Um Estudo sobre a Aplicação de Ordenações Topológicas no Problema de Alcançabilidade em Grafos Grandes

Rodrigo Ferreira da Silva
Departamento de Ciência da Computação - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, Pampulha, Belo Horizonte, Minas Gerais
rfsilva@dcc.ufmg.br

Sebastián Alberto Urrutia
Departamento de Computação - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, Pampulha, Belo Horizonte, Minas Gerais
surrutia@dcc.ufmg.br

RESUMO

O problema de alcançabilidade consiste em, dados um grafo $G = (V, E)$ e dois vértices $u, v \in V$, responder se a partir de u é possível chegar a v. Para grafos grandes é inviável realizar uma busca a cada consulta ou armazenar o fecho transitivo completo. Abordagens intermediárias geram índices que são utilizados durante a execução de consultas. A abordagem FELINE utiliza em seu índice duas ordenações topológicas e, a partir da posição dos vértices, responde sobre a alcançabilidade. Em certos casos, chamados de falsos positivos, é necessário executar uma busca no grafo que pode ser otimizada através de podas com o uso do índice. Neste trabalho, mostramos que muito ainda pode ser feito para melhorar a geração de índices pela abordagem FELINE. Os resultados mostram que o número de falsos positivos é muito superior ao ótimo computado por abordagens exatas. Além disso, a FELINE possui um limite intrínseco em seu desempenho, gerando falsos positivos desnecessários, ocasionando buscas excessivas no grafo original.

PALAVRAS CHAVE. Alcançabilidade em Grafos, Desenho de Dominância Fraca, Ordenação Topológica.

Área principal. Teoria e Algoritmos em Grafos

ABSTRACT

The reachability problem is defined as, given a graph $G = (V, E)$ and two vertices $u, v \in V$, answer whether there is a path from a vertice u to vertice v. For large graphs is not feasible to search the graph on each query or neither store its full transitive closure. Intermediate approaches use indexes during query execution. FELINE uses in its index two topological sorts and take the vertices position into account to answer on reachability queries. In some cases, called false implied paths, it is necessary to perform a search in the graph that can be pruned using the index. In this paper, we show that much can be done to improve the index built by FELINE approach. The results show that the number of false implied paths is much higher than the optimal solution computed by exact approaches. Furthermore, FELINE has an inherent limit on its performance, generating unnecessary false implied paths resulting in excessive searching throughout the graph.

KEYWORDS. Reachability in Graphs, Weak Dominance Drawing, Topological Sort.

Main Area. Theory and Algorithms in Graphs

1. Introdução

Um problema recorrente na área de grafos é, dado um grafo direcionado $G = (V, E)$ e dois vértices $u, v \in V$, avaliar se existe um caminho entre u e v, isto é, responder se a partir de u é possível alcançar v. Esse problema pode ser resolvido de forma trivial, com a execução de uma busca em profundidade. Porém, o problema se torna mais complexo quando se trata de grafos muito grandes como a própria Web, considerando as páginas como vértices e os *hyperlinks* representados como arestas direcionadas [Kumar et al., 2000]. É um grafo gigantesco, com aproximadamente cinco bilhões de vértices e dezenas de bilhões de *hyperlinks*, e parece crescer exponencialmente com o tempo. Para este exemplo, uma consulta sobre alcançabilidade poderia responder, por exemplo, se a partir de uma página Web é possível alcançar outra qualquer somente seguindo *hyperlinks*.

Existem duas abordagens básicas que representam dois extremos quando se trata do projeto de índices para auxiliar consulta de alcançabilidade ([Yildirim et al., 2010, 2012]). Em um dos extremos encontra-se a abordagem que extrai e armazena o fecho transitivo completo do grafo, isto é, gera um grafo que contém uma aresta (u, v) quando existir um caminho de u até v no grafo G. Através desta estrutura é possível responder em tempo $O(1)$ uma pergunta sobre a alcançabilidade entre quaisquer vértices u e v. Por outro lado, o tempo de construção deste índice é da ordem de $O(|V| * |E|)$ e o tamanho do índice armazenado da ordem de $O(|V|^2)$.

No outro extremo, se encontra a abordagem que realiza uma consulta no grafo a cada solicitação sobre alcançabilidade entre dois vértices. Essa consulta é feita através de uma busca no grafo, em largura ou em profundidade, que executa em tempo $O(|V| + |E|)$. Neste caso, não existe a necessidade de nenhum pré-processamento ou etapa de inicialização.

