DYNAGRAPH: Um Modelo de Edição e Representação de Grafos Dinâmicos

Anderson Calixto[1], **Marcos Negreiros**[1]

[1] Universidade Estadual do Ceará (UECE)
Mestrado Profissional em Computação Aplicada
MPCOMP/UECE-IFCE-UFRJ
Av. Paranjana, 1700 – Campus do Itaperi
CEP: 60740-000 – Fortaleza – CE – Brazil

andersonbr@gmail.com, negreiro@graphvs.com.br

Resumo. *Nas mais diversas redes existentes, a sua modificação ao longo do tempo é uma característica comum. O registro otimizado das mudanças que ocorrem na estrutura de representação destas redes não é uma tarefa trivial. Este artigo propõe um modelo de organização de dados que viabiliza de forma otimizada as tarefas de edição e representação da evolução dinâmica de grafos. Para resolver o problema, o grafo que evolui no tempo é definido por sequências de subgrafos de existência limitada a um intervalo de observação, cujos dados de vértices, ligações, atributos e características (*bindings*) são definidos por unidade de tempo de um intervalo. As sequências são então unidas de modo a formar um grande agrupamento que representam a evolução do grafo temporal dado. Apresentamos com detalhes como é realizado este processo, e discutimos a implementação computacional deste recurso, o qual pode ser aplicado: no acompanhamento da evolução de redes sociais, processos epidêmicos, fluxo de tráfego de qualquer natureza (veículos, dados, etc.), e muitas outras.*

Palavras Chave: *Grafo, Rede, Dinâmico. Área de classificação principal: TAG.*

Abstract. *In several existing networks, their modification over time is a common feature. An optimized registering of changes that occur in the structure of these networks is not a trivial task. This article proposes a model of organizing data that enables optimally editing tasks and represent the dynamic evolution of graphs. In the context of solving the problem, the graph that evolves over time is defined by the sequence of subgraphs of being limited to a range of observation, whose vertex data, links, attributes and bindings of its entries are defined for each unit time interval. The sequences are then joined to form a large group of sequences representing the trend graph of the temporal data. We present in detail, how this process is performed, and discuss the computational implementation of this feature, which can be applied in monitoring the evolution of social networks, development of an epidemic desease, traffic flow of any kind (vehicles, data, etc.), and many other dynamic real life applications.*

Keywords: *Graph, Network, Dynamic. Main area: TAG.*

1 Introdução

Redes de avenidas, ruas e estradas possuem diversos serviços como fluxo de veículos, restrições de mobilidade, dentre outras. De forma semelhante, as redes de computadores, por exemplo: uma simples rede cabeada em uma casa ou escritório, usuários de uma rede sem fio, ou mesmo a rede mundial de computadores (web), também possuem vários tipos de serviços como topologia, intensidade de sinal, roteamento, endereço IP, entre outros.

Muitos desses serviços se caracterizam por estarem em constante mudança. Em geral é necessário manter o histórico de movimentação de serviços dinâmicos para que seja possível observar sua evolução e/ou fazer avaliações das mudanças de comportamento inerentes e probabilidades estatísticas associadas às transições de estados. Para se atender a este requisito, torna-se necessário que a estrutura dessas informações estejam dispostas de maneira bem organizada e, preferencialmente, não possuam redundâncias sobre os elementos estruturais do contexto que o definem.

A teoria dos grafos é uma linguagem poderosa e genérica. Grafos e Redes são tratados igualmente do ponto de vista teórico, no entanto alguns pesquisadores consideram os dois temas de formas diferentes. Redes, para a computação, geralmente descrevem estudos sobre uma estrutura física tecnológica (hardware e software) de comunicação entre unidades inteligentes, tendo significado amplo em diversas outras áreas do conhecimento, como nas Ciências Sociais, Biológicas, Médicas, etc. Já grafo é uma estrutura abstrata composta de vértices e ligações entre estes vértices, cujos estudos teóricos avançados, relacionado à combinatória e a Teoria dos Grafos vêm sendo desenvolvidos ao longo dos anos desde [Euler (1736)].

Muito se fez no desenvolvimento de estudos sobre grafos, porém qual é o espaço que sobrou para o desenvolvimento de uma nova ciência sobre Redes? Considera-se que os estudos sobre redes, feito há poucos anos, tomou uma outra forma sobre o que se fazia no passado, distinguindo-se dos trabalhos anteriores em pelo menos três diferentes contextos, [Newman (2006)]:

1. Foco nas propriedades de redes do mundo real, no que se ocupa de questões empíricas e teóricas;
2. É muito frequente a visão de que as redes não são estáticas, mas que evoluem no tempo de acordo com várias regras dinâmicas; e
3. Objetiva, pelo menos ultimamente, a entender redes não apenas como objetos topológicos, mas também como arcabouços sobre os quais sistemas distribuídos dinâmicos são construídos.

