benefits of society come about from the striving of selfish individuals?” [64]. Complex systems studies suggest that in order to understand how the emergent phenomenon of economy’s richness emerges we must investigate the composite interactions between the heterogeneous actors.

Adaptive systems models such as immune networks, neural networks, evolutionary game theory, autocatalytic networks, and classifier systems are possible to be mapped within a common framework [67]. A general review is proposed: [68], [69], [70], [71], [72], and [73].

In 1950 John von Neuman anticipated nowadays trends by saying: “Science and technology will shift from a past emphasis on motion, force, and energy to communication, organization, programming, and control”.

Representing the economy towards a complex systems approach is hardly data driven. The actors within the economy can be identified as the key institutions (banks, firms, financial markets, and households) and so their interaction can be built. The starting point is an agent-based model [74] and [75]. Human beings are the key decision makers and this intrinsically complicates the modelling. Simulation models are at the base of this approach.

As above mentioned the complex system approach stand between a top-down and bottom-up views. This approach aims to represent the interplay between agents (individuals) and institutions.

What is a project?

Buchanan and Boddy (1992): “A project is a unique venture with a beginning and an end, conducted by people to meet established goals with parameter of cost, schedule and quality.”

Turner (1993) in *The Handbook of Project-based Management*: each project is unique, meaning that the following characteristics are embodied:

- One-off, unique, not repetitive;
- Time constrains

- a majority of revolutionary improvements rather than evolutionary ones; creating disequilibrium's states
- Goal-oriented, performance based
- Use transient or novel teams of people
- Risky and with small predictivity because a project starts from a limited experience

Another two propaedeutic definitions for the Thesis regard with the meaning of complex projects. Merrow (1988) defines “mega/large” projects the ones that are worth \$1bn or over and “big” worth \$0.5-1bn. The above-mentioned states refer to monetary aspects and can be considered arbitrary; another characteristic that can be taken into account is the manpower involved in the project.

Morris and Hough (1987), definition of large project: demanding in terms of one of the following aspects: of their size, schedule, complexity and/or demand on know-how or assets.

Fraser (1984) put in the spotlight the riskiness as a defining characteristic; normal projects have the characteristics (among others) that “risk assessment can follow well established procedures as all risks are visible”, “there are no catastrophic risks”, “the scale of individual risks is small compared with the size of the parties involved and therefore there is no completion problem”, but that “none of these characteristics is true of the largest projects”: “in general, beyond a certain size, the risks of projects increase exponentially and this can either be appreciated at the beginning or discovered at the end.”

Lock (1994) defined three projects' categories with different characteristics one form the other demanding different PM approaches

1. Manufacturing projects
2. Civil engineering, petrochemical, construction, mining and other projects requiring external organization
3. Managements projects: change of management, IT and R&D projects.

In order to monitor a project, you have to define the various outcome. The first step in any risk management is to delineate the project and its objectives. It is a fundamental tenet of project management that the targets of a project must be defined (Knoepfel 1990), and the project managed towards achieving those objectives (Turner et al. 1988).

Steiner, in his book Top Management Planning, defined a project as “an organisation of people dedicated to a specific purpose or objective. Projects generally involve large, expensive, unique or high-risk undertaking which have to be completed by a certain date, for a certain amount of money, within some expected level of performance”.

Three criteria of success: meeting cost, schedule and performance targets. Barnes (1988) (with particular reference to construction projects) states that “the client’s objectives are always a combination of the objectives for performance of the completed scheme, for achieving this performance within a named cost or budgetary limit and for getting the project into use by a target date”.

In detail, given that the three criteria are not simple one-dimensional measures:

- We want to know whether we have achieved interim milestones and the final project completion date;
- We want to know not only whether the overall project cost is within budget, but also the cash-flow (Turner 1995), often the unit cost of the final product, and frequently (with schemes such as Build-Operate-Transfer) the Life-Cycle-Cost of the final product.
- We will also in general have a whole series of performance targets to meet, some of which will be more important than others, some of which will be more easily achieved than others; some will be essential for client acceptance, some will have liquidated damages attached, and so on.

The Project Management Institute in USA dedicated a whole conference to the definition and measurement of success (1986); De Wit discussed the following:

1. Priorities of project objectives will change between project phases. At its most simplistic, Avots (1984) suggests that schedule is most important early in the project, while during the project cost becomes most important, and after the project, only technical performance is remembered. Certainly a jaundiced view of projects sees a great deal of activity around the project network in the planning stage, then during the project, cost control becomes a much more dominating feature so that no one can spend more than his section of the project’s budget—but long after a project is complete, we remember the physical outcome, not the project performance.
2. There are often a lot of stakeholders with different objectives seeking success. Salapatras and Sawle (1986) define success to have been achieved only when three groups perceive success: the client (from the viewpoint of performance, budget and reputation), the builder [or contractor] (from the viewpoint of profitability, reputation, client and public satisfaction) and the public (from the viewpoint of environment, reliability, cost, and perhaps safety). Baker et al. (1988) require satisfaction from the parent [sponsor], the client, the users/clientele and the project team itself in order to say the project has been “successful”.

Often the projects objectives themselves change during the project as alternative are shown to be more supportive of the overall organisation’s strategic objectives. Willcocks and Margetts (1993)

study a number of such projects, and conclude that we need “to extend the assessment of risk by focussing on broader possible desirable outcomes than those most often encountered, that is, limited assessments of risk against ‘in time, within budget, to acceptable standards’ type criteria.”

We need tools in order to define, decompose and control the scope of work, the timescale of the project, the cost, and the risk involved. The key to all of these techniques is the ability to decompose the project in a structured way, into manageable sub-sections which together encompass all the content of the project.

Managing the scope:

The key to managing the scope is the Work Breakdown Structure (WBS). This breaks down the work of a project into a hierarchical structure of elements, in order to manage and control the scope. This allows good definition of the work, and allows discrete, coherent packages of work to be estimated and then delegated. The second key technique for managing scope is the Milestone Plan (Turner 1993). Indeed, Goal-Directed Project Management treats this as the key technique for project management with (Andersen et al 1987). As well as defining the work packages—in parallel with or even before defining the work packages—the deliverables or milestones must be defined. As well as giving a tool for managing the project at a strategic level, this ensures that the focus of management is not on carrying out the work but on achieving the purposes of the project.

Managing cost:

Cost Breakdown Structure (CBS) is the tool. Where the organizational structure of the project is also broken down using an Organisational Breakdown Structure (OBS) this gives a three-dimensional matrix (WBS x OBS x CBS) known as the cost-control cube.

The cost breakdown, as well as providing an initial estimate of project cost, can also be used to track spend against time and progress, assuming that activities have been scheduled into a base time-plan, a topic we’ll come to shortly. This tracking uses the ideas of Earned Value Analysis in which, at any point in time, a certain amount of each WBS item has been performed, at a certain cost.

Then,

- the total Budget Cost for the work actually Performed so far is denoted by BCWP;
- similarly, the total Actual Cost for the Work actually Performed so far is denoted as ACWP.

The cost variance is then $BCWP - ACWP$, and this is sometimes scaled up to give an expected cost to competition: $Total\ Budget \times ACWP/BCWP$.

When you also compare the work done against the planned schedule,

- the Budgeted Cost for the Work Schedule (to date) is denoted by BCWS.

This then gives us the schedule slippage, as $BCWP-BCWS$.

Managing time:

Critical Path Method (CPM) developed into the Project Evaluation and Review Technique (PERT) in 1958.

Activity-on-the-arrow networks represent activities as arrows between nodes, which denote points in time. Activity-on-the-node networks use nodes to represent activities, while lines between them denote logical relationships.

By analyzing the network, it is possible to extrapolate the minimum duration for the project and hence the critical path. When resources are taken into consideration, there are well-known heuristics for resource smoothing and levelling (Lockyer and Gordon).

Structured approach: modern project management best practice is epitomized by Turner (1993), who describes managing project scope, cost and time as well as organization and quality through three levels merging the techniques together.

2. “ANT” PROJECT

2.1 Overview

According to the project management maturity model the lower the maturity of the project the higher the probability of failure. In this model, however, there is nothing quantitative and the maturity is reached with ”failure cycles” as the system is not proactive but reactive. The purpose of our research is to establish a quantitative method that can determine the evolution of the project. An approach that seems promising is to exploit the actor network theory (ANT): the agents of a socio-technical system (people, institutions, artifacts etc.) live within an interacting and interconnected system that can be represented through a network, therefore they are considered as the nodes of this network which are connected to each other through edges and in turn these edges represent the relationships between the agents of the system. What kind of relationships between the agents of the governance of a project can be mapped as edges in our network? One way to proceed, since we are in the economy field, is to consider any transactional relationship as edges:

- **Capital transactions** (ex. investor or bank and dcf)
- **Standard economic transaction** (ex. buy and sale)
- **Knowledge transaction** (ex. any information exchange, such as the provision of permits etc.)
- **Economic reports** (ex. input and output)

Within a socio-technical network, however, there are also **collaboration relationships** between the agents, but since these more complicated to quantify they can be treated at a later time and can be treated as a second order effects.

Another fundamental characteristic of this type of network is that it is not static but evolves over time ($G = G(t)$), that is, if photographed at each instant of time it will have a different configuration. It is more convenient to consider the time instants discrete: the initial network $G(t = 0)$ represents the system in the initiate phase, that is, in the phase in which the project resolution is being prepared; at the next instant the network $G(1)$ represents the planning phase and so on.

At this point the crucial questions that this research asks are the following:

1. What is the most convenient way to map the project management process in a net- work?

2. Once the first question is resolved, how important is what happens in the transition from one network to another $G(t) \rightarrow G(t + 1)$?

Ideas and motivations for question 1

Mapping a network for project management processes is fundamental both for viewing and for managing information within the process itself. Such socio-technical processes are often characterized by a high degree of complexity and it is no longer conceivable to attempt to manage them through linear approaches.

Realizing a protocol for the construction of such a network in a specific case could become a useful resource for the construction of a standard protocol in the construction of any managerial network. The mapping is not unique: there are several ways to deal with the characteristics explained above. For example, the various relationships between agents can be treated in an equivalent manner and be positioned within the same network, or a network can be built for each type of relationship (one for knowledge transactions, one for economic transactions and so forth), in this case it would be a multi-layer network and would have completely different structural characteristics; furthermore, it is necessary to decide whether to treat the edges so that they are directional or not, and here too the network would assume very different characteristics; still, nodes can be treated differently depending on their function and so on. So as one can notes, the construction of the network is arbitrary but must be done in such a way as to minimize the amount of information dispersed and therefore needs some attention.

Ideas and motivations for question 2

At this point, once the characteristics of the network are known, its structural metrics must be measured: in its simplest form, a network is characterized by various metrics such as density, degree distribution, diameter etc.

Now, the value of these metrics is not a judgment of the goodness of the network but is simply a description of it. What matters is the dynamics of these metrics, in fact, just as the network varies its shape, they will change value at every moment of time. Question 2 can therefore be formalized as follows: given a metric $d = d(t)$, what does its change in time tell us $\Delta d = d(t + 1) - d(t)$? Or also, what does the variation of its variance tell us $\Delta \sigma_d^2 = \sigma_{d(t+1)}^2 - \sigma_{d(t)}^2$?

Normally, change is associated with risk. Therefore, if a certain acceptability threshold S is appropriately defined, then if $\Delta d \gg S$ then the probability that the project fails is very high. A possible development of such an approach could be to decide, given a certain Δd , which is the best strategy to transform the network at the next time in such a way that the risk of failure (or the

loss of an investment) is the minimum possible. For example, given a certain Δd , it is convenient to adopt a strategy of avoiding, transferring, mitigating or accepting?

In addition, the next step could be to make forecasting: that is, to establish what is the probability of failure given the dynamics or correlation of certain metrics, for example one could see if there is a “point of no return” beyond which the failure or the success are unavoidable.

Methodology

Obviously the goal is that the model is built as close as possible to the real world outcome and once the model is made fairly reliable (with an acceptable ratio (model results) / (real results)), one can try to refine it to make it really predictive.

To verify the reliability of the model it is necessary to clean it through “control parameters” in the real world on which to check its reliability (through the classic “statistical tests”). An attempt is therefore made to make the model dataset “controllable” in such a way that its outcomes are independent of exogenous factors or externalities (such as the second-order effects discussed above). In more familiar terms to the scientific world, we try to make the model normalizable to external conditions. The most important control parameters on which to normalize the model are:

- Industry
- project size
- project lifecycle duration
- and others to be added later

The approach of this research is purely empirical in that the risk = change conjecture is taken for granted and experiments are performed to solidify this conjecture. The study is performed on a population of projects at least 50 unities.

As already mentioned, the model consists in the construction of a network, therefore the inputs in this network are necessarily the nodes (the actors within the management process), the edges between these nodes (the transactional relationships mentioned in the previous sections) and, in the event that one wants to build a weighted network, the quantity of these edges (in terms of capital, price, hours etc.). In particular, the basic inputs are:

- capex transaction, quantified by capital transacted
- opex transaction, quantified by price transacted
- labour transaction, quantified by men/hours
- information transaction

These inputs must be given in discrete time stages, in particular we want to build a network for each canonical stage of management process: casing, initiating, planning, execution and project. For

convenience, let's call these stages with the following notation: $T = \{t_1, t_2, t_3, t_4, t_5\}$. Obviously each stage t_i is not totally static, but we make the reasonable approximation that, if d is any metric and δ is the difference of the metric value between two networks belonging to the same stage, then it is valid

$$\delta d \ll \Delta d ,$$

and therefore we only consider the phase transitions between one canonical stage and another.

With this type of data one can build two types of networks and therefore you will have two types of outcomes

- one can build a temporal multilayer network, i.e. a five-layer network in which these layers are precisely the canonical stages (it is not excluded that the network of each of these stages is itself a layered network) and in this case the output of the model is a vector representing the static metrics d analyzed.
- or, 5 independent networks can be built, each representing a different canonical stage and in this case the model outputs will be vectors representing the Δd analyzed.

In both cases the outputs must be normalized according to the control parameters described above. The ultimate goal is to find a causal link between the model's outputs and real outcomes through, for example, a regression (appropriately considering the control parameters). The usefulness of the results is to see, as already said, the probability of failure/success $P(f/s|\{d_1, d_2, \dots, d_n\})$ conditional on the network metric currents. In other words, a curve is sought that describes the expected outcomes as a function of the structure of the network.

Furthermore, this curve could be used to decide which management strategy to adopt once aware of the structure of the underlying socio-technical network and so the future of this research is likely to fall into machine learning.

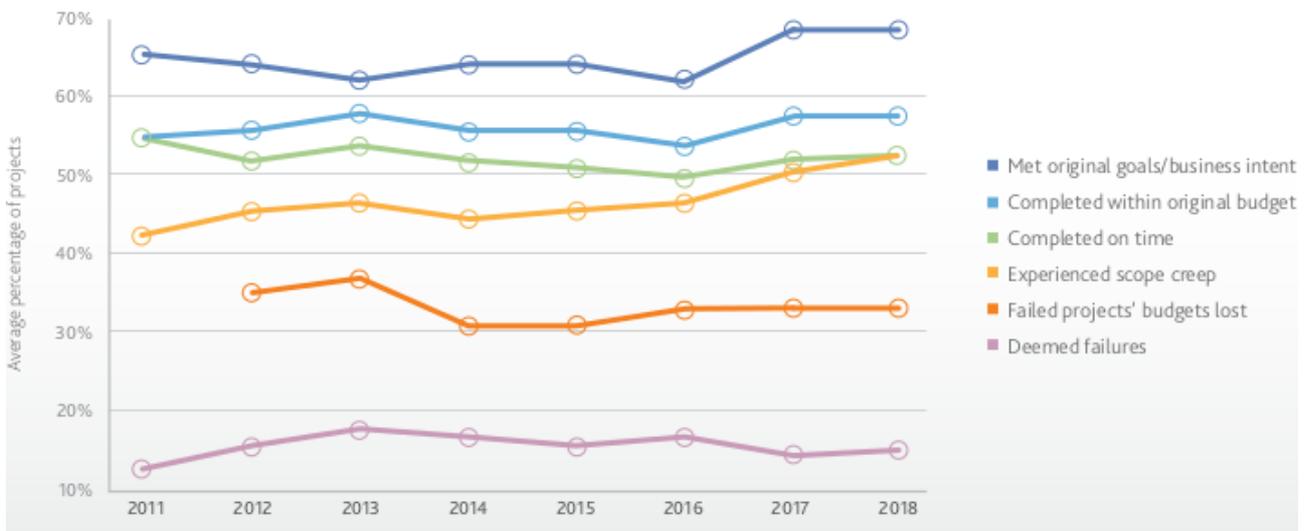
The research aims at modeling project governance through evolving project socio technical network's metrics variances and volatility. Purpose is to gain insight into impacts of structural organizational patterns on projects outcomes.

One of the aims of the Project Management Lab at Politecnico di Torino is to improve effectiveness and efficiency of project based work.

The Project Management Institute, an association of PM practitioners, reckons in a time series the average degree of achievement of these metrics.

On their 2018 pulse of the profession report the series is rendered in figure 6:

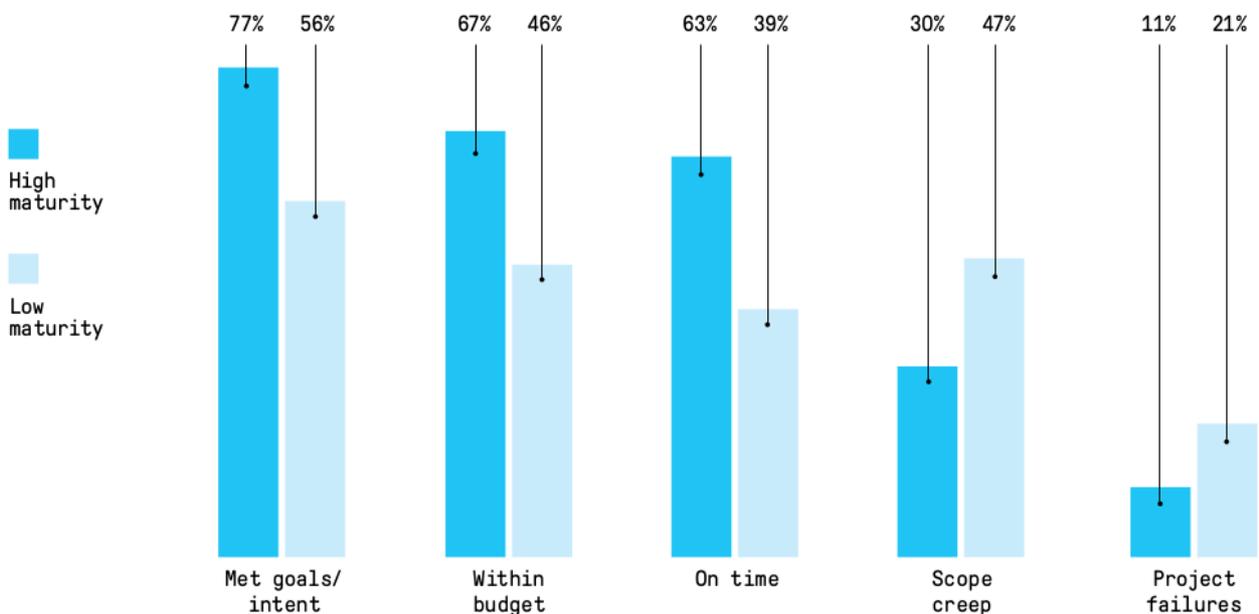
Figure 6: Project Performance Metrics



Where we can see that, on average, nearly 15% of projects are deemed failures, 32% of budget is lost and only 52% of projects are completed on time and, last but not least, only 68% of project deliver the business intent.

The ROI of Maturity

Pulse data show that when it comes to value delivery, organizations that are highly mature in their capabilities outperformed those that are not, across a number of key project metrics:



This chart depicts the high impact of organizational maturity on project outcomes.

Alas, despite the highly graphical impact of the information conveyed, one cannot establish quantitative or statistical inference or correlation of project success metrics with the concept of

Organizational Maturity. Indeed, literature shows that a number of Project Management Maturity Models (PM3s) have been proposed [Backlund2014] and that these model are generally inspired by the Capability Maturity Model (CMM) developed by the Software Engineering Institute of Carnegie-Mellon University between 1986 and 1993. There, the metric of maturity is based and evaluated on the concept that organizations advance through a series of five stages to maturity: initial level, repeatable level, defined level, managed level and optimizing level. These five levels define an ordinal scale for measuring the maturity of an organization's process and for evaluating its process capability. The levels also help an organization prioritize its improvement efforts.

In other words prevailing literature seems to suggest that better project outcomes are related with higher project management competences throughout the organization, both at individual and procedural level. In our opinion we are facing an egg chicken dilemma, to be solved in a reactive way (ie. learning by failing) whereby excellent project are a priori determined by the parent organization excellence in managing projects. This resounds of Seneca the younger stoicism "non est ad astra mollis e terris via" ('there is no easy way from the earth to the stars').

Besides the difficulty in handling highly qualitative metrics, the correlation between the concepts of project success and organizational maturity seems to be established.

A theoretical explanation can be found in the nature of projects being socio-technical systems.

The interdependence of the social and technical systems of organizations was one of the core insights of the sociotechnical systems (STS) tradition associated with the Tavistock School (Trist and Murray, 1990, 1993; Trist et al., 1997). In fact their insight countered the prevailing technological determinism where technology implementations were expected to have direct effects, for example, if a robotic welding system is introduced on an assembly-line, production throughput will increase. They noted that human and organizational outcomes could only be understood when social, psychological, environmental, and technological systems are assessed as a whole. This approach has come to be known as a socio-technical systems (STS) perspective. This perspective assumes that organizations are "made up of people (the social system) using tools, techniques and knowledge (the technical system) to produce goods or services valued by customers (who are part of the organization's external environment). How well the social and technical systems are designed with respect to one another and with respect to the demands of the external environment determines to a large extent how effective the organization will be"[GriffithDougherty2002]

Though this basic insight is now routinely accepted in organization and management theory, in recent years, several authors have questioned the usefulness of the STS tradition as a source of continuing theoretical and practical insight into problems associated with stability and change in complex STS. These authors have argued that the STS tradition — because of an outdated focus on industrial production and industrial relations — has been difficult to apply to the study of topics such as organizational learning and sociotechnical innovation in the emerging organizational forms of the information age.[^Kaghan2001]

Literature suggest that the realm of interaction between social and technical systems should overcome the divide between the "pragmatist/culturalist" and the "rationalist/functionalist".

These two schools differ mostly in their basic assumptions on human intelligence.

Pragmatist/culturalist approaches recognize the situatedness (i.e. cultural specificity) of human intelligence. In this view, human individuals and human groups act pragmatically rather than rationally. That is, people tend to act in a way that they (and others) view as sensible in a particular situation though their behavior may not represent “optimizing” behavior in some universal sense (and may thus be irrational).

Rationalist/functionalist approaches, following Simon (1976), assume that human rationality is bounded and argue that modern economic institutions are structured so as to ensure greater economic efficiency and productivity. Neo-rational choice approaches envision agents with different kinds of cognitive maps or schema operating in and adapting to particular local environments by making “rational choices” within the constraints presented by these local environments.

Getting back to matter, we see that the current school of thought on PM3 relies heavily on the rationalist/functionalist approach: in its essence it stipulates axiomatically that organization accumulate explicit and implicit knowledge on their portfolio of projects and deploy a current project routine that is "rational" and will inevitably lead to project success.

Here we espouse the case for blending traditional project management approaches with a physical modeling technique akin to the pragmatist/culturalist Actor–network theory (ANT) of social systems interaction.

ANT stipulates that everything in the social and natural worlds exists in constantly shifting networks of relationships. It posits that nothing exists outside those relationships. All the factors involved in a social situation are on the same level, and thus there are no external social forces beyond what and how the network participants interact at present. Thus, objects, ideas, processes, and any other relevant factors are seen as just as important in creating social situations as humans. ANT holds that social forces do not exist in themselves, and therefore cannot be used to explain social phenomena. Instead, strictly empirical analysis should be undertaken to "describe" rather than "explain" social activity. Only after this can one introduce the concept of social forces, and only as an abstract theoretical concept, not something which genuinely exists in the world.

This leads us to a wider conceptualization of project initiation and governance. Whereby in traditional standards of pm practice the social system should be rationally and functionally "retrofitted" to the needs of the technical system the project will deliver, here we argue that an holistic project governance should be assumed to reckon, monitor and improve the network of relationship that are instanced by the socio-technical interaction, in a managerial variation of ANT (see [^LeeHassard1999]).

A good metaphor of our endeavor is the representation of a game of chess. The pieces of each player constitute the nodes and the relationships between these pieces (distance from the pieces) constitute the links, the comparison of the evolution of the network metrics of each player with every move proved to be decisive in predicting the winner.

Now, in chess there is a wide body of knowledge based on scripted behavior that abstract from the actual game dynamics (a website[^chess.com] boasts of a database of more than 3000 openings lines). The scripted games are, in our view, a good representation of rationalist/functionalist approach, where by iterations an "optimum" set of play sequences can be defined, whereby, if we can afford the computational power, a pragmatist model of ANT transaction can lead to win. In fact, Big Blue won against Harry Kasparow in 1997.

To improve an overly theoretical discussion let's focus on a reality check and try to construct a network out of a simple building renovation project.

In this project we have a family that using Italian government financial support would renovate their attic to a living / small office home office (SOHO) space.

