entropy

MDPI

Article

Information Technology Project Portfolio Implementation Process Optimization Based on Complex Network Theory and Entropy

Qin Wang [1,2,*], Guangping Zeng [1] and Xuyan Tu [1]

1 School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China; zgp@ustb.edu.cn (G.Z.); tuxuyan@126.com (X.T.)
2 Information Technology Department, Bank of China, Beijing 100818, China
* Correspondence: kj.wangqin@gmail.com

Academic Editors: António M. Lopes and J. A. Tenreiro Machado
Received: 4 April 2017; Accepted: 13 June 2017; Published: 19 June 2017

Abstract: In traditional information technology project portfolio management (ITPPM), managers often pay more attention to the optimization of portfolio selection in the initial stage. In fact, during the portfolio implementation process, there are still issues to be optimized. Organizing cooperation will enhance the efficiency, although it brings more immediate risk due to the complex variety of links between projects. In order to balance the efficiency and risk, an optimization method is presented based on the complex network theory and entropy, which will assist portfolio managers in recognizing the structure of the portfolio and determine the cooperation range. Firstly, a complex network model for an IT project portfolio is constructed, in which the project is simulated as an artificial life agent. At the same time, the portfolio is viewed as a small scale of society. Following this, social network analysis is used to detect and divide communities in order to estimate the roles of projects between different portfolios. Based on these, the efficiency and the risk are measured using entropy and are balanced through searching for adequate hierarchy community divisions. Thus, the activities of cooperation in organizations, risk management, and so on—which are usually viewed as an important art—can be discussed and conducted based on quantity calculations.

Keywords: project portfolio management (PPM); complex network theory; social network analysis; information theory; entropy; cooperation efficiency; risk control; efficiency–risk balance

1. Introduction

The demand for information technology (IT) systems has increased, resulting in enterprises needing to improve efficiency, productivity, and profit. Successful projects save time and budget, while maintaining high quality and enhancing customer satisfaction [1]. However, the failure rates of IT projects have been consistently high for many years. In 1995, the Standish Group provided a report that showed about 31% of software projects were canceled before completion, while more than half of projects overran their budget or were unable to meet the required schedule [2]. After a decade, enterprises are still losing money on failing projects. From 2004 to 2012, only about one-third of the projects were successfully completed on time and within the allocated budget [3]. In Harvard Business Review reports, "one sixth of IT projects had an average cost overrun of 200% and a schedule overrun of 70%" [4]. The United States economy loses \$50–150 billion per year due to failed IT projects, according to the Gallup Business Review [5].

Researchers have conducted analyses to determine the factors contributing to the success or failure of projects. The common reasons are due to the dynamics, competitive environment, difficulties in forecasting future scenarios, lack of information, inadequate resources allocated, non-performing project teams, insufficient risk management analysis, lack of corporate culture, lack of top management involvement, planning, execution, and so on [6].

The ultimate goal of successful projects is to achieve the mission and vision of the enterprise by successfully implementing the strategies established [7]. However, considering only one single project at any time is not practical. Project portfolio management (PPM), as a new management methodology, has been implemented in most large enterprises [8]. PPM aims to do the right things, not just do things right [9]. The core idea of PPM is not studying isolated, local, and individual projects of enterprises, but instead revolves around a focus on the portfolio. This is achieved through a combination of projects that maximizes return and minimizes risk. PPM includes a series of dynamic decision-making processes—such as value assessing, project prioritizing, project selection and resource allocation—which help enterprises quickly adapt to changes in the market environment, improve the success rate of the implementation of enterprise projects and enhance the overall competitive ability of the enterprise [10]. In details, the objectives of project portfolio management are as follows [11,12]. First, an objective is to maximize the portfolio value, which includes two dimensions: the overall success of all projects and the synergies between projects within the portfolio. The others include linking the portfolio to enterprises' strategy, balancing the portfolio, preparing for the future and economic success. Furthermore, the success of project portfolios is also highly related to risk management [13]. The most critical activity in risk management is to identify the risks [14], which includes risk identification, assessment, and management of interdependencies between projects [15,16]. In a portfolio, risks arise from the project itself, while new risks emerge due to the interdependencies between projects [9]. The risks in portfolio include component risks, structural risks, and overall risks [13]. The systematic risk of a portfolio depends on the project elements and their relationships [17]. Some researchers calculate the systematic risk using the Markowitz portfolio theory [17], but this has some inherent limitations in practical applications [18]. Some researchers link this with the structure characteristics of a portfolio, such as size, homogeneity, diversity, and so on [19,20]. Due to PMO managers being concerned more about the relationships between projects—such as synergies, conflicts, and risk spreading—we decide to present this study from the structure view.

Projects in a portfolio may be connected with each other in different aspects and at different levels, including tasks, objectives, alliance, and even at a project level as whole [21]. Aside from these, the interdependencies in an IT project portfolio are even more complex due to their certain characteristics [22]. IT projects are mostly based on software products, which are the results of people's intelligence. Software development is not only a technical activity, but also depends on human skills, such as communication and negotiation. However, the factor of people, which is the main part of an IT project, is usually deemed as an environmental factor. The uncertainties, complexity, and invisibility of an IT project are mainly due to the human factor. The success of IT projects is based on the project team's understanding of customers' needs, implementation effect of human intelligence [23,24] and the team collaboration [25]. Introducing a natural science perspective for PPM provides another dimension and view [26], further enhancing our understanding on how to treat the project, how to express the relationships between projects, as well as how to guide the portfolio management and risk management using a new expression during the implementation. In particular, the relationships between projects are no longer cold and simple lines, but instead lively, understandable, and manageable tunnels. Furthermore, different relationships will bring different effects. For example, if the projects have close relationships, information, knowledge, and even risk can be transferred easily [27]. Therefore, it is important to choose an adequate model to describe these.

There is also a need to have efficient visualization tools to help decision makers to understand and manage the interdependencies [28]. If the project managers establish a comprehensive view of all projects, identify the relationships between projects, and recognize the role of people in a portfolio, they are able to improve the efficiency of information acquisition and clearly define the scope of information and risk transmission among projects. This ultimately achieves effective risk control. Graphical methods provide an efficient alternative method for displaying and evaluating complex data, which helps decision makers to communicate and come to an agreement from a strategic point of view [21,28–32].

Aside from the initial stage of the project portfolio selection, describing, analyzing and optimizing the implementation process of an IT project portfolio with interdependencies are also important for portfolio success [33]. The aim of this paper is to solve two key issues in the implementation stage. The foremost is to fill out the portfolio in a limited time, which requires an improvement in the efficiency of the project portfolio. The second is to maintain a minimal risk level. If projects use the same version, they may probably work together. More cooperation may enhance the efficiency, but also bring more risk due to interdependencies. The balance between them depends on the structure of the portfolio. Thus, the relationships between projects need to be characterized. Based on the analysis above, constructing an adequate model for an IT project portfolio and utilizing the structural characteristics to guide cooperation and risk control are the key points of the solution. Consequently, a managerial method based on the complex network theory [34,35] and entropy [36] for project portfolio implementation process optimization is proposed. The complex network model can afford a structural view of the portfolio, in which the IT project is treated as an agent with life and the IT project portfolio as a biological network. Furthermore, social network analysis is applied to analyze the social role of projects. Following this, the efficiency and the risk are measured by entropy using parameters related to the community structural properties of the model. Finally, the optimization method proposed could provide adequate cooperation ranges through searching communities and evaluation entropies to create a balance. Furthermore, key projects are identified and risk control measures are also given.

A practical example is used to illustrate how to use the managerial method in IT project portfolio implementation scenarios, which could serve to improve the efficiency of project portfolio management, improve the transparency of information, organize cooperation, and control risk.

2. New Lens of "Projects as a Biological Network" for Visualization and Decision

The new methodology of management arises from the revisiting of the traditional managed object. The basic components, management process, and model of a project portfolio all need to be upgraded. In the project management domain, the dominant lenses are "projects as temporary organizations" and "projects as production processes" [37]. Additionally, a biological perspective has been introduced into project management, with the concepts of genotype and phenotype being presented [26]. Furthermore, a lens is proposed for portfolio management, which is "projects as knowledge networks" based on complex networks [29]. In order to fully consider the subjective initiative of humans in the project portfolio, a new lens of "projects as a biology network" is presented based on biological points of view and network models, in which a project is seen as an agent with a life and a portfolio as a network. The emphasis on the biotech of the project portfolio network is that the cooperation between projects depends on the exchange between project implementers.

2.1. Life View on a Single Project

The new IT portfolio lens views IT projects from a new perspective. Traditional project portfolio management is mainly composed of three elements: scope, cost, and time [22], with people as an environmental factor with high uncertainty. In the actual IT portfolio management, the optimal combination of these elements is not easily applied, with the actual consumptions of time and cost usually differing from the theoretical calculations. In the evolutionary process of a portfolio, the human factor is not merely an environmental impact factor, but a dominant factor affecting cooperation and risk transmission. The genotype of a project individual does not just contain project attributes, methodology, and content [26], but also involves the human factor. An IT project, which is often the result of human intelligence, is viewed as an agent having its own objective, organization, and function. Subsequently, the whole portfolio is a biological network, in which each project is an independent individual life connected to each other.

The main reason for adopting this view is that the most important part in management is human management [38]. In a portfolio including multiple project projects, finding the key contributors and stakeholders will improve the efficiency of project management. For example, this can happen if a decision needs to be made when a project is linked to multiple projects and there is a large amount of information being exchanged between them. As it is time-consuming to seek opinions from all project managers, consulting key project managers with extensive knowledge of all projects involved could save an enormous amount of time that could provide sufficient information for decision making.

2.2. New Lens of "Projects as a Biological Network"

These connection factors between projects are often called as project interdependencies (PI) [39], which include resources, market, knowledge, outcomes, and benefits, which will produce multi-topologies [40]. Projects may share or compete for resources, such as hardware, equipment, software, and working environments [21]. Knowledge generated by one project may be transferred to another within a portfolio [41]. The outcome and results from a project are made available and can benefit other projects when it enters into the market [42].

