

# Instructions for Authors of WDCOPIN

Rodrigo Souza<sup>1</sup>

<sup>1</sup>Departamento de Sistemas e Computação – Universidade Federal de Campina Grande (UFCG)  
Caixa Postal 10.106 – CEP 58.109-970 – Campina Grande – PB – Brazil

rodrigo@dsc.ufcg.edu.br

*Abstract.*

*Resumo.*

## 1. Introdução

A divisão conceitual de um sistema de software em módulos é uma informação valiosa durante o seu desenvolvimento. Uma boa organização modular revela subconjuntos de um sistema que podem ser desenvolvidos por equipes trabalhando de forma mais ou menos independente, o que contribui para reduzir o tempo de implementação. Apesar disso, o conhecimento sobre a organização de um sistema muitas vezes é mal documentado e acaba se perdendo à medida que os desenvolvedores são substituídos.

Algoritmos de modularização de software procuram identificar a organização modular de um sistema de software a partir de sua implementação e, por isso, auxiliam a divisão de tarefas entre desenvolvedores. Para realizar a decomposição de um sistema em módulos, muitos desses algoritmos usam heurísticas baseadas nas dependências existentes entre os componentes básicos da implementação do sistema.

Em sistemas implementados em linguagens orientadas a objetos, os componentes básicos são as classes e as interfaces, as quais se relacionam através de mecanismos como herança e agregação. Essas relações estabelecem dependências entre os componentes que, unidas, formam uma rede de dependências, também chamada de rede de software. Formalmente, a entrada de um algoritmo de modularização de software é uma rede de software, modelada como grafo orientado, e a saída é um particionamento dessa rede.

Recentemente alguns pesquisadores analisaram redes de software sob a luz da teoria das redes complexas e encontraram propriedades topológicas marcantes que também estão presentes em redes de interações entre proteínas, redes sociais e muitas outras [cite]. Descobriu-se que nessas redes a distribuição dos graus dos vértices é bem aproximada por uma lei de potência,  $N(k) \propto k^{-\gamma}$  — o número de vértices ligados a exatamente  $k$  arestas é proporcional a uma potência de  $k$  com expoente negativo e constante, como mostra a Figura 1. Redes caracterizadas por essa distribuição de graus específica são chamadas de redes livres de escala.

Para tentar explicar os mecanismos responsáveis pela formação de redes livres de escala em diversos domínios, pesquisadores criaram vários modelos estocásticos. Esses modelos são algoritmos que geram vértices e arestas de maneira probabilística porém de acordo com certas regras que garantem que, quando o número de vértices tende a infinito, a distribuição dos graus dos vértices tende a uma lei de potência. Por essa razão tais modelos podem produzir redes similares a redes de software, ao menos quanto à distribuição de graus.

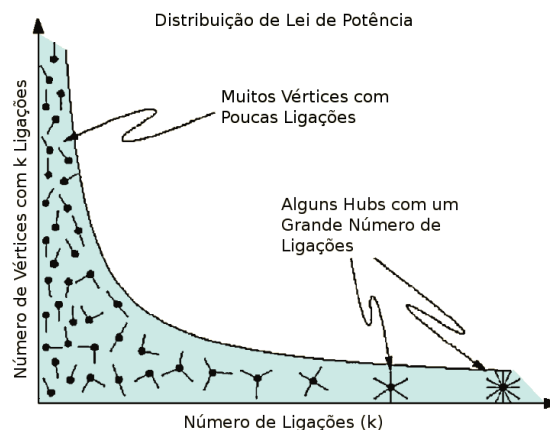


Figure 1. Distribuição de graus como lei de potência. Adaptado de [?].

## 2. Identificação do Problema

Idealmente os algoritmos de modularização de software devem encontrar organizações modulares semelhantes àquelas que seriam encontradas por especialistas nos sistemas analisados. Uma forma de testar os algoritmos consiste, pois, em aplicá-los ao código-fonte de um sistema cuja organização modular esteja documentada e então comparar a organização encontrada com a organização documentada. A diferença entre as organizações pode ser quantificada por uma métrica de distância entre particionamentos [cite Mojo etc].

Infelizmente é difícil encontrar sistemas cuja organização modular é bem documentada, e por isso a avaliação de algoritmos de modularização de software é deficiente. Os trabalhos presentes na literatura sobre a avaliação desses algoritmos se limitam a estudos de caso. [cite]. Para que os resultados sejam estatisticamente significativos e, portanto, generalizáveis, seria necessário realizar estudos em larga escala, com uma grande amostra de sistemas com organização modular conhecida.