Elementos para todos estes assuntos antevêem a recente explosão de interesse em redes. A literatura introduziu esta preocupação em [Newman (2006)]. Nesse contexto, a principal argumentação é a de que a Teoria dos Grafos Aplicada, como o nome sugere, está mais focada em problemas de rede do mundo real e que estão orientadas a planejamento e engenharia. Ao contrário, o foco atual em redes surge naturalmente considerando-se situações que evoluem sem planejamento ou que não evoluem sobre uma estrutura organizacional centralizada. Redes sociais e biológicas são exemplos destes tipos de redes, assim como redes de informação literária e a rede mundial de computadores. As categorias podem ser bem mais amplas, como as redes de transporte, redes de energia elétrica, e a rede física da internet, que parecem terem sido criadas para atender a um propósito simples (transporte, suprimento de energia e comunicação), mas que vem sendo construídas ao longo de anos por empresas privadas ou mesmo públicas. No caso da análise das redes sociais, tudo é feito de forma empírica, mas tende a ser descritiva ao invés de ter natureza construtiva.

Neste artigo, foca-se no problema de definir um modelo computacional que permita, a partir de uma estrutura de dados simples, formalizar o modelo de um Grafo ou Rede Dinâmica, que represente com mínimo custo de armazenamento a evolução desta instância de observação e estudos. O principal desafio é também conduzir um modelo que corresponda ao acompanhamento da evolução dos conjuntos de um grafo: de nós e ligações (arcos e elos), como inserção/retirada ao longo do tempo, e possivelmente mudança de suas características (posição, cor, forma, tamanho, e outras).

Por se tratar de uma abordagem muito nova e necessária, a literatura tem desenvolvido alguns trabalhos que tentaram tratar parte do problema acima descrito. [Kempe (2002)] propuseram um modelo de redes temporais como grafos estáticos onde cada elo é rotulado com o tempo em que a iteração ocorre. [Ferreira (2004)] também considera as redes dinâmicas como uma sequência de grafos estáticos de modo a resolver os problemas de grafos como as medidas de caminho, conectividade da rede, e árvores geradoras mínimas na rede dinâmica formada. [Kostakos (2009)] considera a estrutura de grafos temporais, também como grafos estáticos, no entanto avança sobre as métricas introduzindo conceitos como disponibilidade temporal, proximidade temporal e geodésica, e estuda os seus grafos sobre redes reais. Os estudos destas redes foram feitos a partir de um conjunto de dados de *emails* enviados de dentro da empresa Eron Corp durante o mês de Outubro/2009, e um conjunto de dados que revela os encontros face-a-face entre pessoas, armazenadas por uma camera colocada em uma calçada da Universidade de Bath (Inglaterra) durante a primeira metade do mês de março de 2008.

Destaca-se também um conjunto de três trabalhos apresentados em simpósios do SIGCOMM (2010), ACM (2010) e IEEE (2011), desenvolvidos por Tang e sua equipe (Musolesi, Mascolo, e Latora), *ap ud* [Kim (2012)]. No primeiro trabalho a equipe tenta desenvolver um modelo geral de redes dinâmicas introduzindo a variável que representa a velocidade com que uma mensagem trafega. No segundo propõe-se métricas de temporalidade central baseadas nos caminhos temporais para medir a importância de um nó em uma rede dinâmica. Aqui também investigaram métodos de identificação de *tempo de conciência central* de um nó utilizando vários dados reais. O desempenho desta métrica foi considerado ruim por [Kim (2012)], pois como fraqueza do método tem-se a consideração de conhecimento prévio de cada nó para futuros contatos. No terceiro trabalho eles propuseram uma aplicação baseada na suposição de se ter um conhecimento global dos contatos do passado e futuro.

[Kim (2012)] apresentaram a modelagem de uma rede dinâmica como digrafos orientados ao tempo (*time-ordered graph*), a qual é gerada por uma *colagem* de instantes com arestas direcionadas que unem cada nó ao seu sucessor no tempo, transformando uma rede dinâmica em um grande grafo de fácil análise. O trabalho também aborda a ideia de que uma rede temporal dinâmica é construída por uma sequência de grafos estáticos. Os recursos incluídos permitem que se faça avaliações das métricas de centralidade relacionadas ao grau, fecho, e internalidade de uma forma natural para o caso dinâmico.