According to the ISO standard [ISO21500] the project framework can be construed:

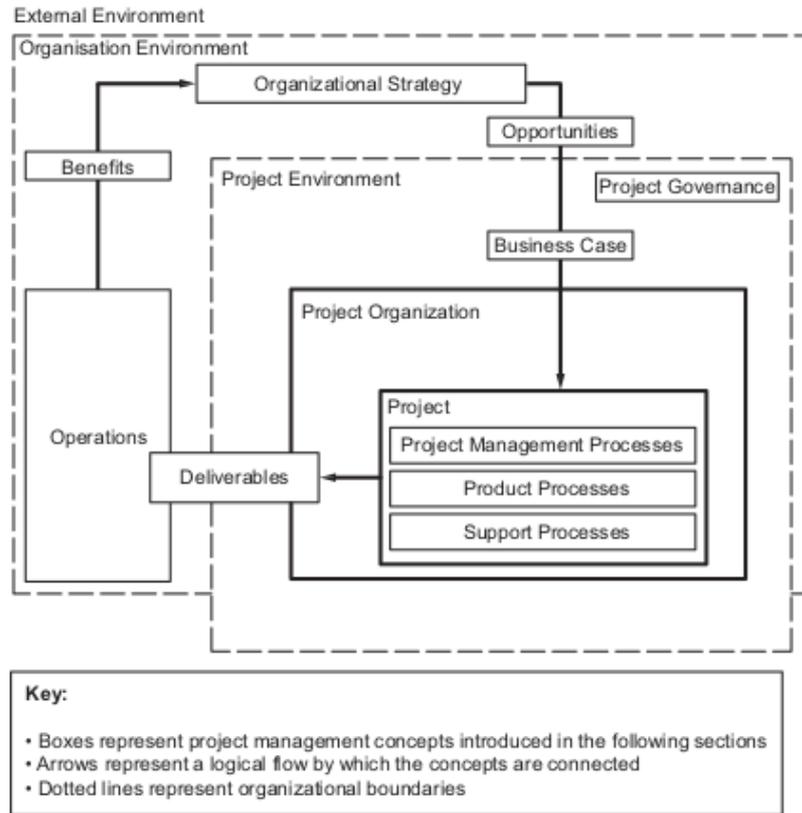


Figure 1 — Overview of project management concepts and their relationships

where:

- **Operations:** daily and social living of a family of 4, the husband smart working, husband and wife running a family business in cosmetics design and selling
- **Benefits:** they seek extra space for a SOHO, a roof terrace / penthouse leisure space, a social functions extra space
- **Strategy and Opportunities:** accrue family estate, diversify wealth portfolio, improve attic space, leverage on post Covid19 tax credits that will be enacted by Italian government, mitigate construction risks by efficient lean construction approach
- **Project governance:** will be discussed later
- **Business case:** here we have some issues. The family has not rationally construed a case comparing alternatives and opportunity costs. They simply adopted their pet project, and a quasi-affective case toward their planned artifact, loaded with totemic features very high in the Maslow pyramid.

- Project and project processes: the family has a very low PM maturity. And this is normal, as building is typically a First-of-a-kind (FOAK) for most families... The result is that the family currently are sustaining a very high market cost trying to accumulate knowledge on the building construction process, mostly by consulting experts, consultants and various tradesmen. Outcome is an informational overflow that seems to increase the distance between their situational as-is and their cherished penthouse.
- Deliverables: throughout this process it has emerged that key deliverables are the building permit, the fitting of a new roofing and interiors, a stair connecting the new space. But there are options for energetic improvement in terms of additional insulation and installation of equipment for renewable energy production.

Project governance is defined[^ISO21500] as:

Governance is the framework by which an organization is directed and controlled. Project governance includes, but is not limited to, those areas of organizational governance that are specifically related to project activities.

Project governance may include subjects such as the following:

- defining the management structure;
- the policies, processes and methodologies to be used;
- limits of authority for decision-making;
- stakeholder responsibilities and accountabilities;
- interactions such as reporting and the escalation of issues or risks.

The responsibility for maintaining the appropriate governance of a project is usually assigned either to the project sponsor or to a project steering committee.

The mostly adopted project management body of knowledge[^PMBOK6] devotes section 1.3 of the standard to linking organizational governance and project governance. There, project governance is defined as:

Project governance is the framework, functions, and processes that guide project management activities in order to create a unique product, service, or result to meet organizational, strategic, and operational goals.

The project governance framework provides the project stakeholders with structure, processes, roles, responsibilities, accountabilities, and decision-making models for managing the project.

The rest of the standard then declines how projects are to be delivered by a scripted process of so called ITTOs, input, tool & techniques, output that, mostly processes written records of information to, in general, advance project knowledge to plan and command the execution of the project itself. But project governance is not the subject of some of the those ITTOs. One is therefore left to bespoke approaches to plan and execute governance tasks.

Project governance of our reality fragment:

In our approach we focus on the last bullet point of ISO standard, ie interactions such as reporting and the escalation of issues or risks, stressing the initial concept, interactions, thus leading us to a direct ANT approach.

We should note that in doing so we somehow subvert the prevailing practice of modeling project management framework. A traditional PMBOK based representation of pm processes rests on the progressive transformation of knowledge, embodied in a standardized set of information fragments:

Table 1-2. Project Management Plan and Project Documents

Project Management Plan	Project Documents	
1. Scope management plan	1. Activity attributes	19. Quality control measurements
2. Requirements management plan	2. Activity list	20. Quality metrics
3. Schedule management plan	3. Assumption log	21. Quality report
4. Cost management plan	4. Basis of estimates	22. Requirements documentation
5. Quality management plan	5. Change log	23. Requirements traceability matrix
6. Resource management plan	6. Cost estimates	24. Resource breakdown structure
7. Communications management plan	7. Cost forecasts	25. Resource calendars
8. Risk management plan	8. Duration estimates	26. Resource requirements
9. Procurement management plan	9. Issue log	27. Risk register
10. Stakeholder engagement plan	10. Lessons learned register	28. Risk report
11. Change management plan	11. Milestone list	29. Schedule data
12. Configuration management plan	12. Physical resource assignments	30. Schedule forecasts
13. Scope baseline	13. Project calendars	31. Stakeholder register
14. Schedule baseline	14. Project communications	32. Team charter
15. Cost baseline	15. Project schedule	33. Test and evaluation documents
16. Performance measurement baseline	16. Project schedule network diagram	
17. Project life cycle description	17. Project scope statement	
18. Development approach	18. Project team assignments	

And this done by processes articulated by macro-process groups and knowledge areas:

Table 1-1. Project Management Process Group and Knowledge Area Mapping

Knowledge Areas	Project Management Process Groups				
	Initiating Process Group	Planning Process Group	Executing Process Group	Monitoring and Controlling Process Group	Closing Process Group
4. Project Integration Management	4.1 Develop Project Charter	4.2 Develop Project Management Plan	4.3 Direct and Manage Project Work 4.4 Manage Project Knowledge	4.5 Monitor and Control Project Work 4.6 Perform Integrated Change Control	4.7 Close Project or Phase
5. Project Scope Management		5.1 Plan Scope Management 5.2 Collect Requirements 5.3 Define Scope 5.4 Create WBS		5.5 Validate Scope 5.6 Control Scope	
6. Project Schedule Management		6.1 Plan Schedule Management 6.2 Define Activities 6.3 Sequence Activities 6.4 Estimate Activity Durations 6.5 Develop Schedule		6.6 Control Schedule	
7. Project Cost Management		7.1 Plan Cost Management 7.2 Estimate Costs 7.3 Determine Budget		7.4 Control Costs	
8. Project Quality Management		8.1 Plan Quality Management	8.2 Manage Quality	8.3 Control Quality	
9. Project Resource Management		9.1 Plan Resource Management 9.2 Estimate Activity Resources	9.3 Acquire Resources 9.4 Develop Team 9.5 Manage Team	9.6 Control Resources	
10. Project Communications Management		10.1 Plan Communications Management	10.2 Manage Communications	10.3 Monitor Communications	
11. Project Risk Management		11.1 Plan Risk Management 11.2 Identify Risks 11.3 Perform Qualitative Risk Analysis 11.4 Perform Quantitative Risk Analysis 11.5 Plan Risk Responses	11.6 Implement Risk Responses	11.7 Monitor Risks	
12. Project Procurement Management		12.1 Plan Procurement Management	12.2 Conduct Procurements	12.3 Control Procurements	
13. Project Stakeholder Management	13.1 Identify Stakeholders	13.2 Plan Stakeholder Engagement	13.3 Manage Stakeholder Engagement	13.4 Monitor Stakeholder Engagement	

And this yields a very complicated data flow diagram shown below:

An AoA network diagram, or Activity on Arrow network diagram, uses circles and arrows. These diagrams are used for Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT), which help optimize the performance of activities in a project.

A simple "Z" configuration exists within an AoN network when a pair of nodes have both a common successor (or predecessor) node and an uncommon successor (or predecessor) node. In terms of matrices this happens when, if M is the adjacency matrix and we fix a node a and a node b where a enters b , we have a configuration Z if this two conditions are satisfied:

$$\begin{aligned} &\exists \text{ a node } i \text{ such that } M_{ib} = 1 \\ &\exists \text{ a node } j \neq i \text{ such that } M_{aj} = 1 \text{ and } M_{ij} = 0 \end{aligned}$$

To remove a Z configuration, just add a "dummy node" that splits the M_{ab} link in two links. Sometimes there can be more than one Z configuration that "overlap". Look at the example in the figure 1.

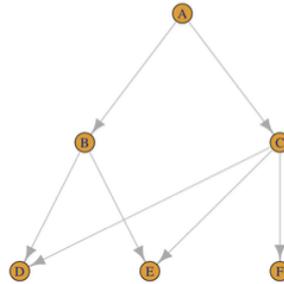


Figure 1: Example of two overlapped Z configurations.

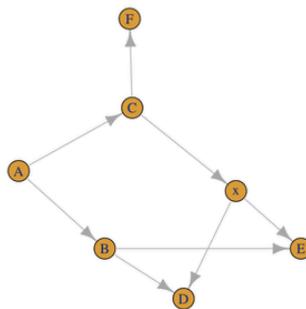


Figure 2: Eliminating the Z configurations with only one dummy node x .

In this case there are 2 overlapping Z configurations: one composed of nodes $\{B, D, C, F\}$, in fact nodes B and C have node D in common and node F not in common; the other is made up of nodes $\{B, E, C, F\}$, in fact nodes B and C have node E in common and node F not in common.

In this example, since the Z s are overlapped, it is not necessary to insert two "dummy nodes" but it is sufficient to insert only one in order to break the "problematic" links. In the example in figure 1, the problematic links are $C \rightarrow D$ and $C \rightarrow E$. The Z s are eliminated by inserting a dummy node x such that $x \rightarrow D$, $x \rightarrow E$ and $C \rightarrow x$, as in figure 2.

A project is the effort dedicated to a specific goal, novelty is an intrinsic character of a project, so all projects are always full of high uncertainty. Project management is a set of rules, methods or techniques, with which to achieve the project goal with a designed duration, cost and quality [126,129].

A project defines a combination of related activities that must be performed in a certain order before the whole activity can be completed. A task in a project is generally seen as a job that requires time and resources to complete.

Project management has evolved as a field with the development of two analytical techniques for planning, scheduling and controlling projects. These are AoN (Activity on Nodes) networks also called Potential Graphs or French Graphs and AoA (Activity on Arcs) networks also called PERT (Program and Evaluation Review Technique), CPM (Critical Path Method) or American Graphs. The design of the AoN network is also easy due to its uniqueness despite the large number of arcs it generates, but the AoA network is more difficult due to the dummy arcs it generates. Project management professionals prefer to work with the AoA network because it is easy to read; each activity is represented by an arc. They give the number of arguments to justify their choice. This is why according to Tsou [126] it is more concise. Furthermore, Hendrickson et al. explains that it is close to the famous Gantt chart. According to Cohen et al. [131,145], the structure of the PERT network is much more suitable for some analytical techniques and optimization formulations [155]. However, the main disadvantage of this method is the existence of dummy arches. Their number is likely to be significantly high, especially if the network size is too large, so the AoA network is not unique .

According to Dechter[126-152] , problems involving time constraints arise in various areas of computing such as planning, program verification and parallel computing. Research into common sense reasoning , natural language understanding and planning has identified new types of temporal reasoning problems specific to AI (Artificial Intelligence) applications. Various formalisms have been proposed to express and reason on temporal knowledge, in particular, Allen's interval algebra , Vilain and Kautz's point algebra [133,134,135], linear inequalities (Malik and Binford [148,149,150], Valdes- Perez[137,138,139]), and Dean and McDermott's time map [141]. Each of these representation schemes is supported by a specialized constraint-oriented reasoning algorithm. At the same time, extensive research has been conducted in recent years on problems involving general constraints (Montanari [129], Mackworth [131] , Gaschnig, Freuder [128] , Nebel and Buckert, Drakengren and Jonsson , Dechter and Pearl , but much of this work has not been extended to problems involving time constraints. Smith and Pyle [have

presented a new heuristic algorithm in the problem of scheduling resource-constrained projects with time windows, Vanhoucke presented a new approach for the treatment of time constraints in project scheduling [126-152].

2.3 The Algorithm

In this work we propose a new approach for modeling time constraints using graphs and a new algorithm for switching from a simple but less used graph (AoN network) containing time constraints to a difficult but widely used graph (AoA network). This transformation was achieved by using line charts and explaining their particular properties.

Let $G = (X, U)$ be a simple or multiple digraph. We construct from G a graph or a line graph indicated with $L(G)$, called a line graph or a line graph of G as follows:

- The nodes of $L(G)$ are in bijective mapping with the nodes of G for reasons of simplicity; we give the same name to the edges of G and to the corresponding nodes of $L(G)$.
- Two nodes u and v of $L(G)$ are connected by an arc of u towards v if and only if the arcs u and v of G are such that the final end of u coincides with the initial end of v , that is $T(u) = I(v)$ []

Suppose the following opposite problem:

Given a direct acyclic graph H , is it the line graph of a direct acyclic graph? In other words, is there a graph G like $L(G)$ isomorphic with H , where $H = L(G)$?

G admits a "Z" configuration if G contains four nodes a, b, c and d as if $(a, c), (b, c)$ and (b, d) were edges of G , then (a, d) is not a arc of G . With a single purpose of simplicity we will give the name of bar of "Z" to the arc (b, c) . The "Z" configuration is displayed when two nodes have common successors and no common successors or by symmetry when two nodes have common predecessors and no common predecessors.

G admits a " Δ " configuration if G contains edges $(a, b), (b, c)$ and (a, c) .

The line charts have been studied but we will present, in this section, the characteristics that interest us and obtained from

1. H is the line graph of a directed acyclic graph if and only if H does not contain any "Z" configuration.
2. H is the line graph of an directed acyclic graph G if and only if the edges of H can be partitioned into a complete bipartite $B_i = (X_i, Y_i), i = 1 \dots, m$, as $X_i \cap X_j = \emptyset$ and $Y_i \cap Y_j = \emptyset \forall i \neq j$. The bipartites B_i of H are therefore in a bijection with the nodes also indicated B_i

which are neither sources nor well, two nodes B_i and B_j of G being connected by an arc from B_i to B_j if and only if the complete bipartite B_i and B_j of H are like $Y_i \cap X_j = \emptyset$.

3. H is the line graph of a non-looped acyclic directed graph if and only if H contains no Z or Δ configuration.
4. H is the line graph of a direct acyclic graph if and only if a pair of nodes with common successors all have their common successors.
5. H is the line graph of a direct acyclic graph if and only if any pair of nodes with common predecessors have all their common predecessors.

Hence, H is not the line graph of any direct acyclic graph if it is only if there is a pair of nodes that have common successors and no common successors or common predecessors and no common predecessors.

So, we want to know how to transform H to get a new graph which is a line graph of a graph.

Given the ease of use of the AoA dag, we must concentrate our efforts on studying the possibility of transforming the AoN dag (a significant number of arcs) into AoA dag (a small number of arcs). So, we want to know how to transform the H graph (which is an AoN dag) to get a new graph which is the line graph (AoA dag). The difficulty that arises is to know if H contains Z configurations or not? If it does not contain Z , it is a line chart and the transformation is immediate. But if it contains Z , we must eliminate the bar from each Z while preserving the succession constraints. Let's analyze each case separately:

We build the AoA dag from AoN dag, assuming it is a line graph. According to the terms of the discussed above, we proceed as follows:

We partition the arcs of the AoN dag into a complete bipartite $B_i = (X_i, Y_i)$.

In the dag AoA we want to construct, each B_i is represented by a node still annotated B_i and will be the center of the star.

The construction of AoA dag is however more complex in general where AoN dag is not a line graph: it does not allow a partition of the complete arcs in bipartite. It is in this case that it must be modified to transform it into an associated direct acyclic graph while preserving the constraints of prior art.

Suppose that activities a_1, \dots , are preceded by activities b_1, \dots, b_n . In the AoN dag, these anteriority constraints are represented by a complete bipartite. In AoA dag, they are represented by a star.

Let's go back to the dummy bow problem in AoA dag. For example, if there are 4 activities, they have a, b, c and d with the following anteriority constraints: c is preceded by a and b , but d is preceded only by b .

In AoN dag, there are no problems representing these activities. But, if there are Z subgraphs in AoN dag (which is regarded as the H line graph), we are obliged to eliminate all "Z" configuration. We therefore introduce, in AoN dag, a fictitious arc f in each "Z". The introduction of fictitious arcs aims to eliminate all the configurations "Z" from the dag AoN, the constraints remain unchanged. We must remember that dummy arcs are not needed in AoN dag but are only introduced to build AoA dag.

Kelley [136] notes that it is advantageous to reduce the length of the computations to build an AoA dag with the minimum number of nodes and dummy activities.

We therefore pose the problem of looking for "Z" in the dag AoN, ie nodes that have common successors and no common predecessors or nodes that have common predecessors and no common successors.

A first elimination technique of Z in Gv consists in replacing the bare (b, c) of each "Z" with two arcs (b, f) and (f, c), according to fig. 8, is the simplest but the worst, the number of "Z" can be arbitrarily large.

Another more effective technique than the previous one eliminates more "Z" at the same time by grouping bars having initial or final ends in the same bipartite set.

Once we have eliminated the "Zs" and added the necessary dummy nodes, we have obtained an AoN network which represents a line graph that can be transformed into a graph. This graph will be our AoA network which will have as links all the AoN tasks plus the dummy nodes.

Call M_z the matrix $(N+t) \times (N+t)$ obtained by eliminating the Zs and adding the dummy nodes. This matrix, having no more Zs, represents a network made up of many complete bipartite sub-networks. Note that a complete bipartite sub-network exists if the following condition holds in the adjacency matrix M_z :

Consider the following complete bipartite sub-network as an example:

$$b_i = \{X_i[v_1, v_2]; Y_i[v_3, v_4, v_5]\}$$

this notation means that nodes $\{v_1, v_2\}$ enter nodes $\{v_3, v_4, v_5\}$. At this point the construction of the AoA network is easily achieved with the following steps:

1. Identify all the complete bipartite sub-networks b_i in the M_z matrix. Each b_i represents a node in the AoN network.
2. In the AoN network, join two nodes b_i and b_j only if $Y_i \cap X_j \neq \emptyset$.

The new network is the AoN network we were looking for. To return to the AoN just build the line graph starting from the AoA graph.

2.4 The Code

Python code for parsing data from .XML file, create an AON graph, cleaning Z_appterns and generate the related AOA graph

XML Parsing

In [53]:

```
import xml.etree.ElementTree as ET
import pandas as pd
import numpy as np
tree = ET.parse('Gantt_Project Spain_Wind.xml')
root = tree.getroot()
```

Build a relationship list with predecessor and successor from xml file

In [34]:

```
relationship = []
for element in root.iter("{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}Relationship"):
    relationship.append(element)

#Take predecessor and successor for each relationship
list_predecessor = []
list_successor = []
list_type = []

for i in range(len(relationship)):
    PREDECESSOR = relationship[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}PredecessorActivityObjectId')
    SUCCESSOR = relationship[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}SuccessorActivityObjectId')
    TYPE = relationship[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}Type')

    list_predecessor.append(PREDECESSOR.text)
    list_successor.append(SUCCESSOR.text)
    list_type.append(TYPE.text)
```

Create a pandas DataFrame with relationships

In [35]:

```
df_relation = pd.DataFrame(list_predecessor, columns=['Predecessors'])
df_relation['Successor']=(list_successor)
df_relation['Type']=list_type
```

Build an activities dataframe from xml with predecessor and successor for each activity

In [36]:

```
list_activity = []
for element in root.iter("{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}Activity"):
    list_activity.append(element)
```

#Collect predecessor and successor for each activity

```
list_ID = []
```

```
list_description = []
```

```
list_start = []
```

```
list_finish = []
```

```
list_priority = []
```

```
list_type = []
```

```
list_planned_start = []
```

```
list_planned_finish = []
```

```
for i in range(len(list_activity)):
```

```
    ID = list_activity[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}ObjectId')
```

```
    DESCRIPTION = list_activity[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}Name')
```

```
    START = list_activity[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}ActualStartDate')
```

```
    FINISH = list_activity[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}ActualFinishDate')
```

```
    PRIORITY = list_activity[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}LevelingPriority')
```

```
    TYPE = list_activity[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}Type')
```

```
    PL_START = list_activity[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}PlannedStartDate')
```

```
    PL_FINISH = list_activity[i].find('{http://xmlns.oracle.com/Primavera/P6/V8.3/API/BusinessObjects}PlannedFinishDate')
```

```
    if ID != None:
```

```
        list_ID.append(ID.text)
```

```
    else:
```

```
        list_ID.append("")
```

```
    if DESCRIPTION != None:
```

```
        list_description.append(DESCRIPTION.text)
```

```
    else:
```

```
        list_description.append("")
```

```
    if START != None:
```

```
        list_start.append(START.text)
```

```
    else:
```

```
        list_start.append("")
```

```
    if FINISH != None:
```

```
        list_finish.append(FINISH.text)
```

```
    else:
```

```
        list_finish.append("")
```

```
    if PRIORITY != None:
```

```
        list_priority.append(PRIORITY.text)
```

```
    else:
```

```
        list_priority.append("")
```

```

if TYPE != None:
    list_type.append(TYPE.text)
else:
    list_type.append("")

if PL_START != None:
    list_planned_start.append(PL_START.text)
else:
    list_planned_start.append("")

if PL_FINISH != None:
    list_planned_finish.append(PL_FINISH.text)
else:
    list_planned_finish.append("")

```

```

df_activity = pd.DataFrame(list_ID, columns=['ID'])
df_activity['DESCRIPTION'] = list_description
df_activity['Start'] = list_start
df_activity['Finish'] = list_finish
df_activity['Planned_Start'] = list_planned_start
df_activity['Planned_Finish'] = list_planned_finish
df_activity['Priority'] = list_priority
df_activity['Type'] = list_type

```

Preprocessing data in order to collect some information about activity and relationship

Parsing start and finish date from string to datetime format

In [37]:

```

from dateutil.parser import parse

list_start = list(df_activity.Start)
list_finish = list(df_activity.Finish)
list_planned_start = list(df_activity.Planned_Start)
list_planned_finish = list(df_activity.Planned_Finish)

list_start_trasf = []
list_finish_trasf = []
list_pl_start_trasf = []
list_pl_finish_trasf = []

for i in range(len(df_activity)):
    a = str(list_start[i])
    af = str(list_finish[i])
    ap = str(list_planned_start[i])
    afp = str(list_planned_finish[i])

    try:
        b = parse(a)

```

```

except:
    b = a
try:
    bf = parse(af)
except:
    bf = af
try:
    bp = parse(ap)
except:
    bp = ap
try:
    bfp = parse(afp)
except:
    bfp = afp

```

```

list_start_trasf.append(b)
list_finish_trasf.append(bf)
list_pl_start_trasf.append(bp)
list_pl_finish_trasf.append(bfp)

```

```

df_activity['Start'] = list_start_trasf
df_activity['Finish'] = list_finish_trasf
df_activity['Planned_Start'] = list_pl_start_trasf
df_activity['Planned_Finish'] = list_pl_finish_trasf

```

```

# Calculate duration of any activity
df_activity['Duration']=(df_activity.Finish-df_activity.Start).dt.days
df_activity.Duration = df_activity.Duration.fillna(0)
df_activity = df_activity.set_index('ID')

```

Save the dataframe for further utilizations

In [38]:

```
df_activity.to_excel('task_AON.xlsx')
```

Add a Start and End node to relationship dataframes

In [39]:

```

# In these steps we look for tasks that doesnt have predecessors and link them to Start node
# Then we look for task that does not have any successors and link them to END node

```

```

list_predecessors = list(df_relation['Predecessors'])
list_successors = list(df_relation['Successor'])

```

```
starting_point = max(df_relation.index)+1
```

```

for i in df_activity.index:
    list_A = []
    if i not in list_predecessors:
        list_A = [i, "End", "No type"]
    if i not in list_successors:

```

```

list_A = ["Start",i, "No Type"]

if len(list_A)>0:
    list_A=pd.DataFrame(list_A, columns= [starting_point], index = ['Predecessors', 'Successor', 'Type'])
    list_A=list_A.transpose()
#     print(lista)
    df_relation = df_relation.append(list_A)

    starting_point = starting_point + 1

```

Create the AON Network from parsed and processed data and transform in to AOA graph

Build AON network and its adjacency matrix

In [45]:

```

import networkx as nx
G = nx.DiGraph()
for i in range(len(df_relation)):
    G.add_edge(df_relation['Predecessors'][i], df_relation['Successor'][i])
# nx.draw(G, with_labels=True)
Matrix_df_aon = nx.to_pandas_adjacency(G)
Matrix_df_aon.to_excel('Staring_AON_Matrix.xlsx')

```

Launch the algorithm for Z patterns removing

In [46]:

```

Matrix = np.array(Matrix_df_aon)

Matrix=np.array(Matrix)
N = np.shape(Matrix)[0]
dummy = 0
list_dummy=[]
lista_Z_ = []
lista_Z_ALL = []
for i in range(N-1): #

    c = 0
    step_ = 0
    internal_z_list=[]
    internal_z_list_ALL=[]
    while c == 0:
        part_equals, superimposed, equals, equals_to_superimposed = np.array([]),np.array([]),np.array([]),np.array([])
        step_ = step_ +1

    for s in range((i+1), (N+dummy)):
#         print(i,s)

        #collect index of superimposed and superimposed number

```

```

if len(np.where((Matrix[i,:] + Matrix[s,:]) == 1)[0]) > 0 and len(np.where((Matrix[i,:] + Matrix[s,:]) == 2)[0])>0
:
#
    part_equals = np.append(part_equals,s) #Vector with superimposed
    superimposed=np.append(superimposed,len(np.where((Matrix[i,:] + Matrix[s,:]) == 2)[0])) #VETTORE CON
IL N DI superimposed

if len(part_equals)>0:
    #collect index of "equals" rows
    for s1 in range((i), (N+dummy)):

        if len(np.where((Matrix[i,:] + Matrix[s1,:]) == 1)[0]) == 0 and len(np.where((Matrix[i,:] + Matrix[s1,:]) == 2)
[0])>0:
            equals = np.append(equals,s1) #equals Index

            equal_to_major_part_equal_n = int(part_equals[int(np.where(superimposed == max(superimposed))[0][0])])
            equal_to_major_part_equal = Matrix[equal_to_major_part_equal_n,:]

            for s2 in part_equals:
                if len(np.where((equal_to_major_part_equal + Matrix[int(s2),:]) == 1)[0]) == 0 and len(np.where((equal_to_
major_part_equal + Matrix[int(s2),:]) == 2)[0])>0:
                    equals_to_superimposed = np.append(equals_to_superimposed,s2)

            dim_equal = np.sum(Matrix[i,:])
            dim_part_equal = np.sum(equal_to_major_part_equal)

            col_superimposed = np.where((Matrix[i,:]+equal_to_major_part_equal)==2)[0]

            dummy_name ="Dummy"+str(dummy)
            list_dummy.append(dummy_name)

            new_row = np.zeros((1,np.shape(Matrix)[1]), dtype=int, order='C')
            Matrix = np.append(Matrix, new_row, axis=0)
            new_column = np.zeros((np.shape(Matrix)[0],1), dtype=int, order='C')
            Matrix = np.append(Matrix, new_column, axis=1)

if dim_equal >= dim_part_equal:
    row_to_reset = equals
else:
    row_to_reset = equals_to_superimposed

for s3 in row_to_reset:
    for s4 in col_superimposed:

        Matrix[int(s3),int(s4)]=0

```

```
Matrix[(N+dummy),int(s4)]=1
Matrix[int(s3),(N+dummy)]=1
```

```
internal_z_list.append(s3)
internal_z_list.append(s4)
```

```
lista_z_unique=[]
for Z_node in internal_z_list:
    if Z_node not in lista_z_unique:
        lista_z_unique.append(Z_node)
```

```
internal_z_list = lista_z_unique
```

```
lista_Z_append(internal_z_list)
```

```
dummy = dummy+1
```

```
else: c = 1
```

```
print("Matrix cleaned by Z Patterns")
print(Matrix)
print("")
print("The algorithm generates:", len(list_dummy), "Dummy nodes")
```

Matrix cleaned by Z Patterns

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

The algorithm generates: 225 Dummy nodes

Create a DataFrame with the adjacency matrix with names of tasks for further utilizations

In [49]:

```
list_nodes = list(Matrix_df_aon.index)+list_dummy
Matrix_df_clean = pd.DataFrame(Matrix, index=list_nodes, columns=list_nodes)
```

In [50]:

```
Matrix_df_clean
```

Out[50]:

	4 9 3	6	4 9 4	7	4 8	5 7	7 8	2 1 5	8 6	9 1	.	Du mm y21 5	Du mm y21 6	Du mm y21 7	Du mm y21 8	Du mm y21 9	Du mm y22 0	Du mm y22 1	Du mm y22 2	Du mm y22 3	Du mm y22 4	
493	0	0	0	0	0	0	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											
6	0	0	0	0	0	0	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											
494	0	0	0	1	0	0	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											
7	0	0	0	0	0	0	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											
48	0	0	0	0	0	1	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											
...
											