The network view of a portfolio provides a new way to express a project portfolio with interdependence factors [21,28–31]. It gives project managers a holistic view of the overall projects. The main factors and their influence on other projects can be determined. Furthermore, it is easier to identify related projects, which could inspire the project managers to work together to communicate ideas, transfer knowledge, and achieve strategic objectives. This also makes it easier for the portfolio management office (PMO) to obtain an understanding with regard to the portfolio risk, which is more than the sum of a single project risk. Based on the role analysis of nodes in a network, it could help the PMO to make a decision on whether a project should be added or removed as well as making arrangements for cooperation between projects [43].

3. Concepts and Methods

Large enterprises tend to implement hundreds of IT projects each year to meet the needs and requirements from regulatory authorities, business from customers and internal management. Each IT project is associated with existing or developing software systems. Different projects are connected with each other because of the same corresponding application system. The pairing of projects and the application software system is indicated as a "multiple to multiple" relationship, as shown in Figure 1.

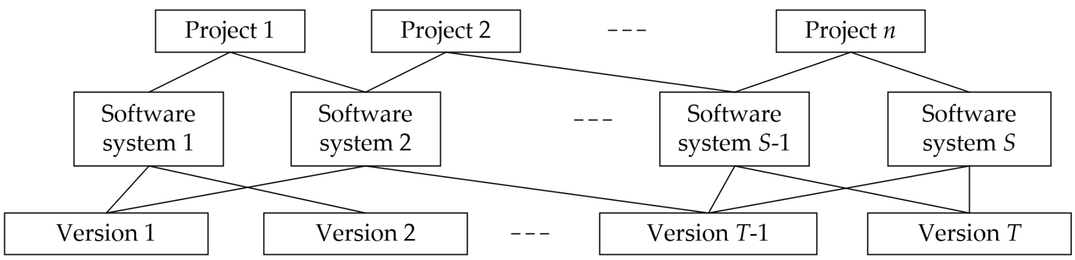


Figure 1. Project relationships dependent on systems/versions.

Furthermore, during the implementation stage in a project, software developers in different project teams are continually changing source codes. Different versions are produced and organized in a "file tree" [44]. The pairing of projects and software versions is also a "multiple to multiple" and dynamic relationship.

Constructing a complex network model of a project portfolio can possibly help managers to improve efficiency and control risk. Based on the model and entropy, a new managerial method for portfolio implementation process optimization is proposed.

3.1. Framework of Portfolio Implementation Process Optimization

The method aims to minimize the risk level of the portfolio and in the meantime, maintains a certain level of cooperation. Furthermore, key projects will be identified for risk control.

The main way to improve the efficiency of the existing portfolio is to strengthen cooperation between projects. Cooperation is based on the close relationship between multiple projects and it mainly happens in a small-world community [45]. In the meantime, more resources involved in cooperation need to be coordinated, with a subsequent increase in the risks arising from cooperation. The spread of this risk depends on the structure of the type of group. If the group is highly homogeneous, the projects may all share similar versions and the risk is easily spread when the risk occurs randomly [45]. Furthermore, if the group is highly heterogeneous, the software systems shared by the projects may considerably differ. This type of structure is relatively resistant to random risks [46].

Through the above analysis, the main idea of the optimization is based on the complex model, which aims to balance the efficiency and the risk through adjusting the structure of the project portfolio network. As the purpose of community partitioning is to identify high-density local networks, essentially to discover small-world networks [47], the first step in the optimization algorithm is to divide the original project portfolio network into a hierarchy community tree, which determines possible scopes of cooperation. The second step is to measure the order of the system, such as the aggregation and heterogeneity according to the entropy [48]. This includes the efficiency entropy and risk entropy, which are established by using the characteristic parameters of the local community. Finally, the efficiency entropy and the risk entropy are balanced by adjusting the sub-community combination. Furthermore, corresponding cooperation advice and risk control means are given. The whole procedure above is shown in Figure 2.

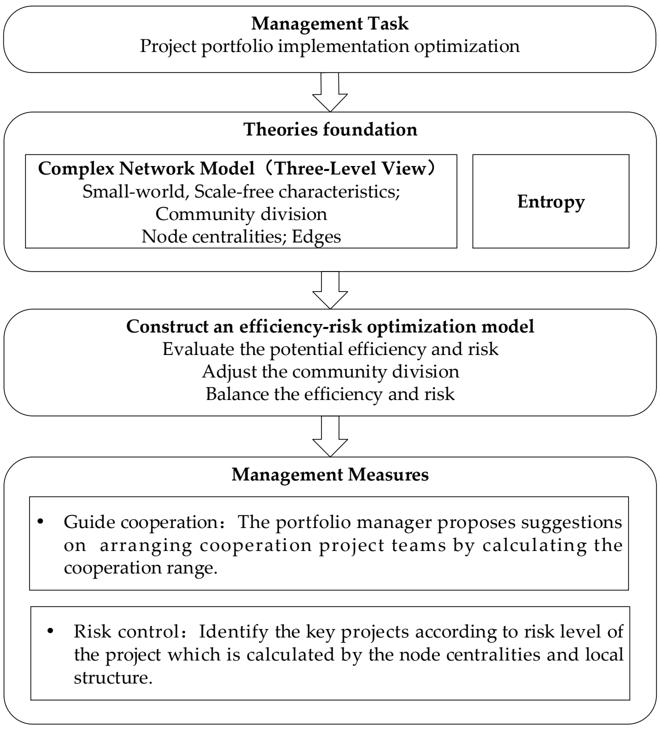


Figure 2. The framework of optimizing the portfolio implementation process.

3.2. Weighted Network Model for an IT Project Portfolio

Based on the weighted network model of an IT project portfolio, statistical indicators of the network will be calculated. In addition, corresponding issues will be discussed, such as the complexity of portfolios, community phenomenon, the roles that projects play within a network, as well as how to balance the cooperation and risk in a portfolio. These analyses will help PMO managers and project managers better understand the portfolio and make decisions.

3.2.1. Weighted Network Model

A weighted network, i.e., edge-weighted graph, denoted as $G_P = G_P$ (V, E, W), is used to extract a portfolio into a complex weighted network [49], where $V = \{v_1, v_2, \ldots, v_n\}$ is the node set of the network, $E = \{e_1, e_2, \ldots, e_m\}$ is the edge set of the network in addition to $W = \{w_{ij}\}$ being the set of edge weights in which w_{ij} is the weight of the connected edges between nodes v_i and v_j $(i, j = 1, 2, \ldots, n)$. In the model, a node represents a project, while the edge weight is related to the shared software systems between projects. The corresponding formula is given by

$$w_{ij} = \frac{r_{ij}^s}{n_s}, \tag{1}$$

where r_{ij}^s represents the number of software systems shared by project v_i and v_j; and n_s is the whole number of software systems. It would be useful to "normalize" the weights by the average weight in the network, with the normalized values then being used in the following experiments.

3.2.2. Statistic Indicators of IT Project Portfolio Network

There are many measures for analyzing the properties of a network. The small-world property [50], scale-free property [51], and the centrality [52,53] of nodes are the concerns. Furthermore, average path length [46], clustering coefficient [50,54], and degree distribution [49] are mainly used to analyze the overall network. Centrality measures are used to analyze the roles of single nodes, including degree centrality (DC) [55], closeness centrality (CC) [56], betweenness centrality (BC) [57], and eigenvector centrality (EC) [58]. These are analogues for "influence", "importance", and "information/knowledge bridges". Based on measurements of the centralities, the important/unimportant nodes, bridge nodes and center nodes can be found for project cooperation and risk control. All the formulas for weighted networks are listed in Table A1.

- Average path length

The shortest path length between two nodes in a network refers to the path with the minimum sum of edges or edge weights. The average path length is defined as the average of shortest path lengths for all-pairs of nodes, which is used to measure the information or mass transport efficiency of a network. A small-world network has a small average path length. In an IT project portfolio network, the edge represents the software system or the version, which means that the two projects need to make changes to the same software system/version. If the path length between two nodes equals to 1, they share at least one same system/version. Knowledge could be shared more easily in two projects with shorter path length, but the risk may increase at the same time. The cooperation and risk must be balanced through the organization of project activities.

- Clustering coefficient

The clustering coefficient measures the degree to which nodes in a network tend to cluster together. There are two versions: the global and the local. The global version gives an overall indication of the clustering in the network, whereas the local indicates the aggregation degree of single nodes. A large cluster coefficient of a project portfolio means the projects in a portfolio are connected with each other closely and vice versa. A high clustering coefficient is another sign of a small-world network.

- Degree distribution and degree centrality

Degree distribution and degree centrality are two close concepts related to the degree of a node, which involves the number of edges connected to that node. The degree distribution emphasizes the probability distribution of these degrees over the whole network, whereas the degree centrality gives an indication of connections of single nodes. Usually, the scale-free property of the network is investigated by these measures. The scale-free property indicates that the network is not evenly distributed, which essentially means that a few nodes have more connections and play a dominant role in the network, while most nodes have only a small number of connections. The weighted degree of a node is similar to the degree, which is the sum of the weight of the edges. A project node with a high degree centrality is definitely connected to many other projects sharing the same software system/version. It may be an information center from which managers may easily know the situations of other projects.

- Closeness centrality

Closeness centrality measures a node's information transformation independence in a network. If one node is closer to others, it can reach other nodes more easily and it has a higher closeness centrality. This indicator is usually defined as the inverse of average shortest path from a node to all other nodes. A project with a high closeness centrality can obtain information from others more easily. This type of project may have several interactions with other projects. Therefore, when the PMO managers want to obtain information quickly, they can consult these project managers on the nodes.

- Betweenness centrality

Betweenness centrality reflects a node's bridge role in a network. It is the frequency of shortest paths from all vertices to all others that pass through that node. A node with a high betweenness centrality may have a large influence within a network because it controls the method of information passing among others. If it is removed from a portfolio, the network connectivity is reduced. As a consequence, the removal will decrease risk, but will also affect cooperation. It is also a communication key node and large amounts of information will pass through the bridge. When PMO managers want to add or remove a project, they could acquire this type of project in its neighbors instead of using all opinions of neighbors to judge the risk influence, which will subsequently save time.

