

# Grafo Adaptativo: uma Proposta de Topologia Adaptativa

Francisco Supino Marcondes<sup>1</sup> e Miguel Ângelo Tancredi Molina<sup>1</sup>

**Abstract**—This paper presents an adaptive version to “normal” graph called adaptive graph. An adaptive graph is graph whose vertices and edges can be modified in order to perform topological variation on it. As concept proof it was chosen the VNE problem, an NP-hard routing problem. The adaptive graph provided an way to reduce computation time aiding the practical use of such technology. Since some edges can be ‘hidden’ on substract network they cannot be considered by routing algorithms decreasing computation effort.

**Keywords:** graph, topoly, VNE, adaptivity

## I. INTRODUÇÃO

Embora o formalismo adaptativo tenha sido proposto para reconhecedores de linguagem, o objeto de interesse neste artigo é estudar sua estrutura topológica na forma de uma grafo. Assim, este estudo trata da adaptatividade sobre grafos a partir de uma adaptação do autômato adaptativo. Os grafos são objeto de interesse em diversas áreas da computação como estrutura de dados, inteligência artificial e redes de computadores.

O objetivo principal deste estudo é apresentar uma proposta de topologia adaptativa com base em um grafo chamado *grafo adaptativo*.

A aplicação escolhida para exemplificar o uso do grafo adaptativo proposto é o mapeamento de enlaces de redes virtuais em sua respectiva estrutura física também chamada de rede de substrato (*substract network*) esse problema é conhecido como VNE (ou *virtual network embedding*). O grafo adaptativo proposto será então aplicado como estrutura de descrição para redes virtuais cuja topologia possa ser alterada visando atender diferentes necessidades ou restrições. Naturalmente tal alteração topológica influi nos percursos sobre o grafo, isto é, nos caminhos de roteamento na rede de substrato. Por isso, a prova de conceito será realizada tendo como base a álgebra de caminhos.

## II. TOPOLOGIA ADAPTATIVA

O termo *topologia* remete à estrutura de um grafo (ou grafo estático). Portanto, o termo *topologia adaptativa* remete ao grafo capaz de alterar sua estrutura. Em outros termos um *grafo adaptativo* consiste em um grafo onde se pode incluir e remover vértices e arestas.

Convém diferenciar a especificação matemática de grafo de sua implementação computacional. Matematicamente um grafo é uma estrutura estática onde não se permite a inclusão ou remoção de vértices e arestas. No entanto, computacionalmente antes de utilizar o grafo é preciso constituí-lo, para isso é necessário associar vértices e arestas à endereços na

memória do computador. Deste tipo de necessidade surgem operações do tipo *addEdge* [8]. Isso não deve ser confundido com a possibilidade de alteração topológica, isto é, não se deve confundir uma necessidade de implementação com uma característica de especificação.

O que se propõe neste estudo é justamente um grafo cuja natureza seja adaptativa, isto é, uma estrutura diversa da proposta pelo grafo estático embora derivada dela.

### A. Breve Revisão da Teoria de Grafos

O grafo é uma forma de representação comum, utilizada para expressar diferentes tipos de autômatos. Formalmente:

Um grafo  $G$  consiste em uma tripla ordenada  $(V(G), E(G), \psi_G)$  onde  $V(G)$  é um conjunto não vazio de vértices,  $E(G)$  um conjunto de arestas disjunto de  $V(G)$  e  $\psi_G$  é uma função de incidência que associa cada aresta com um par não ordenado e não necessariamente distinto de arestas. Se  $a$  é uma aresta e  $u$  e  $v$  são vértices, a função de incidência toma a forma  $\psi_G(a) = uv$ , ou seja, se diz que  $a$  conecta  $u$  e  $v$ ; os vértices  $u$  e  $v$  são chamados extremidades de  $a$ . [1, tradução do autor]

Existem ao menos duas formas equivalentes para descrever a topologia de um grafo: diagramático (círculos e retas) e matricial (adjacências ou incidência). A figura 1 apresenta os dois tipos de representação. Na figura,  $G$  representa a forma diagramática,  $M(G)$  a matriz de incidência e  $A(G)$  a matriz de adjacências. O valor de cada elemento  $a_{ij}$  na matriz de incidência ( $M(G)$ ) representa o número de vezes que uma aresta incidiu sobre um vértice; os na matriz de adjacências ( $A(G)$ ) representa o número de arestas que conecta dois vértices (caso o valor seja 0 significa que os dois vértices não estão conectados).