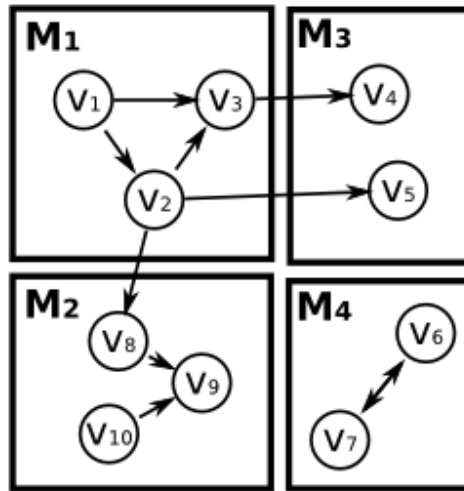
Como não existe um *benchmark* para esses algoritmos, qualquer reivindicação do tipo “o algoritmo A é melhor do algoritmo B” deve ser olhada com desconfiança. Essa lacuna representa uma barreira para a adoção dos algoritmos na indústria e um obstáculo para o avanço das pesquisas.

## 3. Objetivos

Para suprir a escassez de sistemas cuja organização modular é documentada, propomos o uso de modelos estocásticos capazes de sintetizar redes de software organizadas em módulos para servirem como conjunto de teste. Para que essa abordagem seja bem-sucedida, no entanto, é preciso que as redes sintéticas sejam realistas, isto é, semelhantes a redes extraídas de sistemas de software reais.

Assim, o objetivo desta pesquisa é analisar modelos estocásticos que possam ser usados para avaliar algoritmos de modularização de software. Esse objetivo pode ser decomposto nos objetivos específicos apresentados a seguir.

Descobrir modelos estocásticos de redes livres de escala organizadas em módulos. A maioria dos modelos disponíveis na literatura ignora a organização modular.



**Figure 2.**

Desenvolver um método para avaliar o realismo de uma rede. Esse método deve encontrar um grau de realismo alto para redes de software reais e um grau de realismo baixo para redes de outros domínios.

Avaliar o realismo dos modelos estocásticos. Um modelo é realista se ele é capaz de produzir redes realistas.

Avaliar algoritmos de modularização através de sua aplicação a um grande número de redes sintéticas. Os resultados poderão ser comparados com resultados experimentais encontrados na literatura.

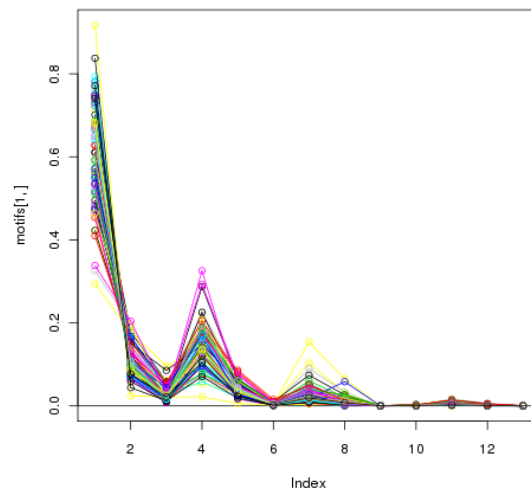
#### **4. Atividades Planejadas e Realizadas**

Foram encontrados na literatura dois modelos que produzem redes organizadas em módulos [cite]. Um terceiro modelo de redes modulares foi desenvolvido a partir da adaptação de um modelo de redes não-modulares [cite].

Para apoiar a avaliação de realismo dos modelos, coletamos um corpo de 65 sistemas de software escritos na linguagem Java. A rede de dependências entre componentes de cada sistema foi extraída através da ferramenta Dependency Finder [footnote]. Coletamos ainda cerca de cem redes de diversos domínios, sobretudo redes sociais e redes biológicas. De cada rede (do domínio de software ou de outros domínios) foram extraídas diversas métricas, tais como número de vértices, número de arestas, número médio de arestas por vértice, o expoente  $\gamma$  da distribuição de graus, entre outras.

Foi encontrada na literatura uma métrica de distância entre duas redes [cite]. Essa métrica foi utilizada em um experimento, descrito em detalhes em um artigo não publicado [footnote], que procurou avaliar o realismo dos modelos. Essa métrica de distância foi, no entanto, descartada, uma vez que indicou que uma rede biológica está mais próxima de redes de software do que as redes de software estão próximas entre si.

No momento estamos procurando outro método para avaliar o realismo de redes



**Figure 3.**

e, conseqüentemente, de modelos.

Com isso pronto, queremos publicar. Afinal, o setup experimental está pronto.

A seguir pretendemos avaliar algoritmos de modularização de software.

E então escrever a dissertação.

Por fim, publicar resultado completo

## **5. Resultados Obtidos**

Como já foi mencionado, um dos resultados da pesquisa foi a concepção e a implementação de um modelo de redes modulares. Ainda não foi possível comparar esse modelo aos dois modelos encontrados na literatura, uma vez que ainda não encontramos uma métrica para quantificar o realismo de uma rede de software. Já descobrimos, no entanto, que a métrica de distância entre redes encontrada em [cite] não é adequada.

Estamos trabalhando em uma nova métrica de realismo e já temos alguns resultados.