- Eigenvector centrality

Eigenvector centrality (also called eigencentrality) measures a node importance through its neighbors' influence in a network. If two nodes have the same connections, the node whose connections are with more links of high importance has a high eigenvector centrality. Google's PageRank is a variant of the eigenvector centrality. A project with a high eigenvector centrality is a potential important or influential project, which may not be found by other measures.

3.2.3. Community Detection of IT Project Portfolio Network

Complex networks often have millions of nodes and edges, so it is difficult to understand their relationships. Retrieving comprehensive information from complex structures could help people to find some representative information [47]. Using community detection, a complex network can be divided into a number of communities (i.e., a set of nodes with the same properties), where the nodes are more highly interconnected than those in other communities. These highly interconnected nodes may have similar characteristics or behaviors or consist of a functional unit. The connection nodes between sub-communities are the key points of network connections.

The community detection is carried out by the Louvain algorithm [59], based on Newman's modularity [60]. A large modularity means high quality of community division and vice versa. At the original stage, n nodes are n different communities. Following this, the algorithm is used to traverse all the nodes in the network, with one node moving each time to one community and calculating the increment of the modularity. This then places the node in the community in order to gain maximum modularity. This process is repeated until no nodes could be moved. This is the first stage of the algorithm and produces a new network. The second stage applies the same procedure to the new network until the modularity no longer increases. The modularity measures the density of links in communities, but not the links between communities.

The algorithm could produce a hierarchical community structure. Projects in the same community are similar, such as possibly sharing similar resources, information or objectives. In the same community, cooperation could be constructed. Between communities, the border could be sketched out to decrease negative effects on each other and prevent risk transfer. Community division can be carried out on the entire network or in the community. Therefore, all the communities can be organized in a tree form, as shown in Figure 3.

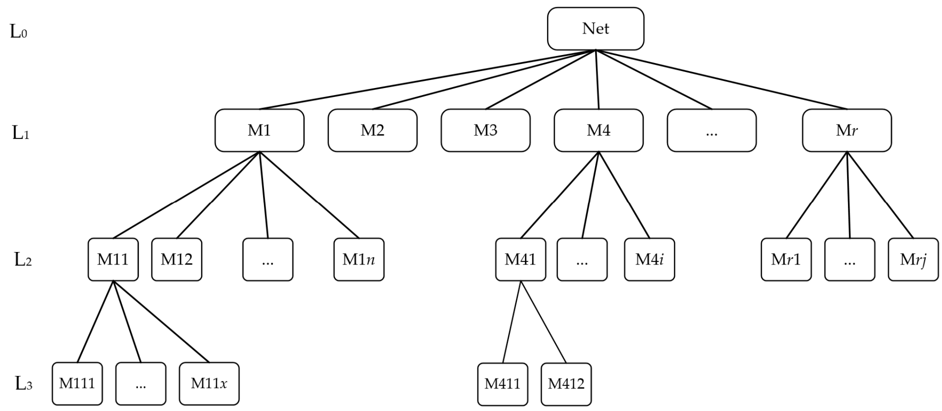


Figure 3. Community division tree.

3.3. Efficiency–Risk Balance Based on the Network Model and Entropy

3.3.1. Efficiency–Risk Optimization Model

The goal of the optimization method proposed in this paper is to determine a set of local project cooperation scopes $\{M_1, M_2, \ldots, M_s\}$ in the implementation of the project portfolio, which minimizes the risk while ensuring a certain level of efficiency. Cooperation usually occurs within the community, so the scope of cooperation is determined mainly through the multi-stage division of the project portfolio network. Thus, the optimization problem can be expressed as

$$\begin{array}{l} \text{Min } R(M_1, M_2, ..., M_s) \\ \text{s.t. } F(M_1, M_2, ..., M_s) \geq F_{m\theta} \end{array}, \tag{2}$$

where $R(\cdot)$ is a risk function; $F(\cdot)$ is an efficiency function; and $F_{m\theta}$ is an efficiency threshold.

When the nodes aggregate closely in a community, the shared software systems are highly similar, resulting in a high potential efficiency of cooperation. However, if a project fails at the same time, the impact on other projects is also large due to the high homogeneity [61]. On the other hand, when the nodes are scattered, the potential efficiency of cooperation is low, while the risk may also be low at the same time due to the heterogeneity. Aggregation and heterogeneity have a negative correlation, but the focus is different. Therefore, we use the aggregation property to measure the efficiency of cooperation and heterogeneity to quantify the risk.

3.3.2. Efficiency Entropy and Risk Entropy

The above measures could describe many properties of a complex network, but still are unable to quantify “how complex is a complex network” [62]. In the information theory proposed by Shannon, the information is ”the reduction of entropy” and ”the reduction of uncertainty of a system” [36]. Entropy is an important concept, which could provide quantitative measurements for the probability distribution. If the probability has a uniform distribution in a complex network, it means that each node has a different state. The system is highly disordered and corresponding entropy increases. On the contrary, if the probability distribution is not uniform, some states have a higher probability. It means that these states could have more chance to be predictable and the uncertainty decreases. The system becomes more orderly and the entropy decreases. Thus, the entropy can describe the state of order in a system [48]. For a complex network, the order means that it has some particular characteristics or has a specific structure. Using entropy to quantify the order could help our understanding of the complexity.

Suppose X is a discrete random variable with possible values $\{x_1, x_2, \ldots, x_n\}$ and probability mass function $\mathrm{P}(X)$. The probability of x_i is denoted by p_i. The entropy can explicitly be written as

$$\mathrm{H}(X) = \sum_{i=1}^{n} p_i \mathrm{I}(x_i) = -\sum_{i=1}^{n} p_i \log_b p_i, \tag{3}$$

where b is the base of logarithm used. Units of entropy are the bit, Hartley, and nat, depending on the base used which are 2, 10, and Euler's number e, respectively. The values defined by different bases can be converted by certain corresponding factors. Here the Euler's number e is used.

In a project portfolio network, the degree of aggregation is low if the nodes are scattered. In the sense of aggregation, it is a disorder. Furthermore, if there are many nodes aggregated together, it follows some order. In addition, if the number of edges of each node is relatively similar to each other, this is a type of structural homogeneity. From the perspective of heterogeneity, it is a disorder. If the edge number greatly differs, it shows an order in the sense of heterogeneity. A heterogeneous network, such as a scale-free network, can resist random attacks. By protecting important nodes, one can effectively control the spread of risk. According to the above discussion on entropy, we use entropy to measure the efficiency and risk. The order in the aggregation sense is used to measure efficiency, which is called efficiency entropy. The order in the heterogeneity sense is used to measure risk, which is called risk entropy.

In order to find an adequate set of ranges for balancing the cooperation efficiency and the risk, the portfolio network is divided into s communities $\{\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_s\}$. The M_j community has n_j projects, $n = \sum_{j=1}^{s} n_j$. Therefore, the efficiency entropy and risk entropy are measured based on the communities and used to realize the efficiency function and risk function.

Efficiency entropy consists of two parts. The first part measures the cooperation in the community in the development phase of a portfolio. It is calculated based on the probability distribution of the sum of the clustering coefficient value and closeness centrality. After that phase, all versions derived from the same software system will be integrated into one version in the software test phase. The second part of the entropy measures the communities' integration efficiency, which depends on the size of each community. Thus, the efficiency entropy is given as

$$\mathrm{H}_E = \sum_{j=1}^{s} \frac{n_j}{n} \mathrm{H}_{\mathrm{EM}_j} + \mathrm{H}_{\mathrm{M}} \tag{4}$$

In the first part, set μ as the sum of clustering coefficient value and closeness centrality. Therefore, for each node v_i, $\mu_i = \beta_1 C_i + \beta_2 C_c(i)$ and $\beta_1 + \beta_2 = 1$. If the values of most μ are large, the nodes aggregate together and the entropy is small. Thus, the probability distribution of μ is used to calculate p. The range of μ is [0, 1], which is divided into 10 intervals $(\Omega_1, \Omega_2, \ldots, \Omega_{10})$, with $p_k = \sum_i p(\mu_i), \mu_i \in \Omega_k$ being the probability of μ in each interval. Following this, the first part of the entropy is given by

$$\mathrm{H}_{EM_j} = -\frac{1}{\psi} \sum_{k=1}^{10} p_k \ln p_k + \gamma, \tag{5}$$

where ψ is a normalized coefficient defined as $\psi = \ln(n_j) \times \mu^* \times \mathrm{num}(\Omega(\mu^*))$, $\mu^* = \max(\mu_i)$; $\mathrm{num}(\cdot)$ is a function to count the number of projects in $\Omega(\mu^*)$; and γ is a correction coefficient related to the

U-shape cost curve [63] in the "economies of scale" theory. When the scale increases, the per-unit cost will decrease. If the scale is above a limit, the per-unit cost begins to increase. A similar situation occurs in IT project cooperation. Here we set γ as

$$\gamma = \frac{\left(n_j - n^*\right)^2}{\left(n - n^*\right)^2}, \tag{6}$$

where n^* is an optimal scale.

The second part is given by

$$\mathrm{H_M} = -\sum_{j=1}^{s} \frac{n_j}{n} \ln \frac{n_j}{n}. \tag{7}$$

According to the definition, the value of H_E is related to the properties of the inner community and the community division. The efficiency may increase in the development phase due to the aggregation being closer in a community, although this will decrease in the test phase due to the reunion of the community. The lower the efficiency entropy is, the higher the efficiency gets. The efficiency entropy is the inverse of the efficiency function.

Risk entropy is based on the weighted degree value [61], which is given by

$$\mathrm{H}_R = \sum_{j=1}^{s} \frac{n_j}{n} \mathrm{H}_{\mathrm{RM}_j}, \tag{8}$$

where $\mathrm{H}_{\mathrm{RM}_j} = -\sum\limits_{i=1}^{n_j} p_i \ln p_i$; and $p_i = \frac{k_i^w}{\sum_{i=1}^{n_j} k_i^w}$, k_i^w is the weighted degree value of node v_i. According to the definition, when the projects in a community share same software versions, the risk is high and the entropy H_R has a high value. If each project adopts a separate version, the risk is very low and the entropy H_R is equal to 0.