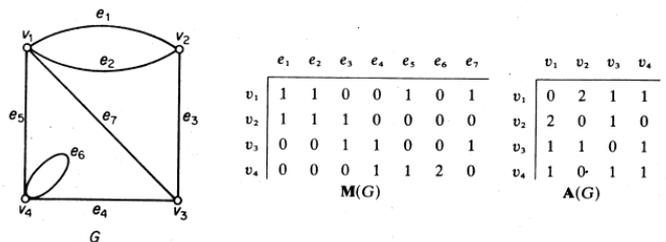


Figura 1. Maneiras para descrever a topologia de um grafo cf. [1].

Quando dois vértices estão ligados por uma aresta são chamados de vértices adjacentes e esta aresta pode ser interpretada como uma relação binária denominada adjacência; a matriz de adjacências é sempre uma matriz simétrica de dimensão  $n \times n$ .

Para [1], é possível transitar por um grafo de três formas: percurso (*walk*), trilha (*trail*) e caminho (*path*). Um percurso

<sup>1</sup> Instituto Federal de Educação, Ciência e Tecnologia - São Paulo {supino|miguel.molina}@ifsp.edu.br

consiste em uma sequência qualquer, porém não vazia, de vértices e arestas alternados, por exemplo,  $v_1e_1v_2e_3v_3\dots e_iv_i$ . Uma trilha consiste em um percurso onde não é permitido a repetição de vértices. Um caminho é uma trilha onde não se admite a repetição de arestas. Portanto, um *caminho* consiste em uma sequência não vazia e sem repetição de vértices e arestas alternados.

Um subgrafo consiste em um subconjunto de vértices e arestas do grafo original. Formalmente, o grafo  $H$  é um subgrafo de  $G$  ( $H \subseteq G$ ) se  $V(H) \subseteq V(G)$ ,  $E(H) \subseteq E(G)$  e se  $\psi_H$  restringir  $\psi_G$ . Se  $H$  é subgrafo de  $G$ , então  $G$  é super-grafo de  $H$ .

### B. Aresta Adaptativa

Um autômato adaptativo consiste em um dispositivo capaz de alterar seu comportamento dinamicamente conforme o estado corrente e o símbolo consumido na cadeia de entrada [7], ou seja, se trata de um formalismo que especifica máquinas de estado auto-modificáveis. Pode-se dividir a operação de um autômato adaptativo em dois recursos distintos. Um deles consiste no reconhecedor subjacente e o outro no dispositivo adaptativo [9]. Em outros termos, o dispositivo adaptativo realiza modificações na estrutura do reconhecedor subjacente. O dispositivo adaptativo pode operar sobre reconhecedores dos diversos tipos de linguagem, por exemplo, pode-se utilizar um autômato finito como dispositivo subjacente ao dispositivo adaptativo [5].

Todo autômato adaptativo pode ser visto como uma máquina de estados que, ao início de sua operação, apresenta uma topologia fixa, predeterminada, denominada máquina de estados inicial  $E_0$ . Na topologia dada por qualquer máquina de estados  $E_i$ , o autômato adaptativo opera como um autômato usual, efetuando uma seqüência de transições não-adaptativas, sendo que tal seqüência é finalizada ou pelo término do reconhecimento da sentença, ou então pela execução de alguma transição adaptativa, responsável por alterar sua topologia. Neste caso, pode-se dizer que o autômato se modifica, assumindo uma nova topologia, representada por alguma outra máquina de estados  $E_{i+1}$ , cujo estado inicial de operação deverá ser determinado pela transição executada. [5]

Autômatos e grafos são conceitos diferentes. No entanto, o fato de ser possível utilizar o grafo para representar o autômato implica na possibilidade de se abstrair características convenientes à proposta do grafo adaptativo.

