Resolução de Problemas de Conversão Irreversível por Meio de Algoritmos Genéticos

Alain A. T. Amaral
Departamento de Computação
CEFET-MG, MG, Brasil
alainandreamaral@gmail.com

Elizabeth F. Wanner
Lab. de Algoritmos,
Metaheurísticas e Otimização
Depto de Computação
CEFET-MG, MG, Brasil
efwanner@decom.cefetmg.br

Vinícius F. dos Santos
Lab. de Algoritmos,
Metaheurísticas e Otimização
Depto de Computação
CEFET-MG, MG, Brasil
vinicius@decom.cefetmg.br

***Resumo*—Diferentes tipos de processos utilizam-se de grafos em sua modelagem matemática. Processos que simulam a disseminação de uma característica dentro de um conjunto de vértices de um grafo, de modo que um vértice detentor de tal característica a possuirá para sempre e passará a ser considerado um disseminador para os demais itens do conjunto são conhecidos como processos de conversão irreversível. Estes processos são ditos convergentes se, dado um conjunto de vértices inicialmente possuidores da característica para determinada simulação, todos os itens do conjunto passarem a possuir a característica após um certo período de tempo. Uma vez que encontrar o número mínimo de vértices de um grafo, que necessitam possuir a característica no inicio do processo, para que um processo de conversão irreversível seja convergente é considerado um problema NP-difícil, encontrar uma solução adequada para o problema em tempo hábil através de métodos determinísticos pode ser uma tarefa inviável para grafos com grande número de vértices. Portanto, este trabalho analisa a capacidade de resolução de tal problema por meio de algoritmos genéticos. Resultados preliminares indicam que a abordagem usada é capaz de resolver o problema de conversão irreversível de forma satisfatória.**

Keywords*—*Grafos, Algoritmos Genéticos, Processos de Conversão Irreversível, Conjuntos Convergentes, Otimização, Disseminação de característica

I. Introdução

Grafos são utilizados para representar uma gama de situações que envolvem relações entre itens de um determinado conjunto [1]. Como exemplos de relações que podem ser representadas matematicamente através de grafos, podemos citar: a distância física entre um conjunto de cidades, lógica entre dispositivos conectados a uma rede, a probabilidade de o fogo passar de uma árvore para outra no caso de um incêndio florestal ou de uma doença ser transmitida de um indivíduo para outro no caso de uma epidemia, o nível de influência de uma pessoa sobre outras dentro de uma comunidade, dentre outros. Por serem capazes de representar problemas de diferentes áreas, grafos, assim como os diferentes processos que são modelados utilizando-se de tal forma representativa, têm sido largamente estudados.

Dentre os diferentes tipos de processos que utilizam-se de grafos em sua modelagem matemática, existe um conjunto de processos denominados por processos de conversão irreversível. Tais processos simulam a disseminação ou transmissão de uma característica ou propriedade dentro de um conjunto de itens relacionados, em um determinado um período de tempo, de modo que um item detentor de tal característica ou propriedade a possuirá para sempre e será considerado um transmissor ou disseminador para os demais itens do conjunto. Estes processos são ditos convergentes para determinada simulação se todos os itens do conjunto possuírem a característica ou propriedade após um certo período de tempo.

A convergência de tais processos depende de diversos fatores, como a quantidade inicial de itens do conjunto que possuem a característica ou propriedade, a susceptibilidade de cada item para adquiri-las, além do tempo total decorrido no processo. Diferentes formas de modelagem abordam o problema de maneiras distintas, algumas tratam a susceptibilidade como uma função variante no tempo e específica para cada item do conjunto, outras a tratam como uma constante válida para todos os itens do conjunto, algumas ainda a analisam como uma probabilidade relacionada ao número de vizinhos quer possuem ou não a característica. Além disso, alguns modelos adicionam restrições de tempo para que o processo ocorra como um todo ou criam restrições de tempo específicas para cada item do conjunto, outros modelos simplesmente não o fazem. Independente da maneira como a questão de susceptibilidade ou tempo é tratada, a determinação de quais e da quantidade mínima de itens inicialmente possuidores da propriedade ou característica necessária para a convergência do processo pode ser vista como um problema de otimização.