3.3.3. Efficiency–Risk Balance Optimization Algorithm

To minimize the risk level, a greedy algorithm is used to find the adequate community combination. It includes several steps:

Step 1: Construct the complex network model.

Step 2: Divide the network into several communities and repeat the division process on each community until the modularity value is less than a threshold. Following this, the division result can be organized as a hierarchical tree. Suppose there are Q layers $L_1, \ldots, L_Q$. Each layer has n_q communities. Layer L_0 is the original network.

Step 3: Search the tree from the top layer to the bottom layer in order to find the best combination to minimize the risk entropy H_R and maintain the efficiency entropy H_E not over the maximum entropy threshold $\mathrm{H}_{E\theta}$.

The procedure is shown as Figure 4 and the optimization communities are $\{\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_s\}$.

The above procedure could specify a set of ranges of cooperation, which provide suggestions to project managers. The efficiency entropy threshold setting depends on the actual situation. For example, if two projects in a cooperation range are limited by urgent completion time and with high risks, cooperation may not be a good choice and the threshold should be set to a large value. If there are plenty of human resources and only a few projects, it is not necessary to cooperate too. The threshold $\mathrm{H}_{E\theta}$ represents an acceptable cooperation level. Here a method is provided to determine it. Firstly, generate a representative and acceptable scale cooperative community, that is an Erdős–Rényi (ER) random network [64] ER_θ with the global clustering coefficient similar to that of the network $\mathrm{G_P}$; then the value of the efficiency entropy $\mathrm{H}_{\mathrm{ER}\theta}$ of ER_θ can be used as an upper limit of the threshold. The scale

n_θ can be decided by n/n_c or based on the user's preference, where n is the size of the network G_P and n_c is the number of communities which can reconstruct the whole original network.

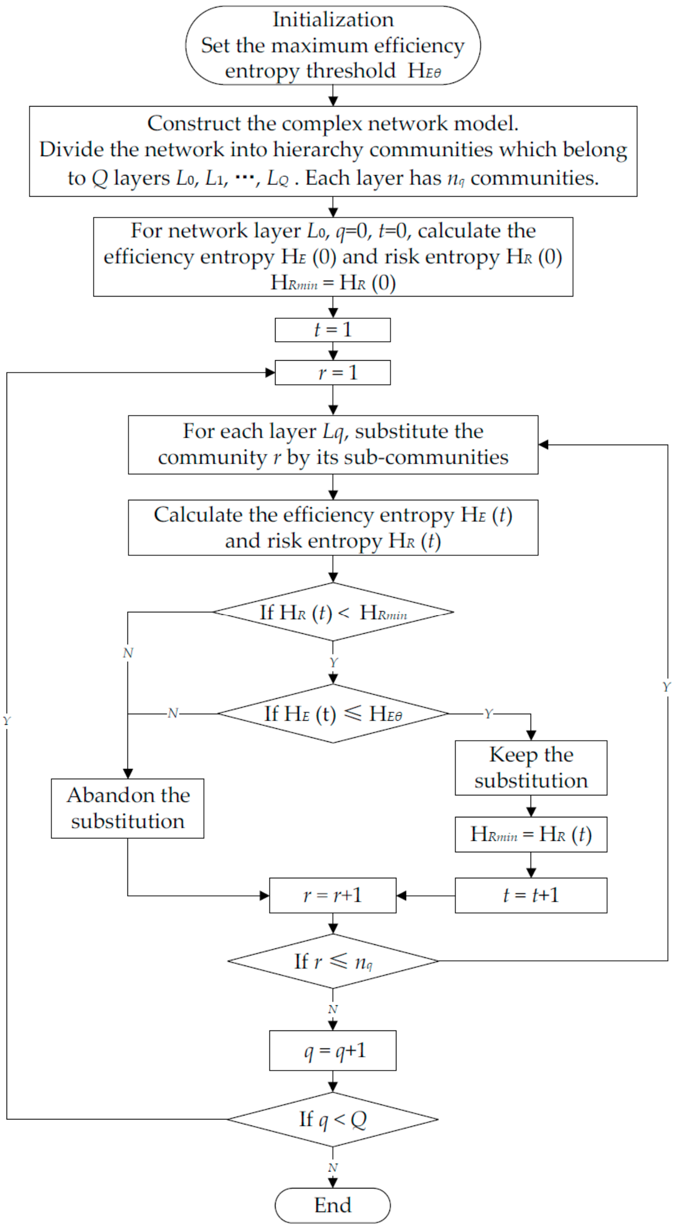


Figure 4. Optimization algorithm for efficiency–risk balance in the portfolio implementation process.

In addition, after evaluating the risk entropy, high-risk projects should also be identified to help managers take measures to control risk. In general, a node with a high degree is critical and accompanied with high risk, although the actual situation is more complex. According to the centrality analysis, the risk type of each node is calculated according to the structural properties and relative parameters. A risk type contains three aspects: global risk measures the effect degree of one project affecting other projects in the entire portfolio; intercommunity risk measures the degree of spreading risk from one community to another; and inner community risk measures the effect degree of one project affecting other projects in the same community. The risk type classification process is as shown in Figure 5.

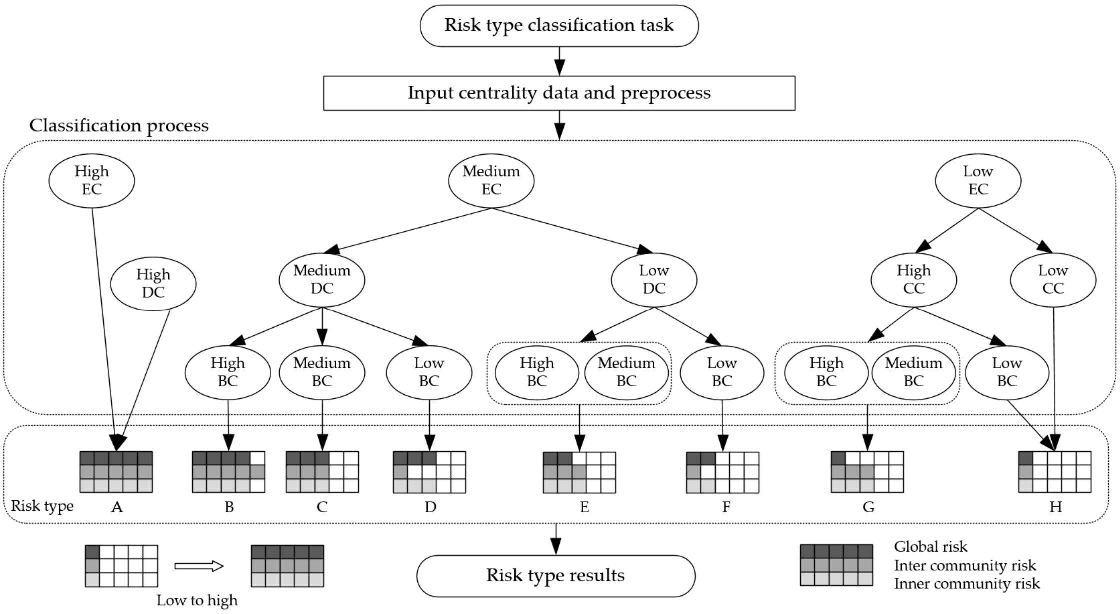


Figure 5. Classification of projects' risk type. Centrality level: Defined by users, such as High −5%; Medium −5–50%; Low–Last 50%.

Projects with a type A risk are mostly key projects, which have an impact on most projects in a portfolio. Nodes with a type B or type C risk should also be dealt with carefully.

In the following experiments, an IT project portfolio example is used to illustrate how to construct a complex model and how the optimization method is used to create an efficiency–risk balance to guide cooperation and control risk.

4. Illustrative Example

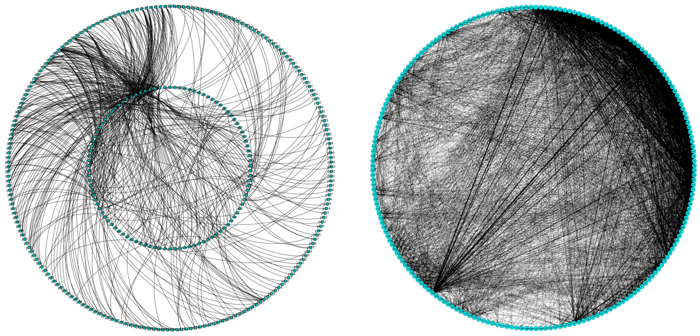
Generally, a typical project portfolio in a large financial enterprise is composed of hundreds of projects of channel interface types, e.g., point-of-sale (POS), automated teller machine (ATM), online banking, call center, the interface connected with external futures companies, securities companies, and so on. These can also be projects of business requirements, e.g., bills, card business, and credit business; as well as projects of internal and external regulatory demand, e.g., updates or reports from risk management, audit management and human resources management. Each project may be associated with a number of software systems, which make projects related to each other. The illustrative example data set is a practical project portfolio, which includes 217 IT projects from regulatory authorities' demands, customer business requirements, and internal management needs.

The relationships between projects dynamically change during their life cycle. In the initial stage, managers can only identify the software systems related to the projects. In the implementation stage, many versions are derived from one version tree of the main software system. Projects with the same software systems may not cooperate, as projects cooperating in the development phase means that they share the same versions. In the software test phase, all those projects that share the same software system should cooperate to integrate a unique version. A complex network model is constructed for the portfolio to represent the interdependencies. Once a community is extracted from the network, the relationships between them are cut off in the development phase and will be reconnected in the test phase. Following this, the following parts will illustrate how to construct a network model, how to form a hierarchy community tree, how to measure the cooperation efficiency and risk using entropy, as well as how to balance them.

4.1. Weighted Network Model of an IT Project Portfolio

The typical portfolio mentioned above includes 217 projects, the number of which denotes the size of the project portfolio (P). There is a total of 103 relative software systems. It is generally a mid-sized scale portfolio. Figure 6a shows the relationships between projects and software systems. The nodes in the outer circle are the projects and those in the inner circle are the software systems.