Du mm y22 0	0	0	0	0	0	0	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											
Du mm y22 1	0	0	0	0	0	0	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											
Du mm y22 2	0	0	0	0	0	0	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											
Du mm y22 3	0	0	0	0	0	0	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											
Du mm	0	0	0	0	0	0	0	0	0	0	.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
											
	0	0	0	0	0	0	0	0	0	0	.											

4	4	7	4	5	7	2	8	9	.	Du	Du	Du	Du	Du	Du	Du	Du	Du	Du	
9	6	9	7	8	7	8	1	6	1	.	mm									
3	4	4	8	7	8	5	5	6	1	.	y21	y21	y21	y21	y21	y22	y22	y22	y22	
											5	6	7	8	9	0	1	2	3	4

```
y22
4
```

549 rows × 549 columns

In [51]:

```
Matrix_df_clean.to_excel('Matrix_AON_Wind_clean.xlsx')
G2 = nx.from_pandas_adjacency(Matrix_df_clean, create_using=nx.DiGraph)
nx.write_gml(G2, 'windAON_clean.gml', stringizer=str)
```

Create AOA network from "cleaned" AON

In [52]:

```
Matrix2 = Matrix
```

```
# grouping all equals rows
# take index of grouped rows and columns == 1
```

```
import numpy as np
```

```
bipartite_list = []
done = []
```

```
for t in range(len(Matrix2)):
```

```
    x = Matrix2.index[t]
    bipartite_list_ = []
    X_bipartite_list = []
    Y_bipartite_list = []
```

```
    for tt in range(t, len(Matrix2)):
```

```
        x1 = Matrix2.index[tt]
```

```
        first_ = (Matrix2.loc[Matrix2.index == x])
        second_ = (Matrix2.loc[Matrix2.index == x1])
        cond = np.all(first_.values == second_.values)
```

```
        if cond == True:
            if x1 in done:
                X_bipartite_list = X_bipartite_list
            else:
                X_bipartite_list.append(x1)
```

```
done.append(x1)
```

```
if len(X_bipartite_list) > 0:  
    bipartite_list_.append(X_bipartite_list)  
    Y_bipartite_list = []  
    row_ = Matrix.loc[Matrix.index==X_bipartite_list[0]]  
    for i in row_.columns:  
        n = row_.index[0]  
        if row_[i][n] == 1:  
            Y_bipartite_list.append(i)  
    bipartite_list_.append(Y_bipartite_list)  
    bipartite_list.append(bipartite_list_)
```

```
bipartites = pd.DataFrame(bipartite_list, columns=['X','Y'])  
bipartites
```

```
# Create AOA links. Link is created where a bipartite Y is == to another bipartite Y
```

```
links=[]  
for i in bipartites.index:  
    for i2 in bipartites['Y'][i]:  
        for i3 in bipartites.index:  
            for i4 in bipartites['X'][i3]:  
                if i2 == i4:  
                    link=[i,i3,i4]  
                    links.append(link)  
links = pd.DataFrame(links, columns=['from','to','task'])  
links['value']=1
```

```
# create network of bipartites, final AOA NETWORK
```

```
import networkx as nx  
G_aoa = nx.DiGraph()  
for i in links.index:  
    G_aoa.add_edge(links['from'][i],links['to'][i], task=links['task'][i])
```

```
## Plotting (Under comment)
```

```
# pos = nx.spring_layout(G_aon)
```

```
# import matplotlib.pyplot as plt
```

```
# plt.figure(figsize=(20,20))
```

```
# nx.draw_networkx_nodes(G_aon, pos, node_size=200, with_labels=True)
```

```
# nx.draw_networkx_edges(G_aon, pos)
```

```
# nx.draw_networkx_edge_labels(G_aon,pos)
```

```
#plt.show()
```

Save graph and adj matrix for further utilization

In []:

```
nx.write_gml(G_aoa, 'windAOA.gml', stringizer=str)
```

```
Matrix_AOA = nx.to_pandas_adjacency(G_aon)
```

```
Matrix_AOA.to_excel('Matrix_AOA_Wind.xlsx')
```

```
Matrix_AOA
```


3. APPLICATION TO STOCK MARKET

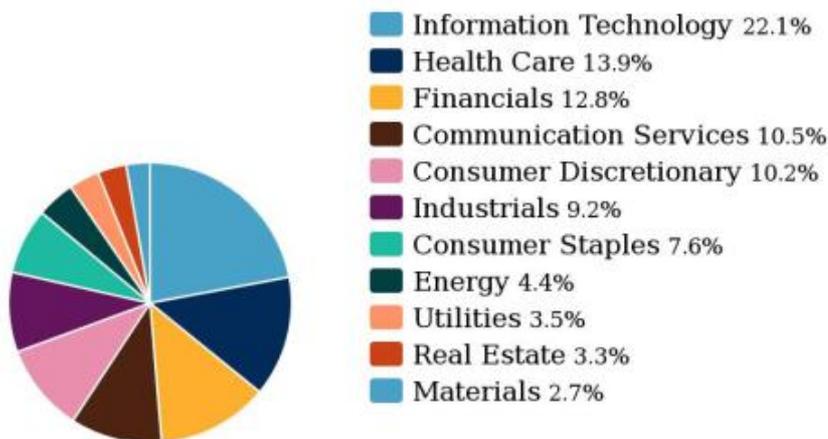
3.1 Database

For the scope of this research, it has been built a database composed of monthly stock prices. The stocks that have been chosen are those present in the Standard & Poor 500 index. It is the index that encompasses 500 publicly listed large companies in the United States that are representative of the US's economy and so its industries. A committee is in charge of selecting the 500 companies, the criteria are the following¹⁴:

- Market cap \geq US\$ 8,2 billion
- Annual dollar value traded to float-adjusted market capitalization is greater than 1.0
- Minimum monthly trading volume of 250,000 shares in each of the six months leading up to the evaluation date

S&P 500 can be considered as a benchmark give its broad scope and wide market breadth of the large-cap companies included in the index.

In terms of data reliability, availability and quality is the most successful. 500 stocks are a significant number for the purpose of this research. Being these large-caps, it is possible to grasp a world economy overview.



Based on GICS® sectors

The weightings for each sector of the index are rounded to the nearest tenth of a percent; therefore, the aggregate weights for the index may not equal 100%.

As of Aug 30, 2019

¹⁴ <https://us.spindices.com/documents/methodologies/methodology-sp-us-indices.pdf>

The period analyzed lasts more than 20 years; specifically, from 1998 to 19/06/2020.

The reasons why it has been chosen such a period are the following:

- i) There are two significant recessions (Dot.com bubble and Subprime mortgage crisis) that have been mainly driven by bubbles, an endogenous factor, and impacted firstly the US economy and then assumed a worldwide seize;
- ii) An exogenous factor (Covid-19) impacted significantly on the world economy;
- iii) European sovereign debt crisis, that is an outsider of the US economy;
- iv) Enough data to process a comprehensive analysis.

The stock prices have been downloaded on a daily basis from Yahoo Finance and then gathered together monthly. It has been used a Python function for importing the daily prices and resampled the monthly.

One assumption has been made regarding the composition of the above-mentioned index: it is assumed the composition of the last day of the period analyzed less those stocks were not present in 1998 and during the analyzed period have been unlisted (Appendix).

In the pursuance of building the network, the windows inspected have been rolling on a semiannual basis; i.e., first window from January 1998 to June 1998, second window from February 1998 to July 1998.

3.2 Network building, link and correlation

In order to build the links between stocks there are several methods as it has been already depicted in

2.3 State of the art: network science applied to financial markets.

It follows the literature works on building edges [107-112]:

- Zero-lag correlation: the change in price of stocks during a defined period Δt is calculated =>

$$G_i(t) = \ln S_i(t) - \ln S_i(t - \Delta t),$$

$S(t)$ is the price of the stock i at time t . Then a normalized return is defined =>

$$g_i(t) = \frac{G_i(t) - \langle G_i \rangle}{\sigma_i},$$

where $\sigma_i = \sqrt{\langle G_i^2 \rangle - \langle G_i \rangle^2}$ is G_i 's standard deviation and $\langle \rangle$ indicates a time average over the interested time frame. It follows the calculation of the equal-time cross-correlation matrix C =>

$$C_{ij} = \langle g_i(t)g_j(t) \rangle$$

C_{ij} is included between $[-1,1]$. Where -1 means perfect anti-correlation, 1 corresponds to perfect correlation and 0 means uncorrelated stocks. The matrix C is symmetric $C_{ij} = C_{ji}$. For each pair of stocks this method has been applied [95].

- Detrended covariance: it has been analyzed both the daily closing values of the Dow Jones and the Nasdaq financial indices together with their corresponding volumes. Absolute values of differences of logarithms for consequently days' time series for both the indices has been analyzed. Their integrated signals: $I(n) = \sum_{i=1}^n (|y_i| - |\bar{y}_i|)$ [98 and 113].
- Time-lag correlations of prices changes over a certain period: by analyzing cross-correlations between price changes of the members of the S&P 500 index, it was found that 98% of the eigenvalue spectrum of the correlation matrix C follows the Gaussian orthogonal ensemble of a perfectly random matrix [114-119]. To quantify long-range collective movements in correlated data sets, it has been applied time-lag random matrix theory (TLRMT) [99].
- Price-volume cross correlation: ("it takes volumes to move stock prices.") Through this approach there have been considered both the difference in logarithm between two consecutive prices and the difference in logarithm between two consecutive values of trading volume [97].
- Cross correlation between x_i and x_j with no time shift [100 and 120]. In building the network a criterion has been applied in order to establish a connection between a pair of nodes. A threshold $\rho=0,9$ has been chosen.

In this research links between stocks have been built upon the correlation of monthly log-returns:

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

Where x and y are two log return, σ_{xy} is the covariance, σ_x and σ_y are the variances of the two stocks x and y .

At each t , the correlation matrix of the log-returns of the stocks present in the basket understudy is calculated.

Therefore, from the correlation matrix, the network adjacency matrix has been built, i.e. a matrix whose elements x_{ij} are null if there is no link between the nodes otherwise the value is equal to one. In our case, it has been decided to build a link between two stocks only if the correlation between those stocks is above a defined threshold S . The elements within the adjacency matrix are equal to one if $C_{ij} > S$ (where C is the correlation matrix).

The above-mentioned adjacency matrix A has its elements A_{ij} as it follows:

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between vertices } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases}$$

Elements in the diagonal are equal to zero, there are no self-edges; the networks are undirected and not weighted, it means that the matrix is binary.

3.3 Stock market's structural measures

Knowing the network's structure, it is possible to compute from it several useful measures and quantities that grasp particular characteristic of network topology. Most of the common measures are born in the social network analysis.

It follows the measures that have been computed in this research.

3.3.1 Modularity

Modularity is the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random. "The value of the modularity for unweighted and undirected graphs lies in the range $[-1/2, 1]$. It is positive if the number of edges within groups exceeds the number expected on the basis of chance. For a given division of the network's vertices into some modules, modularity reflects the concentration of edges within modules compared with random distribution of links between all nodes regardless of modules".

There are different methods for calculating modularity. In the most common version of the concept, the randomization of the edges is done so as to preserve the degree of each vertex. Consider a graph with n nodes and m links (edges) such that the graph can be partitioned into two communities using a membership variable s . If a node v belongs to community 1, $s_v=1$, or if v belongs to community 2, $s_v=-1$. Let the adjacency matrix for the network be represented by A , where $A_{vw}=0$ means there's no edge (no interaction) between nodes v and w and $A_{vw}=1$ means there is an edge between the two. Also, for simplicity we consider an undirected network. Thus $A_{vw}=A_{wv}$ (It is important to note that multiple edges may exist between two nodes, but here we assess the simplest case).

"Modularity Q is then defined as the fraction of edges that fall within group 1 or 2, minus the expected number of edges within groups 1 and 2 for a random graph with the same node degree distribution as the given network.

The expected number of edges shall be computed using the concept of a configuration model. The configuration model is a randomized realization of a particular network. Given a network with n nodes, where each node v has a node degree k_v , the configuration model cuts each edge into two halves, and then each half edge, called a stub, is rewired randomly with any other stub in the network (except itself), even allowing self-loops (which occur when a stub is rewired to another stub from the

same node) and multiple-edges between the same two nodes. Thus, even though the node degree distribution of the graph remains intact, the configuration model results in a completely random network.

Q is called the modularity and is a measure of the extent to which like is connected to like in a network. It is strictly less than 1, takes positive values if there are more edges between vertices of the same type than we would expect by chance, and negative ones if there are less.

The modularity is always less than 1 but in general it does not achieve the value $Q=1$ even for perfectly mixed network, one in which every vertex is connected only to others of the same type [2, 121, 122].

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j).$$

Suppose we have a network in which the vertices are classified according to some characteristic that has a finite set of possible values. The values are merely enumerative—they don't fall in any particular order. For instance, the vertices could represent people and be classified according to nationality, race, or gender. Or they could be web pages classified by what language they are written in, or biological species classified by habitat, or any of many other possibilities.

The network is assortative if a significant fraction of the edges in the network run between vertices of the same type, and a simple way to quantify assortativity would be to measure that fraction. However, this is not a very good measure because, for instance, it is 1 if all vertices belong to the same single type. This is a trivial sort of assortativity: all friends of a human being, for example, are also human beings, but this is not really an interesting statement. What we would like instead is a measure that is large in non-trivial cases but small in trivial ones.

A good measure turns out to be the following. We find the fraction of edges that run between vertices of the same type, and then we subtract from that figure the fraction of such edges we would expect to find if edges were positioned at random without regard for vertex type. For the trivial case in which all vertices are of a single type, for instance, 100% of edges run between vertices of the same type, but this is also the expected figure, since there is nowhere else for the edges to fall. The difference of the two numbers is then zero, telling us that there is no non-trivial assortativity in this case. Only when the fraction of edges between vertices of the same type is significantly greater than we would expect on the basis of chance will our measure give a positive score" [2].

3.3.2 Density

“The degree of a vertex in a graph is the number of edges connected to it. We will denote the degree of vertex i by k_i . For an undirected graph of n vertices, the degree can be written in terms of the

adjacency matrix as:

$$k_i = \sum_{j=1}^n A_{ij}.$$

Every edge in an undirected graph has two ends and if there are m edges in total then there are $2m$ ends of edges. But the number of ends of edges is also equal to the sum of the degrees of all the vertices, so:

$$2m = \sum_{i=1}^n k_i.$$

The mean degree c of a vertex in an undirected graph is:

$$c = \frac{1}{n} \sum_{i=1}^n k_i.$$

The maximum possible number of edges in a simple graph (i.e., one with no multiedges or self-edges) is:

$$\binom{n}{2} = \frac{1}{2}n(n-1).$$

The connectance or density ρ of a graph is the fraction of these edges that are actually present:

$$\rho = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)} = \frac{c}{n-1}.$$

The density lies strictly in the range $0 \leq \rho \leq 1$.

A network for which the density ρ tends to a constant as $n \rightarrow \infty$ is said to be dense. In such a network the fraction of non-zero elements in the adjacency matrix remains constant as the network becomes large. A network in which $\rho \rightarrow 0$ as $n \rightarrow \infty$ is said to be sparse, and the fraction of non-zero elements in the adjacency matrix also tends to zero. In particular, a network is sparse if c tends to a constant as n becomes large”[2].

3.3.3 Average clustering

The clustering coefficient measures how connected a vertex’s neighbors are to one another. More specifically, it is calculated as: (the number of edges connecting a vertex’s neighbors)/(the total number of possible edges between the vertex’s neighbors).

Clustering or transitivity behavior of network connection, which describes the propensity for two neighbors of the same vertex also to be neighbors of each other, is common in real-world systems. The clustering phenomena could be quantified by clustering coefficient C_i which measures the triangle formation in the network. For node i , which has n_i neighbors, the clustering coefficient C_i is defined as the ratio of e_i connected pairs to the number of all possible connections among the n_i neighbors $C_i = 2e_i/n_i(n_i-1)$.

3.3.4 Average shortest path

It is defined as the average number of steps along the shortest paths for all possible pairs of network nodes. It is a measure of the efficiency of information or mass transport on a network. Some examples are the average number of clicks which will lead you from one website to another, or the number of people you will have to communicate through on average, to contact a complete stranger.

The average shortest-path length is defined as follows. Consider a network G with the set of vertices V . Let $\text{dist}(v_1, v_2)$ denote the shortest distance between v_1 and v_2 ($v_1, v_2 \in V$). Assume that $\text{dist}(v_1, v_2)=0$ if $v_1=v_2$ or v_2 cannot be reached from v_1 , $\text{has_path}(v_1, v_2)=0$ if $v_1=v_2$ or if there is no path from v_1 and v_2 , and $\text{has_path}(v_1, v_2)=1$ if there is a path from v_1 to v_2 ; then the average shortest path length ASPL_G is:

$$\text{ASPL}_G = \frac{\sum_{i,j} \text{dist}(v_i, v_j)}{\sum_{i,j} \text{has_path}(v_i, v_j)}$$

Here N denotes the number of nodes in G , $\sum_{i,j} \text{dist}(v_i, v_j)$ is the value of all-pairs shortest-path length of graph G , and $\sum_{i,j} \text{has_path}(v_i, v_j)$ is the number of paths that exist in graph G [123].

3.3.5 Diameter

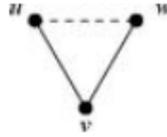
The diameter of a graph is the length of the longest geodesic path (A geodesic path, also called simply a shortest path, is a path between two vertices such that no shorter path exists) between any pair of vertices in the network for which a path actually exists [2].

3.3.6 Network transitivity

If vertex x is connected to vertex y and y is connected to w , then x is also connected to w . Perfect transitivity exists only when in a network each unit is a fully connected clique ("A clique is a maximal subset of the vertices in an undirected network such that every member of the set is

connected by an edge to every other. The word “maximal” here means that there is no other vertex in the network that can be added to the subset while preserving the property that every vertex is connected to every other”).

This measure is useful in social networks, the fact that x knows y and y knows w does not guarantee that also x knows w , but is likely. The friend of my friend is not necessarily my friend, but is far more likely to be my friend than some randomly chosen member of the population.



“The path uvw (solid edges) is said to be closed if the third edge directly from u to w is present (dashed edge).

We can quantify the level of transitivity in a network as follows. If u knows v and v knows w , then we have a path uvw of two edges in the network. If u also knows w , we say that the path is closed—it forms a loop of length three, or a triangle, in the network. In the social network jargon, u , v , and w are said to form a closed triad. We define the clustering coefficient to be the fraction of paths of length two in the network that are closed. That is, we count all paths of length two, and we count how many of them are closed, and we divide the second number by the first to get a clustering coefficient C that lies in the range from zero to one:

$$C = \frac{\text{(number of closed paths of length two)}}{\text{(number of paths of length two)}}.$$

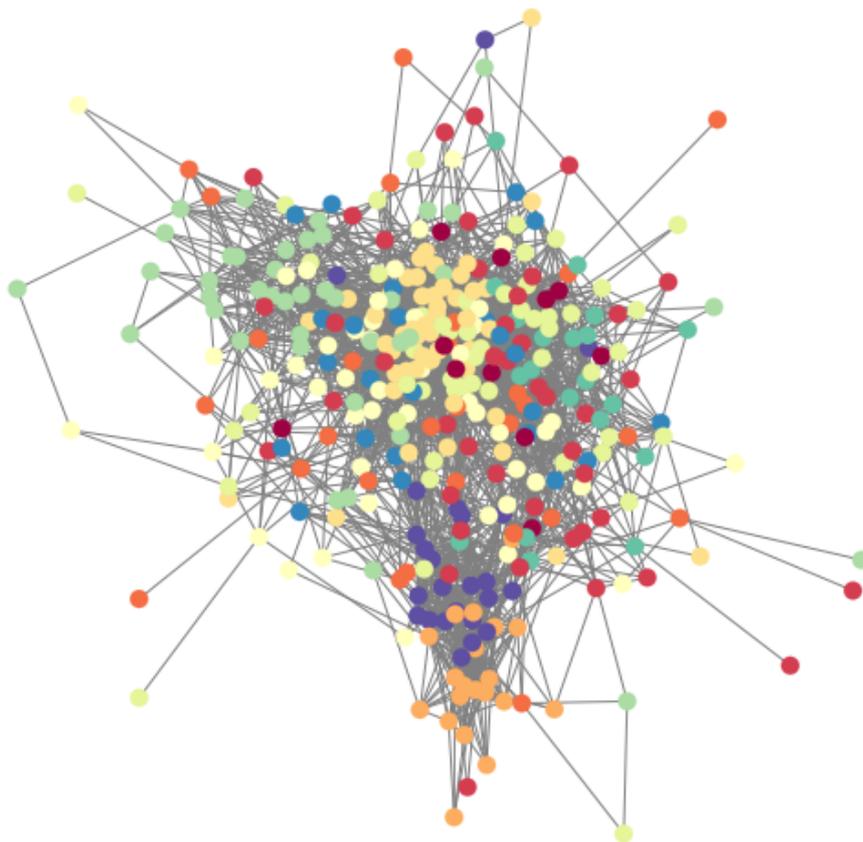
$C = 1$ implies perfect transitivity, i.e., a network whose components are all cliques. $C = 0$ implies no closed triads, which happens for various topologies, such as a tree or a square lattice (which has closed loops with even numbers of vertices only and no closed triads)” [2].

3.4. Results

The structural metrics of the network describe the variance of the analyzed market returns. This is directly related to the fact that the variance is positively correlated with the correlation of the shares making up the index. As the correlation increases, the benefit of diversification is lost.

Plotting some windows of the network is very clear that i) in phases with high volatility the network is more connected, with a smaller diameter and ii) the network is not divided into different communities but there is only a giant component.

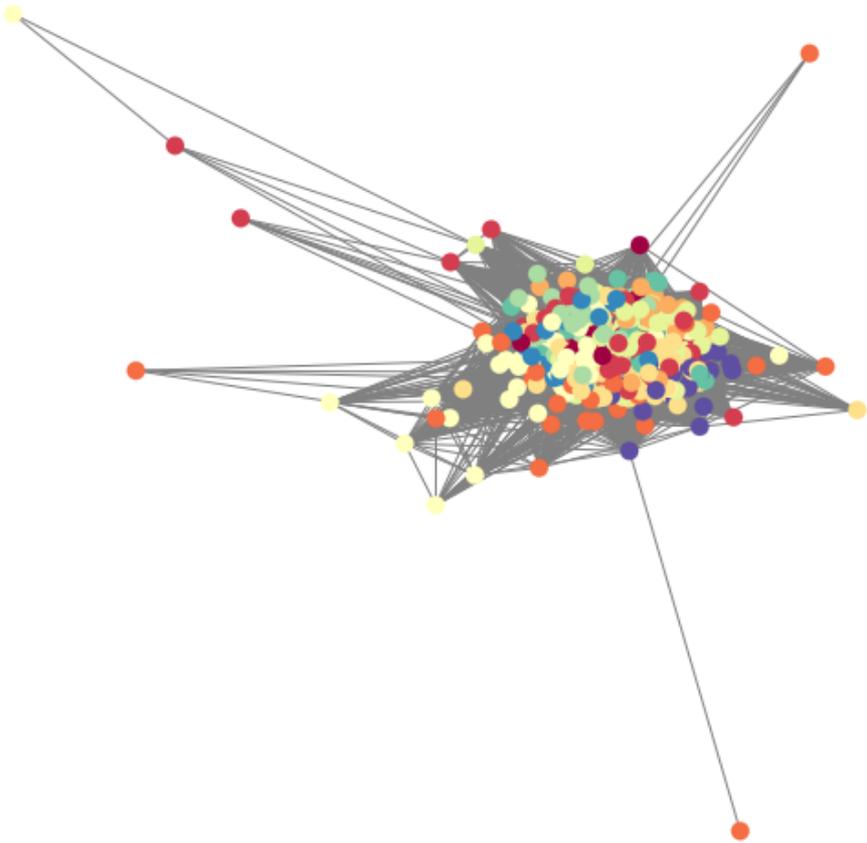
Network of 2000-09-30T00:00:00.000000000 - density= 0.05 - modularity= 0.23



The above figure shows the Dot.com bubble: period of low volatility, high modularity and low density.