Uma função de transição adaptativa, segundo [5], possui a forma:  $(\gamma g, q, s\alpha) : A, \rightarrow (\gamma g', q', s'\alpha), B$  onde  $g$  e  $g'$  representam o conteúdo no topo da pilha, respectivamente, antes e depois da realização da transição;  $q$  e  $q'$  representam respectivamente o estado do autômato antes e depois da ocorrência da transição;  $s$  é o símbolo da cadeia consumido durante a transição e  $s'$  o símbolo inserido na cadeia;  $A$  e  $B$  representam transições adaptativas que ocorrem, respectivamente, antes ou depois da transição.

Nota-se da função de transição adaptativa que o foco de um autômato está no percurso sobre o grafo conforme símbolos da cadeia de entrada são consumidos. Quando se trata de um grafo propriamente dito, não existem símbolos a serem consumidos, isto significa que os elementos  $s$  e  $s'$  podem ser removidos. Além disso, para explicitar a semântica bidirecional da aresta no grafo convém trocar a seta unidirecional na transição adaptativa por uma bidirecional. Ainda neste sentido, o termo estado se mostra inadequado sendo melhor representado pelo termo vértice (troca-se  $q$  por  $v$  na notação da transição adaptativa). O termo *transição adaptativa* também, por isso pode ser renomeado para *aresta adaptativa*. Como a aresta adaptativa não será utilizada para percorrer o grafo, não faz sentido manter elementos que marquem transições de pilha (remove-se o  $\gamma g$  e  $\gamma g'$ ) e nem associar ações adaptativas à eventos do percurso (remove-se  $A$  e  $B$ ).

A forma final da aresta adaptativa fica:  $(v) \leftrightarrow (v')$ .

Dado que o comportamento do autômato adaptativo é passível de representação por um grafo e que todo grafo é passível de ser representado de forma matricial, logo, é possível representar qualquer autômato adaptativo por meio de um estrutura matricial.

### C. Ações Adaptativas sobre Grafos

Diferente do que ocorre com o reconhecedor, pelo fato do controle adaptativo não se dar por meio de percurso, as ações adaptativas são disparadas *ad hoc*. As ações adaptativas aplicáveis à autômatos adaptativos e se mantém no grafo adaptativo: inclusão ( $+[v] \leftrightarrow [v']$ ) e remoção ( $-[v] \leftrightarrow [v']$ ) [5].

Considere como exemplo o grafo  $G_0$  na tabela I. Aplicando sobre ele a ação adaptativa  $+[v_1] \leftrightarrow [v_3]$  se criará no grafo subjacente uma aresta que conecta  $v_1$  e  $v_3$ ; caso  $v_3$  não exista ele é criado (ver  $G_1$ ). Aplicando sobre  $G_1$  a ação adaptativa  $-[v_1] \leftrightarrow [v_2]$  remover-se-a a aresta entre  $v_1$  e  $v_2$ ; caso algum vértice fique desconexo ele é removido (ver  $G_2$ ).

Tabela I  
AÇÕES ADAPTATIVAS SOBRE UM GRAFO SUBJACENTE.

| matriz inicial   | $+[v_1] \leftrightarrow [v_3]$  | $-[v_1] \leftrightarrow [v_2]$   |
|--|---|--|
| $G_0 = \begin{matrix} v_1 & v_2 \\ v_2 & \begin{pmatrix} v_1 & v_2 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix}$ | $G_1 = \begin{matrix} v_1 & v_2 & v_3 \\ v_2 & \begin{pmatrix} v_1 & v_2 & v_3 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{matrix}$ | $G_2 = \begin{matrix} v_1 & v_3 \\ v_3 & \begin{pmatrix} v_1 & v_3 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix}$ |

### D. Grafo Adaptativo

Se diz que se trata de um “grafo adaptativo” se ele puder ser descrito por meio de arestas adaptativas e possuir um conjunto não vazio de *scripts* de adaptação. Os *scripts* de adaptação são também compostos por um conjunto não vazio de ações adaptativas que fazem o grafo assumir diferentes configurações topológicas. A figura 2 apresenta a proposta de estrutura em nível de metamodelo para o Grafo Adaptativo.

