An Improved LeaderRank Algorithm for Identifying Critical Components in Service-Oriented Systems

Shuo Jiang¹,a Xiuguo Zhang²,b* Zhiying Cao³,c
¹Dalian Maritime University
Dalian, China
²Dalian Maritime University
Dalian, China
³Dalian Maritime University
Dalian, China
¹17824853961@163.com ²zhangxg@dlmu.edu.cn ³czysophy@dlmu.edu.cn

ABSTRACT: Identifying critical components of Service-Oriented Systems is of great significance to the overall reliability of the system. As the size of software systems increases, identifying critical components can reduce the number of components that need to be predicted and shorten the prediction time in the process of predicting system reliability. Moreover, predicting the reliability of critical components can also ensure the stability of the system. Therefore, we propose a method for identifying the critical components in Service-Oriented Systems. This method transforms the interaction between service components of Service-Oriented Systems into service dependency graph. An improved weighted LeaderRank algorithm is used to measure the importance of components and obtain the sequence of critical components. Through experimental analysis, the method can accurately and efficiently identify critical components in the system.

CCS Concepts
• Information system→content ranking • Networks→Network components • Theory of computation→Evolutionary algorithms.

1. INTRODUCTION
The reliability prediction of the Service-Oriented Systems can reduce the occurrence of unexpected situations to ensure the stable operation of the Service-Oriented Systems. At present, in order to meet more needs, the scale of the system is gradually expanding, and its complexity is also increasing. However, identifying critical components before predicting systems’ reliability and analyzing the reliability of them not only reduces computational complexity but also reduces evaluation time. Therefore, identifying the critical components of the Service-Oriented Systems is important for maintaining system stability. Although the problem of identifying the critical components of the system has been widely concerned in recent years, the existing methods[1,2] have not combined the two aspects of the relationship between the components and the characteristics of the service itself. In order to pursue better sorting results, this paper proposes an efficient method to identify the critical components of the Service-Oriented Systems. Using the improved weighted LeaderRank algorithm to identify the critical components of the Service-Oriented Systems, which takes into account both service heat and the degree of dependencies between services, is helpful to quickly identify the most critical components.
important components in the systems and ensure their reliability. Taking the tourism Service-Oriented System as the experimental research object, the critical components of the system are identified by analyzing the interaction among the system components. Compared with the existing excellent algorithms, the advantages of the improved LeaderRank algorithm in identifying critical components of the system are verified.

The rest of this paper is organized as follows. Section II presents the related works. Section III provides the process of constructing the system service dependency graph. Section IV presents the approach to identifying the critical components. Section V is mainly about the experimental process, and discusses our results by comparing with related work. Finally, Section VI draws the conclusions of this paper.

2. RELATED WORK
In order to identify the critical components in Service-Oriented Systems, the existing methods are generally studied from two aspects: the importance measurement of system components or the analysis of software system architecture.

There are many methods to measure the importance of components. Zheng et al. [3] thought the significance value of a component was determined by the number of components that invoke this component, the significance values of these components, how often the current component is invoked by other components, and the component characteristics. Then, two component importance ranking methods were proposed. However, this method does not consider the characteristics of components, such as throughput, response time and so on, when calculating link weights.

On the other hand, most of the traditional software key class recognition methods are based on the network model. Ding et al. [4] built software network models, then identified important classes in software systems by h-index method.

In addition, the critical components can be found efficiently and accurately by analyzing the dependencies among software classes. Ioana Şora et al. [5] used fuzzy rules having as inputs class attributes such as size, weighted incoming dependencies, weighted outgoing dependencies and PageRank value to produce the best results.

There are many critical components identification methods of software systems be proposed. However, they are all measured unilaterally. Most of the methods do not combine the two parts for more reasonable research.

3. SERVICES-ORIENTED SYSTEMS SERVICE DEPENDENCY GRAPH MODEL
Services-oriented systems service dependency graph (SSDG) is the basis of identifying critical service components and reliability prediction of Web services. So, the process of building model is a crucial link. For the service dependency graph does not distinguish the differences of dependency relationship, in order to describe the interaction between system services more accurately, the service dependency graph is transformed into a weighted oriented graph. The improved dependency graph can not only contain the dependency relationship among service components, but also express the strength of service dependencies. According to the SSDG, the critical components of Service-Oriented Systems can be identified effectively. This section describes in detail the process of building a service dependency graph for a Services-oriented systems.

3.1 Service Dependency Graph Profile
The traditional service dependency graph is a graph composed of attribute nodes (input/output) and operation nodes (service), which is used to describe all possible input/output dependencies relationship between services[6].