Do ponto de vista prático, quando tratamos de grafos muito grandes nenhuma das duas abordagens é adequada. Na primeira abordagem, o tamanho do índice armazenado é na ordem de $O(|V|^2)$, o que torna o armazenamento infactível para grafos grandes. Na segunda abordagem, no seu pior caso, existe a necessidade de se percorrer todo o grafo para ser possível responder uma única consulta no tempo $O(|V| + |E|)$, o que também inviabiliza a aplicação em grafos muito grandes e consultas recorrentes. Dessa forma, as abordagens existentes e utilizadas na prática se encontram entre esses dois extremos.

Parte das abordagens propostas para o problema de alcançabilidade não são escaláveis, isto é, no contexto de grafos muito grandes os métodos não apresentam desempenho aceitável. Por exemplo, o método proposto em [Simon, 1988] que computa o fecho trasitivo de arestas em um grafo direcionado acíclico.

Entre os métodos escaláveis se destaca o FELINE (*Fast rEfined onLINE search*) [Veloso et al., 2014], inspirado pelo Problema do Desenho de Dominância em um grafo (*Dominance Graph Drawing*) [Eades et al., 1994] e derivada da abordagem definida em [Kameda, 1975]. Essa abordagem, assim como outras abordagens escaláveis na literatura, é valida para grafos direcionados acíclicos, que podem ser construídos a partir dos grafos originais pela condensação dos seus componentes fortemente conectados em um único vértice.

O algoritmo leva em consideração uma propriedade da ordenação topológica sobre um grafo direcionado acíclico qualquer, segundo a qual dados dois vértices u e v, se v ocorre antes de u na ordenação, pode-se garantir que u não alcança v. Se v ocorre depois de u, não é possível afirmação alguma sobre a alcançabilidade, já que neste caso pode representar um falso positivo.

Para diminuir a incidência de falsos positivos, são construídas duas ordenações topológicas. A primeira ordenação topológica é gerada através de um caminhamento no grafo, através de uma busca em profundidade. A segunda ordenação topológica é gerada com a utilização da heurística proposta em [Kornaropoulos e Tollis, 2012a], que tem por objetivo minimizar o número de falsos positivos considerando as duas ordenações topológicas. A partir da posição de cada vértice u nas duas ordenações topológicas são geradas as coordenadas $(X(u), Y(u))$ em um plano Cartesiano, chamadas de desenho de dominância do grafo. Se existir um caminho entre u e v no grafo, então $X(u) \leq X(v)$ e $Y(u) \leq Y(v)$ [Kornaropoulos e Tollis, 2012b]. No entanto, se

$X(u) \leq X(v)$ e $Y(u) \leq Y(v)$, não é possível garantir sobre a alcançabilidade de v a partir de u. Neste caso, é necessário utilizar algoritmos de busca para se chegar a um resultado definitivo. Quando é necessário executar uma busca no grafo, a FELINE utiliza as informações indexadas (ordenações topológicas) para remover partes da árvore de pesquisa que certamente não alcançarão o vértice v.

FELINE [Veloso et al., 2014] é uma abordagem inovadora e promissora que pode ser aperfeiçoada para atingir melhores resultados. Uma das linhas de evolução para FELINE é diminuir os falsos positivos presentes no índice gerado na etapa de inicialização. Essa diminuição pode ser alcançada com a otimização do algoritmo que gera as duas ordenações topológicas para a construção do seu índice. Neste trabalho, vamos mostrar que não só existe a possibilidade de melhoria, mas, que existe uma grande lacuna entre os resultados alcançados por FELINE e o valor ótimo que pode ser alcançado na minimização de falsos positivos entre duas ordenações topológicas para geração do seu índice.

O restante deste trabalho está organizado como se segue. Na Seção 2, são apresentadas abordagens escaláveis com índices baseados em intervalos para o problema de alcançabilidade em grafos grandes. Em seguida, a Seção 3 apresenta o problema de Desenho de Dominância Fraca, que é um dos fundamentos para a geração das ordenações topológicas utilizadas no índice do FELINE e a Seção 4 apresenta mais detalhes sobre a sua abordagem. A Seção 5 apresenta modelos matemáticos para o problema de Desenho de Dominância e para o problema de Desenho de Dominância Fraca com uma Coordenada Fixa, este último sendo uma especialização do primeiro. Na Seção 6 são apresentados os resultados alcançados pela implementação dos modelos e, na seção seguinte, é feita uma discussão sobre a implicação desses resultados. Por fim, são delineadas as conclusões e trabalhos futuros.

2. Abordagens Escaláveis Baseadas em Intervalos para o Problema de Alcançabilidade

Além do FELINE, outros métodos na literatura propõem-se resolver o problema de alcançabilidade em grafos grandes. Todos os métodos criam índices baseados em intervalos que são atribuídos aos vértices do grafo. Entre eles estão o GRAIL [Yildirim et al., 2010], FERRARI [Anand et al., 2013] e Interval-Index [Li et al., 2015], que serão apresentados a seguir.