Diferentes trabalhos, que podem ser encontrados na literatura, tendem a utilizar o processo de conversão irreversível de uma maneira diferente. Em [2], por exemplo, leva-se em conta que a susceptibilidade de um item depende de que a maioria dos seus vizinhos sejam possuidores da propriedade que está sendo transmitida. No mesmo trabalho, é estudada a relação entre a quantidade de itens inicialmente com a característica e o tempo necessário para que ocorra convergência do processo para diferentes topologias conhecidas de rede.

Em [3], é proposta a ideia de ”vacinar”certos itens. Assim, no problema proposto, um item que não possui a característica pode ser vacinado e nunca será possuidor da mesma. Dentro deste contexto, a ideia é de encontrar quais itens do conjunto devem ser vacinados inicialmente ou durante o processo para que a característica nunca chegue a um determinado subconjunto. Ou pode-se pensar em quais itens devem ser vacinados para minimizar a disseminação da característica dado um número limitado de vacinas. No mesmo trabalho, o problema de vacinação também foi proposto no contexto de combate ao fogo. Neste caso, pode-se pensar nos itens como árvores de uma floresta e no ato de vacinar um item como o ato de enviar bombeiros para proteger uma árvore.

O funcionamento de redes de distribuição de energia elétrica também são estudados através da análise da dinâmica de interação entre os itens que compõem a rede. Por exemplo, uma falha de energia em uma estação pode causar sobrecargas nas suas vizinhas que também podem vir a parar de funcionar [4]. Portanto, a dinâmica de interação em uma rede de energia elétrica em casos de falhas no sistema pode ser simplificado, modelado e analisado segundo um processo de conversão irreversível.

Em alguns tipos de problemas, o tempo é um fator importante a ser levado em consideração na convergência de um processo de conversão irreversível e alguns trabalhos o levam em consideração. Por exemplo, dado um processo que modela a disseminação de uma propriedade desejável, então, pode ser necessário que a propriedade atinja determinados itens do conjunto dentro de um determinado tempo. Assim, em [5], é realizado um estudo de forma mais generalista dos processos de conversão irreversível sujeitos a restrições de tempo de convergência. Pode-se notar, assim, que processos de conversão irreversível são estudados por meio diferentes modelagens do problema.

Este trabalho analisa a possibilidade de utilização de algoritmos genéticos na resolução do problema de encontrar o número mínimo de itens de um conjunto, que necessitam possuir a característica no inicio do processo, para que um processo de conversão irreversível seja convergente. Mostramos aqui, de forma geral, que algoritmos genéticos podem ser utilizados na resolução de problemas do tipo para obtenção de soluções em tempo hábil. Uma vez que o problema de decisão associado foi demonstrado ser NP-completo [3], [6], o problema considerado neste artigo pode ser considerado NP-difícil. Assim, encontrar técnicas capazes de gerar boas soluções e até mesmo soluções ótimas para o problema em instâncias de teste relativamente grandes são altamente desejadas. Este trabalho, porém, não tem como objetivo a busca pelos melhores parâmetros ou operadores tais como operadores e parâmetros de seleção, recombinação, mutação, dentre outros, a serem utilizados nos algoritmos genéticos para resolução de problemas com processos de conversão irreversível. Portanto, alguns parâmetros deste trabalho foram definidos de forma empírica durante a modelagem e implementação do algoritmo genético. O objetivo era apenas criar um algoritmo que fosse capaz de encontrar uma solução ótima ou próxima do ótimo em um tempo computacional viável.