Figure 1 shows the dependence between the hotel enquiry service and the hotel reservation service. The rectangle in the figure represents the service, and the circle represents the input and output data objects of the service. As can be seen from the figure below, 'price' and 'hotelName' are both the output data of 'hotelSearch' and the input data of 'hotelReserve'. Because 'hotelSearch' service and
service have service input/output relationship, they have service dependency relationship.

Figure 1. Service dependency graph.

3.2 Service-Oriented Systems Service Dependency Relationship

The service dependency relationship is based on the input/output between services. In the Service-Oriented Systems, the dependency relationship between services can be differentiated according to the interaction between service attributes[7]. Therefore, service dependency relationships are divided into three categories:

Definition 1. (Complete dependency relationship): For two services \( w_i \) and \( w_j \), if the attribute values of \( w_i \) is determined and the attribute values on \( w_j \) is determined accordingly, this is \( w_j \rightarrow w_i \). Service \( w_j \) is said to be completely dependent on service \( w_i \).

Definition 2. (Partial dependency relationship): For services \( w_{m1}, w_{m2}, \ldots, w_m \) and \( w_j \), if \( \{w_{m1}, w_{m2}, \ldots, w_m, w_j\} \rightarrow \{w_j\} \). So, services \( w_j \) are partially dependent on services \( w_{m1}, w_{m2}, \ldots, w_m \).

Definition 3. (Interdependence relationship): For services \( w_i \) and \( w_j \), if \( w_i \rightarrow w_j \) and \( w_j \rightarrow w_i \), so, services \( w_i \) and services \( w_j \) are interdependent.

Because of the difference in the proportion of input/output attributes among service components, these three service relationships have different service dependency intensities. According to formula 1, it is found that only the intensity of partial dependency relationship needs to be calculated according to the attribute proportion, the other intensities are 1.

\[
W_{i,j} = \frac{n}{N} \tag{1}
\]

Among them, \( W_{i,j} \) is the strength of the dependency relationship between service \( i \) and service \( j \), \( n \) is the number of attributes provided by service \( i \) to service \( j \), and \( N \) is the number of service \( j \) attributes.

3.3 SSDG

According to the above analysis, in order to better express the dependency relationship between components, we can build SSDG in the Service-Oriented Systems. Components are regarded as nodes, dependency relationships are regarded as edges, and the weight on edges represents the strength of relationships, which makes constructed model more fully describe the structural characteristics of real systems. SSDG can be described by the following definitions.

Definition 4 SSDG: SSDG = (V, E, W), where \( V = \{v_1, v_2, \ldots, v_n\} \) refers to the set of service components of Service-Oriented Systems, \( E = \{e_1, e_2, \ldots, e_m\} \) refers to the set of dependencies relationship among service components, which can be divided into three categories: complete dependency relationship, partial dependency relationship and interdependence relationship. \( W = \{w_{e_j}\} \) is the corresponding weight set of edge set E, indicating the strength of relationships.

In figure 2, users enter the system to register and login, and the system executes 'register' service and 'login' service accordingly. Users search hotels according to their needs and implement
'hotelSearch' services accordingly. Users find suitable hotels to make reservation, and the system implements 'getUserInfo' service and 'hotelResearch' service accordingly. During the booking process, the system calls the bank service to make payment, and the payment result is returned to the 'hotelReserve' service. As 'register' service and 'login' service, 'login' service and 'getUserInfo' service, 'login' service and 'hotelSearch' service are completely dependency relationships, and 'hotelReserve' service and 'bank' service is interdependence relationship, the weight of all edges is 1. Because 'getUserInfo' service and 'hotelReserve' service, 'hotelSearch' service and 'hotelReserve' service are partially dependency relationships. The weight of edge (getUserInfo, hotelReserve) is 3/5 according to formula 1, and the weight of edge (hotelSearch, hotelResearch) is 2/5.

Figure 2. Hotel reservation system SSDG.

4. CRITICAL SERVICES IDENTIFICATION METHOD

The purpose of identification method is to identify critical service components in Service-Oriented Systems according to SSDG. The critical components of a Service-Oriented Systems generally refer to components that are important to maintaining the structure and functionality of the entire system. Based on the SSDG, this paper uses the improved LeaderRank algorithm to identify the critical components of the Service-Oriented Systems.