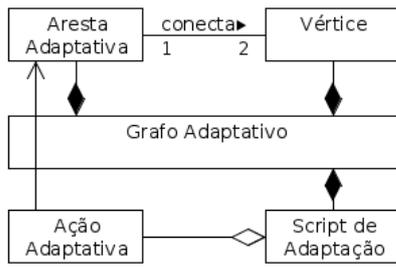


Figura 2. Metamodelo proposto para o Grafo Adaptativo.

E. Script de Adaptação

O *script* de adaptação pode operar no grafo subjacente de duas maneiras distintas: invariável e complexa. Um *script* de adaptação invariável garante que qualquer que seja a configuração topológica do grafo subjacente num dado momento, sua aplicação resulta sempre em uma mesma configuração topológica. Por outro lado, um *script* de adaptação complexo aplica de maneira incremental seu conjunto de ações sobre a estrutura topológica do grafo subjacente no instante da aplicação, isto significa que a aplicação de um mesmo *script* de adaptação complexo deve resultar em topologias diferentes e eventualmente desconexas. Por isso o *script* de adaptação complexo exige por parte do idealizador certo esforço para evitar este tipo de situação. Naturalmente, ao aplicar os algoritmos de percurso no grafo adaptativo, o resultado irá variar conforme a topologia do grafo no momento.

A figura 3 mostra como a topologia do grafo G pode variar conforme são aplicados diferentes *scripts* de adaptação. A tabela II apresenta um exemplo de aplicação de um *script* de adaptação invariável e a tabela III apresenta um exemplo de aplicação de um *script* de adaptação complexo. Pode-se dizer também que as adaptações sejam subgrafos de um super-grafo G ideal.

Tabela II  
EXEMPLO DE APLICAÇÃO DE UM MESMO *script* INVARIÁVEL.

|                  | Exemplo 1  | Exemplo 2   |
|------------------|--|---|
| grafo inicial    | $G = \begin{matrix} v_1 & v_3 & v_4 \\ v_2 & \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{matrix}$ | $H = \begin{matrix} v_8 & v_9 \\ v_8 & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix}$ |
| grafo resultante | $G = \begin{matrix} v_1 & v_2 \\ v_2 & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix}$                            | $H = \begin{matrix} v_1 & v_2 \\ v_2 & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix}$ |

Tabela III  
EXEMPLO DE APLICAÇÃO DE UM MESMO *script* COMPLEXO.

|                  | Exemplo 1   | Exemplo 2   |
|------------------|---|---|
| grafo inicial    | $G = \begin{matrix} v_1 & v_3 & v_4 \\ v_2 & \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{matrix}$                                    | $H = \begin{matrix} v_8 & v_9 \\ v_8 & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix}$   |
| grafo resultante | $G = \begin{matrix} v_1 & v_2 & v_3 & v_4 \\ v_2 & \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$ | $H = \begin{matrix} v_1 & v_2 & v_8 & v_9 \\ v_2 & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$ |

Quando se toma rede como serviço se torna necessário atender à diversas propriedades de qualidade do serviço (*Quality of Service* - QoS) – percepção subjetiva de um grupo de *stakeholders* em relação à qualidade do serviço oferecido por uma rede. Dado que uma mesma infraestrutura de rede pode atender diferentes *stakeholders*, o comum é que tais propriedades sejam conflitantes. Esse problema costuma ser administrado por meio das chamadas redes virtuais [3]. Em ambientes de redes virtuais diferentes provedores podem criar redes virtuais heterogêneas e assim prestar serviço de forma personalizada.