II. Fundamentação Teórica

A. *Processo de Conversão Irreversível*

Com base na definição formal dada em [5], temos G como um grafo finito, simples e não-direcionado, $V(G)$ como o conjunto de vértices e $E(G)$ o conjunto de arestas de G. Definimos a ordem $n(G)$ de G como a cardinalidade de $V(G)$, a vizinhança $N_G(u)$ do vértice u de G como o conjunto $\{v \in V(G) : uv \in E(G)\}$ e o grau $d_G(u)$ de um vértice u em G como a cardinalidade de $N_G(u)$. Representamos o conjunto dos inteiros não negativos por $\mathbb{Z}_+$ e o conjunto dos inteiros positivos por $\mathbb{Z}_+^*$.

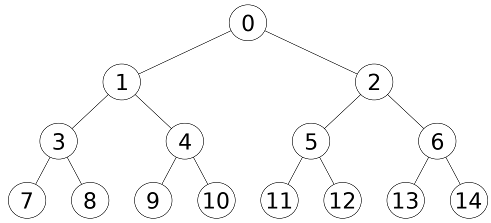

Figura 1. Enumeração dos vértices da árvore binária com $n(G) = 15$.

O processo de conversão irreversível será analisado aqui como uma sequência infinita (Equação (1)) de rótulos binários dos vértices de um grafo (Equação (2)) sendo que, sua evolução será guiada por uma função limiar (Equação (3)) que representa a sensibilidade de cada vértice. De modo análogo ao trabalho realizado em [6], o processo será analisado aqui sem levar em consideração possíveis restrições de tempo inerentes ao problema. Assim, a definição formal do processo, neste estudo, não conta com uma restrição temporal como em [5], além disso, a função limiar considerada aqui é independente do tempo.

$$\mathcal{C} = (c_0, c_1, c_2 \ldots) = (c_t)_{t \in \mathbb{Z}_+} \tag{1}$$

$$c_t : V(G) \to \{0, 1\} \tag{2}$$

$$f : V(G) \to \mathbb{Z}_+ \tag{3}$$

O processo iterativo de conversão irreversível em um grafo se dá de modo que, dado um grafo G, a rotulação binária inicial c_0 dos vértices de G e uma função limiar f, para todo $t \in \mathbb{Z}_+^*$, o t-ésimo rótulo c_t é tal que para todo vértice u de G, temos $c_t(u) = 1$ se e somente se $c_{t-1}(u) = 1$ ou se o número de vizinhos de u com $c_{t-1}(v) = 1$, para $v \in N_G(u)$, é maior ou igual ao valor da função limiar $f(u)$ para o vértice u.

Na Figura 1 podemos observar o grafo G definido como uma árvore binária com $n(G) = 15$. Já na Figura 2, é possível notar o instante de tempo t em que cada vértices de G é convertido com $f : V(G) \to \{2\}$, $c_0 = 1$ para o conjunto de vértices pretos $\{7, 8, 9, 10, 11, 12, 13, 14\}$ e $c_0 = 0$ para o conjunto dos demais vértices. Analisando o processo iterativo de conversão irreversível teremos $c_1 = 1$ para o conjunto $V(G) - \{0, 1, 2\}$ de vértices em $t = 1$, $c_2 = 1$ para o conjunto $V(G) - \{0\}$ de vértices em $t = 2$ e $c_i = 1$ para o conjunto $V(G)$ de vértices em $t = i$ com $i \geq 3$. Assim, pode-se observar que o processo se estabiliza a partir de $t = 3$ com todos os vértices de G convertidos.

Para a Figura 3 temos o mesmo grafo G da Figura 1 sujeito à mesma função limiar $f : V(G) \to \{2\}$ e com $c_0 = 1$ para o conjunto de vértices pretos $\{1, 5, 9, 10, 13, 14\}$ e $c_0 = 0$ para o conjunto dos demais vértices. Analisando o processo de conversão irreversível de forma iterativa com o novo c_0 dado na Figura 3, pode-se observar que o processo

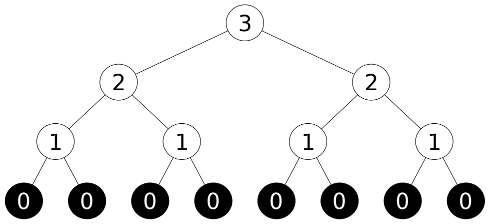

Figura 2. Tempo de conversão dos vértices, inciando o processo com conjunto convergente mínimo para $f : V(G) \to \{2\}$.