O primeiro método escalável para o problema de alcançabilidade em grafos grandes foi o GRAIL (*Graph Reachability indexing via rAndomized Interval Labeling*) [Yildirim et al., 2010], que propõe a utilização de um índice simples e escalável, baseado na ideia de rotulação de intervalos aleatórios. Cada vértice u do grafo é rotulado com um intervalo $L_u = [r_x, r_u]$, no qual r_u indica o valor atribuído a u em um caminhamento pós-ordem. Já a r_x é atribuído o menor valor entre os vértices filhos de u na árvore resultante do caminhamento. Com isso, se a partir de u é possível alcançar um vértice v, então $L_v \subseteq L_u$.

No entanto, essa abordagem está sujeita à ocorrência de falsos positivos, isto é, quando $L_v \subseteq L_u$ mas u não alcança v. Dessa forma, quando um resultado da comparação entre os intervalos é negativo, $L_v \nsubseteq L_u$ é possível garantir que o vértice v não é alcançável a partir de u. Porém, no caso contrário, $L_v \subseteq L_u$, não é possível saber de antemão se o resultado é verdadeiro ou um falso positivo. Nesses casos, é necessário executar uma busca no grafo para confirmar o resultado, que é otimizada com esquemas de poda e direcionamento usando as informações armazenadas nos índices.

Outro método que merece ser mencionado, similar ao GRAIL, é o FERRARI (*Flexible and Efficient Reachability Range Assignment for gRaph Indexing*) [Anand et al., 2013] que também utiliza intervalos para obter indicações sobre a alcançabilidade entre vértices no grafo. Esse algoritmo apresenta melhorias na técnica de rotulação na qual cada nodo é codificado de forma compacta como uma coleção de intervalos de identificadores. Nos casos de falsos positivos, as informações indexadas são utilizadas para podar e direcionar a busca no grafo.

Por fim, mais recentemente em [Li et al., 2015] foi proposta uma nova abordagem que cria uma coleção de partições do grafo, cada partição sendo uma subárvore do grafo. São criados

intervalos para as partições de acordo com o índice atribuído em cada vértice ao ser percorrer o grafo. Dessa forma, executando-se duas buscas binárias, uma no índice e outra na partição do vértice de destino, é possível responder a uma consulta sobre a alcançabilidade entre dois vértices. Na prática, essa abordagem se mostrou mais eficiente que a FELINE, pois, não é necessário percorrer o grafo quando o índice não consegue responder uma consulta, já que o índice é completo.

3. O Problema de Desenho de Dominância Fraca

A abodargem utilizada pela FELINE para computar o seu índice foi inspirada pelo Problema do Desenho de Dominância [Eades et al., 1994], ou *Dominance Drawing* (DD), que consiste em obter duas ordenações topológicas, X e Y, de um grafo G tais que se dados dois vértices quaisquer u e v, se u aparece antes de v nas duas ordenações topológicas, então u alcança v. Dessa forma, é possível desenhar os vértices do grafo em um plano, a partir das coordenadas estabelecidas pelas posições $X(v)$ e $Y(v)$ de cada vértice v nas ordenações topológicas.

A partir desse desenho é possível identificar visualmente a alcançabilidade entre quaisquer dois vértices, dado que se u alcança v, então $X(u) < X(v)$ e $Y(u) < Y(v)$. Caso contrário, isto é, se $X(u) > X(v)$ ou $Y(u) > Y(v)$, é possível afirmar que u não alcança v. Com isso, se for necessário, por exemplo, estabeler quais são todos os vértices alcançáveis a partir de um vértice v, basta observar apenas o quadrante superior à $X(v)$ e $Y(v)$.

A aplicação do Desenho de Dominância está fortemente relacionada ao conceito de dimensão do grafo ($dim(G)$) que é definida como o menor valor de k para o qual um Desenho de Dominância com a dimensão k pode ser obtido [Eades et al., 1994]. O problema de decisão equivalente foi provada ser NP-Completo em [Brightwell e Massow, 2013].