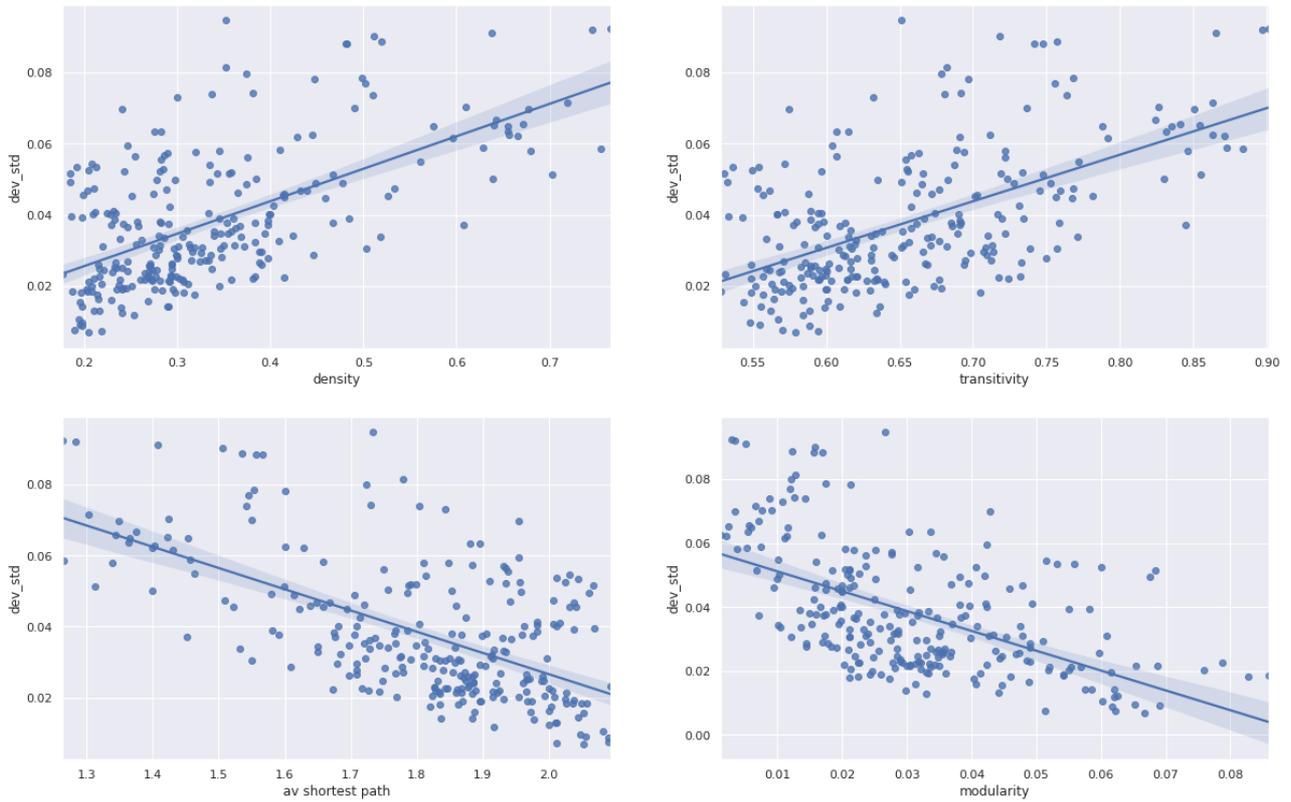
Network of

2009-03-31T00:00:00.000000000 - density= 0.76 - modularity= 0.01

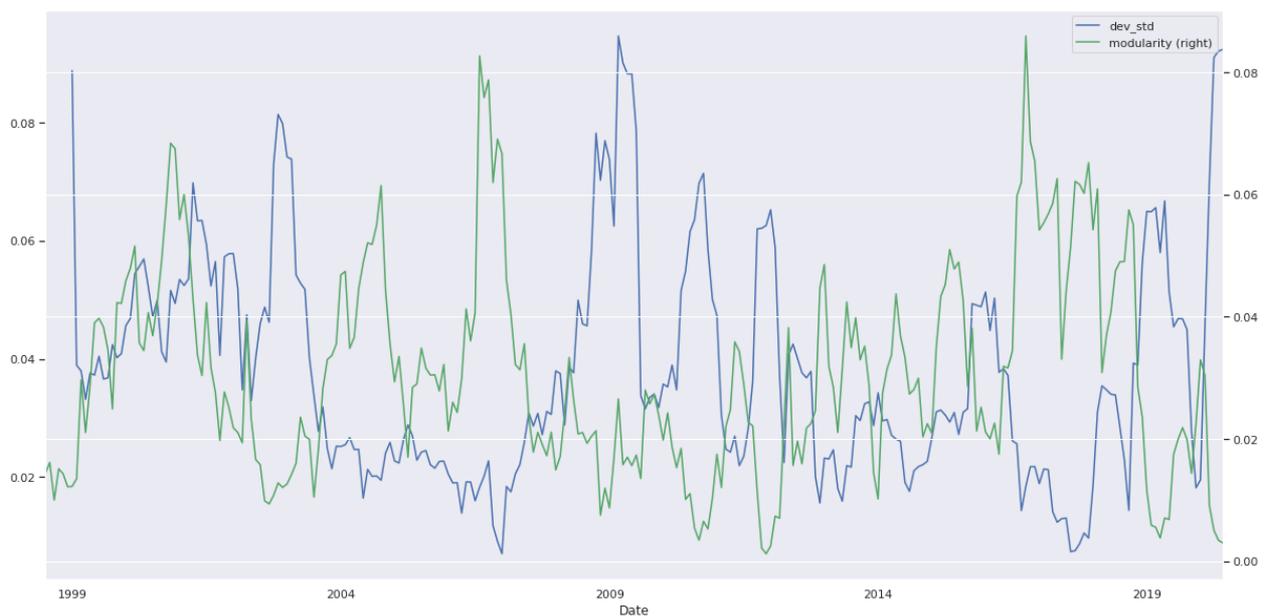


The above figure shows the tail of the Subprime mortgage crisis: period of high volatility, low modularity and high density.

From the network realized through the 6-month rolling windows procedure, an important correlation emerges with the standard deviation of the reference index log yields (direct, as far as density and transitivity are concerned, and inverse as far as average shortest path and modularity are concerned).



The metrics of the network built on the correlations between the performance of the stocks making up the index seem to be a good indicator of market risk. The interesting thing that emerges from the study is how, although the structural metrics give us information similar to that of the classic standard deviation, they are not perfectly superimposable. In fact, the measurement of the correlation network provides us with information on an implicit risk, linked to correlation, which in our opinion deserves to be studied in depth as a complementary risk indicator.



Modularity, in particular, is the most interesting metric because it provides us with important information on how effective sectoral diversification is at that particular moment in history.

Obviously, a similar indicator can be built for any parameter of aggregation of the shares that make up a hypothetical market index, both static (defensive against cyclical, geographical areas, etc.) and dynamic over time (exposure to factors, fundamental aggregation parameters, etc.).

The analysis of the historical series of structural metrics also shows a link with the price direction of the index itself, as well as with the volatility of returns.



As can be seen from the two previous graphs, the current period, from the beginning of 2019 onwards, represents (at least at present) an exception to the historical series, since the trend of the index has continued to reach new highs despite the risk indicators (both the standard deviation and the structural metrics) have shown increasing risks.

4. DISCUSSIONS AND FUTURE RESEARCH

The work carried out shows a new approach to the study of line graphs and projects' network. Compared to classical work on the subject, this research takes into account the "complexity" of the project, i.e. the interactions between players. We would like to pursue this research in order to understand if and how the structural measures of evolving network can assess the riskiness of both a single node and overall project. Our aspiration is the possibility to find measures that are predictive to success of the project. We are still working with our counterpart. A further step is to build multilayer networks

The natural continuation of this research is the Learning machine with which it is possible to make inferences because analytically it would be too narrow.

This ambition until a few years ago would have been only speculation, but the advent of big data and the billions of data available today make this kind of analysis possible.

The network approach we use lends itself to being the most appropriate technique for the analysis of projects complexity beyond historical data.

We hope that the research carried out in this work can represent a starting point for future studies and interest.

5. BIBLIOGRAPHY

- [1] Luciano da Fontoura Costa, Osvaldo N. Oliveira Jr. , Gonzalo Travieso , Francisco Aparecido Rodrigues , Paulino Ribeiro Villas Boas , Lucas Antiqueira , Matheus Palhares Viana & Luis Enrique Correa Rocha (2011) Analyzing and modeling real-world phenomena with complex networks: a survey of applications, *Advances in Physics*, 60:3, 329-412.
- [2] Newman, M. E.J. 2010. *Networks: An Introduction*, NY: Oxford University Press.
- [3] Newman, M., Barabasi, A.-I. and Watts, D.J. 2006. *The Structure and Dynamics of Networks*, Princenton, NJ: Princenton University Press.
- [4] https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg
- [5] Erdos and Renyi 1960; Rapoport 1963.
- [6] Scott, J. P. 2000. *Social Network Analysis: A Handbook*, 2, London: Sage Publications.
- [7] D. J. Watts, S. H. Strogatz. Collective dynamics of "small-world" networks. *Nature* 393(1), 440–442 (1998).
- [8] Telesford, Qawi & Joyce, Karen & Hayasaka, Satoru & Burdette, Jonathan & Laurienti, Paul. (2011). The Ubiquity of Small-World Networks. *Brain connectivity*. 1. 367-75.
- [9] Strogatz, *Nature* 410 (2001) 268.
- [10] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. 1999. On power-law relationships of the Internet topology.
- [11] Albert, R., Jeong, H. and Barabási, A.-L. 2000. *Nature*, 406: 378–382.
- [12] Barabási, A.-L. and Oltvai, Z. N. 2004. *Nat. Rev. Genet.*, 5(2): 101–113.
- [13] Costa, L. da F., Sporns, O., Antiqueira, L., Nunes, M. G.V. and Oliveira Jr, O. N. 2007. *Applied Physics Letters*, 91: 054107.
- [14] Costa, L. da F., Rodrigues, F. A. and Travieso, G. 2007. *Phys. Rev.*, E 76(4): 46106
- [15] Boccaletti, S., Latora, V., Moreno, Y., Chaves, M. and Hwang, D.-U. 2006. *Phys. Rep.*, 424(4–5): 175–308.
- [16] Costa, L. da F. and Rodrigues, F. A. 2008. What is there between any two nodes in a complex network?, Preprint arXiv:0801.4068
- [17] M.E.J. Newman, *SIAM Rev.* 45 (2003) 167.
- [18] Prigogine, I., Allen, P. M. G., & Herman, R. (1977). Long term trends and the evolution of complexity.
- [19] James Ladyman, James Lambert, Karoline Wiesner. What is a Complex System?.
- [20] Editorial. No man is an island. *Nature Physics*, 5:1, 2009.
- [21] R.S. MacKay. Nonlinearity in complexity science. 2008.
- [22] Pablo Arrighi, Gilles Dowek. *Csusal graph dynamics*.
- [23] Richard Feynman. *Feynman Lectures on Computation*. Westview Press, 2000.
- [24] Jurgen Schroder, *Emergence: Non-Deducibility or Downwards Causation?*
- [25] Herbert A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962.
- [26] P.W. Anderson. More is different: Broken symmetry and the nature of the hierarchical structure of science. *Science*, 177, 1972.
- [27] Andrea Roli, *An introduction to Complex System Science*
- [28] James P. Crutchfield. The calculi of emergence: computation, dynamics and induction. *Physica D: Nonlinear Phenomena*, 75(1-3):11–54, 1994.
- [29] James P. Crutchfield and Cosma R. Shalizi. Thermodynamic depth of causal states: objective complexity via minimal representations. *Physical Review E*, 59(1):275–283, 1999.
- [30] James P. Crutchfield and Karl Young. Inferring statistical complexity. *Physical Review Letters*, 63:105, 1989.
- [31] Cosma R. Shalizi and Christopher Moore. What is a macrostate? subjective observations and objective dynamics. *cond-mat/0303625*, 2003.
- [32] Yanis Varoufakis, Talking to my faughter about the economy.

- [33] John Maynard Keynes, *The General Theory of Employment, Interest and Money*.
- [34] Ping Chen, *Mathematical Representation in Economics and Finance: Philosophical Preference, Mathematical Simplicity, and Empirical Relevance*.
- [35] Léon Walras, *Elements of Pure Economics: The Theory of Social Wealth*.
- [36] Miroslav N. Jovanovic, *Evolutionary Spatial Economics: Understanding Economic Geography and Location Over Time*.
- [37] Boulton, J. G., Allen, P. M., & Bowman, C. (2015). *Embracing Complexity: Strategic Perspectives for an Age of Turbulence*. Oxford, U. K.: Oxford University Press.
- [38] George R. Feiwel, *Joan Robinson and Modern Economic Theory*.
- [39] Paul Ormerod, *The Death of Economics*.
- [40] Edward Fullbrook, *A Guide to What's Wrong with Economics*.
- [41] Joon Chang, *23 Things They Don't Tell You About Capitalism*.
- [42] Fullbrook 2009
- [43] Stiglitz 2002
- [44] Thorstein Veblen 1898
- [45] Boulding 1981
- [46] Arthur 1989
- [47] Metcalfe 1998
- [48] Hadgson 2002
- [49] Farmer and Geanakoplos 2009
- [50] Beinhocker 2007
- [51] Bronk 2009
- [52] Ormerod 2005
- [53] Room 2011
- [54] Buchanan 2014
- [55] Marshall, Arthur 1994
- [56] Knight 1940
- [57] Shackle 1953
- [58] Hayek 1958
- [59] Common 1995
- [60] Scott-Cato 2009
- [61] Sen 1987
- [62] Meadows et al. 1972
- [63] M. Jackson, *Social and Economics Networks*. Princeton: Princeton University Press, 2008.
- [64] J. Doyne Farmer, *Economics needs to treat the economy as a complex system*.
- [65] Henrik Bruus, *Many-Body Quantum Theory in Condensed Matter Physics: An Introduction*.
- [66] Adam Smith, *The Theory of Moral Sentiments*.
- [67] J. D. Farmer, *A rosetta stone for connectionism*.
- [68] M. E. J. Newman, "The structure and function of complex networks," *Siam Journal on Applied Mathematics*, vol. 45, no. 2, pp. 167–256, 2003.
- [69] M. Mitchell, "Complex systems: Network thinking," *Artificial Intelligence*, vol. 170, no. 18, pp. 1194–1212, 2006.
- [70] E. Beinhocker, *The Origin of Wealth: Evolution, Complexity and the Radical Remaking of Economics*. Harvard Business School Press.
- [71] *Complexity: A Guided Tour*. Oxford University Press, 2009.
- [72] M. E. J. Newman, "Complex systems: A survey," *American Journal of Physics*, vol. 79, pp. 800–810, 2011.
- [73] A. Kirman, *Complex Economics: Individual and Collective Rationality*. Routledge, 2010.
- [74] J. D. Farmer and D. Foley, "The economy needs agent-based modelling," *Nature*, vol. 460, pp. 685–686, 2009.

- [75] J. D. Farmer, M. Gallegati, C. Hommes, A. Kirman, P. Omerod, S. Cincotti, A. Sanchez, and D. Helbing, “A complex systems approach to constructing better models for managing financial markets and the economy,” tech. rep., FuturICT, 2012.
- [76] N. Vandewalle et al., Non-random topology of stock markets.
- [77] N. Vandewalle et al., Self-organized critical topology of stock markets.
- [78] R.N. Mantegna, Hierarchical structure in financial markets.
- [79] G. Bonanno et al., Networks of equities in financial markets.
- [80] Dror Y. Kenett et al., Partial correlation analysis: Applications for financial markets
- [81] H. Markowitz, Portfolio Selection: Efficient Diversification of Investment (J. Wiley, New York, 1959).
- [82] E. J. Elton and M. J. Gruber, Modern Portfolio Theory and Investment Analysis (J. Wiley and Sons, New York, 1995).
- [83] J. Y. Campbell, A. W. Lo, and A. C. MacKinlay, The Econometrics of Financial Markets (Princeton University Press, Princeton, 1997).
- [84] L. Laloux, P. Cizeau, J.-P. Bouchaud, and M. Potters, Phys. Rev. Lett. 83, 1468 (1999).
- [85] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. N. Amaral, and H. E. Stanley, Phys. Rev. Lett. 83, 1471 (1999).
- [86] D. B. Panton, V. Parker Lessig, and O. M. Joy, Journal of Financial and Quantitative Analysis 11, 415 (1976).
- [87] L. Kullmann, J. Kertesz and R. N. Mantegna, Physica A 287, 412 (2000).
- [88] M. Bernaschi, L. Grilli, L. Marangio, S. Succi and D. Vergni cond-mat/0003025
- [89] L. Giada and M. Marsili, Phys. Rev. E 63, 061101 (2001).
- [90] M. Marsili, Quantitative Finance 2, 297 (2002).
- [91] Wenyue Sun et al., Network Analysis of the Stock Market.
- [92] V. Boginski, S. Butenko, and P. M. Pardalos. Statistical analysis of financial networks. Computational statistics and data analysis, 48(2):431–443, 2005.
- [93] P. Gopikrishnan, B. Rosenow, V. Plerouand, and H. E. Stanley. Quantifying and interpreting collective behavior in financial markets. Physical Review E, 64(3):035–106, 2001.
- [94] W. Q. Huang, X. T. Zhuang, and S. Yao. A network analysis of the chinese stock market. Physica A: Statistical Mechanics and its Applications, 2009.
- [95] A. Namaki, A.H. Shirazi, R. Raei, and G.R. Jafari. Network analysis of a financial market based on genuine correlation and threshold method. Physica A: Statistical Mechanics and its Applications, 2011.
- [96] M. E. Newman. Power laws, pareto distributions and Zipf ’s law. Contemporary physics, 46(5):323–351, 2005.
- [97] B. Podobnik, D. Horvatic, A. M. Petersen, and H. E. Stanley. Cross-correlations between volume change and price change. PNAS, 106(52):22079–22084, 2009.
- [98] B. Podobnik and H. E. Stanley. Detrended correlation analysis: A new method for analyzing two non-stationary time series. Physical review letters, 100(8):084–102, 2008.
- [99] B. Podobnik, D. Wang, D. Horvatic, I. Grosse, and H. E. Stanley. Time-lag cross-correlations in collective phenomena. Europhysics Letters, 2010.
- [100] C. K. Tse, J. Liu, and C. M. Lau. A network perspective of the stock market. Journal of Empirical Finance, 2010.
- [101] Bonanno, G., Lillo, F., Mantegna, R.N., 2001. High-frequency cross-correlation in a set of stocks. Quantit. Finan. 1, 96–104.
- [102] Bonanno, G., Caldarelli, G., Lillo, F., Mantegna, R.N., 2003. Topology of correlation-based minimal spanning trees in real and model markets. Phys. Rev. E 68,046103.
- [103] Onnela, J.P., Chakraborti, A., Kaski, K., 2003. Dynamics of market correlations: taxonomy and portfolio analysis. Phys. Rev. E 68, 056110.
- [104] Onnela, J.P., Kaski, K., Kertesz, J., 2004. Clustering and information in correlation based financial networks. Euro. Phys. J. B 38, 353–362.

- [105] R. D. Smith. The Spread of the Credit Crisis: View from a Stock Correlation Network. *Journal of Korean Physical Society*, 54:2460, June 2009.
- [106] B. Rosenow, V. Plerou, P. Gopikrishnan, and H. E. Stanley. Portfolio optimization and the random magnet problem. *Europhysics Letters*, 59(4):500, 2002.
- [107] R.N. Mantegna, *The European Physical Journal B* 11 (1999) 193.
- [108] H.J. Kim, Y. Lee, B. Kahng, I.M. Kim, *Journal of Physical Society of Japan* 71 (2002) 2133.
- [109] J.P. Onnela, K. Kaski, J. Kertesz, *The European Physical Journal B* 38 (2004) 353.
- [110] M. Tumminello, T. Aste, T.D. Matteo, R.N. Mantegna, *Proceeding of National Academy of Sciences* 102 (2005) 10421.
- [111] M. Tumminello, T.D. Matteo, R.N. Mantegna, *The European Physical Journal B* 55 (2007) 209.
- [112] V. Boginski, S. Butenko, P.M. Pardalos, *Computational Statistics & Data Analysis* 48 (2005) 431.
- [113] L. Laloux et al. *Phys. Rev. Lett.* 83, 1467 (1999); V. Plerou et al., *ibid.* 83, 1471 (1999).
- [114] Potters M., Bouchaud J.-P. and Laloux L., *Acta Phys. Pol. B*, 36 (2005) 2767.
- [115] Kwapien J., Drozd S., Gorski A. Z. and Oswiecimka P., *Acta Phys. Pol. B*, 37 (2006) 3039.
- [116] Mayya K. B. K. and Amritkar R. E., *AAPPS Bull.*, 17 (2007) 96.
- [117] Thurner S. and Biely C., *Acta Phys. Pol. B*, 38 (2007) 4111.
- [118] Bouchaud J.-P., Laloux L., Miceli M. A. and Potters M., *Eur. Phys. J. B*, 55 (2007) 201.
- [119] Nakamura T. and Small M., *Physica A*, 383 (2007) 96.
- [120] Cohen, J., Cohen, P., West, S.G., Aiken, L.S., 2003. *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*, 3rd ed. Lawrence Erlbaum Associates, Hillsdale, NJ.
- [121] Newman, M. E. J., Mixing patterns in networks, *Phys. Rev. E* 67, 026126 (2003).
- [122] Newman, M. E. J. and Girvan, M., Finding and evaluating community structure in networks, *Phys. Rev. E* 69, 026113 (2004).
- [123] Guoyong Mao, and Ning Zhang; Analysis of Average Shortest-Path Length of Scale-Free Network
- [124] Nicholas Huang, Leo Keselman, Vincent Sitzmann; Beyond correlation networks
- [125] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, Mason A. Porter, Multilayer networks, *Journal of Complex Networks*, Volume 2, Issue 3, September 2014, Pages 203–271,
- [126] Tsou C-M, On the project management scheduling based on agent technology and theory of constraint, *International Journal of Electronic Business Management*, Vol. 10, No. 4, pp. 286-295, 2012.
- [127] Henderckson C., *Project Management for Construction, Version 2.2*, Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, 152-154, 2008.
- [128] Y. Cohen, A. Sadeh, A New Approach for Constructing and Generating AoA Networks, *Journal of computer science*, Vol. 1, No. 1, 2007.
- [129] N. E. Mouhoub, Benhocine A., An efficient algorithm for generating AoA networks, *IJCSI* Vol. 9, No 1, 2012.
- [130] Dechter R., Temporal constraint networks, *Artificial intelligence Journal*, Vol. 49, pp. 61-95, 1991.
- [131] S. S. Hanks S. S., McDermott D. V., Default reasoning, nonmonotonic logics, and the frame problem, In *Proceedings AAAI, philadelphia*, pp. 328-333, 1986.
- [132] Shoham Y., *Reasoning about change: Time and causation from the stand point of artificial intelligence* (MIT Press, Cambridge, MA), 1988.
- [133] Allen J.F., Towards a general theory of action and time, *Artif. Intell.* Vol. 23 No. 2, pp. 123-154, 1984.
- [134] K. Kahn K. Gorry G. A., Mechanizing temporal knowledge, *Artif. Intell.* Vol. 9, pp. 87-108, 1977.

- [135] D.V. McDermott, A temporal logic for reasoning about process and plan, *Cogn. Sci*, Vol 6, pp. 101-155, 1982. Allen J. F., Maintaining knowledge about temporal intervals, *Commun. ACM*, Vol. 26 No. 11, pp. 832-843, 1983.
- [136] M. Vilain M., Kautz H., Constraint propagation algorithms for temporal reasoning, In *Proceedings AAAI*, Philadelphia, pp. 377-382, 1986.
- [137] Malik J. Inford J.O., Reasoning in time and space, *Proceedings IJCAI*, Vol. 83, Karlsruhe, pp. 343-345, 1983.
- [138] R.E. Valdes-Perez R.E., Spatio-temporal reasoning and linear inequalities, *Artificial Intelligence Laboratory*, AIM-875, MIT, Cambridge, MA, 1986.
- [139] T.L Deanand T. L., McDermott D. V., Temporal data base management, *Artif. Intell. Vol.* 32, pp. 1-55, 1987.
- [140] U. Montanari U., Networks of constraints: fundamental properties and applications to picture processing, *Inf. Sci*, Vol. 7, pp. 95-132, 1974.
- [141] Mackworth A.K., Consistency in networks of relations, *Artif. Intell. Vol. 8*, No. 1, pp. 99-118, 1977.
- [142] Gashnig J., Performance measurement and analysis of certain search algorithms, *Tech. Rept. CMU-CS-79-124*, Carnegie-Mellon University, Pittsburg, 1979.
- [143] Freuder E. C., Synthesizing constraint expressions, *Commun. ACM*, Vol. 21, No. 11, pp. 958-965, 1978.
- [144] Nebel B., Buckert H. J., Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra, *Journal of ACM Vol. 42*, No. 1, pp. 43 – 66, 1995.
- [145] T. Drakengren T., Jonsson P., Towards a Complete Classification of Tractability in Allen's Algebra. *IJCAI*, pp. 1466-1475, 1997.
- [146] T.B. Smith T. B., Pyle J.M., An effective algorithm for project scheduling with arbitrary temporal constraints, *American association for artificial intelligence*, 2004.
- [147] R. Dechter R. Pearl J., Tree clustering for constraint networks, *Artif. Intell.*, Vol. 38, No. 3, pp. 353-366, 1989.
- [148] Vanhoucke M., *Project management with dynamic scheduling: Baseline scheduling, Risk analysis and Project control*, Springer, 2005.
- [149] N. E.Mouhoub, Belouadah H. and Boubetra A., Algorithme de construction de graphe PERT à partir d'un graphe des potentiels donné, *Studia Univ. Babes Bolyai, Informatica*, Vol. 51, No. 2, pp. 340-348, 2006.
- [150] C. Heuchenne, Sur une certaine correspondance entre graphes, *Bull. Soc. Roy. Sci. Liege* 33(1964) 743-753.
- [151] N. E. Mouhoub, S. Akrouf, Approche optimisée dans les problèmes d'ordonnancement de projet, *ANL'07, Actes du colloque international sur l'analyse non linéaire*, Sétif university, Algeria, 2007.
- [152] J. E. Kelley, Critical path planning and scheduling- Mathematical basis, *Operations research*, vol. 9, n° 3, 1961, p. 296-320.