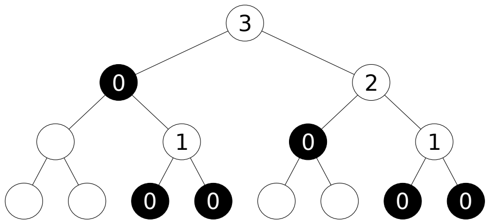

Figura 3. Tempo de conversão dos vértices, inciando o processo com conjunto não convergente para $f : V(G) \to \{2\}$.

também se estabiliza a partir de $t = 3$ assim como ocorre na Figura 2. Entretanto, na Figura 3, o processo se estabiliza sem que o conjunto de vértices $\{3, 7, 8, 11, 12\}$ do grafo G seja convertido.

B. Conjuntos Convergentes

Um processo de conversão irreversível é considerado convergente se, visto como uma sequência (Equação (1)), existir um $t_0 \in \mathbb{Z}_+$ tal que $c_{t_0}(u) = 1, \forall u \in V(G)$. Dada uma rotulação binária inicial c_0 dos vértices de G que permita a convergência do processo de conversão irreversível, um conjunto convergente é definido como o conjunto $\{u \in V(G) : c_0(u) = 1\} = c_0^{-1}(1)$.

Um conjunto convergente conhecido para o processo, dado qualquer grafo G e qualquer função limiar f, é o próprio conjunto $V(G)$, pois se $c_0(u) = 1, \forall u \in V(G)$, então o processo de conversão irreversível converge com $t = 0$. Portanto, sabemos que existe pelo menos um conjunto convergente dado qualquer grafo G e qualquer função limiar f. Porém, sabemos que outros conjuntos convergentes de menor cardinalidade podem existir.

Dado um grafo G e uma função limiar $f : V(G) \to \{k\}$, o problema de decisão associado ao problema de encontrar o menor conjunto convergente foi provado ser NP-completo para $k \geq 3$ [3] e posteriormente para $k \geq 2$ [6]. Por ser $k = 2$ o menor valor de k para o qual o problema pode ser dito NP-difícil, este trabalho, é focado na análise da capacidade de algoritmos genéticos para encontrar conjuntos convergentes mínimos em processos que possuem função limiar do tipo $f : V(G) \to \{2\}$.

Na Figura 4 pode-se observar um conjunto convergente formado pelo conjunto de vértices pretos. Pois, ao iniciar o processo de conversão irreversível com $c_0 = 1$ para tais vértices e $c_0 = 0$ para os demais vértices, o processo atinge a estabilidade em $t = 9$ com todos os vértices convertidos. Neste caso, o conjunto convergente é o de menor cardinalidade possível para o grafo em questão, portanto, pode-se dizer que o

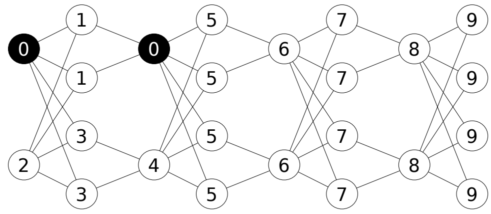

Figura 4. Tempo de conversão dos vértices, inciando o processo com conjunto convergente mínimo para $f : V(G) \to \{2\}$.