Uma direção natural para a extensão desse problema é tentar obter um Desenho de Dominância de um grafo G no plano com a relaxação de algumas restrições. Dessa forma, foi proposto em [Kornaropoulos e Tollis, 2012b] o problema de Desenho de Dominância Fraca (*Weak Dominance Drawing*) que consiste em, dado um grafo direcionado acíclico $G(V, E)$, determinar duas ordenações topológicas que minimizem o número de falsos positivos. Dadas duas ordenações topológicas $X = [x_1, x_2, ..., x_n]$ e $Y = [y_1, y_2, ..., y_n]$ de G, um falso-positivo é caracterizado quando temos os vértices $v, u \in V$, sendo que v não alcança u em G, e v aparece antes de u em ambas ordenações topológicas, X e Y. Com isso, se observadas apenas as duas ordenações topológicas, poderia-se pensar que u alcança v, o que não é verdade. A interseção entre duas ordenações topológicas é definida como o conjunto $I = \{(u, v)|t_X(u) < t_X(u) \wedge t_Y(u) < t_Y(u)\}$ ([Kornaropoulos e Tollis, 2012b]). Quanto menor o número de falsos-positivos, menor a interseção entre as duas ordenações topológicas.

A versão de decisão do problema consiste em, dado um grafo G e um número inteiro positivo k, estabelecer se existem duas ordenações topológicas tais que o número de falsos-positivos seja no máximo k, e foi provado ser NP-Completo em [Kornaropoulos e Tollis, 2012b].

4. Alcançabilidade em Grafos Grandes usando FELINE

Para computar o seu índice, o FELINE utiliza uma heurística para o Problema de Desenho de Dominância Fraca. A heurística recebe uma ordenação topológica qualquer do grafo como entrada e gera uma segunda ordenação topológica. A heurística para geração da segunda ordenação topológica é apresentada no Algoritmo 1, chamado de *Maximum-Rank* [Kornaropoulos e Tollis, 2012a] pois, a cada iteração, escolhe o vértice com o ranking máximo considerando a ordenação topológica dada como entrada.

O conjunto S_G é inicializado contendo todos os vértices fonte do grafo G, que são os vértices para os quais o grau de entrada é 0. A ordenação topológica Y, na qual será armazenado o resultado do algoritmo, é inicialmente vazia. Em seguida, a cada iteração o algoritmo escolhe o vértice com o ranking máximo no conjunto S_G através da função $maxRank_X(S_G)$. Essa função retorna o vértice de $u \in S_G$ que está na maior posição na ordenação topológica X. O vértice selecionado é removido do grafo, juntamente com as suas arestas incidentes e também removido

Algorithm 1: Maximum-Rank

Input: G, X
Output: Y
Inicializa S_G ;
$Y \leftarrow Nil$;
for $i \leftarrow 1$ *to* n **do**
  $u \leftarrow maxRank_X(S_G)$;
  $G \leftarrow G - u$;
  Atualiza S_G ;
  $Y[i] = u$;
end

de S_G. O conjunto S_G é atualizado incluindo todos os vértices fonte do grafo G resultantes com a remoção do vértice u. O vértice u é adicionado à ordenação topológica Y.

É importante ressaltar que a escolha do vértice $u \in S_G$ a ser inserido em Y realizada a cada iteração é localmente ótima entre todos os vértices de S_G pois a posição escolhida para o vértice em Y não gera falsos positivos entre os pares de S_G [Kornaropoulos e Tollis, 2012a], já que os vértices $S_G - \{v\}$ serão adicionados posteriormente em Y.

Conforme mencionado, a heurística apresentada foi utilizada na abordagem FELINE como aproximação para o Problema de Desenho de Dominância Fraca. No entanto, fixando-se a primeira ordenação topológica, esta representada por X no Algoritmo 1, o problema a resolvido é distinto do original que consiste em encontrar duas ordenações topológicas que possuam o menor número de falsos positivos entre si. A solução proposta com a heurística passa a ser uma solução viável para o problema de se encontrar uma ordenação topológica Y do grafo G que minimize o número de falsos positivos em relação à ordenação topológica X fornecida como entrada.

Esse novo problema é chamado aqui de Desenho de Dominância Fraca com uma Coordenada Fixa (DDFCF) e o seu correspondente problema de decisão é definido a seguir:

Desenho de Dominância Fraca com uma Coordenada Fixa (*1-Fixed Dominance Drawing*) 1.
INSTÂNCIA: Um grafo direcionado acíclico $G = (V, E)$, uma ordenação topológica X, e um inteiro positivo C tal que $|E| \leq C \leq \frac{|V|(|V|-1)}{2}$.
PERGUNTA: Existe uma ordenação topológica Y de G cuja interseção tenha cardinalidade igual ou menor que C?

Até onde sabemos, não é conhecido um algoritmo polinomial ótimo para este problema e também, por outro lado, não foi provado ser NP-Completo. Dessa forma, a complexidade do Problema de Desenho de Dominância Fraca com uma Coordenada Fixa é uma questão ainda em aberto.