Assim, a tecnologia de virtualização de redes é um paradigma que propicia à múltiplas redes virtuais (*Virtual Networks* - VNs) compartilharem de uma mesma infraestrutura denominada rede de substrato (*Substrate Network* - SN). Talvez o maior desafio na implementação deste conceito seja justamente embutir a rede virtual na rede de substrato (*Virtual Network Embedding* - VNE) [6].

O VNE costuma ser realizado aplicando algoritmos sobre a topologia da rede de substrato o qual considera as restrições durante a computação. Neste estudo se propõe que a topologia apresentada ao algoritmo seja um grafo adaptativo, com isso é possível remover antecipadamente os enlaces que ferem determinado conjunto de restrições do *stakeholder*.

A. Virtualização de Redes

Uma rede virtual é uma coleção de enlaces virtuais mapeados em diferentes enlaces físico interconectados por caminhos na rede de substrato. Como diversas redes virtuais podem depender de um mesmo conjunto de recursos físicos, é preciso que o *embedding* (mapeamento ou atribuição) de cada rede

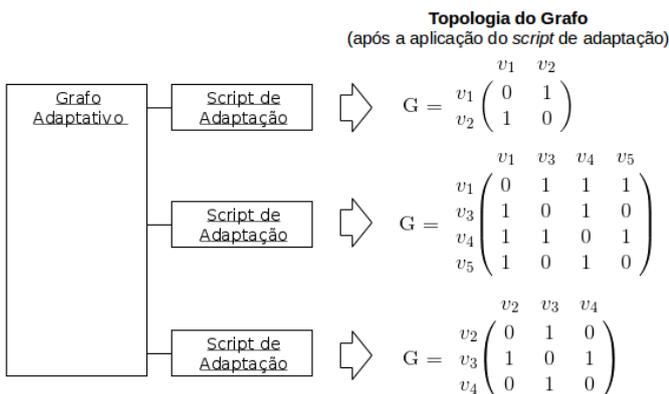


Figura 3. Adaptação topológica do grafo G.

III. VIRTUAL NETWORK EMBEDDING ADAPTATIVO

Ambientes de rede heterogêneos são aqueles compostos por uma diversidade de tecnologias de rede interconectadas [3].

virtual *on-line* seja eficiente visando o uso apropriado dos recursos da rede [3]. A figura 4 apresenta o problema de VNE por meio de duas redes virtuais e uma de substrato.

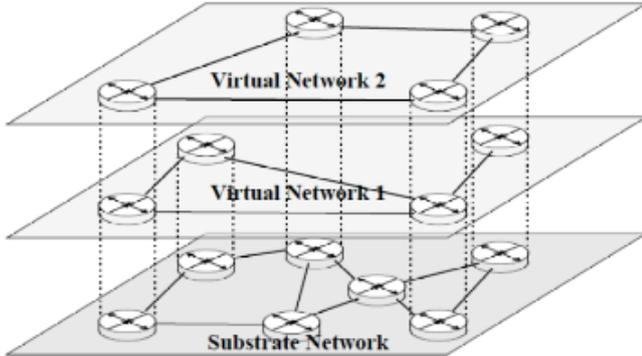


Figura 4. Ilustração de *Virtual Network Embedding* [2].

Os algoritmos ou heurísticas que propiciam o VNE devem garantir a convergência e eliminação de laços de roteamento nessas redes, obedecendo a tipos diferentes de métricas de QoS. Se diz desafio pois o VNE nesses termos é considerado multi-restritivo e por isso computacionalmente, um problema *NP-hard*. Isto, por sua vez, torna inevitável o uso de algoritmos heurísticos e aproximados para mapear nós e enlaces virtuais nos dos estágios (virtual e substrato) de forma independente ou coordenada [6].

Uma requisição de rede virtual (*Virtual Network Requisition* - VNR) se traduz em uma rede virtual com nós e enlaces virtuais com certas necessidades de rede, garantia de QoS e segurança. Os elementos da rede virtual (links e nós) devem ser mapeados sobre elementos concretos da rede de substrato com a preocupação de se obter uma solução otimizada no sentido de atender a demanda da requisição e que consuma a menor quantidade de recursos físicos da rede de substrato.