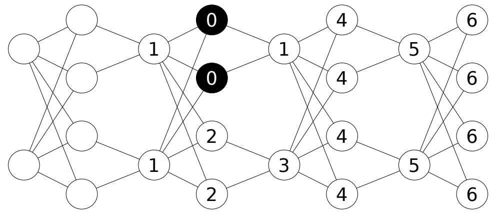

Figura 5. Tempo de conversão dos vértices, inciando o processo com conjunto não convergente para $f : V(G) \to \{2\}$.

conjunto de vértices pretos é um conjunto convergente mínimo para o processo de conversão irreversível dado o grafo da Figura 4, sujeito a uma função limiar do tipo $f : V(G) \to \{2\}$. Já na Figura 5 podemos observar um conjunto não convergente formado pelos vértices pretos. Este conjunto é não convergente, pois, ao iniciar o processo com $c_0 = 1$ para os vértices pretos e $c_0 = 0$ para os demais vértices, o processo atinge a estabilidade em $t = 6$ sem ter todos os vértices convertidos no processo.

C. Algoritmos Genéticos

Algoritmos genéticos são um tipo específico de algoritmos evolutivos, idealizados por Holland [7] e popularizados por Goldberd [8]. Usando técnicas baseadas em seleção natural, mutações, cruzamentos e combinações de genes, estes algoritmos avaliam as soluções candidatas a cada iteração, mantêm as mais bem avaliadas e criam novas possíveis soluções para o problema a fim de obter a convergência na busca de uma solução ótima. O processo de desenvolvimento de um algoritmo genético envolve a escolha de diversos detalhes de implementação como a representação do indivíduo, o modo de mutação, de recombinação, de seleção de indivíduos, dentre outros.

1) Representação: Em um algoritmo genético, cada indivíduo representa uma solução candidata para o problema. Existem diversas formas de representar uma solução candidata como um indivíduo em um algoritmo genético. Indivíduos podem ser representados como um vetor de números inteiros, de números reais, de caracteres, de bits, dentre outros. A escolha da forma de representação depende diretamente do problema que está sendo resolvido e é fundamental para o desenvolvimento de um algoritmo genético eficiente [9].

Para representação das soluções candidatas do problema de encontrar um conjunto mínimo convergente, foi proposta a representação binária das imagens de c_0 candidatas à solução, que consiste em vetores de bits de tamanho $n(G)$ onde cada item do vetor, ou gene, representa um vértice do grafo G. Nesta representação, um item do vetor com valor 1 indica que o vértice representado por aquele item está inicialmente convertido e um item do vetor com valor 0 indica que o vértice não está convertido.

A representação binária dos indivíduos foi escolhida, neste caso, devido à própria natureza binária do problema. Um conjunto convergente é um conjunto de vértices v que possuem $c_0(v) = 1$. Assim sendo, encontrar o conjunto convergente mínimo, pode ser visto como encontrar a imagem de c_0 com a menor quantidade de bits 1, capaz de fazer com que o processo seja convergente.

2) População: Neste trabalho, foram utilizados valores variados para o tamanho da população uma vez que observou-se empiricamente que valores fixos não traziam bons resultados para determinados tamanhos de grafos. Deste modo, o tamanho da população, nos experimentos, foi definido como igual à ordem do grafo para o qual se busca o conjunto convergente mínimo. Um indivíduo que representa o conjunto convergente trivial, com todos os vértices convertidos no instante inicial, foi sempre adicionado à população na primeira geração do algoritmo genético. Por fim, os demais indivíduos da população foram gerados aleatoriamente.

3) Seleção: Diferentes métodos de seleção de indivíduos são normalmente empregados em algoritmos genéticos. Como exemplos, podemos citar: seleção por roleta, seleção por *ranking* e seleção por torneio. Neste trabalho, a seleção dos indivíduos utilizados na recombinação foi realizada por meio de seleção por torneio. Os indivíduos filhos gerados na recombinação de uma geração são os indivíduos da geração subsequente. Além disso, foi utilizado elitismo entre as gerações, assim, o melhor indivíduo da população em uma geração foi sempre mantido na população da geração seguinte.