5. Formulações Matemáticas para os Problemas de Desenho de Dominância Fraca e com uma Coordenada Fixa

A solução de um problema de Desenho de Dominância Fraca consiste em duas ordenações topológicas quaisquer do grafo que minimizem o número de falsos positivos. Já no problema de Desenho de Dominância Fraca com uma Coordenada Fixa, é fornecida uma das ordenações topológicas como entrada e deve ser gerada a outra ordenação que minimize o número de falsos positivos. Para permitir a comparação entre os dois problemas, foram construídos modelos matemáticos e, a partir desses modelos, foram executados testes que apresentam a diferença entre eles na prática. Os testes referenciados são apresentados na Seção 6.

A seguir, será apresentado um modelo para o Problema Desenho de Dominância Fraca. As variáveis X representam a primeira ordenação topológica responsável por representar as coordenadas X no plano. Se o vértice i está antes do vértice j, ou $i \prec j$, na ordenação topológica X,

então $X_{ij} = 1$, caso contrário $X_{ij} = 0$. A ordenação topológica Y é definida de forma semelhante à X, conforme apresentado abaixo.

1. $X_{ij} = \begin{cases} 1 & \text{se } i \prec j \text{ na ordenação topológica } X \\ 0 & \text{c.c.} \end{cases}$
2. $Y_{ij} = \begin{cases} 1 & \text{se } i \prec j \text{ na ordenação topológica } Y \\ 0 & \text{c.c.} \end{cases}$

No modelo abaixo, a função objetivo minimiza a interseção entre as duas ordenações topológicas que é o caso no qual $X_{ij} = 1$ e $Y_{ij} = 1$. De forma simétrica, também são minimizados os casos em que $X_{ij} = 0$ e $Y_{ij} = 0$, já que, nessa situação, necessariamente $X_{ji} = 1$ e $Y_{ji} = 1$, pela restrição (2). A restrição (1) fixa a ordem de vértices para os quais existem arestas no grafo, enquanto a restrições (3) e (4) garantem que não ocorrerão ciclos na ordenação topológica X. As restrições (6), (7) e (8) tem as mesmas funções das restrições (3), (4) e (5), porém, relativas à ordenação topológica Y.

$$\text{Minimizar} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij}Y_{ij} + (1 - X_{ij})(1 - Y_{ij})$$

sujeito a:

$$X_{ij} = 1 \qquad \forall (i,j) \in E \qquad (1)$$
$$X_{ij} + X_{ji} = 1 \qquad \forall i,j \qquad (2)$$
$$X_{ij} + X_{jk} + X_{ki} \leq 2 \qquad \forall i,j,k \qquad (3)$$
$$X_{ij} + X_{jk} + X_{ki} \geq 1 \qquad \forall i,j,k \qquad (4)$$
$$Y_{ij} = 1 \qquad \forall (i,j) \in E \qquad (5)$$
$$Y_{ij} + Y_{ji} = 1 \qquad \forall i,j \qquad (6)$$
$$Y_{ij} + Y_{jk} + Y_{ki} \leq 2 \qquad \forall i,j,k \qquad (7)$$
$$Y_{ij} + Y_{jk} + Y_{ki} \geq 1 \qquad \forall i,j,k \qquad (8)$$
$$X_{ij}, Y_{ij} \in \{0,1\} \qquad \forall i,j \qquad (9)$$

O modelo definido anteriormente é não-Linear, pois inclui o produto $X_{ij}Y_{ij}$ na sua função objetivo. Esse mesmo modelo pode ser linearizado com a introdução da variável Z, a alteração da função objetivo e a introdução da restrição (10). A variável Z_{ij} é igual a 1 sempre que as variáveis $X_{ij} = 1$ e $Y_{ij} = 1$.

$$\text{Minimizar} \quad \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} 2Z_{ij} - X_{ij} - Y_{ij}$$

sujeito a:

$$Z_{ij} \geq X_{ij} + Y_{ij} - 1 \qquad \forall i,j \qquad (10)$$
$$X_{ij} = 1 \qquad \forall (i,j) \in E \qquad (11)$$
$$X_{ij} + X_{ji} = 1 \qquad \forall i,j \qquad (12)$$
$$X_{ij} + X_{jk} + X_{ki} \leq 2 \qquad \forall i,j,k \qquad (13)$$
$$X_{ij} + X_{jk} + X_{ki} \geq 1 \qquad \forall i,j,k \qquad (14)$$
$$Y_{ij} = 1 \qquad \forall (i,j) \in E \qquad (15)$$
$$Y_{ij} + Y_{ji} = 1 \qquad \forall i,j \qquad (16)$$
$$Y_{ij} + Y_{jk} + Y_{ki} \leq 2 \qquad \forall i,j,k \qquad (17)$$
$$Y_{ij} + Y_{jk} + Y_{ki} \geq 1 \qquad \forall i,j,k \qquad (18)$$
$$X_{ij}, Y_{ij}, Z_{ij} \in \{0,1\} \qquad \forall i,j \qquad (19)$$