A ideia proposta por [Kim (2012)] é tomada como base deste trabalho, porém é feito o uso de sequências temporais para nós e ligações, suas características modificáveis, assim como o relacionamento entre elas, formando um grafo com as informações necessárias para qualquer instante no tempo. A partir disto, desenvolveu-se um software denominado de **DYNAGRAPH**, que permite visualizar o comportamento do grafo ao longo de um período de tempo, e editá-lo. A ferramenta construída permite visualizar previsões e processos dinâmicos em vários contextos e realizar simulações preditivas sobre estes eventos.

Este artigo está dividido do seguinte modo, na seção 2, considera-se a definição e o passo a passo da construção do modelo computacional do Grafo Dinâmico. A seguir, na seção 3 descrevemos detalhadamente uma visão computacional do grafo dinâmico e sua estrutura de dados, na seção 4 propõe-se os principais recursos necessários para o objeto grafo dinâmico. Por fim, na seção 5 apresenta-se a interface gráfica de edição e reprodução de grafos dinâmicos no DYNAGRAPH.

2 Definição e Construção de Grafo Dinâmico

Um grafo dinâmico pode ser definido em um período $T = [t_i, t_f]$, como sendo G_{t_i,t_f} = $(S^{s,d}_{a,b})$, onde t_i é o tempo inicial e t_f é o tempo final, ou período de sua existência. S é o conjunto de sequências temporais $\{S^{s_0,d_0}_{a_0,b_0}, S^{s_1,d_1}_{a_1,b_1} \ldots, S^{s_n,d_n}_{a_n,b_n}\}$ que ocorrem durante o tempo que inicia em a e vai até b. Neste caso, $a \geq t_i$ e $b \leq t_f$, identificadas por s e diretamente dependente da sequência d.

A figura 1 apresenta o modelo geral do grafo dinâmico aqui definido, onde as camadas indicam as sequências de subgrafos, onde cada uma representa uma informação da rede que varia nos intervalos de tempo do período $[a, b]$.

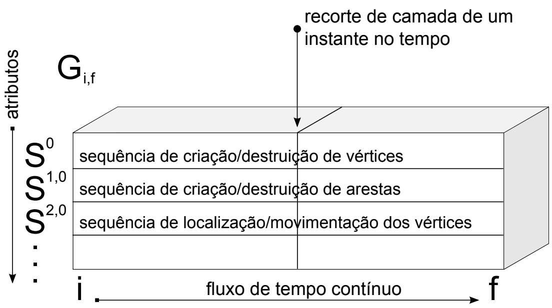

Figura 1. Modelo computacional de construção de um grafo dinâmico

A sequência temporal é representada por $S^{s,d}_{a,b}$ = $(V_{k,l})$, sendo:

- *S* - conjunto das sequências;
- *s* - índice da sequência;
- *d* - sequência a qual este *S* depende, para poder existir. d deve ter o índice de uma sequência anterior ou estar vazio caso não haja dependência;
- *a* - tempo inicial;
- *b* - tempo final (quando não informado, significa que a caracteristica em questão continua sendo observada e o tempo está aberto);
- *V* - conjunto de elementos que representam modificações durante o período $[k, l]$ para o atributo representado pela sequência:
  $\{v_{k_0,l_0}, v_{k_1,l_1} \ldots, v_{k_n,l_n}\}$, onde $k \leq k_0 \ldots k_n$ e $l \geq l_0 \ldots l_n$.

Quando um elemento $v_{k_n,l_n} \in V$ de uma sequência S^x deixa de existir no tempo l_n, todos os elementos de *V* de uma outra sequência $S^{y,x}$ que tem este mesmo elemento como referência também deixam de existir nos tempos $> l_n$, e tempos iniciais e finais não precisam ser observados, pois a observação do tempo seguirá a prioridade *dependencia* $\rightarrow$ *dependente*.

[Kim (2012)] considera como atributos os vértices e arcos de uma mesma sequência, onde vértices permanecem estáticos e as arestas mudam no tempo. Esta proposta torna a rede totalmente dinâmica, onde qualquer elemento, atributo ou característica do grafo em evolução podem ser modificados no tempo.

2.1 Sequência de criação dos vértices S^0

Para ilustrar a construção de uma rede dinâmica pretendida, considera-se um tempo finito, começando inicialmente em *a* = 0 até *b* = 2. A sequência temporal $S^0_{0,2}$ define o tempo de vida de cada vértice no grafo dinâmico.

Vértice	Intervalo
(A)	[0,2]
(B)	[1,2]
(C)	[2,2]

Tabela 1. Descrição textual da evolução da existencia de vértice de um exemplo de grafo dinâmico.