APPENDIX

First code, window of one month:

```
import pandas_datareader as pdr
import datetime
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from pandas import DataFrame
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
import networkx as nx
import community
import seaborn as sns; sns.set(color_codes=True)

""""#Estrae le quotazioni storiche da Yahoo Finance""""
#questa parte è sotto commento perchè per comodità importiamo dal file excel nel blocco
successivo, si usa solo la prima volta per acquisire i dati.

# # # url = "https://raw.githubusercontent.com/datasets/s-and-p-500-
companies/master/data/constituents.csv"
# # # df_1 = pd.read_excel(url)
# df_1 = pd.read_excel('lista_stocks_presenti_al_genn98.xlsx')
# df_1 = df_1.loc[df_1.BIG==1]
# df_1.Sector = pd.factorize(df_1.Sector)[0]
# tickers = list(df_1['ticker'])
# sector = list(df_1.Sector)

# for i in range (len(tickers)):
#   if tickers[i]=='(BF.B)':
#     tickers[i]="(BF-B)"
#   if tickers[i]=='(BRK.B)':
#     tickers[i]="(BRK-B)"

# dizionario = dict(zip(tickers,sector)) # in questo dizionario ci sono i settori fattorizzati che
verranno usati come communities per il network e per la misurazione

# SP500_selezionati_a = pdr.get_data_yahoo(tickers, start = '1998-1-1')['Close']
# SP500_selezionati_a = pd.DataFrame(SP500_selezionati_a)

""""#Per evitare i tempi di importazione da server importiamo i dati direttamente da un file .xlsx he
abbiamo creato in precedenza ('quotazioni_storiche_sp500.xlsx')""""
```

```

# url = "https://raw.githubusercontent.com/datasets/s-and-p-500-
companies/master/data/constituents.csv"
# df_1 = pd.read_excel(url)
df_1 = pd.read_excel('lista_stocks_presenti_al_genn98.xlsx')
# df_1 = df_1.loc[df_1.BIG==1]
df_1.Sector = pd.factorize(df_1.Sector)[0]
tickers = list(df_1['ticker'])
sector = list(df_1.Sector)

dizionario = dict(zip(tickers,sector)) # in questo dizionario ci sono i settori fattorizzati che verranno
usati come communities per il network e per la misurazione

SP500_selezionati_a = pd.read_excel('quotazioni_storiche_sp500.xlsx')
SP500_selezionati_a = pd.DataFrame(SP500_selezionati_a)
SP500_selezionati_a = SP500_selezionati_a.set_index('Date',1)

SP500_selezionati_a

SP500_selezionati = SP500_selezionati_a

""""#scarica la serie mensile di sp500""""

Indice_completo = pdr.get_data_yahoo('^GSPC', start = '1998-1-1')['Close']
Indice_completo = pd.DataFrame(Indice_completo)
Indice_completo = Indice_completo.resample('M').last()

Indice_completo

Indice_completo.to_excel('sp.xlsx')

""""#Bug Fixing""""

colonne = list(SP500_selezionati.columns.values)
SP500_selezionati = pd.DataFrame(SP500_selezionati, columns=colonne)

""""#Crea un DataFrame che contiene i log-rendimenti per il successivo calcolo delle correlazioni""""

df = SP500_selezionati
daily_close_px = df

#crea il dataframe
df = pd.DataFrame(daily_close_px)
#crea un dataframe con le variazioni percentuali per ogni unità di tempo sull' unità precedente
dfb = np.log(df / df.shift(1))

""""#crea un contatore che evidenzi i mesi in ordine crescente sulla serie daily""""

SP = dfb.resample('M').last()
SP['contatore']= range(0, len(SP))
SP = SP.resample('D').bfill()

```

```

SP = pd.DataFrame(SP.contatore)
dfb=dfb.join(SP)
dfb.contatore = dfb.contatore.fillna(0)
dfb.contatore[:60].plot()

```

```

""""#Genera le matrici di correlazione. Le matrici saranno pari ai mesi ed ogni correlazione
superiore alla treshold sarà un link del network""""

```

```

#listm sarà la lista che contiene le matrici di correlazione
listm =[]
n_finestre = int(max(dfb.contatore)+1)
for b in range(n_finestre):
    finestra = dfb.loc[dfb.contatore==int(b)]
    nome = str(b)
    #print (data)
    correlations = finestra.corr()
    correlations
    listm.append(correlations)
print("verranno calcolate ", b+1, " finestre temporali")

```

```

cut = 0.5

```

```

""""#Genera i networks statici che rappresentano lo stato del network dinamico nei vari T, calcola le
metriche e le inserisce in una lista che verrà poi inserita in un dataframe

```

Attenzione all' ultimo punto: Disegna il network di ogni finestra, ma ovviamnente rallenta infinitamente il codice, eventualmente mettere sotto commento in caso di networks con molti nodi

```

""""

```

```

#definiamo le liste che ospiteranno le metriche
lista = []
listb = []
listc = []
listd = []
liste = []
listf = []
listg = []
listnet=[]

```

```

for t in range(n_finestre): #n_finestre
    #NOTA 1 - questa istruzione è per salvare il plot delle singole finestre, eventualmente mettere
sotto commento
    graph_ = str(t)
    gra = int(graph_)
    t_ = int(t)
    #crea la adajacency matrix per ogni metriche di correlazione
    links = listm[gra].stack().reset_index()
    links.columns = ['var1', 'var2','value']
    links
    #filtra i link (scalino)

```

```

links_filtered=links.loc[ (links['value'] > cut) & (links['var1'] != links['var2']) ]
links_filtered
links_filtered.drop_duplicates()

links_filtered = links_filtered.drop('value',1)
links_filtered['value']=1

#genera il network
G=nx.from_pandas_edgelist(links_filtered, 'var1', 'var2')
listnet.append(G)

#alcune metriche che finiscono in altrettante liste - implementare altre:

#density a
Den = nx.density(G)
lista.append(Den)

#av clustering b
try:
    av_cl = nx.average_clustering(G)
    listb.append(av_cl)
except:
    listb.append(0)

#communities c - le communities sono rappresentate dai serrori di appartenenza delle stocks
comm = dizionario.values()
try:
    mod = max(comm)
    listc.append(mod)
except:
    listc.append(0)

# average sh path d. Qualora l'algoritmo rilevi che ci sono più networks non collegati tra di loro
la sh. path viene calcolata sul network con il maggior numero di nodi.

try:
    if nx.is_connected(G) ==True:
        av_sp = nx.average_shortest_path_length(G)
    else:
        components = nx.connected_components(G)
        largest_component = max(components, key=len)
        subgraph = G.subgraph(largest_component)
        av_sp = nx.average_shortest_path_length(subgraph)
except:
    av_sp = ("inf")
listd.append(av_sp)

#diameter. Qualora l'algoritmo rilevi che ci sono più networks non collegati tra di loro il diametro
viene calcolato sul network con il maggior numero di nodi.

try:

```

```

if nx.is_connected(G) == True:
    diameter = nx.diameter(G)
else:
    components = nx.connected_components(G)
    largest_component = max(components, key=len)
    subgraph = G.subgraph(largest_component)
    diameter = nx.diameter(subgraph)
except:
    diameter="inf"
liste.append(diameter)

#transitivity f

transitivity_ = nx.transitivity(G)
listf.append(transitivity_)

print("elaborata la finestra numero: ", t)

#modularity
#crea una partizione basata sul settore di appartenenza

fb = pd.DataFrame(tickers, columns = ['Symbol'])
fb['Sector']=sector
fb = fb.set_index('Symbol',1)

fa = pd.DataFrame(index=list(G.nodes))
fa = fa.join(fb)
fa = fa.reset_index()

try:
    n_communities = len(fa.Sector.unique())
except:
    n_communities = 1
listaz=[]
listaliste=[]
for i in range (n_communities):
    df_comm = fa.loc[fa.Sector==i]
    listaz = list(df_comm['index'])
    listaliste.append(listaz)
partition = listaliste
try:
    modularity = nx.algorithms.community.quality.modularity(G, partition)
except:
    modularity = 0
listg.append(modularity)

## #NOTA 1 - disegna e salva l'immagine di ogni finestra EVENTUALMENTE ELIMINARE
# nx.draw(G, with_labels=True, node_color=list(comm), node_size=400, edge_color='red',
linewidths=0.1, font_size=10)

```

```

# plt.savefig(graph_)
# plt.close()

#disegna le serie storiche delle metriche
plt.figure(figsize = (15,8))
plt.plot(lista)
plt.plot(listb)
# plt.plot(listc)
plt.plot(listd)
plt.plot(liste)
plt.plot(listf)
plt.plot(listg)
plt.legend(['Density', 'Average Clustering', 'Average shortest path', 'Diameter', 'Network transitivity',
'Modularity'])
plt.show()

print("Network all' ultima finestra temporale")
# plt.figure(figsize=(10,10))
# nx.draw(G, with_labels=True, node_color='red', node_size=400, edge_color='red',
linewidths=0.1, font_size=10, cmap='Accent')
# plt.show()

""""#Crea un DataFrame (export) dove verranno ospitate le metriche per l'export in foglio di
calcolo""""

dfb_mensile = dfb.resample('M').last()
indice_export = dfb.resample('M').last().index
export = pd.DataFrame(index = indice_export[:n_finestre])

export['density']=lista
export['av_clustering']=listb
export['communities'] = listc
export['av shortest path'] = listd
export['diameter'] = liste
export['transitivity'] = listf
export['modularity']=listg
export=export.rolling(12).mean()

# aggiugiamo l'andamento di SP500 per avere un parametro su cui ragionare inseguito in base alle
metriche osservate
export['SP500_Price_Index']= Indice_completo.Close[0:]
export['log_rend_forward']= np.log(export.SP500_Price_Index.shift(-
1)/export.SP500_Price_Index.shift(0))
export['log_rend']= np.log(export.SP500_Price_Index.shift(0)/export.SP500_Price_Index.shift(1))
export['dev_std'] = export.log_rend.rolling(10).std()
export['roc6']= (export.SP500_Price_Index.shift(-6)/export.SP500_Price_Index.shift(0))

name = "Metriche__m"+"__"+str(cut)+".xlsx"

```

```

export.to_excel(name)
Indice_completo.to_excel('dati_azioni_16giu2020.xlsx')

""""#per importare i risultati, poi da togliere

Verifichiamo la correlazione di alcuni fattori
""""

export.columns

data1 = pd.DataFrame(export)

fig = plt.subplots(figsize=(20,20))
fig = plt.subplot(321)
ax = sns.regplot(x="density", y="dev_std", data=data1)
fig = plt.subplot(322)
ax = sns.regplot(x="transitivity", y="dev_std", data=data1)
fig = plt.subplot(323)
ax = sns.regplot(x="av shortest path", y="dev_std", data=data1)
fig = plt.subplot(324)
ax = sns.regplot(x="diameter", y="dev_std", data=data1)
fig = plt.subplot(325)
ax = sns.regplot(x="modularity", y="dev_std", data=data1)
plt.show()

data1 = pd.DataFrame(export)

fig = plt.subplots(figsize=(20,10))
fig = plt.subplot(221)
ax = sns.regplot(x="density", y="log_rend", data=data1)
fig = plt.subplot(222)
ax = sns.regplot(x="transitivity", y="log_rend", data=data1)
fig = plt.subplot(223)
ax = sns.regplot(x="av shortest path", y="log_rend", data=data1)
fig = plt.subplot(224)
ax = sns.regplot(x="modularity", y="log_rend", data=data1)
plt.show()

data1 = pd.DataFrame(export)

fig = plt.subplots(figsize=(20,10))
fig = plt.subplot(221)
ax = sns.regplot(x="density", y="roc6", data=data1)
fig = plt.subplot(222)
ax = sns.regplot(x="transitivity", y="roc6", data=data1)
fig = plt.subplot(223)
ax = sns.regplot(x="av shortest path", y="roc6", data=data1)
fig = plt.subplot(224)
ax = sns.regplot(x="modularity", y="roc6", data=data1)

```

```
plt.show()
```

```
"""Parrebbe che le metriche del network presentino una buona correlazione con la standard deviation dei log-rendimenti e nessuna correlazione apprezzabile con i log rendimenti
```

```
#Plotta alcune finestre  
"""
```

```
n_finestre
```

```
lista_finestre= [31,62,98,128,196,215,236,269]
```

```
for finestra in lista_finestre:
```

```
    df_coloring_1 = pd.DataFrame(index = tickers)  
    df_coloring_1['colore']=sector  
    df_nodi = pd.DataFrame(index=listnet[finestra].nodes)  
    df_coloring = df_nodi.join(df_coloring_1)
```

```
    fig = plt.subplots(figsize=(20,20))  
    fig = plt.subplot(221)  
    ax = nx.draw(listnet[finestra], with_labels=False, node_color=list(df_coloring.colore),  
node_size=100, edge_color='grey', linewidths=0.1, font_size=10, cmap='Spectral')  
    A = ("network della finestra "+str(dfb_mensile.index.values[finestra])+" - density=  
"+str(round(export.density[finestra],2))+" - modularity=  
"+str(round(export.modularity[finestra],2)))  
    plt.title(A, fontsize=15)
```

```
export
```

```
"""#Plotta una legenda per i settori"""
```

```
url = "https://raw.githubusercontent.com/datasets/s-and-p-500-companies/master/data/constituents.csv"
```

```
df_1 = pd.read_csv(url)  
df_1['Sector_factorized'] = pd.factorize(df_1.Sector)[0]
```

```
lista_controllo = []  
lista_1 = []  
for i in range (len(df_1)):  
    A = df_1.Sector_factorized[i]  
    if A in lista_controllo:  
        A = A  
        lista_1.append(0)  
    else:  
        lista_controllo.append(A)  
        lista_1.append(1)
```

```
df_1['chek']=lista_1
```

```

df_legenda = df_1.loc[df_1.chek==1]
df_legenda = df_legenda.drop(['Symbol', 'Name', 'chek'],1)

df_legenda.style.background_gradient(cmap='Spectral')

export['modularity_MA']=export['modularity'].rolling(12).mean()
plt.figure(figsize=(20,10))
export.SP500_Price_Index.plot(legend=True)
export.modularity.plot(secondary_y=True, linewidth=0.1, legend=True)
export.modularity_MA.plot(secondary_y=True, legend=True)
plt.show()

export['density_MA']=export.density
plt.figure(figsize=(20,10))
export.SP500_Price_Index.plot(legend=True)
export.density.plot(secondary_y=True, linewidth=0.15, legend=True, color='red')
export.density_MA.plot(secondary_y=True, legend=True)
plt.show()

export['Shortest_p_MA']=export['av shortest path'].rolling(12).mean()
plt.figure(figsize=(20,10))
export.SP500_Price_Index.plot(legend=True)
export['av shortest path'].plot(secondary_y=True, linewidth=0.15, legend=True, color='red')
export.Shortest_p_MA.plot(secondary_y=True, legend=True)
plt.show()

tendenze = pd.DataFrame(index=export.index)
tendenze['%change 12 m (trend) SP500'] =
np.log(export.SP500_Price_Index/export.SP500_Price_Index.shift(12))
tendenze['%change 12 m (trend) modularity'] =
np.log(export.modularity_MA/export.modularity_MA.shift(12))
plt.figure(figsize=(20,10))
plt.title("Trend MA modularity VS trend SP500 price", fontsize=20)
ax = sns.regplot(x="%change 12 m (trend) SP500", y="%change 12 m (trend) modularity",
data=tendenze)

tendenze = pd.DataFrame(index=export.index)
tendenze['%change 12 m (trend) SP500 Forw'] = np.log(export.SP500_Price_Index.shift(-
1)/export.SP500_Price_Index.shift(11))
tendenze['%change 12 m (trend) modularity'] =
np.log(export.modularity_MA/export.modularity_MA.shift(12))
plt.figure(figsize=(20,10))
plt.title("Trend MA modularity VS trend SP500 price forward1", fontsize=20)
ax = sns.regplot(x="%change 12 m (trend) SP500 Forw", y="%change 12 m (trend) modularity",
data=tendenze)

export['modularity_MA']=export['modularity'].rolling(12).mean()
export['std_MA']=1/export['dev_std'].rolling(12).mean()
plt.figure(figsize=(20,10))
export.SP500_Price_Index.plot(legend=True)
# export.modularity.plot(secondary_y=True, linewidth=0.1, legend=True)

```

```

# export.modularity_MA.plot(secondary_y=True, legend=True)
export.std_MA.plot(secondary_y=True, legend=True)
plt.show()

export['modularity_MA']=export['modularity'].rolling(12).mean()
export['std_MA']=export['dev_std'].rolling(12).mean()
plt.figure(figsize=(20,10))
# export.SP500_Price_Index.plot(legend=True)
# export.modularity.plot(secondary_y=True, linewidth=0.1, legend=True)
export.modularity_MA.plot(secondary_y=True, legend=True)
export.std_MA.plot(secondary_y=True, legend=True)
plt.show()

```

Second code, window rolling 6 months:

```

import pandas_datareader as pdr
import datetime
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from pandas import DataFrame
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
import networkx as nx
import community
import seaborn as sns; sns.set(color_codes=True)

""""#Estrae le quotazioni storiche da Yahoo Finance""""

# url = "https://raw.githubusercontent.com/datasets/s-and-p-500-companies/master/data/constituents.csv"
# df_1 = pd.read_csv(url)
# df_1.Sector = pd.factorize(df_1.Sector)[0]
# tickers = list(df_1.Symbol)
# sector = list(df_1.Sector)

# for i in range (len(tickers)):
#   if tickers[i]=='(BF.B)':
#     tickers[i]="(BF-B)"
#   if tickers[i]=='(BRK.B)':
#     tickers[i]="(BRK-B)"

# dizionario = dict(zip(tickers,sector)) # in questo dizionario ci sono i settori fattorizzati che
# verranno usati come communities per il network e per la misurazione

# SP500_selezionati_a = pdr.get_data_yahoo(tickers, start = '1998-1-1')['Close']
# SP500_selezionati_a = pd.DataFrame(SP500_selezionati_a)

```

```
""""#Per evitare i tempi di importazione da server importiamo i dati direttamente da un file .xlsx he abbiamo creato in precedenza ('quotazioni_storiche_sp500.xlsx')""""
```

```
df_1 = pd.read_excel('lista_stocks_presenti_al_genn98.xlsx')  
# df_1 = df_1.loc[df_1.BIG==1]  
df_1.Sector = pd.factorize(df_1.Sector)[0]  
tickers = list(df_1['ticker'])  
sector = list(df_1.Sector)
```

```
dizionario = dict(zip(tickers,sector)) # in questo dizionario ci sono i settori fattorizzati che verranno usati come communities per il network e per la misurazione
```

```
SP500_selezionati_a = pd.read_excel('quotazioni_storiche_sp500_mensile.xlsx')  
SP500_selezionati_a = pd.DataFrame(SP500_selezionati_a)  
SP500_selezionati_a = SP500_selezionati_a.set_index('Date',1)
```

```
""""#resampling""""
```

```
SP500_selezionati = SP500_selezionati_a.resample('M').last()
```

```
Indice_completo = pdr.get_data_yahoo('^GSPC', start = '1998-1-1')['Close']  
Indice_completo = pd.DataFrame(Indice_completo)  
Indice_completo = Indice_completo.resample('M').last()
```

```
Indice_completo.to_excel('sp.xlsx')
```

```
""""#Bug Fixing""""
```

```
colonne = list(SP500_selezionati.columns.values)  
SP500_selezionati = pd.DataFrame(SP500_selezionati, columns=colonne)
```

```
""""#Crea un DataFrame che contiene i log-rendimenti per il successivo calcolo delle correlazioni""""
```

```
df = SP500_selezionati = SP500_selezionati.resample('M').last()  
daily_close_px = df
```

```
#crea il dataframe  
df = pd.DataFrame(daily_close_px)  
#crea un dataframe con le variazioni percentuali per ogni unità di tempo sull' unità precedente  
dfb = np.log(df / df.shift(1))
```

```
""""#Imputa la lunghezza della finestra temporale su cui viene calcolata la matrice di correlazione: ogni finestra sarà un T dell' evoluzione del network""""
```

```
#parametri di input: permette di inserire la lunghezza della finestra temporale (in unità di tempo) e lo scalino del network  
print("Inserire i parametri!")  
a_ = input("lunghezza Finestra")  
cut_ = input("treshold")  
cut = float(cut_)
```

```

b_ = 0
size = dfb.index.size #parametro da usare in alternativa a d e c, discrezionali
#(d1 = size-a) #non utilizzato, eventualmente al posto di d con c = d1 e d = d1
import numpy
a = int(a_) #finestra
b = int(b_) #incrementa da 0 (o dal numero scelto) a finestra finale
c = size - a #matrici da generare
d = size - a #matrici da trasformare in graph

#####Genera le matrici di correlazione. Le matrici saranno pari ai T definiti in precedenza ed ogni
correlazione superiore alla treshold sarà un link del network#####

```

```

#listm sarà la lista che contiene le matrici di correlazione
listm = []
for b in range(c):
    nome = str(b)
    data = dfb[:(a + b)][b:]
    #print (data)
    correlations = data.corr()
    correlations
    listm.append(correlations)
print("verranno calcolate ", b, " finestre temporali")

```

```

#####Genera i networks statici che rappresentano lo stato del network dinamico nei vari T, calcola le
metriche e le inserisce in una lista che verrà poi inserita in un dataframe

```

Attenzione all' ultimo punto: Disegna il network di ogni finestra, ma ovviamnete rallenta infinitamente il codice, eventualmente mettere sotto commento in caso di networks con molti nodi

```

#####

```

```

#definiamo le liste che ospiteranno le metriche
lista = []
listb = []
listc = []
listd = []
liste = []
listf = []
listg = []
listnet=[]

```

```

for t in range(d):
    #NOTA 1 - questa istruzione è per salvare il plot delle singole finestre, eventualmente mettere
sotto commento
    graph_ = str(t)
    gra = int(graph_)
    t_ = int(t)
    #crea la adajacency matrix per ogni metriche di correlazione
    links = listm[gra].stack().reset_index()
    links.columns = ['var1', 'var2','value']
    links

```

```

#filtra i link (scalino)
links_filtered=links.loc[ (links['value'] > cut) & (links['var1'] != links['var2']) ]
links_filtered
links_filtered.drop_duplicates()
#genera il network
G=nx.from_pandas_edgelist(links_filtered, 'var1', 'var2')
listnet.append(G)

#alcune metriche che finiscono in altrettante liste - implementare altre:

#density a
Den = nx.density(G)
lista.append(Den)

#av clustering b
try:
    av_cl = nx.average_clustering(G)
    listb.append(av_cl)
except:
    listb.append(0)

#communities c - le communities sono rappresentate dai serrori di appartenenza delle stocks
comm = dizionario.values()
try:
    mod = max(comm)
    listc.append(mod)
except:
    listc.append(0)

# average sh path d. Qualora l'algorithmo rilevi che ci sono più networks non collegati tra di loro
la sh. path viene calcolata sul network con il maggior numero di nodi.

try:
    if nx.is_connected(G) ==True:
        av_sp = nx.average_shortest_path_length(G)
    else:
        components = nx.connected_components(G)
        largest_component = max(components, key=len)
        subgraph = G.subgraph(largest_component)
        av_sp = nx.average_shortest_path_length(subgraph)
except:
    av_sp = ("inf")
listd.append(av_sp)

#diameter. Qualora l'algorithmo rilevi che ci sono più networks non collegati tra di loro il diametro
viene calcolato sul network con il maggior numero di nodi.

try:
    if nx.is_connected(G) ==True:
        diameter = nx.diameter(G)
    else:

```

```

    components = nx.connected_components(G)
    largest_component = max(components, key=len)
    subgraph = G.subgraph(largest_component)
    diameter = nx.diameter(subgraph)
except:
    diameter=("inf")
liste.append(diameter)

#transitivity f

transitivity_ = nx.transitivity(G)
listf.append(transitivity_)

print("elaborata la finestra numero: ", t)

#modularity
#crea una partizione basata sul settore di appartenenza

fb = pd.DataFrame(tickers, columns = ['Symbol'])
fb['Sector']=sector
fb = fb.set_index('Symbol',1)

fa = pd.DataFrame(index=list(G.nodes))
fa = fa.join(fb)
fa = fa.reset_index()

n_communities = max(fa.Sector)
listaz=[]
listaliste=[]
for i in range (n_communities+1):
    df_comm = fa.loc[fa.Sector==i]
    listaz = list(df_comm['index'])
    listaliste.append(listaz)
    partition = listaliste
modularity = nx.algorithms.community.quality.modularity(G, partition)
listg.append(modularity)

### #NOTA 1 - disegna e salva l'immagine di ogni finestra EVENTUALMENTE ELIMINARE
# nx.draw(G, with_labels=True, node_color=list(comm), node_size=400, edge_color='red',
linewidths=0.1, font_size=10)
# plt.savefig(graph_)
# plt.close()

#disegna le serie storiche delle metriche
plt.figure(figsize = (15,8))
plt.plot(lista)
plt.plot(listb)
# plt.plot(listc)

```

```

plt.plot(listd)
plt.plot(liste)
plt.plot(listf)
plt.plot(listg)
plt.legend(['Density', 'Average Clustering', 'Average shortest path', 'Diameter', 'Network transitivity',
'Modularity'])
plt.show()

```

```

print("Network all' ultima finestra temporale")
# plt.figure(figsize=(10,10))
# nx.draw(G, with_labels=True, node_color='red', node_size=400, edge_color='red',
linewidths=0.1, font_size=10, cmap='Accent')
# plt.show()

```

```

#####Crea un DataFrame (export) dove verranno ospitate le metriche per l'export in foglio di
calcolo#####

```

```

export = pd.DataFrame(index = dfb.index[a:])

```

```

export['density']=lista
export['av_clustering']=listb
export['communities'] = listc
export['av shortest path'] = listd
export['diameter'] = liste
export['transitivity'] = listf
export['modularity']=listg

```

```

# aggiungiamo l'andamento di SP500 per avere un parametro su cui ragionare inseguito in base alle
metriche osservate

```

```

export['SP500_Price_Index']= Indice_completo.Close[a:]
export['log_rend_forward']= np.log(export.SP500_Price_Index.shift(-
1)/export.SP500_Price_Index.shift(0))
export['log_rend']= np.log(export.SP500_Price_Index.shift(0)/export.SP500_Price_Index.shift(1))
export['dev_std'] = export.log_rend.rolling(a).std()
export['roc6']= (export.SP500_Price_Index.shift(-6)/export.SP500_Price_Index.shift(0))

```

```

name = "Metriche__m"+str(a)+ "__"+str(cut)+".xlsx"

```

```

export.to_excel(name)