### B. Álgebra de Caminhos

Uma alternativa à abordagem heurística geralmente utilizada para tratar problemas de roteamento em redes virtuais, tem se estudado abordagens algébricas como a “Álgebra de Caminhos” [6]. A álgebra de caminhos, por exemplo, como proposta por [4] consiste na quadrupla  $\langle M, F, \bar{S}, \preceq_{ML} \rangle$ , onde  $M$  é o conjunto de métricas de roteamento;  $F$  é o conjunto de combinação de métricas;  $\bar{S}$  é o conjunto de operações binárias denominadas sínteses; e  $\preceq_{ML}$  é a relação de ordenação léxica multi-dimensional.

Grosso modo, as métricas ( $M$ ) enumeram as dimensões de interesse para a escolha do caminho, os quais podem ser combinados por meio de uma função ( $F$ ). Um exemplo é a função  $QoS = f(THR, PDT, PDV, PLR)$  com  $QoS \in F$ , onde  $THR$  é o *throughput* da rede;  $PDT$  o *packet delay transfer*;  $PDV$  o *packet delay variation*; e o  $PLR$  o *packet loss rate*. A síntese consiste em, dado os vértices de origem e destino, reduzir os caminhos possíveis à um valor com base em um conjunto de funções. Quando o caminho é analisado por

uma única função se diz que ele é mono-restritivo, em caso contrário multi-restritivo.

Nesta álgebra existem quatro sínteses: a) *minimizativa* ou  $\bar{S}_1$ , o menor valor atribuído às aresta de um caminho; b) *maximitativa* ou  $\bar{S}_2$ , o maior valor atribuído às arestas de um caminho; c) *aditiva* ou  $\bar{S}_3$ , soma dos valores atribuídos às arestas do caminho; e d) *multiplicativa* ou  $\bar{S}_4$ , multiplicação dos valores atribuídos às arestas do caminho. Caso a síntese resultante de um caminho hipotético  $\alpha$  seja maior do que a síntese resultante de outro caminho hipotético  $\beta$  se diz que  $\bar{S}[\bar{C}(\alpha_{O,D})] \preceq_{ML} \bar{S}[\bar{C}(\beta_{O,D})]$ , isto é, o caminho  $\beta$  é melhor que o  $\alpha$ ; desta comparação se obtém a ordenação léxica multi-dimensional ( $\preceq_{ML}$ ).

1) *Dígrafo de Rede Virtual*: A álgebra de caminhos opera sobre uma grafos orientados (ou dígrafos) com propriedades específicas, neste estudo chamado de Dígrafo de Rede Virtual. Por ser um dígrafo, a matriz de adjacências, embora se mantenha quadrada e de dimensão  $n \times n$ , não é necessariamente simétrica (pode existir  $\psi_G = uv$  mas não  $\psi_G = vu$ ). O Dígrafo de Rede Virtual aceita apenas uma aresta na conexão de dois vértices ( $\exists \psi_G(a) = uv \neg \psi_G(\delta) = uv$  onde  $\delta$  é o conjunto de todas as outras arestas do grafo). Além disso, não é permitida a existência de arestas que liguem um mesmo vértice ( $\psi_G(a) = uu$ ); isso força que a diagonal da matriz de adjacências possua sempre valor 0.

2) *Dígrafo de Rede Virtual Adaptativo*: Um dígrafo adaptativo é um grafo adaptativo orientado. De maneira geral se mantém as propriedades do grafo adaptativo com a diferença de que as ações adaptativas são unidirecionais, ou seja, inclusão ( $+[v] \rightarrow [v']$ ) e remoção ( $-[v] \rightarrow [v']$ ). A ação adaptativa altera a aresta adaptativa no sentido da seta mas não no sentido oposto. A equivalência com a ação sobre o grafo adaptativo se dá na forma:  $+[v] \leftrightarrow [v'] \Leftrightarrow (+[v] \rightarrow [v']) \wedge (+[v'] \rightarrow [v])$  e  $-[v] \leftrightarrow [v'] \Leftrightarrow (-[v] \rightarrow [v']) \wedge (-[v'] \rightarrow [v])$ . Soma-se à esta propriedades as restrições impostas pelo Dígrafo de Rede Virtual, isto é, aceita apenas uma aresta que incide sobre dois vértices e rejeita arestas que incidem sobre um mesmo vértice.