Following this, the weighted network model G_P is constructed based on the interdependent factor of software systems. After detecting the connections, nine projects are isolated with no linkage with others. The other 208 projects construct a partially connected network, which consists of two fully connected sub-networks. One of them is a large network, which includes 199 nodes and 2949 edges. The other network includes 9 nodes and 20 edges. The model is shown in Figure 6b, which is created using Gephi software [65]. It can be seen from the graph that there are very complex relationships between the projects within a portfolio, which brings great difficulties into the actual management.



(a) (b)

Figure 6. The complex network model of a project portfolio using Circular Layout. (**a**) Relationships between projects and software systems; (**b**) Relationships between projects.

Additionally, some indicators and measured values will be compared to those of an ER random network ER1 with similar nodes and edges to show the properties of G_P.

4.1.1. Properties of the Network

Overall Properties

The small-world property and scale-free property of a project portfolio network model are observed and analyzed through the clustering coefficient and degree distribution, which are shown in Figures 7 and 8. Here, the cumulative degree distribution (CDF) is used to show the degree distribution.

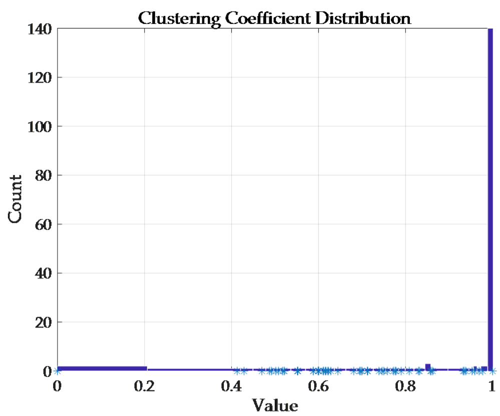


Figure 7. Clustering coefficient distribution.

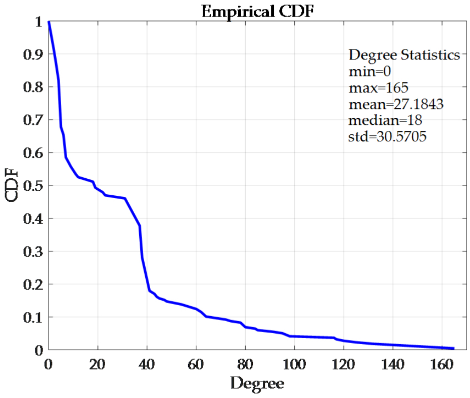


Figure 8. Degree distribution.

Other indicators are listed in Table 1.

Table 1. Indicators.

Net	Average Path Length	Global Clustering Coefficient	Average Clustering Coefficient
G_P	2.3550	0.6735	0.9052
ER1	1.4111	0.1347	0.1344

According to the measure of small-worldness [66] in Table A1, S_{Global} is 2.9960 >> 1 and S_{Local} is 4.0356 >> 1. From the statistics, G_P exhibits the small-world property [50], in which the average path length is small and the clustering coefficient is large. The small-world property indicates that there is a high degree of nodal aggregation in this network, which is the basis of cooperation.

According to the power-law distribution function, the fitting coefficient is 0.8865. It is slightly lower than the standard power-law distribution coefficient $\eta \in [1, 2]$ for cumulative probability distribution [46]. It indicates that the degree distribution of the network approximately follows a power-law distribution. There are few dominant nodes in the network that have a high degree and a large number of node connections, but the scale-free property of this network is not very typical. The reason is that the network has several nearly completely connected societies with a number of nodes. Therefore, it is necessary to take relevant risk measures according to the different community properties. Those high degree nodes in the network play more important roles than the others. They may be the key projects with high risk to which managers should pay more attention.

Centrality Results of Nodes

The indicators of centrality identify the importance of nodes in the network G_P. The box plot is used to show the dispersion of each centrality and details of normalized centralities are shown in Figure 9.

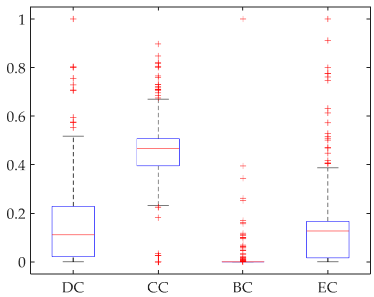


Figure 9. Distribution of centrality values.

The correlation coefficients are used to measure the dependence between each pair of centralities. The results are shown in Figure 10.

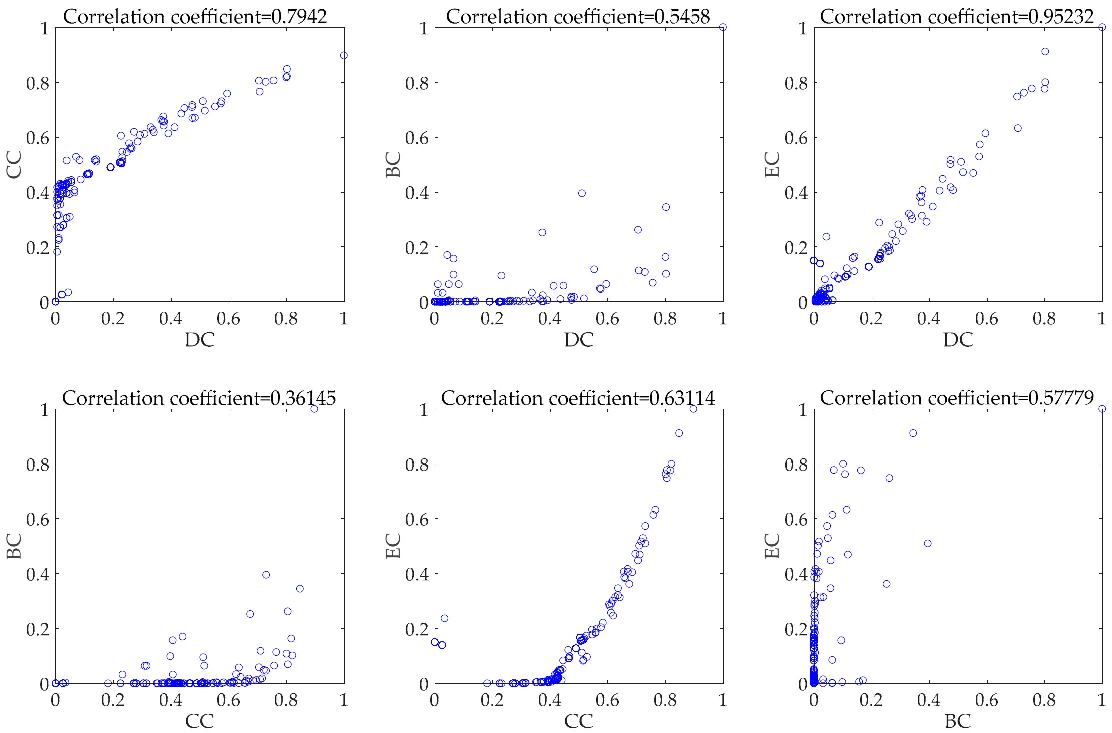


Figure 10. Correlations of centrality pairs.

From Figure 9, it is found that each indicator has high dispersion, with values being widely scattered around the average value. That means the properties of these nodes vary widely, with the existence of key nodes. From Figure 10, the correlations between centrality pairs are all positive, but not high in most pairs. These four indicators of each node are not consistent. Some nodes with a high degree may have low betweenness centrality. Therefore, to achieve risk control, the importance of these key nodes should be identified further according to the risk classification process in Section 3.3.

4.1.2. Community Division for G_P

Gephi software (version 0.9.1) is used by the Louvain algorithm for network community division. This work will be conducted iteratively. Firstly, the network is divided into several communities and each community may be divided into small ones. The community division of the first step is shown in Figure 11 by Yifan Hu Proportional network layout [67]. This layout provides a multi-level force directed algorithm for large graphs using a tree structure and nodes with force, which can easily express the multi-level and strength of the links. Furthermore, in the graph, the size of a node is proportional to the degree. The color of nodes identifies the community.

The resolution is set to 1, which should be lower to obtain smaller communities and larger to obtain bigger communities. As shown in Figure 11, the network G_P can be divided into eight communities (different communities can be distinguished by different colors) and the modularity is 0.318. It is obviously larger than the modularity of the random network ER1, which is 0.141. It is feasible to implement cooperation within the community.

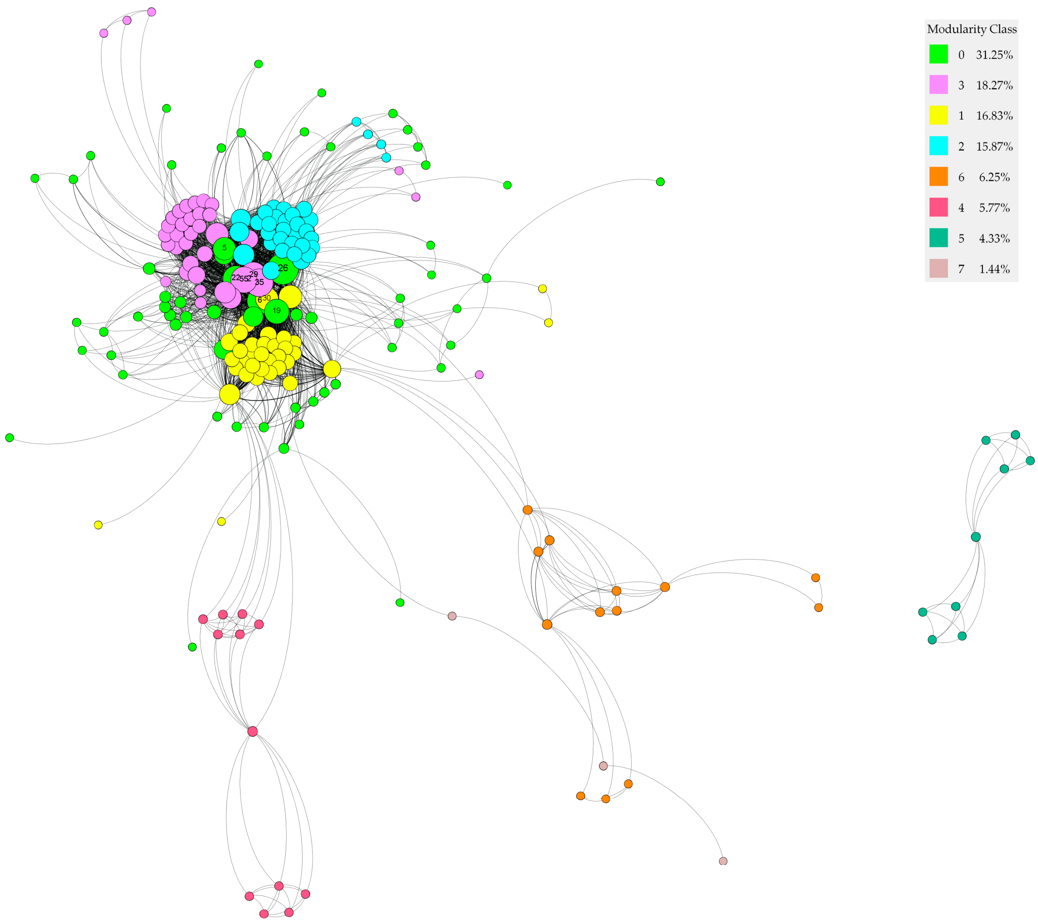


Figure 11. Community detection of layer L_1.