```

```
Indice_completo.to_excel('dati_azioni_16giu2020.xlsx')
```

```
""""Verifichiamo la correlazione di alcuni fattori""""
```

```
export.columns
```

```
data1 = pd.DataFrame(export)
```

```
fig = plt.subplots(figsize=(20,20))  
plt.title("Metrics VS Standard Deviation")
```

```
fig = plt.subplot(321)  
ax = sns.regplot(x="density", y="dev_std", data=data1)  
fig = plt.subplot(322)  
ax = sns.regplot(x="transitivity", y="dev_std", data=data1)  
fig = plt.subplot(323)  
ax = sns.regplot(x="av shortest path", y="dev_std", data=data1)  
fig = plt.subplot(324)  
ax = sns.regplot(x="modularity", y="dev_std", data=data1)
```

```
plt.show()
```

```
data1 = pd.DataFrame(export)
```

```
fig = plt.subplots(figsize=(20,10))  
fig = plt.subplot(221)  
ax = sns.regplot(x="density", y="log_rend", data=data1)  
fig = plt.subplot(222)  
ax = sns.regplot(x="transitivity", y="log_rend", data=data1)  
fig = plt.subplot(223)  
ax = sns.regplot(x="av shortest path", y="log_rend", data=data1)  
fig = plt.subplot(224)  
ax = sns.regplot(x="modularity", y="log_rend", data=data1)  
plt.show()
```

```
data1 = pd.DataFrame(export)
```

```
fig = plt.subplots(figsize=(20,10))  
fig = plt.subplot(221)  
ax = sns.regplot(x="density", y="roc6", data=data1)  
fig = plt.subplot(222)  
ax = sns.regplot(x="transitivity", y="roc6", data=data1)  
fig = plt.subplot(223)  
ax = sns.regplot(x="av shortest path", y="roc6", data=data1)  
fig = plt.subplot(224)  
ax = sns.regplot(x="modularity", y="roc6", data=data1)  
plt.show()
```

```
""""Parrebbe che le metriche del network presentino una buona correlazione con la standard deviation dei log-rendimenti e nessuna correlazione apprezzabile con i log rendimenti
```

```

#Plotta alcune finestre
"""

lista_finestre=[0, 134, 250]

for finestra in lista_finestre:

    df_coloring_1 = pd.DataFrame(index = tickers)
    df_coloring_1['colore']=sector
    df_nodi = pd.DataFrame(index=listnet[finestra].nodes)
    df_coloring = df_nodi.join(df_coloring_1)

    fig = plt.subplots(figsize=(20,20))
    fig = plt.subplot(221)
    nx.draw(listnet[finestra], with_labels=False, node_color=list(df_coloring.colore), node_size=100,
    edge_color='grey', linewidths=0.1, font_size=10, cmap='Spectral')
    A = ("network della finestra "+str(df.index.values[finestra])+" - density=
"+str(round(export.density[finestra],2))+" - modularity=
"+str(round(export.modularity[finestra],2)))

    plt.title(A)

"""#Plotta una legenda per i settori"""

url = "https://raw.githubusercontent.com/datasets/s-and-p-500-companies/master/data/constituents.csv"
df_1 = pd.read_csv(url)
df_1['Sector_factorized'] = pd.factorize(df_1.Sector)[0]

lista_controllo = []
lista_1 = []
for i in range (len(df_1)):
    A = df_1.Sector_factorized[i]
    if A in lista_controllo:
        A = A
        lista_1.append(0)
    else:
        lista_controllo.append(A)
        lista_1.append(1)

df_1['chek']=lista_1

df_legenda = df_1.loc[df_1.chek==1]
df_legenda = df_legenda.drop(['Symbol', 'Name', 'chek'],1)

df_legenda.style.background_gradient(cmap='Spectral')

export['modularity_MA']=export['modularity'].rolling(12).mean()
plt.figure(figsize=(20,10))
export.SP500_Price_Index.plot(legend=True)

```

```

export.modularity.plot(secondary_y=True, linewidth=0.1, legend=True)
export.modularity_MA.plot(secondary_y=True, legend=True)
plt.show()

```

```

export['density_MA']=export.density.rolling(12).mean()
plt.figure(figsize=(20,10))
export.SP500_Price_Index.plot(legend=True)
export.density.plot(secondary_y=True, linewidth=0.15, legend=True, color='red')
export.density_MA.plot(secondary_y=True, legend=True, color='green')
plt.show()

```

```

export['Shortest_p_MA']=export['av shortest path'].rolling(12).mean()
plt.figure(figsize=(20,10))
export.SP500_Price_Index.plot(legend=True)
export['av shortest path'].plot(secondary_y=True, linewidth=0.15, legend=True, color='red')
export.Shortest_p_MA.plot(secondary_y=True, legend=True, color='green')
plt.show()

```

```

tendenze = pd.DataFrame(index=export.index)
tendenze['%change 12 m (trend) SP500'] =
np.log(export.SP500_Price_Index/export.SP500_Price_Index.shift(12))
tendenze['%change 12 m (trend) modularity'] =
np.log(export.modularity_MA/export.modularity_MA.shift(12))
plt.figure(figsize=(20,10))
plt.title("Trend MA modularity VS trend SP500 price", fontsize=20)
ax = sns.regplot(x="%change 12 m (trend) SP500", y="%change 12 m (trend) modularity",
data=tendenze)

```

```

from scipy import stats
tendenze = pd.DataFrame(index=export.index)
tendenze['%change 12 m (trend) SP500 Forw'] = np.log(export.SP500_Price_Index.shift(-
1)/export.SP500_Price_Index.shift(11))
tendenze['%change 12 m (trend) modularity'] =
np.log(export.modularity_MA/export.modularity_MA.shift(12))
tendenze = tendenze.dropna()
plt.figure(figsize=(20,10))
plt.title("Trend MA modularity VS trend SP500 price forward1", fontsize=20)
slope, intercept, r_value, p_value, std_err = stats.linregress(tendenze['%change 12 m (trend) SP500
Forw'],tendenze['%change 12 m (trend) modularity'])
ax = sns.regplot(x="%change 12 m (trend) SP500 Forw", y="%change 12 m (trend) modularity",
data=tendenze)

```

```

plt.show()

```

```

r_value

```

```

stats.linregress()

```

```

export

```

```

export['modularity_MA']=export['modularity'].rolling(12).mean()

```

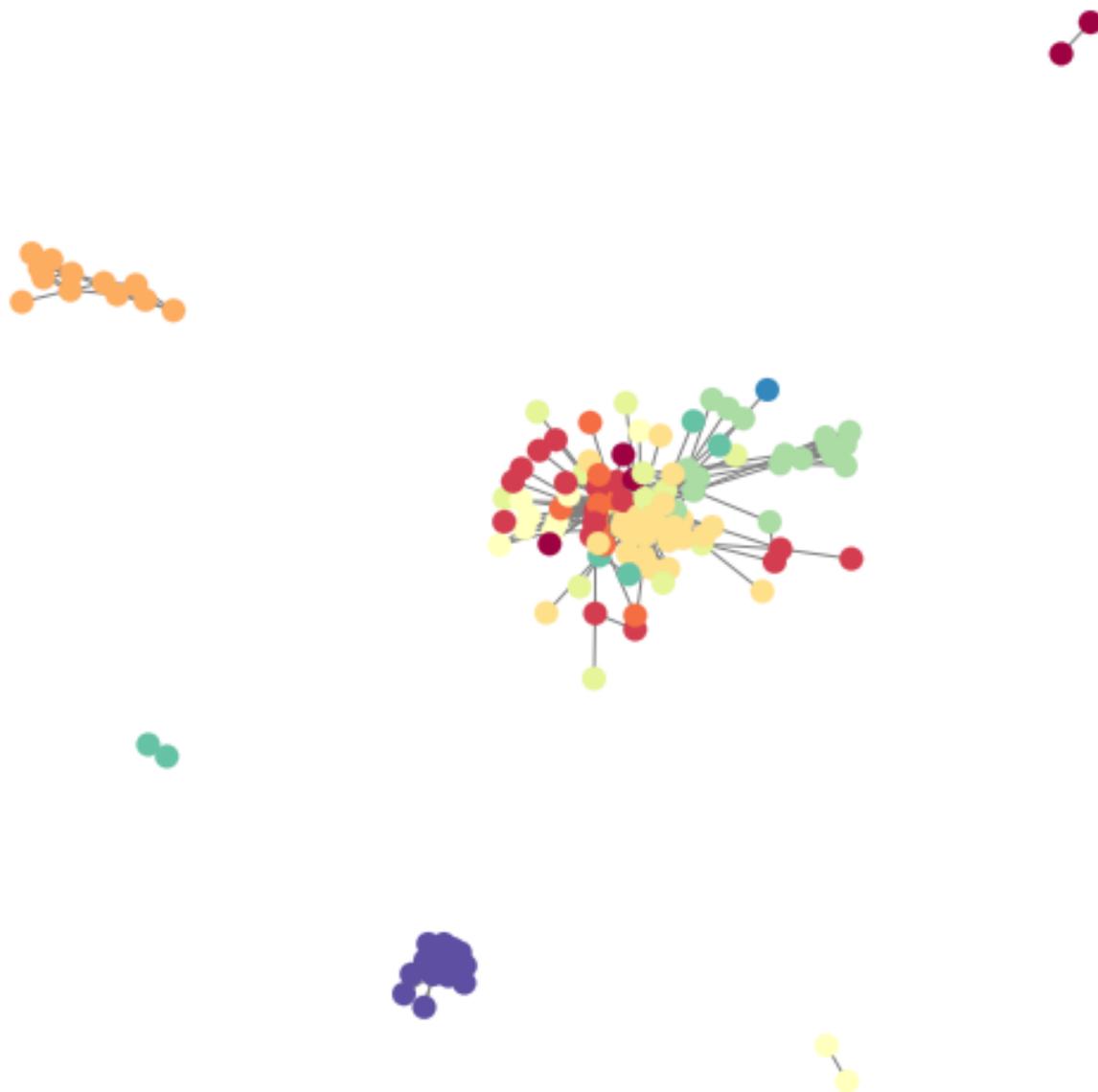
```
export['ST_deviation_MA']=export['dev_std'].rolling(12).mean()
plt.figure(figsize=(20,10))
export.ST_deviation_MA.plot(legend=True)
export.modularity.plot(secondary_y=True, linewidth=0.1, legend=True)
export.modularity_MA.plot(secondary_y=True, legend=True)
plt.show()
```

```
export['modularity_MA']=export['modularity'].rolling(12).mean()
export['ST_deviation_MA']=export['dev_std'].rolling(12).mean()
plt.figure(figsize=(20,10))
export.SP500_Price_Index.plot(legend=True)
export.modularity.plot(secondary_y=True, linewidth=0.1, legend=True)
export.modularity_MA.plot(secondary_y=True, legend=True)
plt.show()
```

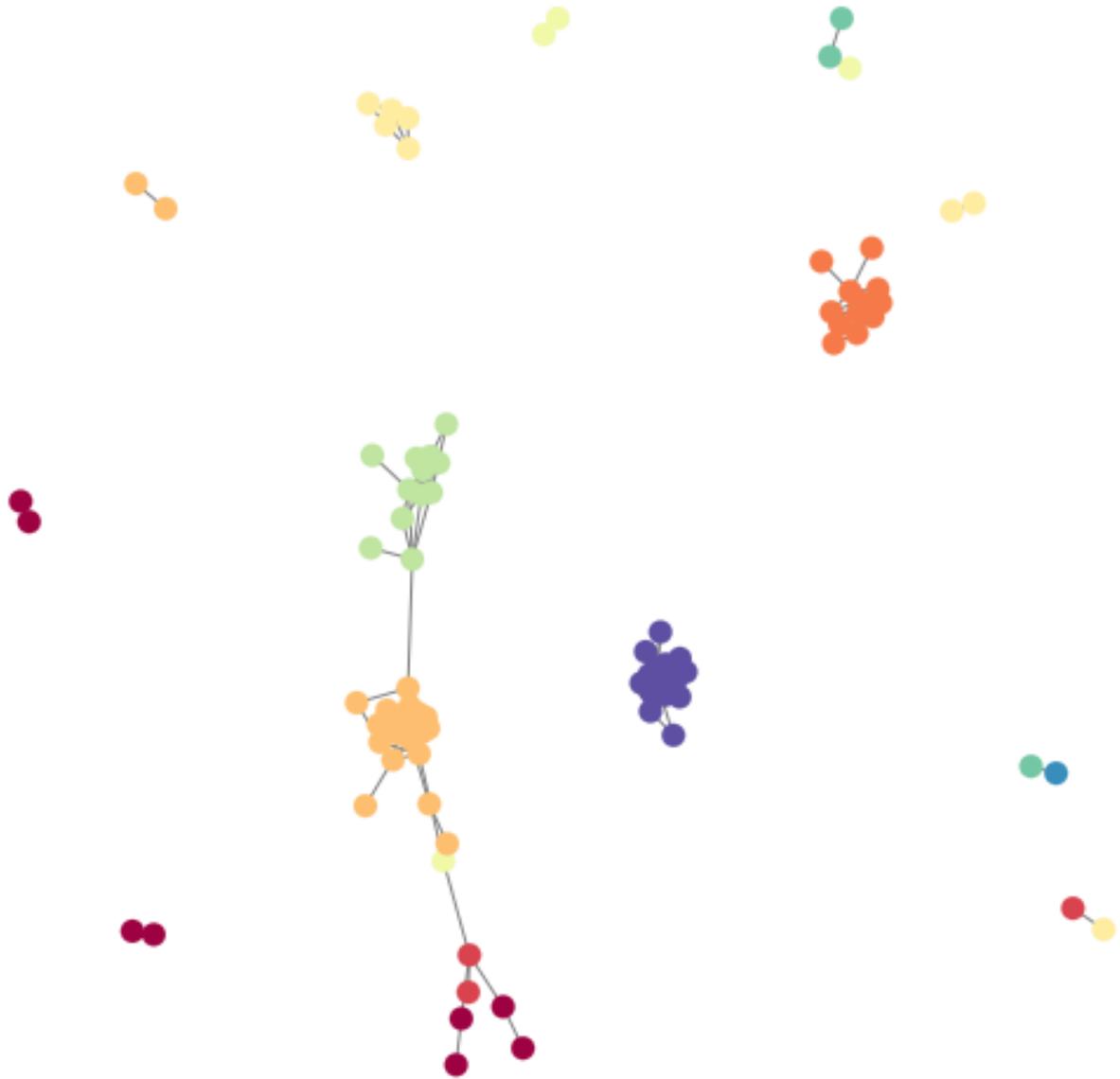
```
export['modularity_MA']=export['modularity'].rolling(12).mean()
export['ST_deviation_MA']=export['dev_std'].rolling(12).mean()
plt.figure(figsize=(20,10))
export.SP500_Price_Index.plot(legend=True)
export.modularity.plot(secondary_y=True, linewidth=0.1, legend=True)
export.modularity_MA.plot(secondary_y=True, legend=True)
plt.show()
```

It follows a polt for each network from 1998 to 2020:

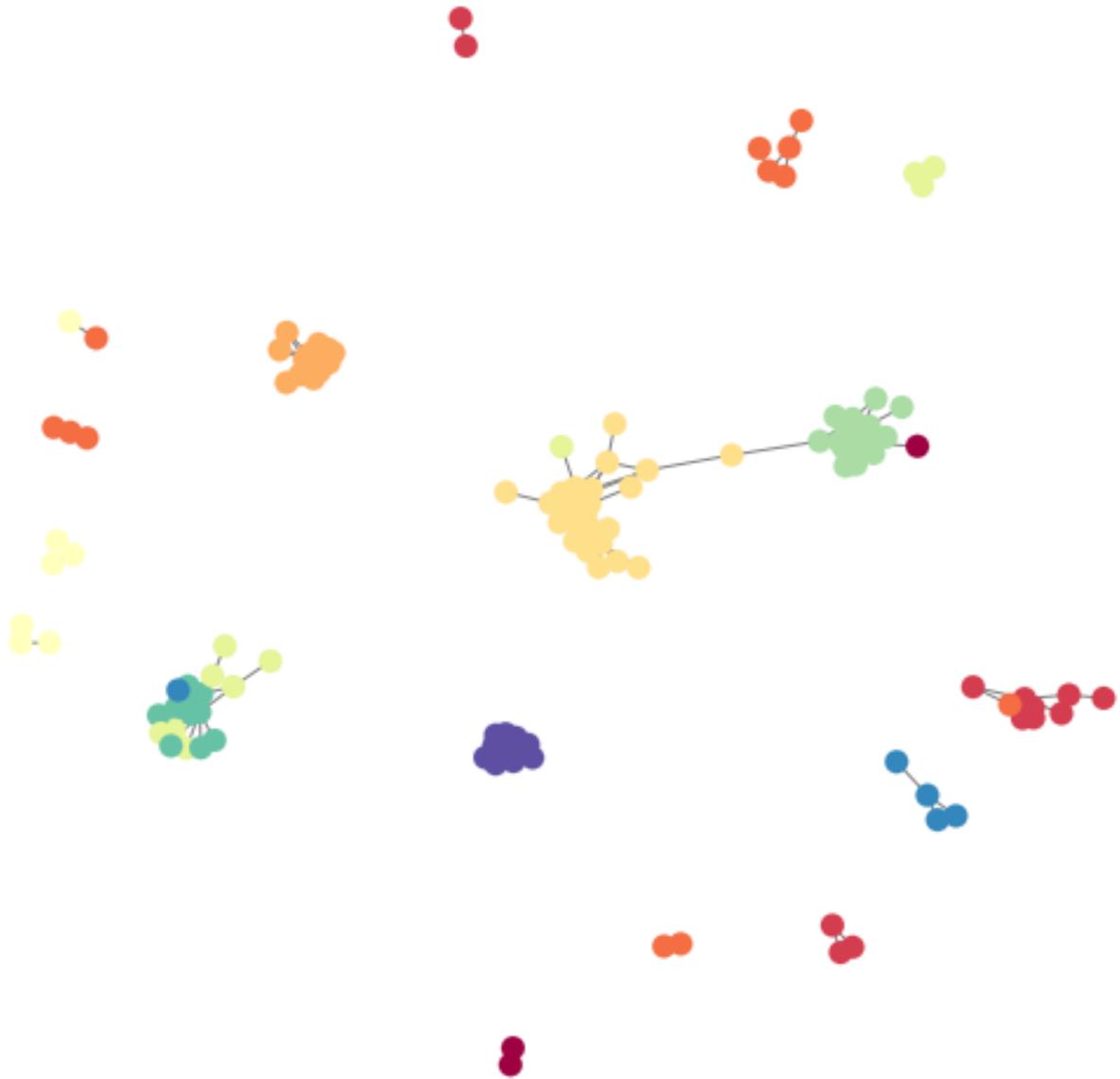
network della finestra 1998 - density= 0.06 - modularity= 0.73



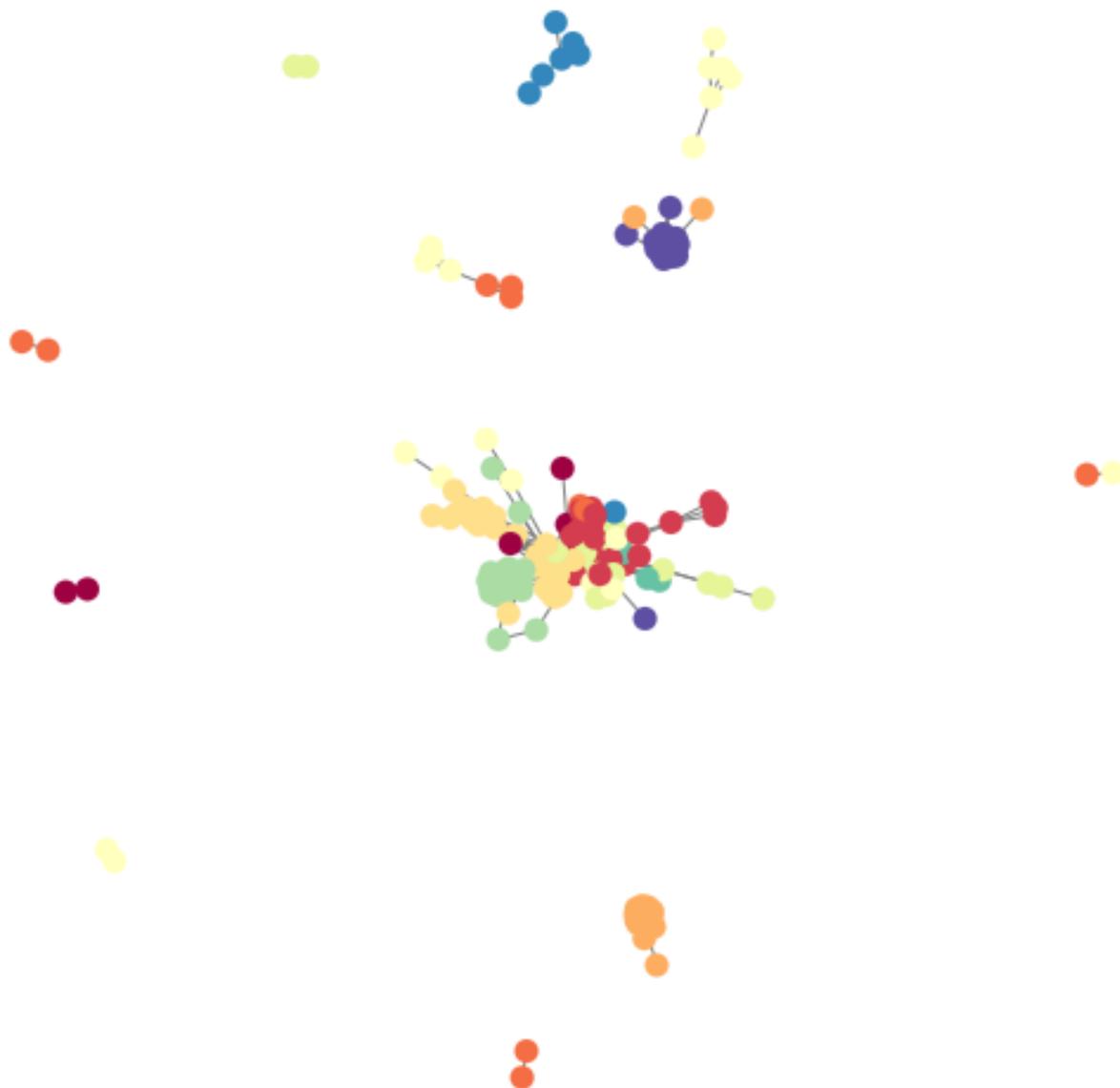
network della finestra 1999 - density= 0.07 - modularity= 0.75



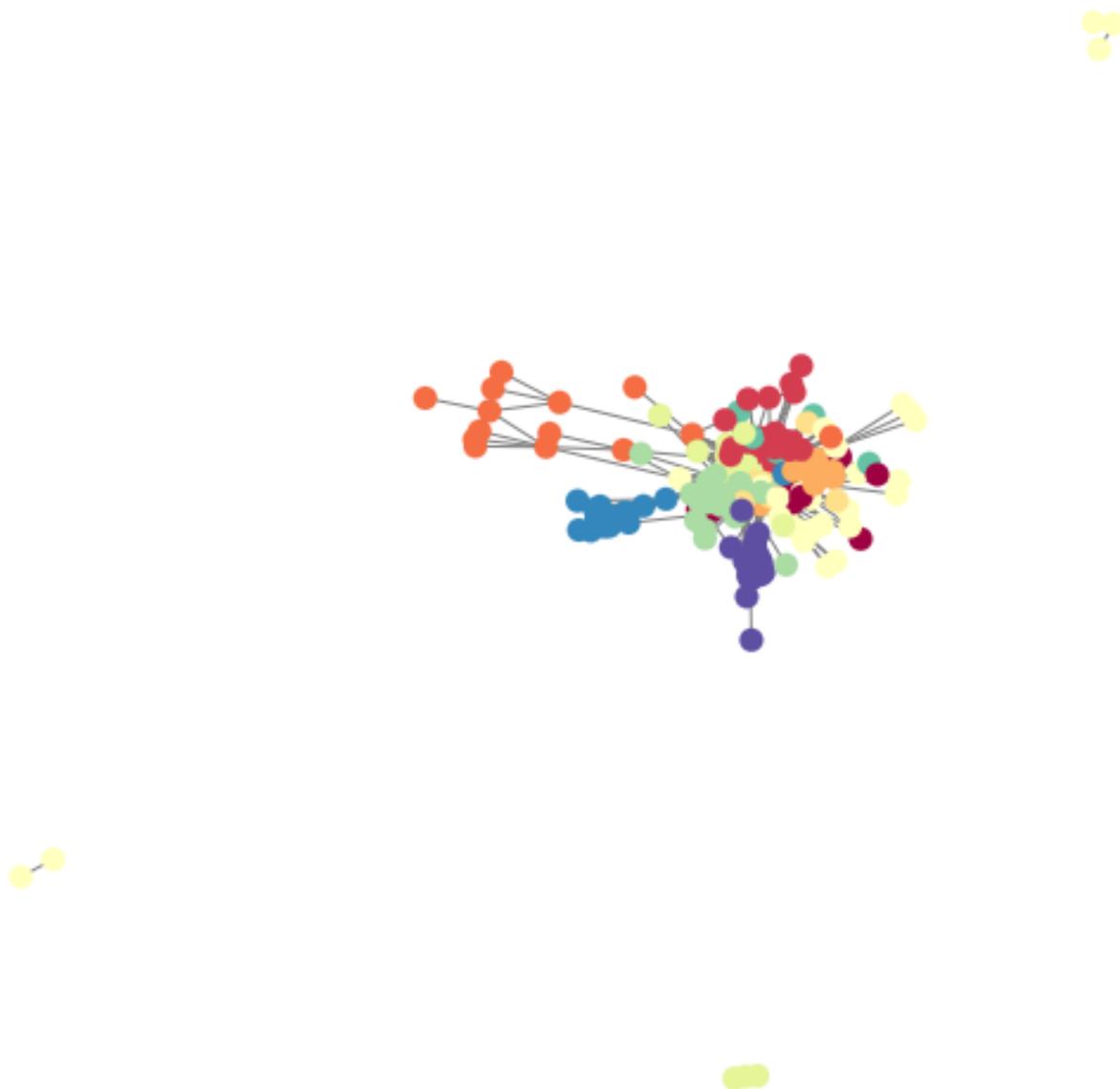
network della finestra 2000 - density= 0.05 - modularity= 0.73