The idea of the LeaderRank algorithm [8] is to add a common node (Ground Node) to the known directed network G(N, M), and add a reverse connection to the unidirectional connection to obtain a directional with N+1 nodes, Network G (N+1, M+2N). Since the public node has a two-way connection with other nodes in the network, G constitutes a strong connected graph, which can avoid isolated nodes in the complex network and ensure the convergence of the algorithm.

In order to improve the accuracy, many papers have proposed the improved LearnerRank algorithm. Li et al. [9] believe that nodes with more indegrees get more points from the Ground node. Therefore, the weight of the edge between the node and the Ground node is related to the degree of entry of the node itself.

In summary, the LeaderRank algorithm and the weighted LeaderRank algorithm play an important role in node ordering. Although many documents[10,11] are properly improved, these methods are still not comprehensive enough. For example, when combining the propagation ability, the characteristics of the node are not analyzed. Combine the two to achieve better measurement results. And when considering the influence of multiple factors, the calculation time is long, and the balance point of accuracy and efficiency is not achieved.

4.1 Improved weighted LeaderRank Algorithm

When the Service-Oriented System identifies critical components, the service node can be regarded as a web page, and the dependency between services is regarded as a web page link. In the SSDG, the service $w_i$ points to the service $w_j$, that is $w_i \rightarrow w_j$, represents the service $w_j$ depend on service $w_i$.

Therefore, In order to calculate the degree to which components are dependent, the SSDG adjacency matrix should be transposed before identifying critical components. In the iterative process of the LeaderRank algorithm, the LeaderRank value of the node is equally divided to the nodes that pointing to the node. However, in the actual Service-Oriented Systems, node values will be biased towards a node which strongly relies on the node, that is, a service with higher weight. At the same time, the service component with high heat is often more important than the service with low heat. For example, in the shared bicycle service in the travel system, the service has a high number of people and frequency in the spring and summer, and the service is hot at this stage. Therefore, sharing bicycle
services at this stage is more important. Therefore, considering service enthusiasm and the different dependencies between services, an improved LeaderRank algorithm is proposed, which is more suitable for identifying critical components of the Service-Oriented Systems. Its calculation formula is as follows:

\[
LR_i(t+1) = \frac{\sum_{j=1}^{N} a_{ji} \cdot w_{ij} \cdot LR_j(t) \cdot \gamma \cdot H(i)}{\sum_{j=1}^{N} a_{ji} \cdot w_{ij}}
\] (2)

\[
LR_g(t+1) = \sum_{j=1}^{N} \frac{a_{rg}}{N} LR_j(t) \cdot \gamma \cdot H(g)
\] (3)

\[
LR_i = LR_i(t_c) + \frac{LR_g(t_c)}{N}
\] (4)

Equation (2) is the iterative process of the algorithm, where \(w_{ji}\) and \(w_{ij}\) are the weights of the edge between the nodes, and the Ground node is interdependent with other nodes, so the weight of the edge is 1. When node \(i\) and node \(j\) have edges, the \(a_{ij}\) value is 1, otherwise the value is 0, \(H(i)\) is the current heat of service \(w_i\), and \(\gamma\) is the heat adjustment factor.

When calculating the Service LeaderRank value, since each edge has a weight, the values cannot be evenly distributed. Actually, it should be distributed according to the weight. The larger the weight is, the larger the value of the service is. The weight distribution calculation process can be seen from the first half of formula (2) (formula (5)).

\[
LR_i(t+1) = \sum_{j=1}^{N} \frac{a_{ji} \cdot w_{ij} \cdot LR_j(t)}{\sum_{j=1}^{N} a_{ji} \cdot w_{ij}}
\] (5)

The above iteration formula relies on the value of service \(w_j\) pointing to service \(w_i\), when calculating the service \(w_i\) LR value. Each time the score obtained from service \(w_j\) is the partial value of the LeaderRank value of \(w_j\). That is, the ratio of edge \((w_i, w_j)\) weight to the sum of edge weights between node \(w_j\) and all outgoing nodes.

\(H(i)\) is the heat of service. Considering that the value of the service component with high heat is larger, the service heat is added in the iterative process, so that make the hot service get a certain upward in the order. The degree of influence of the service heat on the importance of the service can be adjusted by \(\gamma\), and the appropriate ratio is selected according to the actual situation of the Service-Oriented Systems. Its calculation formula is

\[
H_i = \frac{\sum_{j=1}^{N} u_j}{\sum_{j=1}^{N} u_j}
\] (6)

\(u_j\) is the number of uses of service \(w_j\) is used, that is, the number of times the service \(w_j\) interface is called. \(N\) is the number of components in the system. \(\sum_{j=1}^{N} u_j\) is the total number of times all components which were called. Statistical time intervals can be selected based on component usage and system component structure changes, ranging from approximately one week to several months.