Na representação da tabela 1, o elemento $(v_{k,l})$ que representa um vértice, de rótulo *A* existiu no tempo $k = 0$ até $l = 2$. Os rótulos *B* e *C* seguem comportamento semelhante, nos respectivos intervalos de tempo. A figura 2 mostra a representação gráfica conceitual da mudança temporal, onde é possível perceber com maior facilidade a criação ou destruição de vértices durante o intervalo de tempo [0, 2].

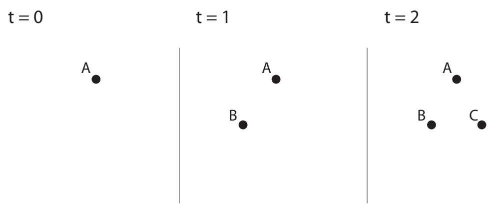

Figura 2. Representação gráfica da evolução do grafo descrito na tabela 1.

2.2 Sequência de posição $S^{1,0}$

Após um vértice ser criado, ele possui uma posição, que poderá ser modificada diversas vezes até sua existencia se findar. Para isto ser descrito computacionalmente é necessário que se represente o elemento $v_{k,l}$ nesta sequência, que é definido por $[(v \in S^0), x, y]_{k,l}$, sendo x e y as coordenadas do vértices no plano cartesiano. Uma extensão destes atributos é a inclusão de uma terceira coordenada (z) para representar o grafo num contexto tridimensional.

A tabela 1 descreve o comportamento ao longo do tempo devido a mudança de posição do vértice A do grafo exemplo, enquanto a figura 2 mostra o que ocorreu no tempo. Usando a representação adotada, é possível observar que durante a existência dos três vértices, eles permaneceram na mesma posição na figura 2 e acontece o deslocamento do

$[(v \in S^0), x, y]$	Intervalo $[k, l]$
(A, [1, 1])	[0,2]
(B, [1, 4])	[1,2]
(C, [4, 4])	[2,2]

Tabela 2. Descrição de aparecimento dos vértices de G relativo à tabela 1

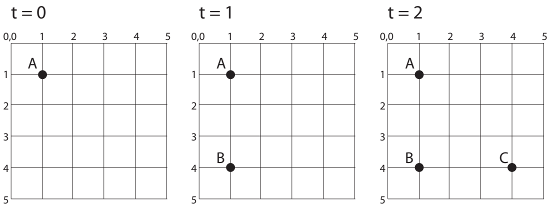

Figura 3. Visualização de criação do grafo G ao longo do tempo.

vértice A conforme se vê na figura 4. A tabela 2 descreve a movimentação do vértice A do grafo exemplo, e a necessidade de existência e sua ocorrência é mostrada na descrição. Nota-se que o vértice A movimenta-se de t=0 para t=1, e os vértices A e B de t=1 para t=2, incluindo-se C em t=2.

$[(v \in S^0), x, y]$	Intervalo $[k, l]$
(A, [1, 1])	[0,0]
(A, [2, 2])	[1,1]
(B, [1, 4])	[1,1]
(A, [3, 3])	[2,2]
(B, [1, 3])	[2,2]
(C, [4, 4])	[2,2]

Tabela 3. Descrição da movimentação de um vértice no tempo.

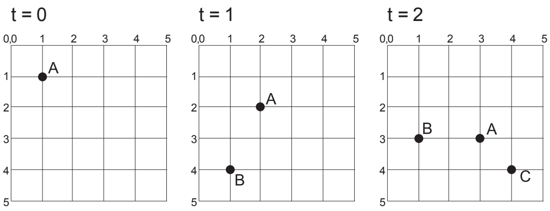

Figura 4. Movimentações dos vértices A e B no tempo.

2.3 Sequência de arestas em forma de arcos $S^{2,0}$

Para se definir a existência de arcos, é introduzido o atributo $v_{k,l}$ nessa sequência, o qual é definido por $[(v_o \in S^0), (v_t \in S^0)]_{k,l}$, sendo o a origem e t o destino. A tabela 4 mostra a descrição de eventos de inclusão de arcos no grafo da figura 2, enquanto temos a visualização dos acontecimentos na figura 5.

2.4 Consolidação das sequências em grafo

Para consolidar as diversas sequências, é possivel gerar uma descrição completa do grafo, com todas informações de evolução em um determinado período. A tabela 5 apresenta uma

Arco	Intervalo
(B,A)	[1,1]
(A,B)	[1,2]
(C,A)	[2,2]
(B,C)	[2,2]

Tabela 4. Descrição da existência de vértices no grafo exemplo.

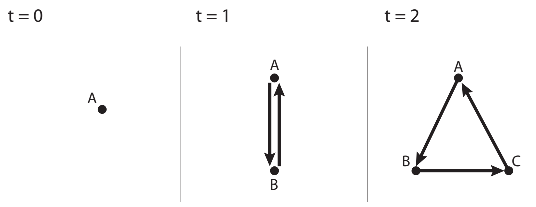

Figura 5. Visualização das ocorrências de aparecimento dos arcos do grafo exemplo.

descrição do que ocorreu no contexto, e a figura 6 mostra a evolução dos eventos.