### C. Prova de Conceito

Considere como exemplo a topologia da SN apresentada na figura 5 e sua representação matricial na tabela IV. Supondo que se queira ir do vértice  $F$  ao vértice  $D$ , se obtém os caminhos  $F,E,B,D$  e  $F,E,C,A,B,D$ . Para esses caminhos os valores das sínteses  $F,E,B,D$  são:  $\bar{S}_1[\alpha_{F,D}] = 3$ ,  $\bar{S}_2[\alpha_{F,D}] = 65$ ,  $\bar{S}_3[\alpha_{F,D}] = 50$  e  $\bar{S}_4[\alpha_{F,D}] = 1700$ . Os valores para  $F,E,C,A,B,D$  são:  $\bar{S}_1[\beta_{F,D}] = 5$ ,  $\bar{S}_2[\beta_{F,D}] = 60$ ,  $\bar{S}_3[\beta_{F,D}] = 50$  e  $\bar{S}_4[\beta_{F,D}] = 2400$ . Portanto,  $\bar{S}[\bar{F}(\beta_{F,D})] \preceq_{ML} \bar{S}[\bar{F}(\alpha_{F,D})]$ , isto é, o caminho  $\alpha$  é mais eficiente que o  $\beta$ .

Suponha agora que por uma restrição qualquer do *stakeholder* determinadas tecnologias não possam ser utilizadas no serviço de rede a ser prestado. Naturalmente não se espera, por exemplo, que tais tecnologias sejam removidas da rede de substrato mas apenas que a rede virtual não as utilize. Isto significa que o caminho que passa por um determinado enlace

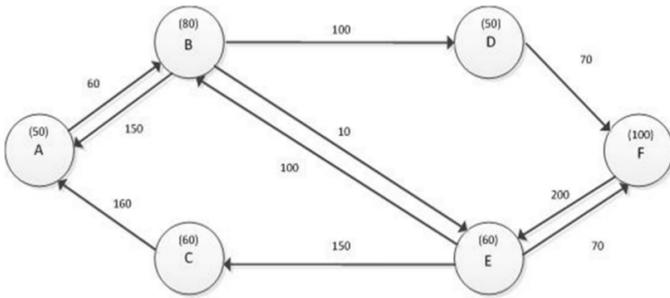


Figura 5. Exemplo de topologia de rede [6].

Tabela IV

REPRESENTAÇÃO MATRICIAL DO EXEMPLO DE TOPOLOGIA DE REDE.

$$R_0 = \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} \begin{pmatrix} A & B & C & D & E & F \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad R_1 = \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} \begin{pmatrix} A & B & C & D & E & F \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

deverá ser eliminado, no entanto, isso costuma ser efetuado após a listagem de todos os caminhos.

Aplicando o grafo adaptativo pode-se remover tais enlaces antes de se listar cada um dos caminhos possíveis e assim reduzir o esforço computacional envolvido. Suponha, por exemplo, que a tecnologia do enlace  $E-C$  da figura 5 seja considerado indesejável pelo stakeholder; aplicando sobre  $R_0$  na tabela IV a ação adaptativa  $-(E \rightarrow C)$  resulta em  $R_1$ . Esta modificação faria com que o caminho  $F,E,C,A,B,D$  não pudesse mais ser encontrado pela rede virtual.

Naturalmente se o enlace  $F-E$  fosse removido não seria possível encontrar nenhum caminho; este tipo de preocupação deve ser parte da especificação do *script* de adaptação.