We repeat the above process, so that each community is divided into sub-communities until the modularity value is less than 0.09. In this way, the original network can be divided into several smaller communities. The portfolio implementation process optimization aims to find the best sub-community combination to balance the efficiency and risk. The division result is organized as a hierarchical tree form. The whole network can be divided into 4 layers containing 35 sub nodes, as shown in Figure 12. The complete network can be reconstructed by up to 26 communities, that is, n_c = 26.

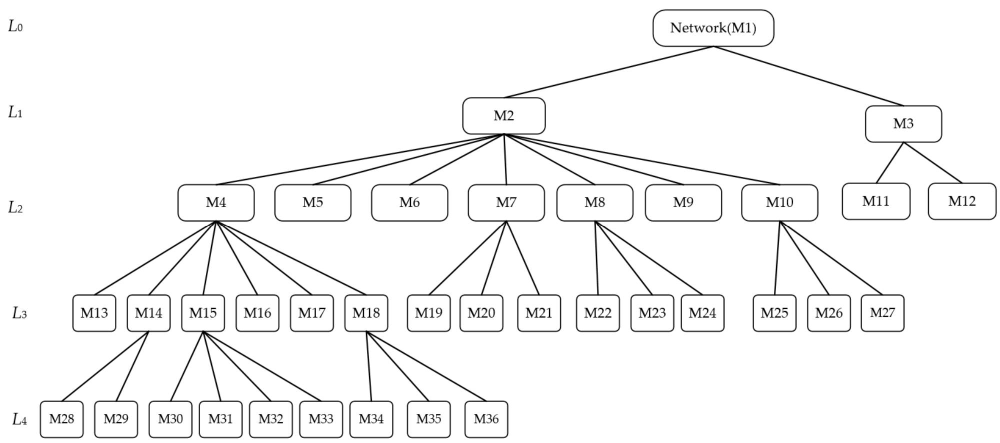
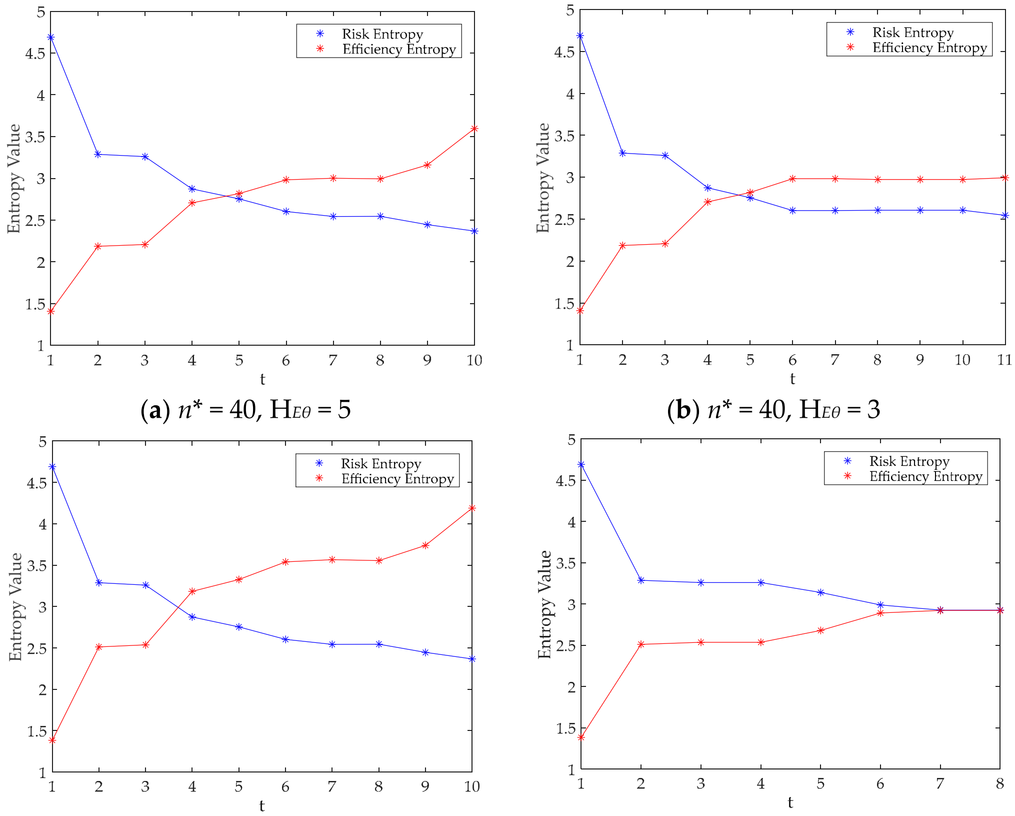


Figure 12. Community detection.

The projects in the same community will share the same versions, which are concrete, and operational relations. Through the efficiency–risk balance procedure below, project managers will know the adequate cooperation range and the risk level. Thus, they will be able to make decisions more easily.

4.2. Project Portfolio Implementation Process Optimization Results

Based on the complex model and the optimization algorithm in Figure 4, the efficiency entropy and risk entropy can be calculated in order to make a balance between them. In this project portfolio optimization process simulation, the optimal scale n^* and the threshold $H_{E\theta}$ are set to different values. The threshold $H_{E\theta}$ is determined according to the method mentioned in Section 3.3.3. The scale of the random network ER_θ is 8 (n/n_c = 208/26 = 8) and the global clustering coefficient is 0.664. If the optimal scale n^* is set to 40, the efficiency entropy $H_{ER\theta}$ is 5.69; and if n^* is set to 100, $H_{ER\theta}$ is 6.39. So the threshold $H_{E\theta}$ is set to 5 or less in simulations. The results are shown in Figure 13.



(**a**) n^* = 40, $H_{E\theta}$ = 5 (**b**) n^* = 40, $H_{E\theta}$ = 3

(**c**) n^* = 100, $H_{E\theta}$ = 5 (**d**) n^* = 100, $H_{E\theta}$ = 3

Figure 13. Optimization process.

From the results, the values of efficiency entropy and risk entropy are related to the size of the community and the aggregation level of nodes. In the process, the extraction of a community from its root level actually cuts the software version relationship between them in the development phase. They are only related due to the interdependence of software systems, instead of versions, which decreases the risk of the portfolio. During the search process, the risk entropy decreases and the efficiency entropy gradually increases. The search process can be controlled by appropriate parameter settings. If the value of n^* is large, it is better to organize large scale cooperation. Therefore, when the network is divided into several small communities, the efficiency entropy increases faster. Under the premise of controlling risk, the efficiency entropy can be controlled by setting the threshold.

After that, the risk type of each project is also assessed according to the risk classification process. Projects of risk Types A, B, and C are marked in Figure 14.

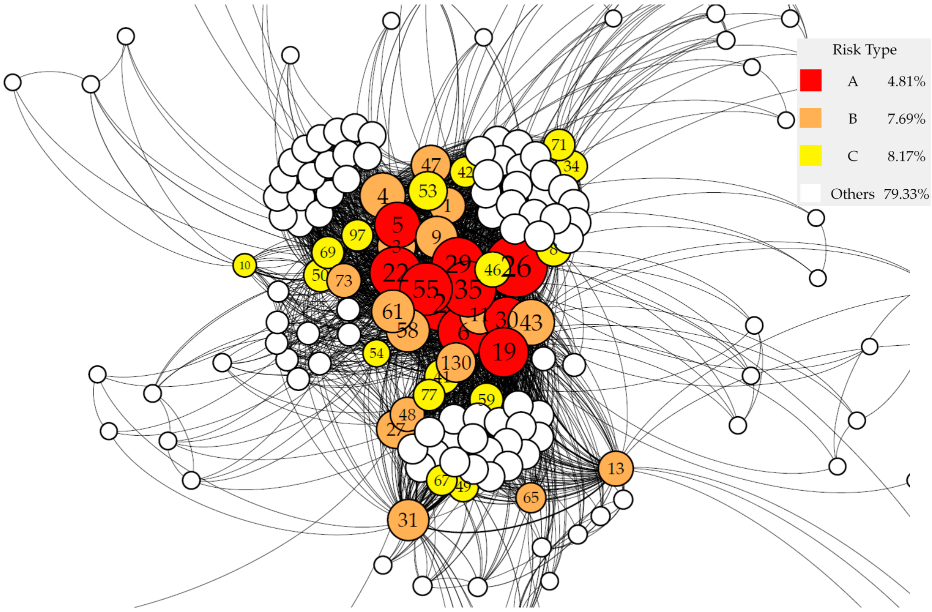


Figure 14. Key projects.