	Nó	Posição	Arco	Intervalo
1	(A)	[1,1]		[0,0]
2	(A)	[1,1]	(A,B)	[1,2]
3	(B)	[1,4]	(B,A)	[1,1]
4	(B)	[1,4]	(B,C)	[2,2]
5	(C)	[4,4]	(C,A)	[2,2]

Tabela 5. Descrição da evolução geral de um grafo dinâmico

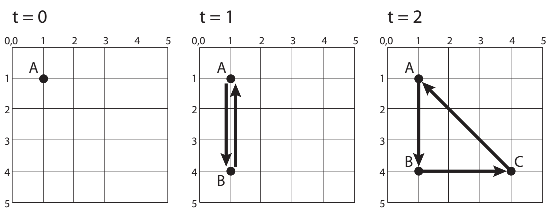

Figura 6. Visualização da evolução geral de um grafo dinâmico

Dispondo das informações da tabela consolidada, é possível facilmente filtrar o estado do grafo em um determinado momento. Por exemplo, para saber como o grafo estava no tempo t = 2, observa-se que esta unidade de tempo está dentro do intervalo de tempo correspondentes às linhas 2, 4 e 5 da tabela 5. Com as informações destas linhas é possível montar o grafo. Cada grafo gerado trata-se de um recorte demonstrado na figura 1 e a transição entre cada recorte possível gera a visualização da movimentação sobre o grafo.

Nesta abordagem, nota-se também que o arco $(A \rightarrow B)$ surge simultaneamente ao arco $(B \rightarrow A)$, o que força o conhecimento prévio da existência destes vértices A e B em $t = 1$.

O grafo dinâmico anteriormente descrito, mantém as propriedades de um grafo estático que evolui no tempo, e acrescenta o conceito de mudança de atributos ao longo do tempo. Deste modo, para que seja possível um processo de edição e representação visual do contexto de evolução do mesmo, é necessário que se inclua uma estrutura de dados pertinente para isto.

3 Modelo Computacional do Grafo Dinâmico

A estrutura de dados lista de vetores pode ser usada para representar um grafo dinâmico, como segue:

1. Estrutura de *Bindings*

Código-fonte 1. Estrutura Binding

```
Binding = Record
Case Status of
 Vertex :
  begin
     forma   : integer; ( forma de um vertice - circulo, quadrado,
         estrela, tec.)
     fill_cor       : integer; ( cor de preenchimento da forma )
     borda_cor : integer; ( cor da borda da forma )
  end;
 Ligacao :
  begin
    traco   : integer; ( forma de uma ligacao - curva, linha, etc)
    traco_cor : integer; ( cor do traco )
    traco_lg : integer;  ( largura do traco )
  end;
```

2. Estrutura Vértice Dinâmico

Código-fonte 2. Estrutura Vertex

```
Vertex = Record  ( Lista de Sequencia )
Rot  : integer; ( Indice do Vertice )
a,b  : integer; ( a - tempo do nascimento; b - tempo da morte )
x,y  : integer; ( atributos de coordenadas do vertice )
at   : integer; ( atributo de um vertice - peso, valor, etc)
St   : integer; ( Estatus do vertice )
Bd   :  Binding; ( Caracteristicas do Vertice )
end;
```

3. Estrutura Ligação Dinamica

Código-fonte 3. Estrutura Ligacao

```
Ligacao = Record  ( Ligacoes )
vi, vj : integer; ( vertices da ligacao )
TpL    : integer;  ( Tipo da ligacao: 0 - Elo, 1 - Arco )
a,b    : integer; ( a - tempo do nascimento; b - tempo da morte )
Valor  : integer; ( valor da ligacao )
Bd     : Binding; ( Caracteristicas da Ligacao )
end;
```

4. Estrutura Elemento Dinâmico (entry)

Código-fonte 4. Estrutura Entry

```
Entry = Record  ( Elementos )
 Status : (Vertice, Ligacao);
 Case Status of
```

```
  Vertice   : Vertex;
  Lig       : Ligacao;
end;
```

5. Estrutura Sequência de Partes de Grafo

Código-fonte 5. Estrutura Sequencia

```
Sequencia = Record  ( Lista de Sequencia )
LastCreated  : Array of Entry; ( Elementos criados nesta sequencia )
LastRemoved  : Array of Entry; ( Ultimos elementos retirados );
end;
```