O Problema de Dominância Fraca com uma Coordenada Fixa recebe como entrada uma das ordenações topológicas, que foi representada pela ordenação topológica y, constante no modelo. Dessa forma, o modelo inicial pode ser simplificado com a remoção das restrições relativas às variáveis Y_{ij}. A função objetivo passa a ser linear já que y é uma constante.

$$\text{Minimizar} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} y_{ij} + (1 - X_{ij})(1 - y_{ij})$$

sujeito a:

$$X_{ij} = 1 \qquad \forall (i,j) \in E \tag{20}$$

$$X_{ij} + X_{ji} = 1 \qquad \forall i, j \tag{21}$$

$$X_{ij} + X_{jk} + X_{ki} \leq 2 \qquad \forall i, j, k \tag{22}$$

$$X_{ij} + X_{jk} + X_{ki} \geq 1 \qquad \forall i, j, k \tag{23}$$

$$X_{ij} \in \{0, 1\} \qquad \forall i, j \tag{24}$$

6. Resultados Computacionais

Para ser possível produzir uma comparação, os modelos de Programação Linear Inteira apresentados na seção anterior foram implementados com a utilização do IBM CPLEX 12.6 e executados em uma máquina com processador Intel Core i5 de 2.8 GHz e 12 GB de memória RAM. Além disso, para fins de comparação, foi executada a heurística proposta em [Kornaropoulos e Tollis, 2012a] utilizada pela FELINE [Veloso et al., 2014]. Para simplificar, chamamos o modelo desenvolvido para o Problema de Desenho de Dominância Fraca de DDF e o modelo para o Problema de Dominância Fraca com uma Coordenada Fixa de DDFCF.

O DDFCF e a heurística requerem uma das ordenações topológicas como entrada para a sua execução e, para fins de comparação, nos testes apresentados foi utilizada a mesma ordenação topológica nas duas abordagens. Esta ordenação topológica foi gerada através de um caminhamento por busca em profundidade no grafo, assim como é gerada na abordagem FELINE.

Tabela 1: Comparação do número de falsos positivos gerados pelas abordagens exatas para o Problema de Dominância Fraca e Problema de Dominância Fraca com uma Coordenada Fixa, e a heurística apresentada em [Kornaropoulos e Tollis, 2012a] e utilizada por FELINE

Vértices	Arestas	FPs DDF	FPs DDFCF	FPs Heurística
10	25	1 (1,0%)	3 (3,0%)	3 (3,0%)
20	62	1 (0,2%)	6 (1,5%)	7 (1,8%)
30	86	1 (0,1%)	12 (1,3%)	20 (2,2%)
40	108	5 (0,3%)	21 (1,3%)	29 (1,8%)
50	132	14 (0,6%)	80 (3,2%)	89 (3,6%)
60	177	22 (0,7%)	222 (6,2%)	331 (9,2%)
70	235	38 (0,7%)	326 (6,7%)	335 (6,8%)
80	258	57 (0,8%)	374 (5,8%)	691 (10,8%)
90	341	64 (0,7%)	562 (6,9%)	1.167 (14,4%)
100	442	91 (0,9%)	660 (6,6%)	1.647 (16,5%)

A Tabela 1 apresenta os resultados alcançados durante a execução dos algoritmos usando um grafo direcionado acíclico criado a partir do Arxiv[1], que é uma instância de um grafo real denso que representa um arquivo de impressões eletrônicas de artigos científicos nos campos da

[1] arxiv.org

matemática, física, ciência da computação, biologia quantitativa, finança quantitativa e estatística, que podem ser acessados via Internet. A instância possui um total de 6.000 vértices e 66.707 arestas. Porém, para esse teste, foram utilizadas instâncias reduzidas que contém apenas uma fração desse grafo, para que fosse possível executar os algoritmos exatos. Para gerar as instâncias reduzidas, foram selecionados os primeiros N vértices do grafo e mantidas as arestas originais entre eles.

Além do número absoluto de potenciais falsos positivos, é apresentado também um percentual em relação no número total de possíveis potenciais falsos positivos que, no pior caso, representam a situação em que todos os vértices estão na mesma ordem relativa nas duas ordenações topológicas ou, mais especificamente, as ordenações topológicas são idênticas.