It can be seen that most projects of risk type A are key projects with a high degree centrality or eigencentrality, which occupy the center place in the network. These connect with multiple projects, involving a wide range of systems or versions. Controlling the risk of these projects can greatly control the spread of risk throughout the entire portfolio. Projects of risk type B also have a considerable number of connections, with some of them even being the main bridges between different communities. For example, node 31 is the bridge between the small community 4 and the large communities 0 and 1 at layer L_1. Projects of risk type C are further apart from the center, having a moderate number of connections and playing bridge roles. Controlling the risk of these projects can ensure that the risk does not spread between specific local communities.

5. Conclusions

The complex network model could provide portfolio managers with three levels of cognition, namely, macroscopic network, meso-community, and micro-node. Furthermore, this could help project managers to know their own roles and neighbors. These will help to develop IT portfolio governance, promote cooperation between projects, and control the spread of risk.

- Macro-level

Macro-level means from the view of the overall structure of the network and its statistical properties to form an overall direction of the portfolio management. If the network has the small-world property with highly aggregated nodes, the possibility of cooperation is large. Resources could be allocated to unification, considering the cooperation in the communities. If the network has a scale-free feature, it is important to pay special attention to the success rate of those key projects for risk control.

- Meso-level

Through the community division, portfolio managers could understand the size of the community, the location of the community in the network, and the mutual influence between communities. This could help to organize effective cooperative relations within the community, which will save resources. Community detection could clarify the scope of the risk occurrence.

- Micro-level

On the micro-level, the main work is to analyze the properties of specific nodes, understand the location and role of nodes in the community through nodes' centralities in addition to the impact on risk.

The network model provides a structural description of a portfolio. Following this, some indicators can be used to quantify the cooperation efficiency and the risk to some extent. Through the hierarchy community division, a potential set of cooperation ranges is provided for searching in order to balance the efficiency and the risk. Through the possibility distribution of the coefficient and closeness centrality, the efficiency entropy can describe the aggregation property, which is the basis of cooperation. In addition, with regards to the scale of project portfolio in a community, the efficiency entropy also considers the scale economics in actual situations. The risk entropy based on weighted degrees can provide a description of the risk in a community. It is related to the heterogeneity property, which could help to make decisions on taking measures for risk control.

The optimization method is used in a given portfolio. More work should be done in the future, such as combining with the portfolio selection process, dealing with dynamic portfolio changes, and so on. The complex network model is constructed based on the software system/version in this paper, but there are many other interdependencies between projects. Determining how best to take these factors into account and describe the relationship between different networks is still a challenge.

Acknowledgments: The authors would like to thank Yan Shi for the help in the understanding of social network analysis.

Author Contributions: Qin Wang provided the data, designed the algorithm, performed the experiments, and wrote the paper; Guangping Zeng designed the experiments and provided the software tool; Xuyan Tu introduced the complex network. Qin Wang and Xuyan Tu completed the discussion. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Mathematical definitions of complex network measures.

Measures	Weighted Definitions
Basic concepts and notation	n is the total node of the network. $V = \{v_1, v_2, \ldots, v_n\}$ is the node set of the network. $E = \{e_1, e_2, \ldots, e_m\}$ is the edge set of the network. $W = \{w_{ij}\}$ is the set of edge weights in which w_{ij} is the weight of the connected edges between nodes v_i and v_j ($i, j = 1, 2, \ldots, n$). All the weights are normalized by the average of the weights. (i, j) is a link between v_i and v_j. a_{ij} is the connection status between v_i and v_j: $a_{ij} = 1$ when link (i, j) exists; $a_{ij} = 0$ otherwise.
Degree	Degree of node v_i [53], $k_i = \sum_{j \neq i} a_{ij}$. Weighted degree or the strength of v_i [53], $s_i = k_i^w = \sum_{j \neq i} w_{ij}$
Shortest path length	Shortest path length (distance) between v_i and v_j [53], $d_{ij}^w = \sum_{a_{uv} \in g_{i \leftrightarrow j}} \frac{a_{uv}}{(w_{uv})^\alpha}$, where $g_{i \leftrightarrow j}$ is the shortest path (geodesic) between v_i and v_j, α is a positive tuning parameter set by the users. Note that $d_{ij}^w = \infty$ for all disconnected pairs (i, j).
Average path length	Average path length [53], $L_A = \frac{\sum_{j \neq i} d_{ij}^w}{n(n-1)}$
Global clustering coefficient	Global clustering coefficient [54], $C_{Global} = \frac{\text{Total value of closed triplets}}{\text{Total values of triplets}} = \frac{\sum_{\tau_\Delta} w}{\sum_{\tau} w}$, where τ_Δ is the closed triplet and τ is any form triplet.

Table A1. *Cont.*

Measures	Weighted Definitions
Local clustering coefficient	Local clustering coefficient of a node [68], $C_i = \frac{1}{k_i(k_i-1)}\sum_{j,k}\frac{1}{\langle w_i\rangle}\frac{w_{ij}+w_{jk}}{2}a_{ij}a_{jk}a_{ik}$, where $\langle w_i\rangle = \sum_j w_{ij}/k_i$. Clustering coefficient of a network, $C_{Local} = \frac{1}{n}\sum_i C_i$
Degree distribution	Cumulative degree distribution of the network [51], $P(k) = \sum_{k'\geq k} p(k')$, Cumulative weighted degree distribution of the network [69], $P(k^w) = \sum_{k'\geq k^w} p(k')$, where $p(k')$ is the probability of a node having degree k'.
Degree centrality	Degree centrality [53], $C_D(i) = k_i \times \left(\frac{s_i}{k_i}\right)^{\alpha} = k_i^{1-\alpha} \times s_i^{\alpha}$ where α is a positive tuning parameter set by the users, here set $\alpha = 0.5$. Normalized version divides simple degree by the maximum value possible.
Closeness centrality	Closeness centrality of a node [53], $C_c(i) = \sum_j \frac{1}{d_{ij}^w}$ (set $\frac{1}{\infty} = 0$). Normalized version divides each value by n-1.
Betweenness centrality	Betweenness centrality [55], $C_b(i) = \frac{\sum_{s<t} g_{st,i}}{n_{st}}$, where $g_{st,i}$ is the number of shortest paths from v_s to v_t that pass through v_i , n_{st} is the total number of shortest paths from v_s to v_t. Normalized version divides each value by the maximum value possible.
Eigenvector centrality	Eigenvector centrality [70], $C_e(i) = \lambda_i^{-1}\sum_{j\neq i,j=1}^{n} w_{ij}e_j$, where $\widetilde{W}$ is the adjacency matrix constructed by w_{ij}, in which each eigenvalue λ_i has its own eigenvector e_i. Note that the algorithm computes the eigenvector centrality individually for each disconnected component. The centrality score of disconnected nodes is $1/n$. Normalized version divides each value by the maximum value possible.
Modularity	Modularity of a network [71], $Q = \frac{1}{2m}\sum_{i,j}\left[w_{ij} - \frac{k_i^w k_j^w}{2m}\right]\delta\left(c_i, c_j\right)$, where c_i denotes the community to which node v_i belongs. $\delta\left(c_i, c_j\right)$ is a simple delta function which denotes whether v_i and v_j are in the same community, if they are in the same community, then the value is 1, otherwise is 0; $m = \frac{1}{2}\sum_{i,j} w_{ij}$.
Measure of network small-worldness	Network small-worldness [66], $S_{Global} = \frac{C_{Global}/C_{Global}^{rand}}{L_A/L_A^{rand}}$ or $S_{Local} = \frac{C_{Local}/C_{Local}^{rand}}{L_A/L_A^{rand}}$, where C and C^{rand} are the clustering coefficients and L_A and L_A^{rand} are the average path lengths of the tested network and a random network. Small-worldness networks often have $S_w \gg 1$.
Measure of network scale-free property	A scale-free network is a network whose degree distribution follows a power law [46], $P(k) \sim k^{-\eta}$, where η is a parameter whose value is typically in the range, $1 < \eta < 2$ for the cumulative degree distribution.

References

1. Meskendahl, S. The influence of business strategy on project portfolio management and its success—A conceptual framework. *Int. J. Proj. Manag.* **2010**, *28*, 807–817. [CrossRef]
2. Clancy, T. The Standish Group Report. Available online: https://www.projectsmart.co.uk/white-papers/chaos-report.pdf (accessed on 14 June 2017).
3. Manifesto, C. Think Big, Act Small. Available online: http://athena.ecs.csus.edu/~buckley/CSc231_files/Standish_2013_Report.pdf (accessed on 10 Octorber 2016).
4. Flyvbjerg, B.; Budzier, A. Why Your IT Project May Be Riskier Than You Think. Available online: https://hbr.org/2011/09/why-your-it-project-may-be-riskier-than-you-think/ar/1 (accessed on 20 October 2016).