Equation (3) is an iterative process of the common node ground. Since the ground node and the existing node become weighted by 1, the weight \(w_{rg}\) is reduced to 1 in the calculation, and the formula (3) is obtained.
Equation (4) divides the value of the common node ground to other nodes after the iteration ends. The process of improving the LeaderRank algorithm is shown in Table 1.

Table 1. Improved weighted LeaderRank algorithm flow

| Algorithm 1: Improved weighted LeaderRank |
|------------------------------------------|
| **Input:** SSDG Adjacent Transfer Matrix, Number of Service Calls, Service Thermal Coefficient $\gamma$ |
| **Output:** components LeaderRank Value sequence |
| 1. Calculate the heat of each service component according to formula (6) |
| 2. $\Delta \leftarrow 1$ |
| 3. while ($\Delta > 10^{-5}$) do |
| 4. $LR(t+1) = \sum_{j} \frac{a_{ji} \cdot w_{ji}}{\sum_{j} \frac{a_{ji} \cdot w_{ji}}{2}} LR_j(t) \cdot \gamma \cdot H(i)$ |
| 5. $\Delta = \text{abs}(LR_i(t+1) - LR_i(t))$ |
| 6. $t \leftarrow t + 1$ |
| 7. End |
| 8. $LR_i = LR_i(t_e) + \frac{LR_i(t_e)}{N}$ |
| 9. return $LR_i$ |

The improved weighted LeaderRank algorithm can get the ranking of service components in the Service-Oriented Systems, and the more advanced the service, the more important it is. In order to get the candidate critical components, filter the service component ranking list and select the top-$k$ component as the candidate critical component.

5. EXPERIMENTS

5.1 Experimental Data
This paper takes the tourism Service-Oriented Systems as the experimental object. The system is designed and implemented in our laboratory. It mainly includes map navigation, hotel reservation, scenic spot introduction, ticket purchase and other functions. Its system service components are mainly obtained from service websites, like programmable web (https://www.programmableweb.com). The SSDG consists of 58 nodes and 136 edges.

5.2 Experimental Results and Analysis

5.2.1 Improved Weighted LeaderRank Algorithms to Identify the Effectiveness of Critical Components in System
In order to compare the difference in the importance of identification between this method and other methods, recall rate and precision rate are used as evaluation criteria. Its calculation formula is as follows:

\[
\text{recall} = \frac{W \cap K}{K} \quad (7)
\]

\[
\text{precision} = \frac{W \cap K}{W} \quad (8)
\]

In formula (7) (8), $W$ is the critical component set identified by the evaluation method and $K$ is the critical component set known in the system. The recall rate is the ratio of the number of identified correct critical components to the number of all known critical components in the system. Precision rate is the ratio of the number of identified correct critical components to the number of required
critical components. As can be seen from the formula, the two evaluation criteria are only different in denominator.

The improved weighted LeaderRank (ICLederRank) algorithm proposed in this paper is compared with the following excellent algorithms: FTCloudel [3], FLC-4 input [5], G-NWD [4], weighted LeaderRank (W-LeaderRank) [9].

Figure 3 shows the changes in recall rates of various methods for identifying critical service components in tourism systems. As can be seen from the figure, ICAN and W-LeaderRank are ineffective in identifying critical components. The reason is that these methods only focus on whether there are dependency relationships among components and do not further analyze the differences of relationships, so they can not better distinguish the importance of components in ranking. The recall rate of FLC-4 input method is higher than FTCloudel. Overall, these two methods have better recognition effect. The results of ICLederRank method in the first half are slightly better than those of other algorithms, However, there is no significant difference in recall rates between 60% and 100% component sequences.

![Figure 3. Method recall rate.](image1)

The comparison of accuracy rate about five methods of identifying the critical components of tourism system is shown in Fig. 4. We can clearly see that ICAN and W-LeaderRank have low accuracy, which is closely related to the inadequacy of the two algorithms. However, the accuracy of the first 20% sequence of W-LeaderRank algorithm is significantly higher than that of ICAN method, which is close to that of FTCloudel algorithm. But overall, the accuracy of FTCloudel algorithm in identifying critical components is higher than the first methods. FLC-4 input and ICLederRank have the best accuracy and are superior to other algorithms.