6. Estrutura Grafo Dinâmico

Código-fonte 6. Estrutura DYNGrafo

```
DYNGrafo = Record
Rot        : String; ( Rotulo do Grafo)
NVertex    : Integer; ( Nr de Vertices )
NLig       : Integer; ( Nr de Ligacoes )
NSeq       : Integer; ( Nr de seq. que descrevem o grafo dinamico )
Ti, Tf     : Integer; ( Tempo inicial/final de existencia do grafo )
S          : array of Sequencia; ( Lista de seqs. que contem o grafo )
end;
```

Na estrutura de dados indicada acima, permite-se descrever os elementos que existem no grafo dinâmico, deste modo ele pode ser construído sem a preocupação formal de caracterizar e tratar cada um dos seus vértices e ligações como elementos específicos. Esta abstração faz com que se torne transparente o elemento que será apresentado ou editado futuramente. Estes elementos serão identificados convenientemente e tratados da mesma forma, maximizando o desempenho de construção e a possível edição do grafo.

4 O Processo de Edição de Grafos Dinâmicos

O modelo de Estrutura de Dados apresentado na seção 3, cada rótulo está associado a um vetor de vértices. Se se pretende que os rótulos sejam únicos, então, tem-se que atualizá-los a cada edição. Esta edição pode ocorrer de duas formas: Dinâmica e Estática. Na primeira, Edição Dinâmica, os vértices e ligações são incluídos e retirados ao longo do tempo de vida do grafo, à medida que o tempo evolui. Enquanto a segunda, Edição Estática, acontece em tempos específicos, após a criação de grafo num período de análise. Neste caso, retrocede-se na linha do tempo ao tempo em que se deseja alterar/incluir/retirar dados do grafo em determinado tempo t[k]. O presente e o futuro são afetados aqui, pois inclui-se no processo a possibilidade de remodelagem do grafo.

Para tratar o caso Estático, deve-se considerar o problema da Edição em Inserção (Inclusão e Alteração de vértices e ligações) e a Edição em Deleção (retirada em t[k] de vértices e ligações).

4.1 Inserção

Operações relacionadas:

1. Inserção Corrida
   Aqui preocupa-se com a inserção de um elemento em uma unidade de tempo do processo de construção dinâmica do grafo. Isto é feito de forma bastante simples.

É adicionada uma nova sequência no tempo em que se iniciou a edição t_i, e um um novo elemento é incrementado a esta sequência. Se um vértice é incluído, o elemento inserido na sequência é este e carrega os atributos que estão configurados para isto. Se uma ligação é incluída, inclui-se os atributos e características da ligação, indicando-se também a origem e o destino. O tempo de início de existência da ligação é marcado a partir da seleção de um dos seus vértices. Esta operação é $O(1)$;

2. Inserção Estática
Neste caso, em uma unidade de tempo do intervalo de existência do grafo, ou melhor, do período já observado, pretende-se inserir um novo elemento no grafo (Alteração do grafo dinâmico num tempo do período já construído).
Se houve uma inserção em um processo de edição no tempo, é porque o grafo gerado anteriormente não correspondia ao que deveria ser mostrado, assim, a alteração no tempo significa que o presente da edição e o futuro a partir dela também serão afetados. Caso não se queira rerotulá-los, não há o que fazer.
Isso é uma tarefa bem mais complexa, quando se quer manter os rótulos do grafo mantendo a numeração dentro do número de vértices que existe a cada unidade de tempo:
   (a) Ao ser inserido um novo vértice no tempo t[k], este vértice deverá assumir o ($rot + 1$) do último vértice inserido neste grafo neste tempo. Os demais vértices maiores que (rot) que foram criados nos tempos seguintes t[$k+1$]..t[t_f] devem ser rerotulados em +1, e assim os vértices das ligações correspondentes devem ser rerotuladas, esta operação é O(n) onde n é o número de vértices do grafo.
   (b) Note também que, ao longo do tempo, a partir de t[$k + 1$], haverá vértices que foram deletados dentro do período de existência do Grafo. Neste caso, os vértices deletados que tiverem rótulos menores que (rot) se mantêm com o mesmo rótulo, e os que tiverem rótulos maiores que (rot), devem ser atualizados em ($rot + 1$).
   (c) As ligações retiradas devem ser rerotuladas em conformidade, pois o que há é uma referência a seu rótulo, que foi alterado no tempo, esta operação é $O(m)$.