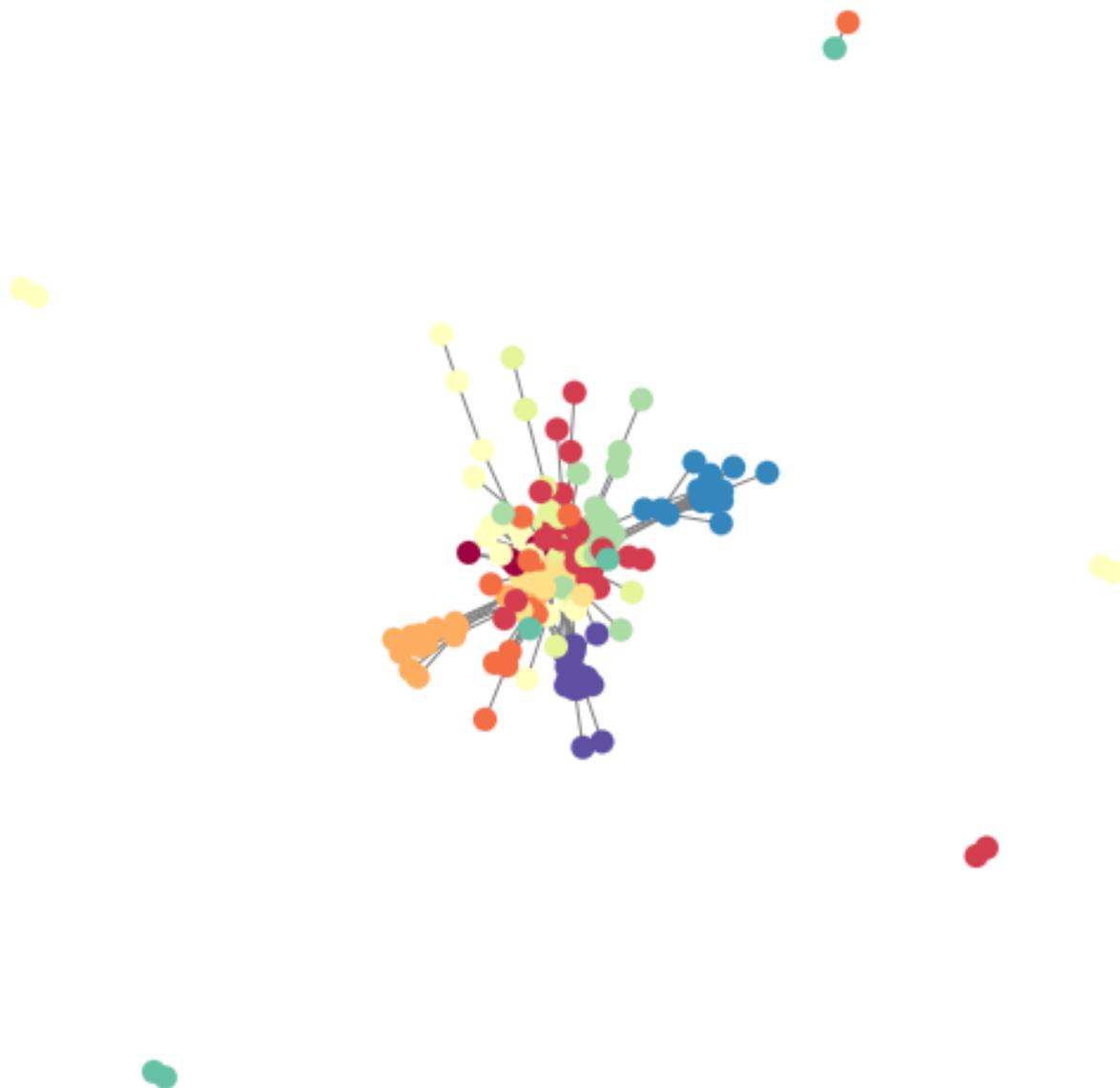
network della finestra 2001 - density= 0.07 - modularity= 0.49



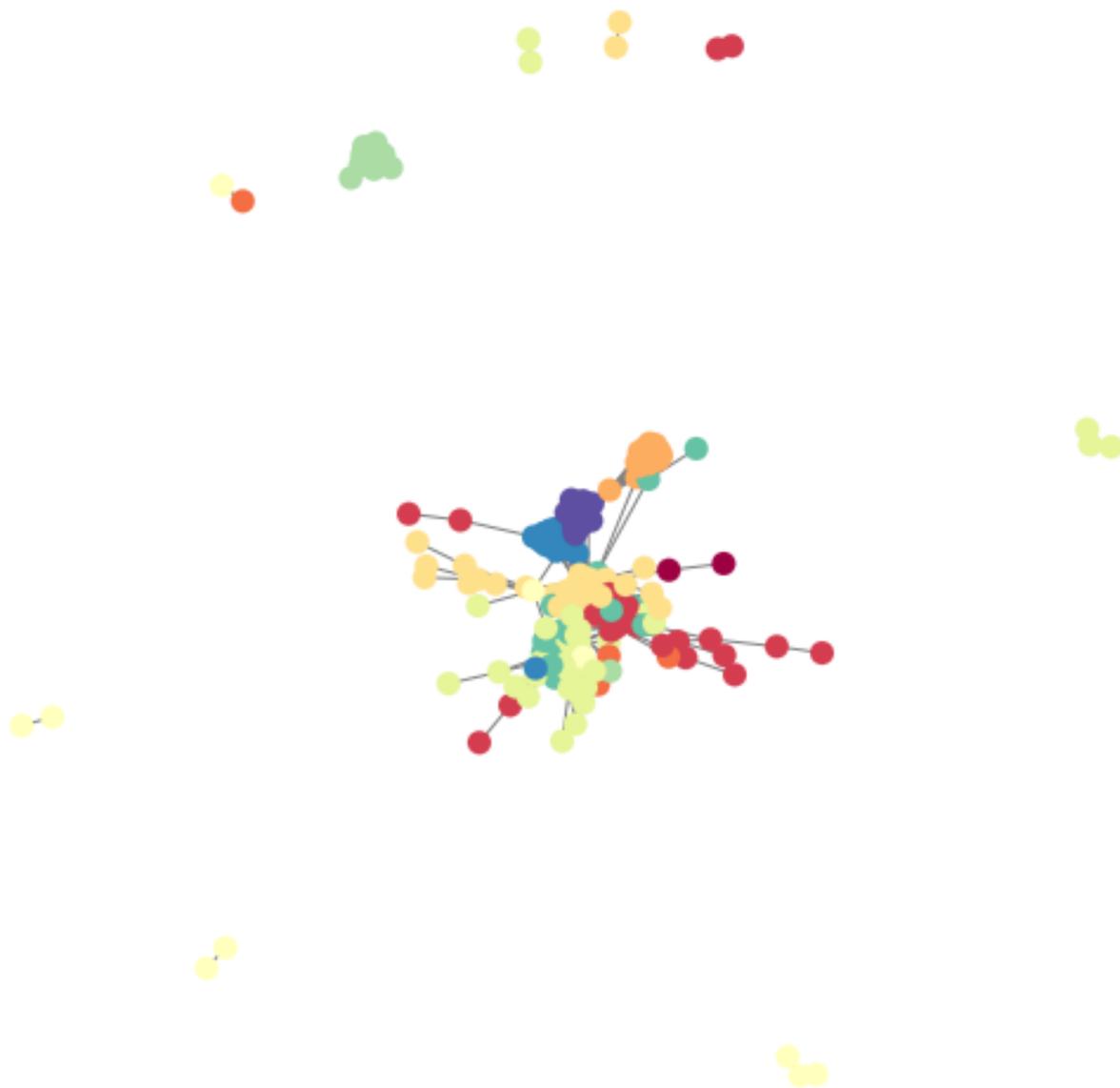
network della finestra 2002 - density= 0.14 - modularity= 0.17



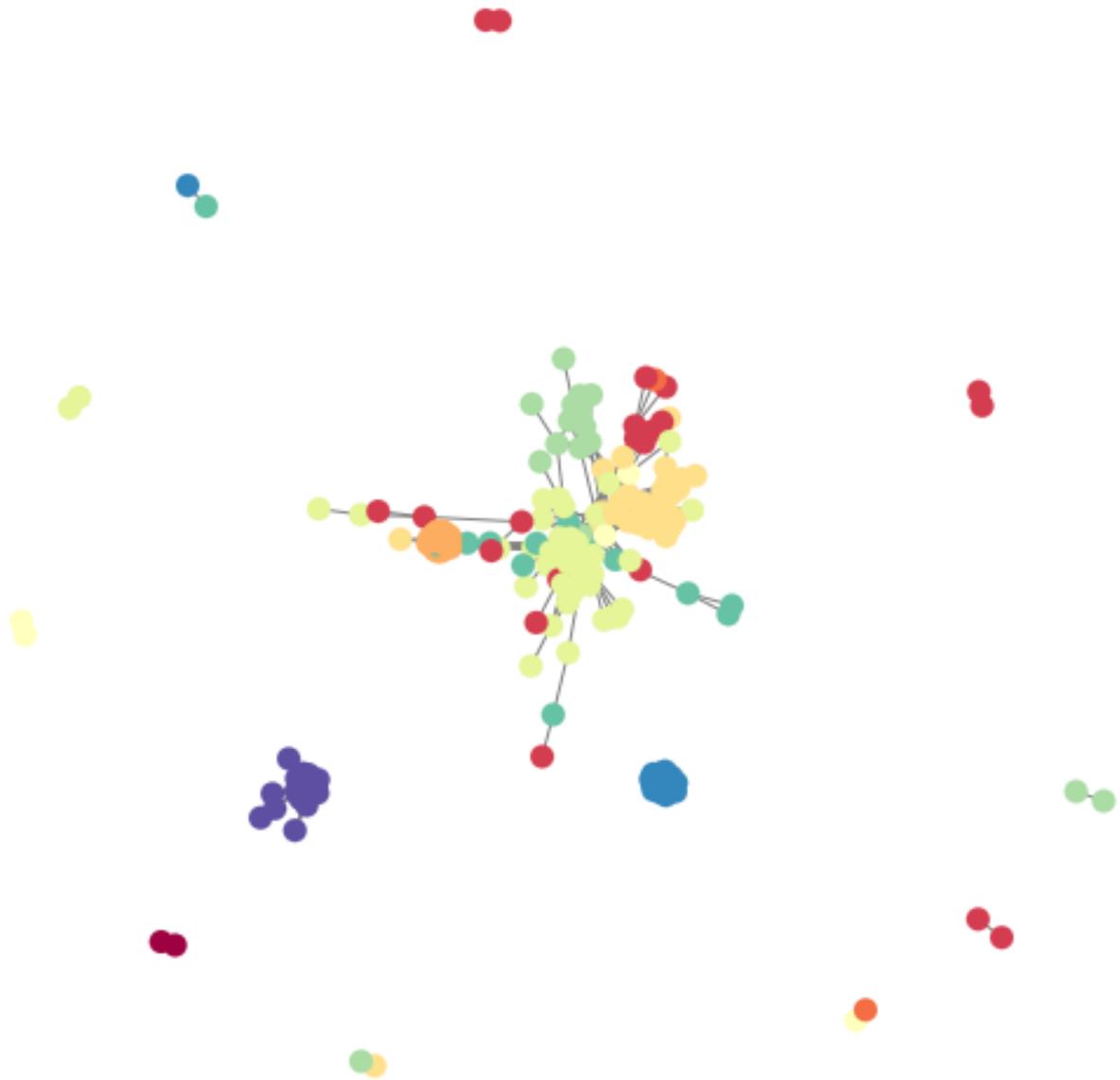
network della finestra 2003 - density= 0.11 - modularity= 0.16



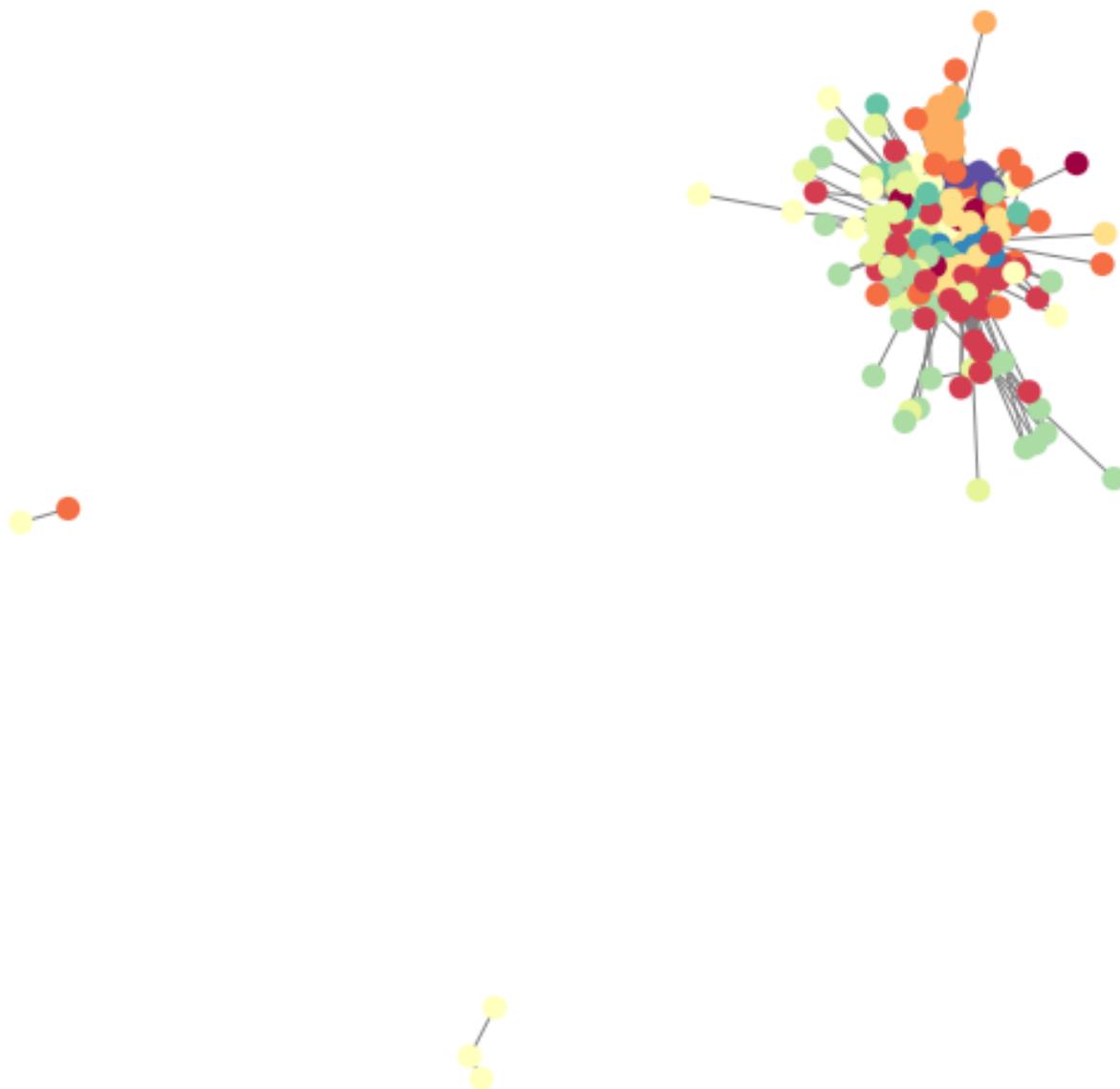
network della finestra 2005 - density= 0.06 - modularity= 0.59



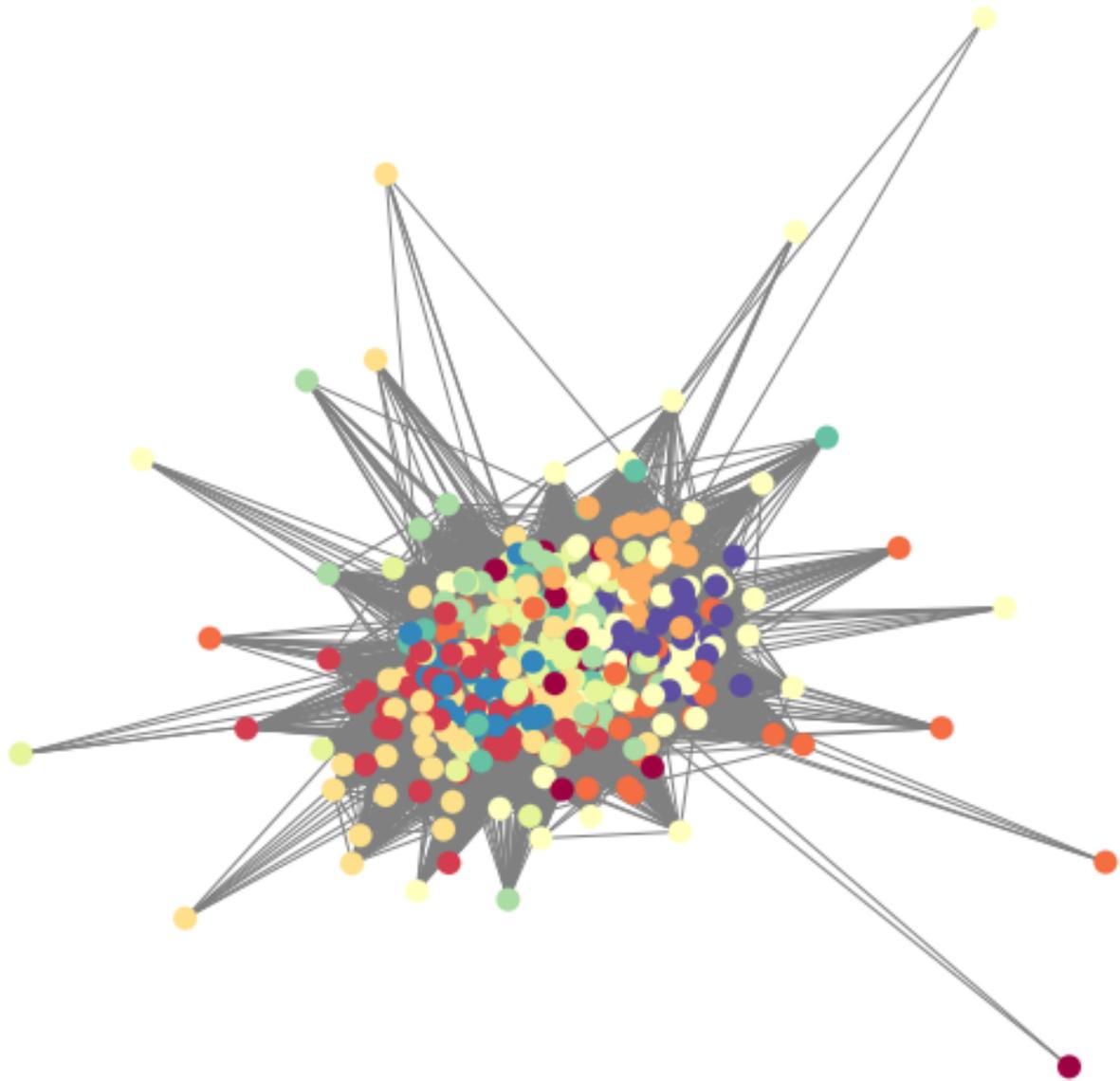
network della finestra 2006 - density= 0.05 - modularity= 0.69



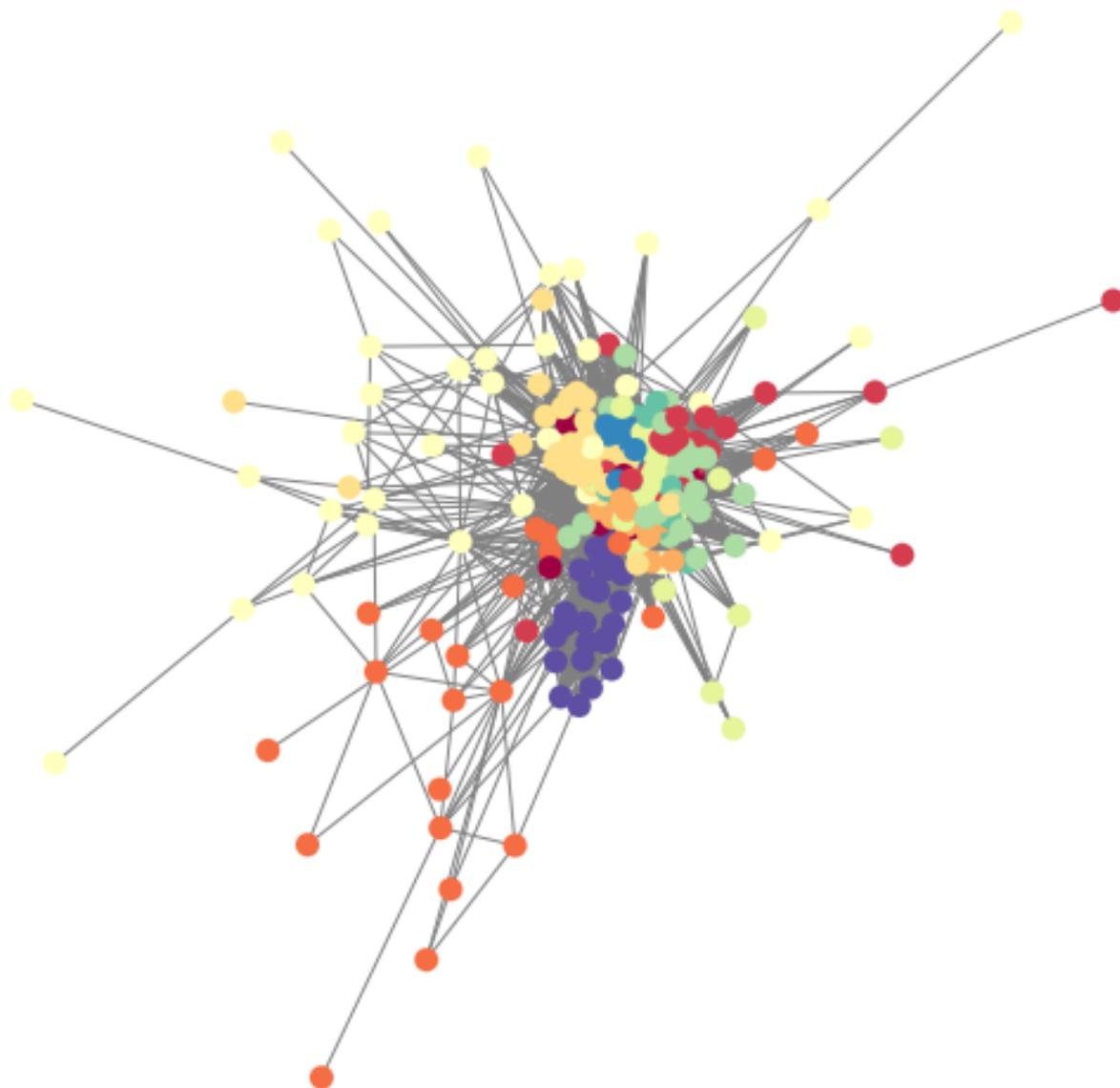
network della finestra 2007 - density= 0.14 - modularity= 0.15



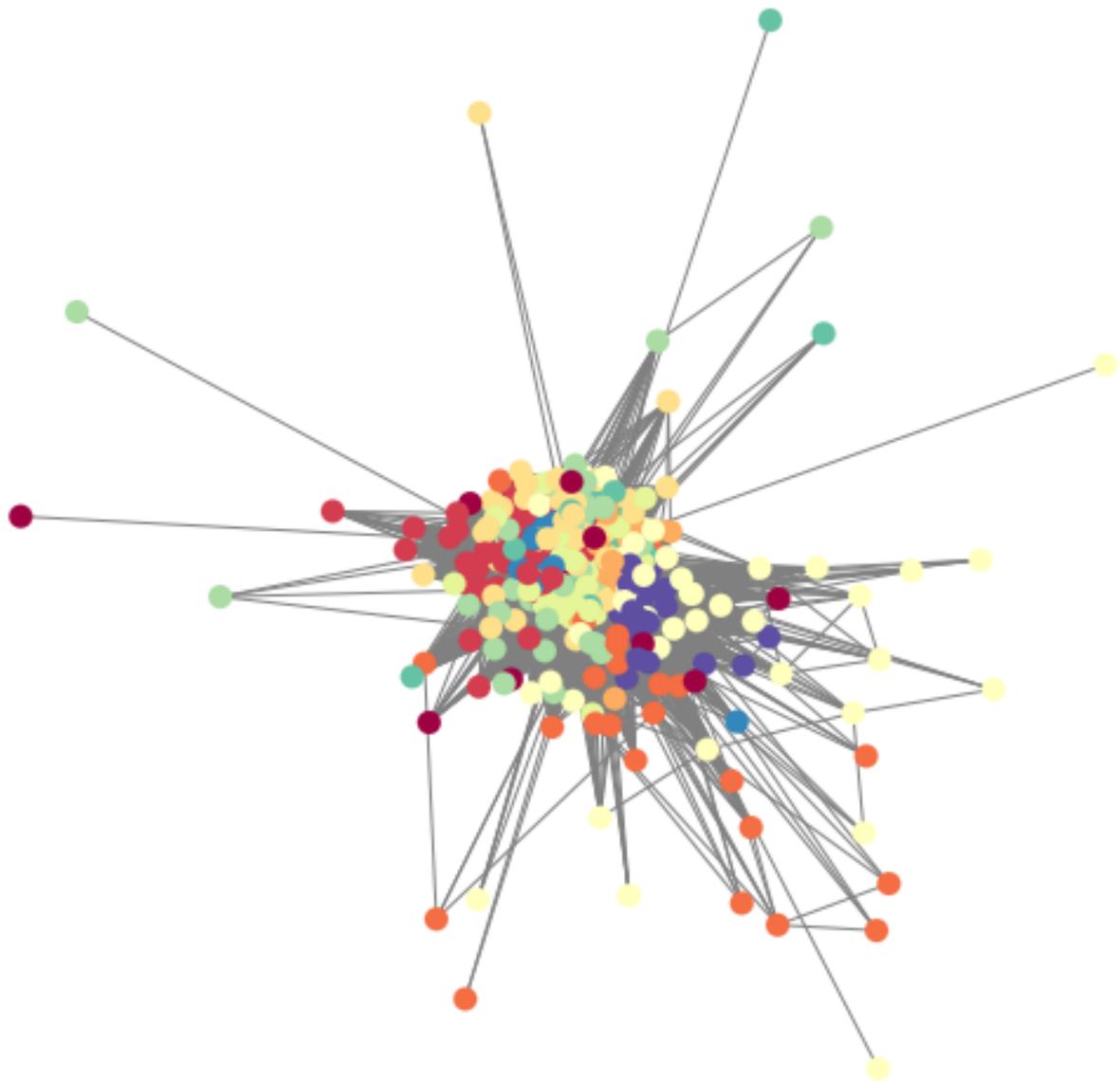
network della finestra 2008 - density= 0.6 - modularity= 0.03