Comparando-se as soluções ótimas geradas para o DDF e para DDFCF, é possível observar que, conforme o tamanho da instância aumenta, o número de potenciais falsos positivos para o DDF é significativamente inferior ao número de potenciais falsos positivos para o DDFCF. Neste caso particular, para instâncias acima de 60 vértices, o número de falsos positivo é pelo menos 6 vezes menor.

A heurística gera resultados significativamente distantes do limite mínimo possível, que é o resultado para a abordagem exata implementada no DDFCF. Além disso, essa diferença percentual aumenta com o tamanho das instâncias, isto é, o resultado da heurística piora com o aumento do tamanho das instâncias.

Na Tabela 2 é apresentado o tempo de execução de cada algoritmo para a mesma instância anterior. O tempo de execução do DDF cresce rapidamente com o tamanho da instância enquanto que para o DDFCF o tempo cresce em uma velocidade bastante inferior. Para todas as instância computadas, a heurística foi executada em menos de 1 milissegundo.

Tabela 2: Comparação do tempo de execução das abordagens exatas para o Problema de Dominância Fraca e Problema de Dominância Fraca com uma Coordenada Fixa

Vértices	Arestas	Tempo DDF (s)	Tempo DDFCF (s)
10	25	1,40	0,12
20	62	1,25	0,25
30	86	1,15	0,44
40	108	16,13	1,11
50	132	80,76	1,77
60	177	131,10	1,71
70	235	449,90	2,86
80	258	3.839,90	5,16
90	341	5.604,30	7,17
100	442	23.167,86	9,92

7. Discussão sobre os Resultados

Na Seção anterior foram apresentados os resultados alcançados para os modelos implementados para os Problemas de Desenho de Dominância Fraca (modelo DDF) e para a sua variante com uma Coordenada Fixa (modelo DDFCF), que foram executados em instâncias específicas construídas com base em um grafo real. Os dois resultados alcançados pelo DDF e DDFCF foram comparados com a heurística proposta em [Kornaropoulos e Tollis, 2012a] e utilizada pela FELINE.

A partir dos resultados gerados, foi possível observar que o DDF produziu um número de potenciais falsos positivos significativamente menor que o DDFCF. Devido a esta diferença, é possível concluir que a estratégia de fixar uma ordenação topológica específica e fornecê-la como entrada para o algoritmo, o caso do DDFCF, diminui significativamente as chances de se obter uma boa solução em comparação com o DDF, mesmo com a utilização de uma abordagem exata.

É importante ressaltar que os resultados do DDFCF dependem fortemente da ordenação topológica fornecida como entrada. Caso seja fornecida como entrada para o DDFCF uma ordenação topológica específica que foi gerada como resultado pelo DDF, o DDFCF encontrará o resultado ótimo para o DDF, com o mínimo de potenciais falsos positivos. No entanto, a chance disso acontecer é muito pequena, devido ao número exponencial de possíveis ordenações topológicas para uma grafo direcionado acíclico, no pior caso. FELINE utiliza um procedimento simples para a geração da primeira ordenação topológica, que consiste em realizar uma busca em profundidade e acrescentar os vértices na ordem de finalização ao final da ordenação topológica. Esse procedimento foi o mesmo utilizado para a geração da ordenação topológica para o DDFCF, que gerou os resultados apresentados, muito distantes do mínimo para o DDF.

Mantendo-se a mesma ordenação topológica como entrada, o DDFCF apresenta um limite inferior para o número de potenciais falsos positivos gerados pela heurística utilizada em FELINE. Com relação às instâncias apresentadas, os resultados alcançados pela heurística se encontram distantes do seu limite inferior (DDFCF) e, este último, conforme já mencionado, se encontra distante do resultado para o DDF.

No entanto, a heurística procura resolver o problema apresentado no DDF, e não o problema apresentado no DDFCF que é o seu limite inferior. Em outras palavras, mesmo que a heurística conseguisse chegar próximo ao seu limite inferior para alguma instância específica, o resultado ainda seria distante do ótimo desejado (DDF). Com isso, é possível concluir que a abordagem utilizada em FELINE de fixar uma das ordenações topológicas, esta gerada através de uma busca em profundidade, não é uma boa estratégia para criar uma heurística para o DDF.

A partir do que foi apresentado, fica latente a necessidade de novas heurísticas para o problema DDF, que tenham como limite inferior o próprio DDF e que alcancem resultados mais próximos dos resultados ótimos apresentados para o problema. Outra questão que deve ser considerada para a geração dessa nova heurística é a sua complexidade computacional, que deve ser preferencialmente linear em relação ao tamanho do grafo de entrada, já que o problema original resolvido em FELINE implica na execução do algoritmo em grafos grandes.