4.2 Deleção

1. Deleção Dinâmica: A morte de um vértice e/ou de uma ligação no tempo, em um processo contínuo de acompanhamento de um grafo, significa prioritariamente a alteração de seus estados, e a rerotulação do vértice a partir deste tempo. Esta operação já foi demonstrada antes, na seção 2, e é $O(n+m)$, onde n, m é o número de vértices e ligações respectivamente presente em todo o grafo dinâmico.
2. Deleção Estática: Se a preocupação de rerotulação for necessária, a deleção estática de um vértice é um problema complicado que se resolve em 3 etapas sequencialmente. Na primeira deve-se remover todas as ligações associadas ao vértice que sai. Na segunda deve-se remover o vértice desejado, e na terceira deve-se rerotular todos os vértices da rede com rótulo maior que o rótulo de remoção. Esta visão pode ser implementada para o grafo que não varia no tempo, só cresce em vértices e ligações. A complexidade é a mesma da deleção dinâmica.
Na edição em um tempo t[k] a deleção deve manter as mesmas etapas, no entanto cuidando também do tempo de vida dos vértices e ligações do futuro.

Deve-se então verificar quais são as ligações deste tempo de edição até o final da vida do grafo, que incluem o vértice que será deletado. Se houver ligações a este vértice, removê-las fisicamente, e proceder com a atualização dos rótulos das arestas nos tempos iguais ou posteriores a t[k].

5 O DYNAGRAPH

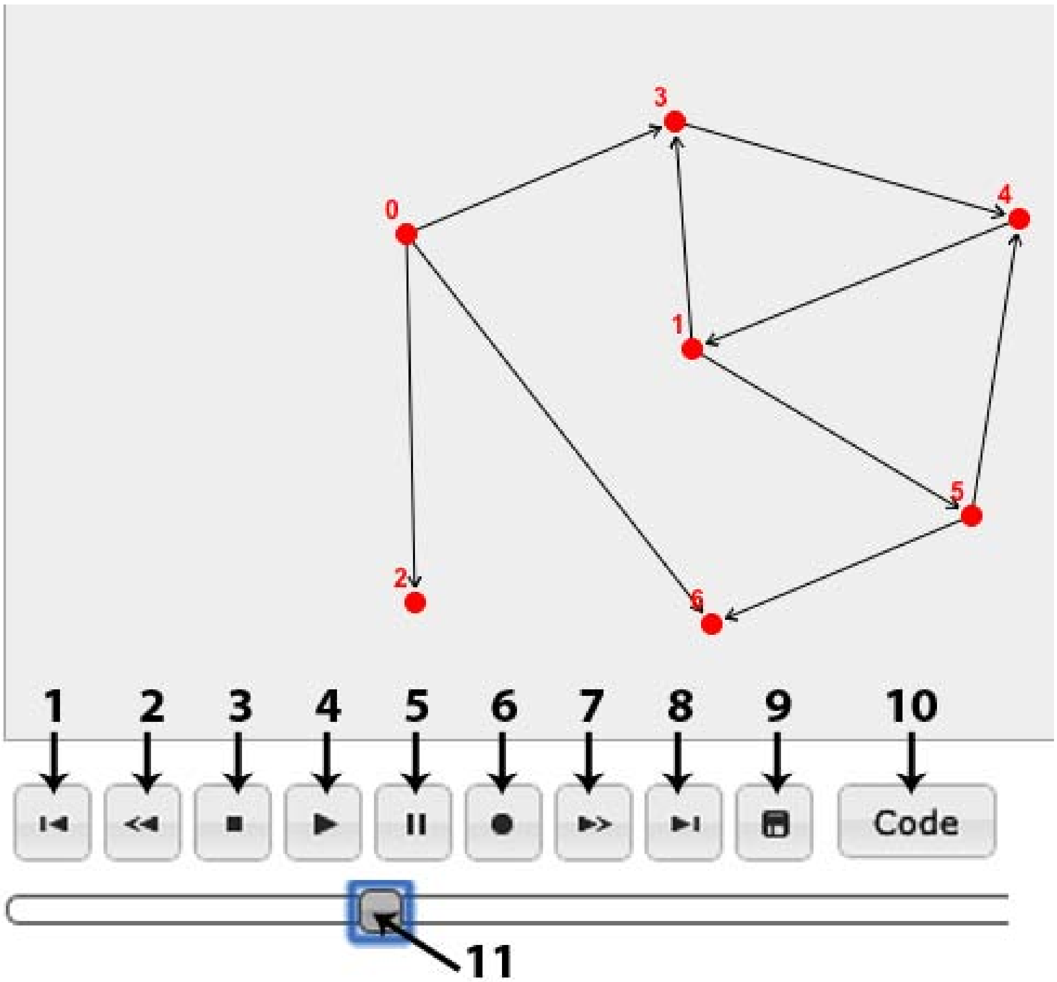

Figura 7. Controles de reprodução das ações do grafo dinâmico.