5. Hardy-Vallee, B. The Cost of Bad Project Management. Available online: http://www.gallup.com/businessjournal/152429/cost-bad-management.aspx#1 (accessed on 1 November 2016).
6. Whittaker, B. What went wrong? Unsuccessful information technology projects. *Inf. Manag. Comput. Secur.* **1999**, *7*, 23–30. [CrossRef]
7. Crawford, J.K. *The Strategic Project Office*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2010.
8. Morris, P.; Pinto, J.K. *The Wiley Guide to Project, Program, and Portfolio Management*; John Wiley & Sons: Hoboken, NJ, USA, 2010; Volume 10.
9. Institute, P.M. *The Standard for Portfolio Management*, 3rd ed.; Project Management Institute, Inc.: Newtown Square, PA, USA, 2013.
10. Killen, C.P.; Jugdev, K.; Drouin, N.; Petit, Y. Advancing project and portfolio management research: Applying strategic management theories. *Int. J. Proj. Manag.* **2012**, *30*, 525–538. [CrossRef]
11. Cooper, R.G.; Edgett, S.J.; Kleinschmidt, E.J. *Portfolio Management for New Products*, 2nd ed.; Basic Books: New York, NY, USA, 2002.
12. Jonas, D.; Kock, A.; Gemünden, H.G. Predicting project portfolio success by measuring management quality—A longitudinal study. *IEEE Trans. Eng. Manag.* **2013**, *60*, 215–226. [CrossRef]
13. Teller, J.; Kock, A. An empirical investigation on how portfolio risk management influences project portfolio success. *Int. J. Proj. Manag.* **2013**, *31*, 817–829. [CrossRef]
14. Kwak, Y.H.; Stoddard, J. Project risk management: Lessons learned from software development environment. *Technovation* **2004**, *24*, 915–920. [CrossRef]
15. Sanchez, H.; Robert, B.; Bourgault, M.; Pellerin, R. Risk management applied to projects, programs, and portfolios. *Int. J. Manag. Proj. Bus.* **2009**, *2*, 14–35. [CrossRef]
16. Chang Lee, K.; Lee, N.; Li, H. A particle swarm optimization—Driven cognitive map approach to analyzing information systems project risk. *J. Am. Soc. Inf. Sci. Technol.* **2009**, *60*, 1208–1221. [CrossRef]
17. Beer, M.; Wolf, T.; Zare Garizy, T. Systemic Risk in IT Portfolios—An Integrated Quantification Approach. In Proceedings of the Thirty Sixth International Conference on Information Systems, Fort Worth, TX, USA, 13–16 December 2015.
18. Drake, J.R.; Byrd, T.A. Risk in information technology project portfolio management. *J. Inf. Technol. Theory Appl.* **2006**, *8*, 1–11.
19. Hofman, M.; Grela, G. Project Portfolio Risk Identification—Application of Delphi Method. *J. Bus. Econ.* **2015**, *6*, 1857–1867. [CrossRef]
20. Chien, C.F. A portfolio—Evaluation framework for selecting R&D projects. *R D Manag.* **2002**, *32*, 359–368.
21. Killen, C.P.; Kjaer, C. Understanding project interdependencies: The role of visual representation, culture and process. *Int. J. Proj. Manag.* **2012**, *30*, 554–566. [CrossRef]
22. Schwalbe, K. *Information Technology Project Management*, 8th ed.; Cengage Learning: Boston, MA, USA, 2015.
23. Peffers, K.; Gengler, C.E.; Tuunanen, T. Extending critical success factors methodology to facilitate broadly participative information systems planning. *J. Manag. Inf. Syst.* **2003**, *20*, 51–85.
24. Salmeron, J.L.; Herrero, I. An AHP-based methodology to rank critical success factors of executive information systems. *Comput. Stand. Interfaces* **2005**, *28*, 1–12. [CrossRef]
25. Grabher, G. Cool projects, boring institutions: Temporary collaboration in social context. *Reg. Stud.* **2002**, *36*, 205–214. [CrossRef]
26. Joslin, R.; Müller, R. New Insights into Project Management Research: A Natural Sciences Comparative. *Proj. Manag. J.* **2015**, *46*, 73–89. [CrossRef]
27. Cowan, R.; Jonard, N. Network structure and the diffusion of knowledge. *J. Econ. Dyn. Control* **2004**, *28*, 1557–1575. [CrossRef]
28. Killen, C.P. Evaluation of project interdependency visualizations through decision scenario experimentation. *Int. J. Proj. Manag.* **2013**, *31*, 804–816. [CrossRef]
29. Durant-Law, G. Network Project Management: Visualising Collective Knowledge to Better Understand and Model a Project-Portfolio. Ph.D. Thesis, University of Canberra, Canberra, Australia, 2012.
30. Mikkola, J.H. Portfolio management of R&D projects: Implications for innovation management. *Technovation* **2001**, *21*, 423–435.
31. Warglien, M.; Jacobides, M.G. The power of representations: From visualization, maps and categories to dynamic tools. In Proceedings of the Academy of Management Meeting, Montreal, QC, Canada, 6 August 2010.

32. Fang, C.; Marle, F.; Zio, E.; Bocquet, J.-C. Network theory-based analysis of risk interactions in large engineering projects. *Reliab. Eng. Syst. Saf.* **2012**, *106*, 1–10. [CrossRef]
33. Unger, B.N.; Gemünden, H.G.; Aubry, M. The three roles of a project portfolio management office: Their impact on portfolio management execution and success. *Int. J. Proj. Manag.* **2012**, *30*, 608–620. [CrossRef]
34. Cook, A.; Zanin, M. Complex Network Theory. In *Complexity Science in Air Traffic Management*; Routledge: London, UK, 2016; pp. 9–22.
35. Scott, J. *Social Network Analysis*, 3rd ed.; SAGE Publications Ltd.: London, UK, 2012.
36. Shannon, C.E. A mathematical theory of communication. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2001**, *5*, 3–55. [CrossRef]
37. Winter, M.; Szczepanek, T. *Images of Projects*; Gower Publishing, Ltd.: Surrey, UK, 2009.
38. Patterson, M.G.; West, M.A.; Lawthom, R.; Nickell, S. *Impact of People Management Practices on Business Performance*; Institute of Personnel and Development London: London, UK, 1997.
39. Arifin, R.S.; Moersidik, S.S.; Soesilo, E.T.B.; Hartono, D.M.; Latief, Y. Dynamic project interdependencies (PI) in optimizing project portfolio management (PPM). *Int. J. Technol.* **2015**, *5*, 828–837. [CrossRef]
40. Tasevska, F.; Toropova, O. Management of Project Interdependencies in a Project Portfolio. Master Thesis, Umeå University, Umeå, Sweden, 2013.
41. Rungi, M.; Hilmola, O.-P. Interdependency management of projects: Survey comparison between Estonia and Finland. *Balt. J. Manag.* **2011**, *6*, 146–162. [CrossRef]
42. Zuluaga, A.; Sefair, J.A.; Medaglia, A.L. Model for the selection and scheduling of interdependent projects. In Proceedings of the 2007 IEEE Systems and Information Engineering Design Symposium, Charlottesville, VA, USA, 27 April 2007.
43. Quimby, P. The Portfolio of Programs and Projects as a Network. Available online: http://www.caskllc.com/2014/05/the-portfolio-of-programs-and-projects-as-a-network/ (accessed on 1 November 2016).
44. Pilato, C.M.; Collins-Sussman, B.; Fitzpatrick, B. *Version Control with Subversion*, 2nd ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2008.
45. Uzzi, B.; Spiro, J. Collaboration and creativity: The small world problem 1. *Am. J. Sociol.* **2005**, *111*, 447–504. [CrossRef]
46. Albert, R.; Barabási, A.-L. Statistical mechanics of complex networks. *Rev. Mod. Phys.* **2002**, *74*. [CrossRef]
47. Blondel, V.D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
48. Park, K.; Yilmaz, A. A social network analysis approach to analyze road networks. In Proceedings of the ASPRS Annual Conference, San Diego, CA, USA, 26–30 April 2010.
49. Boccaletti, S.; Latora, V.; Moreno, Y.; Chavez, M.; Hwang, D.-U. Complex networks: Structure and dynamics. *Phys. Rep.* **2006**, *424*, 175–308. [CrossRef]
50. Watts, D.; Strogatz, S. Collective dynamics of 'small-world' networks. *Nature* **1998**, *393*, 440–442. [CrossRef] [PubMed]
51. Barabási, A.-L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512. [PubMed]
52. Borgatti, S.P.; Everett, M.G. A graph-theoretic perspective on centrality. *Soc. Netw.* **2006**, *28*, 466–484. [CrossRef]
53. Opsahl, T.; Agneessens, F.; Skvoretz, J. Node centrality in weighted networks: Generalizing degree and shortest paths. *Soc. Netw.* **2010**, *32*, 245–251. [CrossRef]
54. Opsahl, T.; Panzarasa, P. Clustering in weighted networks. *Soc. Netw.* **2009**, *31*, 155–163. [CrossRef]
55. Freeman, L.C. Centrality in social networks conceptual clarification. *Soc. Netw.* **1979**, *1*, 215–239. [CrossRef]
56. Bavelas, A. Communication patterns in task-oriented groups. *J. Acoust. Soc. Am.* **1950**, *22*, 725–730. [CrossRef]
57. Brandes, U. A faster algorithm for betweenness centrality. *J. Math. Sociol.* **2001**, *25*, 163–177. [CrossRef]
58. Bonacich, P. Some unique properties of eigenvector centrality. *Soc. Netw.* **2007**, *29*, 555–564. [CrossRef]
59. Fortunato, S.; Castellano, C. Community structure in graphs. In *Computational Complexity: Theory, Techniques, and Applications*; Springer: New York, NY, USA, 2012; pp. 490–512.
60. Newman, M.E.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113. [CrossRef] [PubMed]
61. Tan, Y.-J.; Wu, J. Network structure entropy and its application to scale-free networks. *Syst. Eng. Theory Pract.* **2004**, *6*, 1–3.
62. Bianconi, G. Entropy of network ensembles. *Phys. Rev. E* **2009**, *79*, 036114. [CrossRef] [PubMed]

63. Benston, G.J.; Hanweck, G.A.; Humphrey, D.B. Scale economies in banking: A restructuring and reassessment. *J. Money Credit Bank.* **1982**, *14*, 435–456. [CrossRef]
64. Erdös, P.; Rényi, A. On random graphs I. *Publ. Math. Debrecen* **1959**, *6*, 290–297.
65. Gephi Software. Available online: https://gephi.org/ (accessed on 15 October 2016).
66. Humphries, M.D.; Gurney, K. Network 'small-world-ness': A quantitative method for determining canonical network equivalence. *PLoS ONE* **2008**, *3*, e0002051. [CrossRef] [PubMed]
67. Hu, Y. Algorithms for visualizing large networks. *Comb. Sci. Comput.* **2011**, *5*, 180–186.
68. Barrat, A.; Barthelemy, M.; Pastor-Satorras, R.; Vespignani, A. The architecture of complex weighted networks. *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 3747–3752. [CrossRef] [PubMed]
69. Rubinov, M.; Sporns, O. Complex network measures of brain connectivity: Uses and interpretations. *Neuroimage* **2010**, *52*, 1059–1069. [CrossRef] [PubMed]
70. Newman, M.E. Analysis of weighted networks. *Phys. Rev. E* **2004**, *70*, 056131. [CrossRef] [PubMed]
71. Newman, M.E. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [CrossRef] [PubMed]