8. Conclusão

Neste trabalho, foi apresentada a estratégia da abordagem FELINE para responder perguntas de alcançabilidade entre dois vértices em um grafo direcionado acíclico. FELINE utiliza a heurística proposta em [Kornaropoulos e Tollis, 2012a] para a geração das ordenações topológicas que compõem o seu índice.

Essa heurística utiliza uma das ordenações topológicas fornecida como entrada, gerada a partir de uma busca em profundidade no grafo, e o seu papel é encontrar a segunda ordenação topológica que minimize o número de potenciais falsos positivos entre essas duas ordenações. A fixação de uma das ordenações topológicas altera o problema fundamental a ser resolvido que passa do Problema de Dominância Fraca [Kornaropoulos e Tollis, 2012b] para o Problema de Dominância Fraca com uma Coordena Fixa.

Foram apresentadas formulações matemáticas para o Problema de Dominância Fraca e para a sua variante com uma Coordena Fixa, além do algoritmo referente à heurística utilizada por FELINE. As implementações dos modelos e da heurística foram comparadas em termos dos resultados gerados em uma instância de problema criada a partir de um grafo real.

Por fim, foi possível observar que a heurística aplicada na geração dos índice em FELINE não utiliza uma boa estratégia ao fixar uma ordenação topológica que é fornecida como entrada. Ao fixar uma das ordenações topológicas, o limite inferior do número de potenciais falsos positivos pode crescer significativamente, comprometendo o resultado final. Com isso, é necessária uma nova heurística que tenha como limite inferior o Problema de Dominância Fraca, e não o Problema de Dominância Fraca com uma Coordena Fixa, conforme proposto por FELINE.

Como trabalhos futuros, pretende-se trabalhar na concepção dessa nova heurística que consiga, ao mesmo tempo, fornecer resultados mais próximos do limite inferior para o problema

e também executar em tempo computacional linear, para que seja possível ser aplicada em grafos suficientemente grandes. Com isso pretende-se melhorar consideravelmente o desempenho de FELINE.

Referências

Anand, A., Seufert, S., Bedathur, S., e Weikum, G. (2013). Ferrari: Flexible and efficient reachability range assignment for graph indexing. In *Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013)*, ICDE '13, p. 1009–1020, Washington, DC, USA. IEEE Computer Society.

Brightwell, G. R. e Massow, M. (2013). Diametral pairs of linear extensions. *SIAM J. Discrete Math.*, 27(2):634–649.

Eades, P., ElGindy, H., Houle, M., Lenhart, B., Miller, M., Rappaport, D., e Whitesides, S. (1994). Dominance drawings of bipartite graphs.

Kameda, T. (1975). On the vector representation of the reachability in planar directed graphs. *Information Processing Letters*, 3(3):75 – 77.

Kornaropoulos, E. M. e Tollis, I. G. (2012a). Overloaded orthogonal drawings. In *Proceedings of the 19th International Conference on Graph Drawing*, GD'11, p. 242–253, Berlin, Heidelberg. Springer-Verlag.

Kornaropoulos, E. M. e Tollis, I. G. (2012b). Weak dominance drawings for directed acyclic graphs. In *Graph Drawing - 20th International Symposium, GD 2012, Redmond, WA, USA, September 19-21, 2012, Revised Selected Papers*, p. 559–560.

Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tompkins, A., e Upfal, E. (2000). The web as a graph. In *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '00, p. 1–10, New York, NY, USA. ACM. ISBN 1-58113-214-X.

Li, F., Yuan, P., e Jin, H. (2015). *Knowledge Science, Engineering and Management: 8th International Conference, KSEM 2015, Chongqing, China, October 28-30, 2015, Proceedings*, chapter Interval-Index: A Scalable and Fast Approach for Reachability Queries in Large Graphs, p. 224–235. Springer International Publishing, Cham.

Simon, K. (1988). An improved algorithm for transitive closure on acyclic digraphs. *Theor. Comput. Sci.*, 58(1-3):325–346.

Veloso, R. R., Cerf, L., Jr., W. M., e Zaki, M. J. (2014). Reachability queries in very large graphs: A fast refined online search approach. In *Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, March 24-28, 2014.*, p. 511–522.

Yildirim, H., Chaoji, V., e Zaki, M. J. (2010). Grail: Scalable reachability index for large graphs. *Proc. VLDB Endow.*, 3(1-2):276–284.

Yildirim, H., Chaoji, V., e Zaki, M. J. (2012). Grail: A scalable index for reachability queries in very large graphs. *The VLDB Journal*, 21(4):509–534.