O software DYNAGRAPH foi criado contendo os recursos computacionais aqui descritos, com o objetivo de acompanhar a evolução dos grafos dessa natureza. Pode ser testado através do site *http://www.andersoncalixto.com.br/grafos*. O software foi desenvolvido em JavaScript para web, e sua interface, vista nas figuras 7 e 8, possui uma série de controles para interação com o usuário. São elas:

- Controles de reprodução, numerados na figura 7:

  1. Retornar ao início das ações registradas;
  2. Reproduzir em sentindo inverso as ações registradas, em velocidade dobrada;
  3. Parar a reprodução, posicionando a barra de progresso no início;

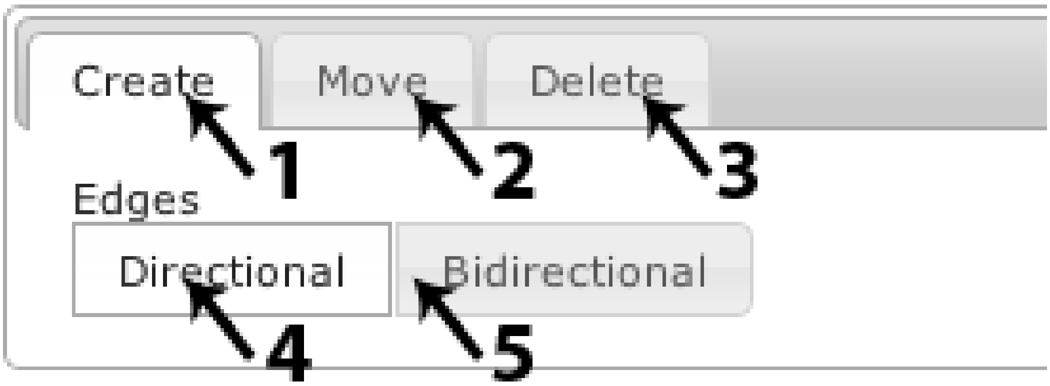

Figura 8. Controles para manipulação dos elementos do grafo.

4. Reproduzir as ações registradas;
5. Pausar reprodução;
6. Gravar novas ações do usuário, no tempo definido;
7. Reproduz em velocidade dobrada;
8. Posiciona a barra de progresso das ações na última ação registrada;
9. Grava a estrutura de dados do grafo dinâmico para ser carregada posteriormente;
10. Exibe a estrutura de dados atual do grafo;
11. Barra de progresso de reprodução das ações do grafo.

- Quando o controle de reprodução estiver em estado de gravação, são acionados os seguintes controles para edição: criação, movimentação e remoção, numerados na figura 8:

  1. Define a ação para criação;
  2. Define a ação para movimentação: mover vértices;
  3. Define a ação para remover vértices e arestas;
  4. Em momento de criação, define a criação das arestas como arco;
  5. Em momento de criação, define a criação das arestas como elo.

6 Conclusão

Este artigo ampliou um modelo computacional que resolve o problema de descrição de um grafo dinâmico nas suas diversas formas. O modelo computacional adotado, permite ao grafo possuir qualquer característica, e que mudanças nessas características sejam armazenadas em sequências de dados que possuem grau de dependência. Assim, o grafo se tornará totalmente dinâmico, permitindo inclusive o desaparecimento de vértices e ligações ao longo do tempo.

O modelo apresentado permite o armazenamento e recuperação de dados temporais em grafos dinâmicos, podendo observar o estado anterior de atributos, e fazer análises para antever eventos futuros.

O trabalho sobre o DYNAGRAPH continuará abordando as suas caracteristicas de centralidade (tempo médio de vida e morte de vértices e ligações, vizinhança e cardinalidade temporal, fecho transitivo, conectividade e outras), será ajustado para trabalhar com geoprocessamento (aplicações no controle de doenças epidêmicas, evolução de redes elétricas, fluxo de tráfego), ajustando-se a arquivos no padrão XML.

Referências

[Euler (1736)] Euler, L. (1736). Solutio problematis ad geometriam situs pertinentis. *Graph Theory 1736-1936*.

[Ferreira (2004)] Ferreira, A. (2004). Building a reference combinatorial model for manets. 18:1–24.

[Kempe (2002)] Kempe, D., Kleinberg, J., and Kumar, A. (2002). Connectivity and inference problems for temporal networks. 64:820–842.

[Kim (2012)] Kim, H. and Anderson, R. (2012). Temporal node centrality in complex networks. page 8.

[Kostakos (2009)] Kostakos, V. (2009). Temporal graphs. 388:1007–1023.

[Newman (2006)] Newman, M., Barabási, A.-L., and Watts, D. J. (2006). The structure and dynamics of networks.