![Figure 4. Method Accuracy rate.](image2)

According to Figure 3 and Figure 4, we can see that the recall rate increases and the accuracy rate decreases with the increase of component ranking ratio. Therefore, we need to select the appropriate Top-k component as the critical components of the system. Through experiments of Literature[12], it is found that for all systems, the most important classes always rank in the top 10, almost independent of tools. In our experiments, we found that top10 component is a better choice, and it is suitable to be the critical component of the system. The results of Table 2 show that the recall rate of each top5 method is low and some important service components can not be identified. So the reliability of the system as a whole can not be guaranteed. Although the recall rate of top15 has increased, it has not been greatly improved, while the accuracy rate has been significantly reduced. Recall rate and
accuracy rate results of top10 are good, so we can use ICLederRank method to measure the results of top 10 as the critical component of the tourism Service-Oriented Systems.

Table 2. Comparing Statistical Results of Experiments

| method          | recall(%) | precision(%) |
|-----------------|-----------|--------------|
|                 | top5      | top10 | top15 | top5 | top10 | top15 |
| FTCloude1       | 27.3      | 45.5   | 54.5  | 60.0 | 50.0  | 40.0  |
| FLC-4 input     | 36.4      | 63.6   | 63.6  | 80.0 | 70.0  | 46.7  |
| G-NWD           | 18.2      | 45.5   | 54.5  | 40.0 | 50.0  | 40.0  |
| W-LeaderRank    | 27.3      | 54.5   | 54.5  | 60.0 | 60.0  | 40.0  |
| ICLederRank     | 36.4      | 72.7   | 72.7  | 80.0 | 80.0  | 53.3  |

6. CONCLUSION

In this paper, service heat and the degree of dependencies between services are integrated into the LeaderRank algorithm, and an ICLederRank algorithm is proposed to identify the critical components of the Service-Oriented Systems. The main contribution of this paper is as follows:

1. Because of the different dependency intensities among services, an SSDG is constructed to represent the interaction between system components by adding weights to dependency relationships. Using this graph can better measure components importance.

2. In this paper, ICLederRank algorithm is proposed to identify the critical components of Service-Oriented Systems. Through this method, the critical components sequence can be obtained effectively, the prediction time can be reduced and the system reliability can be guaranteed.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 51479021, 51679025) and the Fundamental Liaoning Provincial Natural Science Foundation of China(Grant No.20170520196).

REFERENCES

[1] Wang J , Ai J , Yang Y , et al. 2018. Identifying key classes of object-oriented software based on software complex network International Conference on System Reliability & Safety. IEEE.

[2] Mala D J , Praba M R. 2012. Critical components identification and verification for effective software test prioritization Third International Conference on Advanced Computing. IEEE.

[3] Zheng Z , Zhou T C , Lyu M R , et al. 2012. Component Ranking for Fault-Tolerant Cloud Applications. IEEE Transactions on Services Computing,5(4):540-550.

[4] Ding X, Li B, Cheng C, et al. 2017(4). A Method of Measuring the Importance of Software Network Nodes Based on H Index Variant. Engineering Science and Technology.

[5] Ioana Șora, Todinca D. 2016. Using fuzzy rules for identifying key classes in software systems. IEEE International Symposium on Applied Computational Intelligence & Informatics.

[6] Guo Y , Chen S , Feng Z. 2011. Composition oriented web service semantic relations research International Joint Conference on Service Sciences. IEEE.

[7] Zhang Y M , Ye C L, Huang LY , et al. 2018. Research on Data Service Dependency Graph Model and Automatic Composition Method.Journal of Chinese Mini-Micro Computer Systems.

[8] Lü L, Zhang Y C, Chi H Y, et al. 2011. Leaders in Social Networks, the Delicious Case. Plos One, 6(6):e21202.

[9] Li Q , Zhou T , Lü, Linyuan, et al. 2014. Identifying influential spreaders by weighted LeaderRank. Physica A: Statistical Mechanics and its Applications, 404:47-55.

[10] Shuang X , Pei W , Chun-Xia Z , et al. 2018. Spectral Learning Algorithm Reveals Propagation Capability of Complex Networks. IEEE Transactions on Cybernetics:1-9.
[11] Zhang Z H, Jiang G P, Song Y R, et al. 2017. An Improved Weighted LeaderRank Algorithm for Identifying Influential Spreaders in Complex Networks IEEE International Conference on Computational Science & Engineering. IEEE.

[12] Sora I. 2015. A PageRank based recommender system for identifying key classes in software systems IEEE Jubilee International Symposium on Applied Computational Intelligence & Informatics.