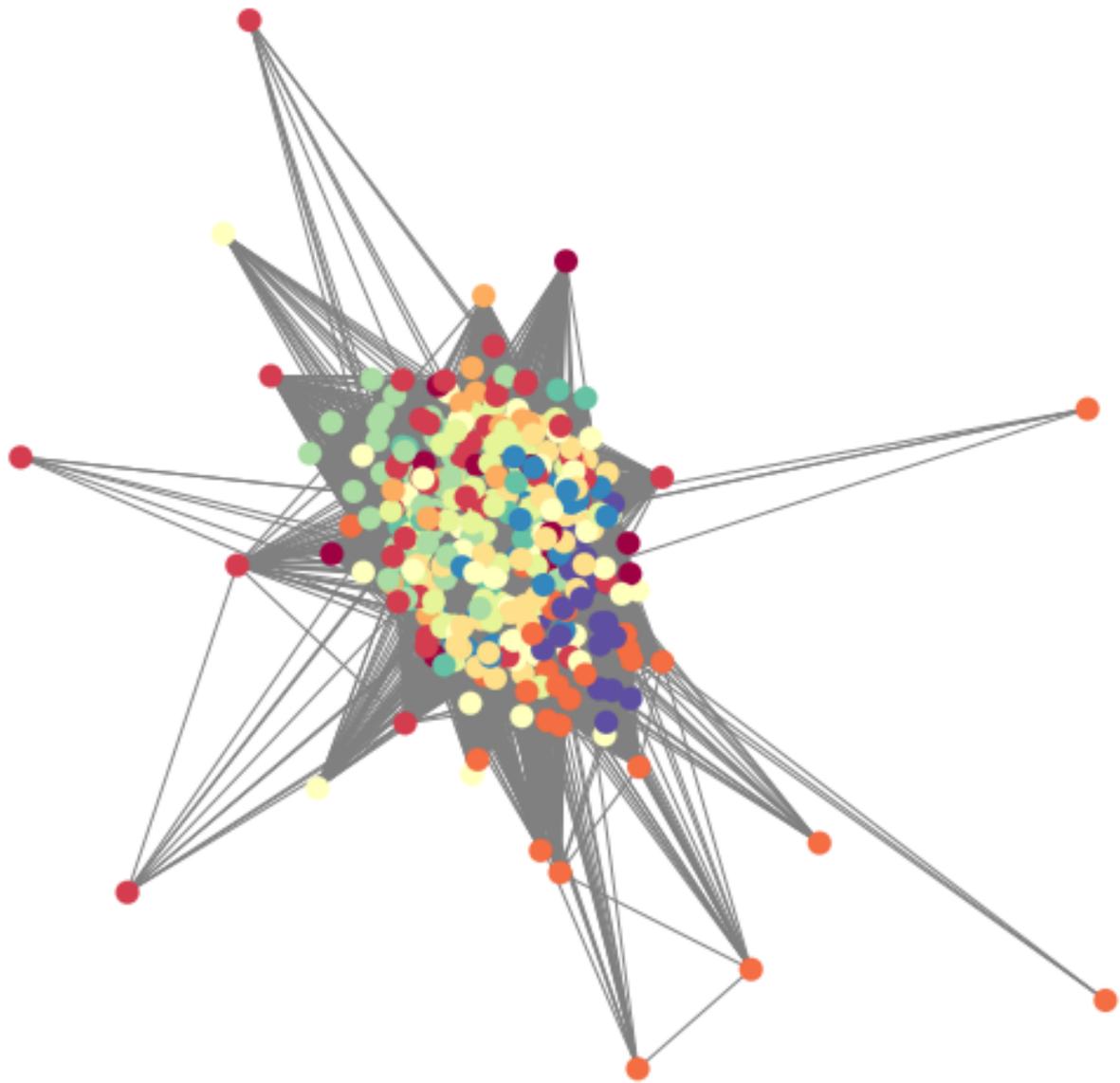
network della finestra 2009 - density= 0.37 - modularity= 0.05



network della finestra 2010 - density= 0.5 - modularity= 0.02



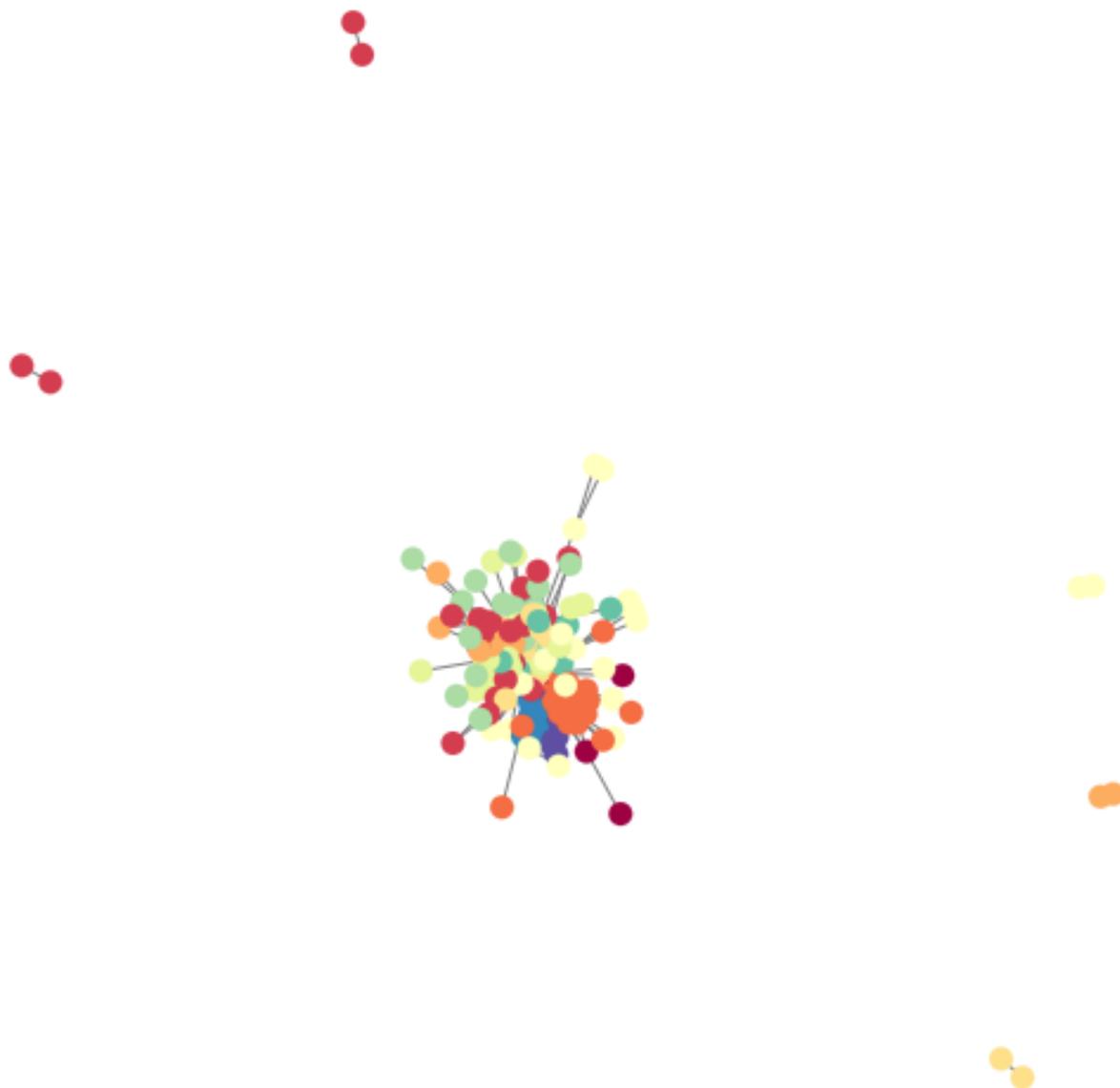
network della finestra 2011 - density= 0.8 - modularity= 0.0



network della finestra 2012 - density= 0.16 - modularity= 0.16



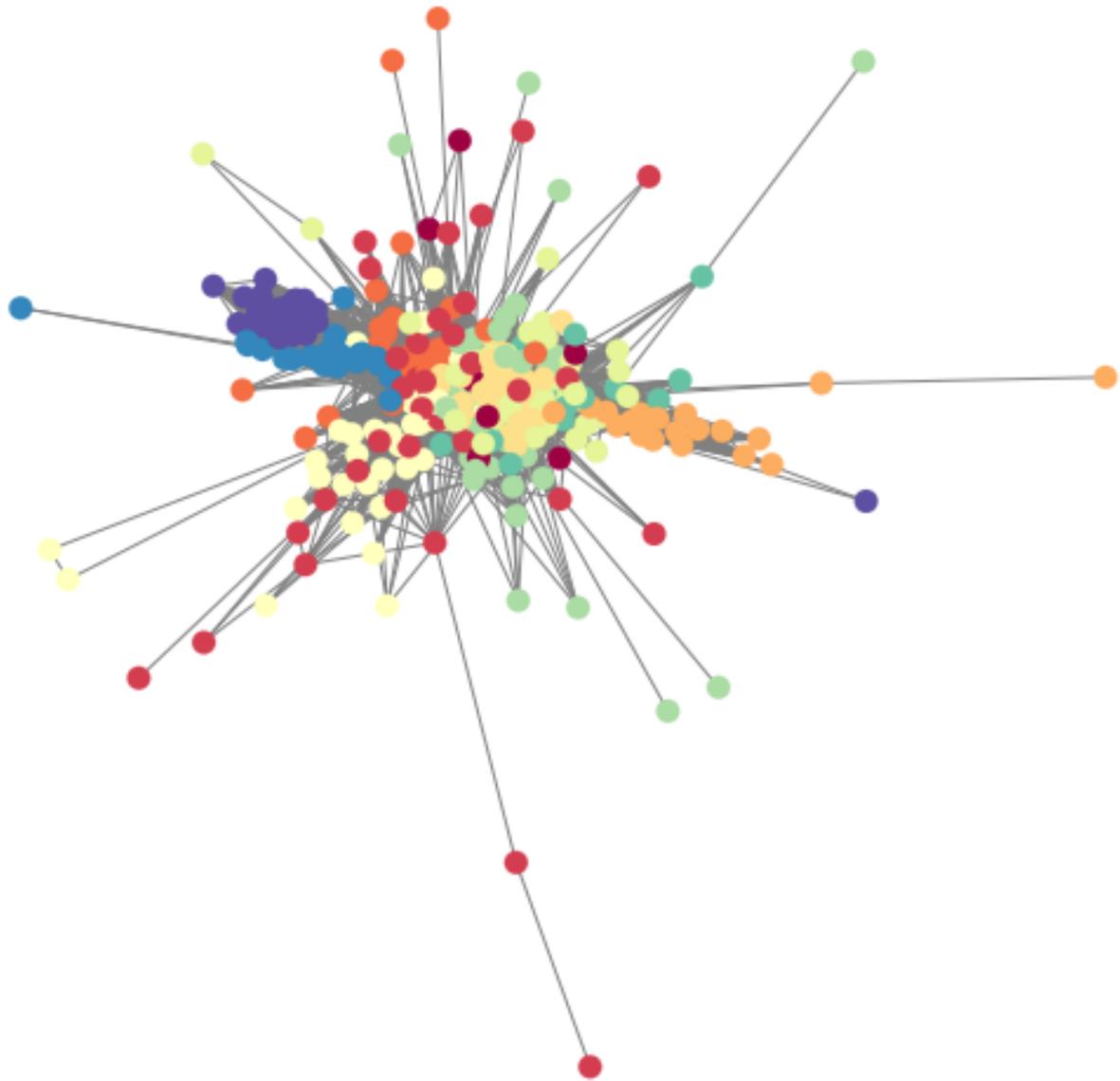
network della finestra 2013 - density= 0.13 - modularity= 0.21



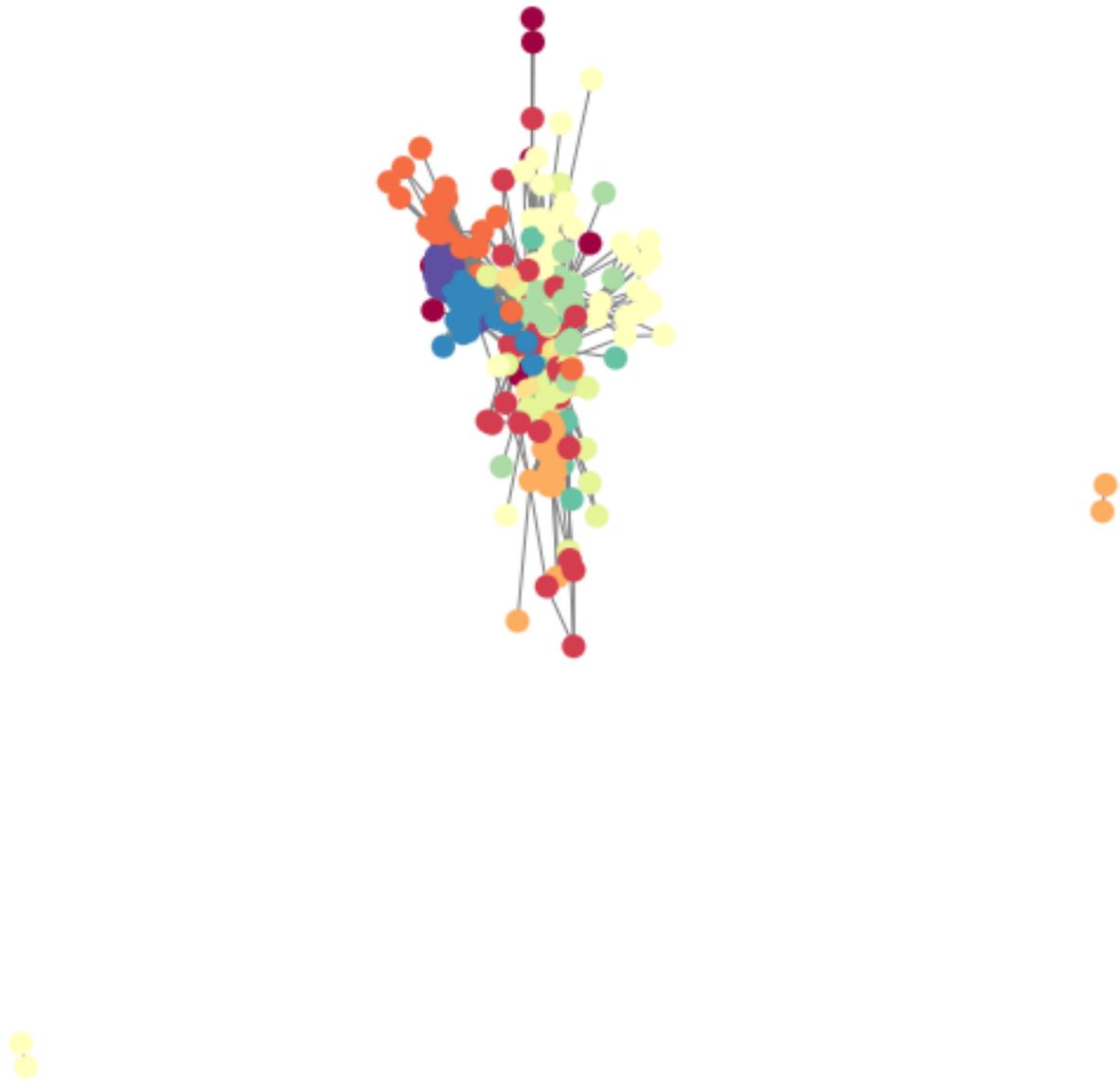
network della finestra 2014 - density= 0.12 - modularity= 0.22



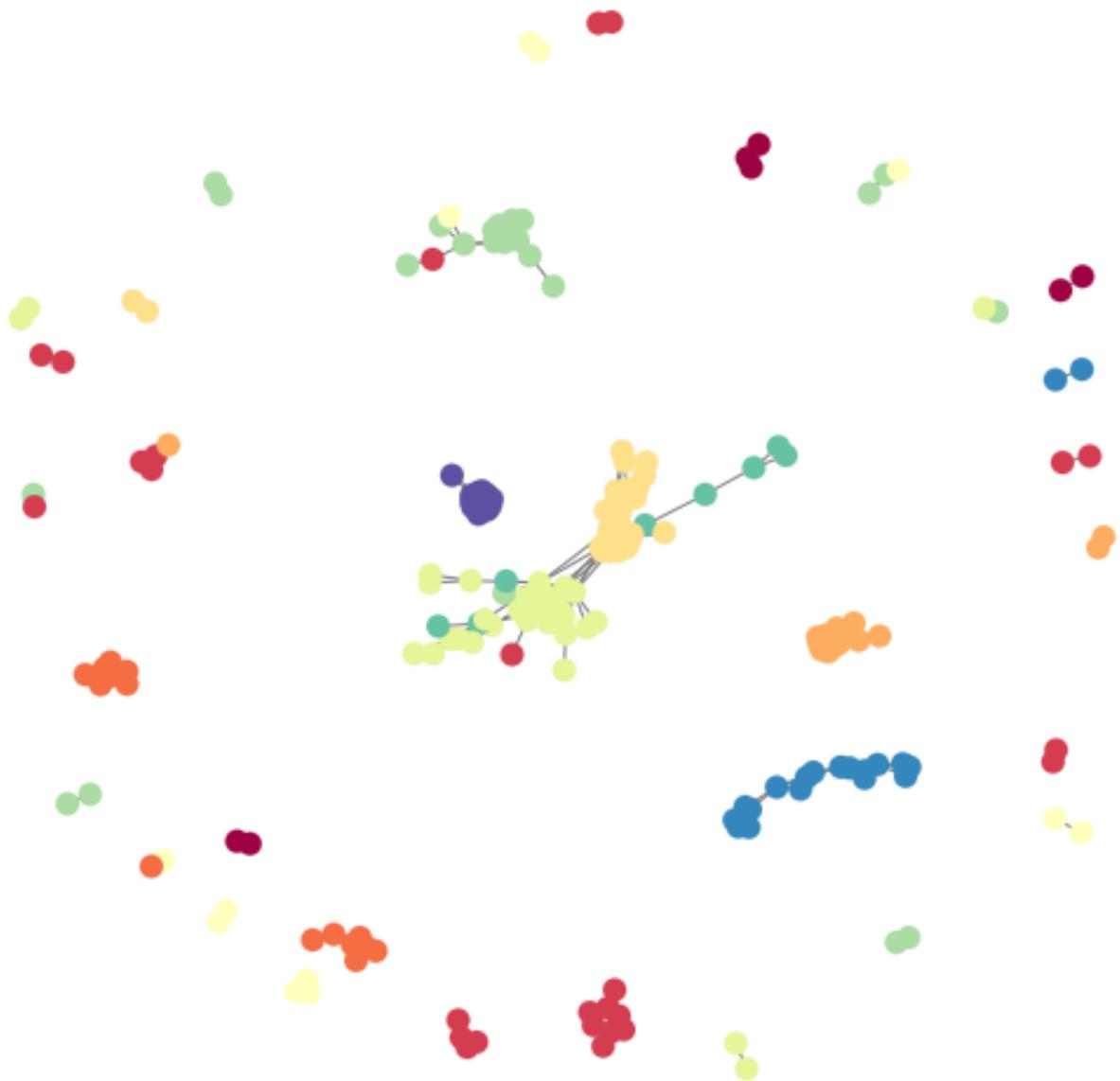
network della finestra 2015 - density= 0.25 - modularity= 0.1



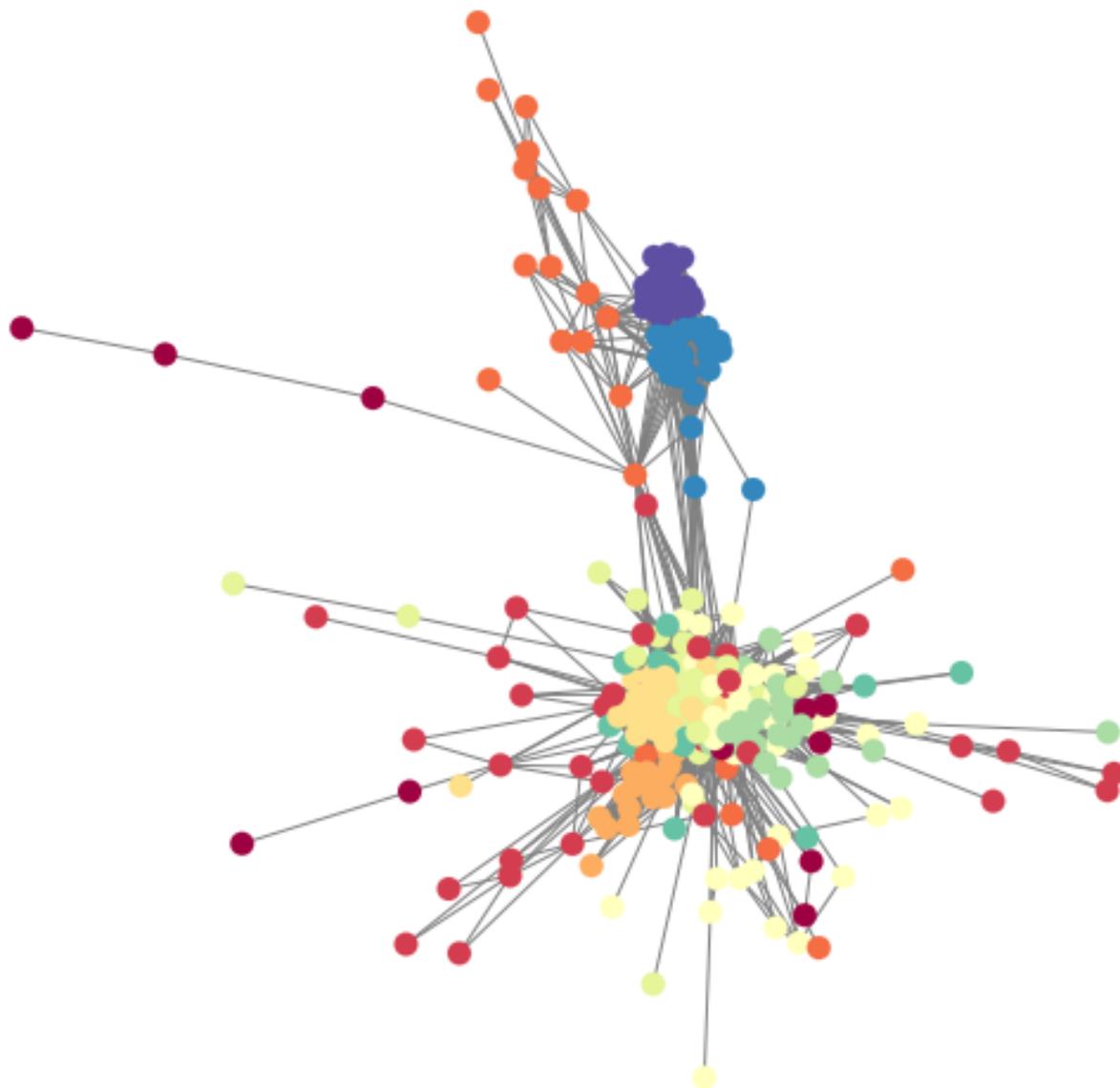
network della finestra 2016 - density= 0.13 - modularity= 0.2



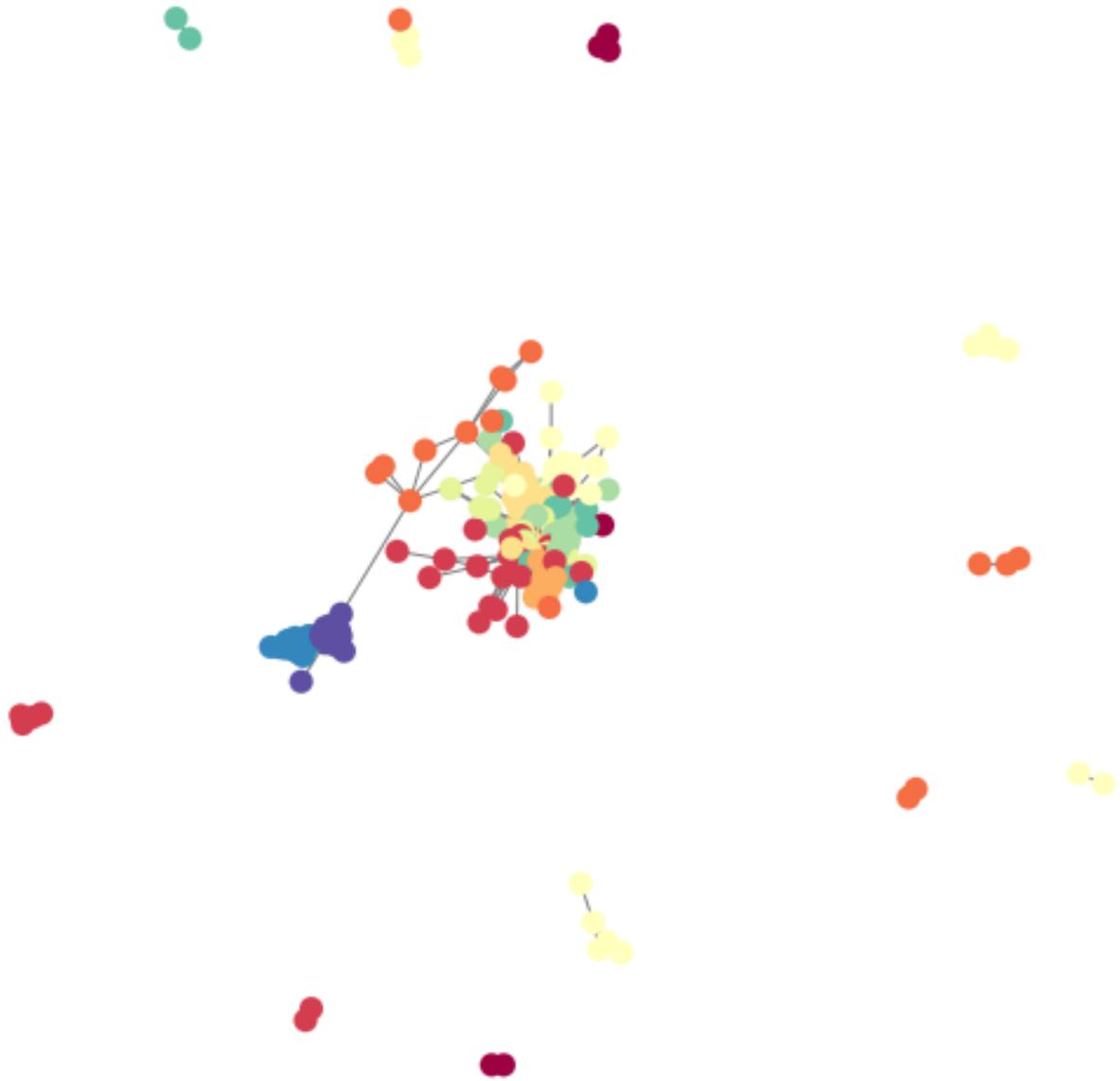
network della finestra 2017 - density= 0.04 - modularity= 0.68



network della finestra 2018 - density= 0.15 - modularity= 0.21



network della finestra 2019 - density= 0.09 - modularity= 0.27



network della finestra 2020 - density= 0.8 - modularity= 0.01