Com esta estratégia é possível ter diferentes *scripts* de adaptação os quais ajudam a atender diferentes especificações de QoS.

#### D. Uma consideração sobre Complexidade

Dado que álgebra de caminhos trata cada caminho possível entre o vértice de origem e o de destino, se pode afirmar que a complexidade algorítmica envolvida é  $O(n^n)$  [6], ou seja, se trata de uma problema *NP-hard*. Naturalmente a tecnologia adaptativa não altera a complexidade intrínseca do problema mas ao propiciar a redução no número vértices e/ou arestas assegura a diminuição no tempo efetivo de computação. O algoritmo PAViLiM (Paths Algebra Virtualization Link Mapping) proposto por [6] consiste em uma implementação computacional da álgebra de caminhos orientada a solução do problema VNE.

Considere como exemplo, a execução deste algoritmo em uma máquina padrão tendo como base um grafo hipotético com nove vértices requer cinco horas e meia de computação [6]; a execução deste mesmo algoritmo na mesma máquina mas tendo como base um grafo hipotético com dez vértices requer quase dois dias de computação [6]. Por outro lado,

a execução deste mesmo algoritmo na mesma máquina mas tendo como base um grafo hipotético com oito vértices requer cerca de trinta minutos [6]. Isso sugere que a escolha qualitativa dos vértices e arestas a serem considerados durante cada VNE pode reduzir significativamente o tempo de computação necessário. A escolha qualitativa implica em alterar a topologia existente no grafo que representa a rede de substrato, daí a necessidade do grafo adaptativo.

#### IV. CONCLUSÃO

Este estudo propôs uma estrutura de topologia adaptativa chamada *grafo adaptativo*. A apresentação foi informal e teve a intenção de sugerir suas principais propriedades. Como trabalho futuro sugere-se propor a formalização da estrutura proposta.

Embora informal, é possível vislumbrar o potencial da estrutura proposta tanto em seu aspecto teórico quanto aplicado ao contexto de VNE proposto. Também como trabalho futuro sugere-se aprofundar na investigação do uso desta estrutura junto com algoritmos de roteamento e em estruturas topológicas mais próximas do real. Sugere-se também investigar quais outras aplicações de grafo podem se beneficiar de uma estrutura adaptativa, entre outras, estruturas de dados, complexidade e redes neurais artificiais.

#### REFERÊNCIAS

- [1] Bondy, J. A. Murty U. S. R. (1982). *Graph Theory with Applications*. North-Holland.
- [2] Botero, J. F. Molina, M. A. T., Serra, X. H. Amazonas, J. R. (2013). A novel paths algebra-based strategy to flexibly solve the link mapping stage of VNE problems. *Journal of Network and Computer Applications* v. 36, i. 6, Elsevier.
- [3] Chowdhury, N. M. M. K. Rahman, M. R. Boutaba, R. (2009). Virtual Network Embedding with Coordinated Node and Link Mapping. *IEEE INFOCOM*.
- [4] Herman, W. P. (2008). Formulação algébrica para a modelagem de algoritmos de roteamento multi-restritivo hop-by-hop. *Tese - Universidade de São Paulo*.
- [5] Neto, J. J. (2003). Autômatos em engenharia de Computação - uma visão unificada. *Primera Semana de Ciencia y Tecnología de la Sociedad Chotana de Ciencias y la Red Mundial de Científicos Peruanos*.
- [6] Molina, M. A. T. (2013). Utilização da álgebra de caminhos para realizar o mapeamento de requisições virtuais sobre redes de substrato. *Tese - Universidade de São Paulo*.
- [7] Ramos, M. V. M. Neto, J. J. Vega, I. S. (2009). *Linguagens Formais: teoria, modelagem e implementação*. Editora Bookman.
- [8] Sedgewick, R. Wayne, K. (2011). *Algorithms*. Addison-Wesley Professional, 4th edition.
- [9] Vega, Í. S. (2008). An adaptive automata operational semantic. *IEEE Latin America Transactions*, v. 6, n. 5