Na seleção de indivíduos por meio de torneio, seleciona-se aleatoriamente um número $\mathcal{T}$ de indivíduos da população, e dentre estes, é escolhido o melhor indivíduo. O processo é então repetido tantas vezes quanto forem necessárias realizar a seleção de um indivíduo da população [10]. Na implementação do algoritmo de seleção por torneio, deste trabalho, foi utilizado o parâmetro $\mathcal{T} = 5$. Os indivíduos filhos foram colocados na geração seguinte e os pais descartados. Uma exceção é o melhor indivíduo da população, que é sempre mantido na população da geração seguinte.

A seleção por torneio não requer o conhecimento global dos valores de aptidão. Em vez disso, a seleção se baseia somente na comparação entre os $\mathcal{T}$ indivíduos selecionados aleatoriamente da população. Esta abordagem mostra-se interessante para populações de tamanhos elevados, para as quais, calcular o valor de aptidão de todos os indivíduos da população pode ser demorado e ter um custo computacional elevado [9].

4) Recombinação: O processo de recombinação, em algoritmos genéticos, geralmente inicia com dois indivíduos pais e gera dois indivíduos filhos. Em alguns casos, somente um dos indivíduos filhos é considerado. Há três formas de recombinação, geralmente utilizadas, com representações binárias de indivíduos. Cruzamento de 1-ponto, cruzamento de n-pontos, e cruzamento uniforme. As duas primeiras dividem os pais em um número de seções de genes contíguos e remonta estas seções para produzir os indivíduos filhos. Já a terceira forma cria um indivíduo filho, tratando cada gene de forma independente, fazendo uma escolha aleatória de qual indivíduo pai o gene em questão será herdado [9].

Dados: $ind \leftarrow$ Indivíduo a ser mutado
Resultado: $ind \leftarrow$ Indivíduo após operação de mutação
início
    para $i \leftarrow 1$ *até* 3 **faça**
        se *valor aleatório do intervalo* $[0, 1) < 0,5$ **então**
            altera o valor de um gene aleatório de ind
        fim
    fim
    retorna ind
fim

Figura 6. Algoritmo que implementa o operador de mutação.

Pelas características do problema de encontrar um conjunto mínimo convergente, tais como a relação aleatória entre os vértices do grafo e a desconhecida dependência entre eles, decidiu-se por realizar o processo de recombinação por meio do processo de cruzamento uniforme, com a característica particular da implementação de se considerar apenas um filho.

5) Mutação: Ao operador que aplica alguma modificação aleatória na representação do indivíduo é dado o nome genérico de mutação [9]. Em representações binárias, a mutação de um indivíduo é normalmente feita de acordo com uma probabilidade de mutação p_m. Assim, realizar a mutação de um indivíduo com probabilidade p_m consiste em, alterar o valor de cada gene do mesmo com probabilidade p_m. Geralmente, a probabilidade da ocorrência de mutação em um indivíduo é muito pequena, em analogia ao mundo dos organismos vivos onde mutações raramente ocorrem [11].

Utilizar um valor fixo de p_m em um algoritmo genético que busca por conjuntos convergentes mínimos em grafos de tamanhos variáveis pode ser um problema. Isto se dá devido ao fato de que, para um grafo G_a com $n(G_a)$ muito grande, um valor de p_m que se mostrou ideal para encontrar um conjunto convergente mínimo em um grafo G_b com $n(G_b)$ pequeno pode acabar por modificar excessivamente as possíveis soluções para o grafo G_a. Por exemplo, o valor de $p_m = 0,015$ pode ser ideal para um grafo G_{100} com $n(G_{100}) = 100$, pois, com este valor de probabilidade de mutação, somente 1,5 genes de cada possível solução serão alterados em média. Entretanto, para um grafo G_{1000} com $n(G_{1000}) = 1000$, este valor de p_m acaba alterando 15 genes em média em cada possível solução. Estas alterações excessiva nas soluções candidatas do problema, acabam por fazer com que o algoritmo genético não convirja para uma solução desejada.

Para contornar o problema de utilizar um operador de mutação com valor fixo de p_m, foi implementado um operador de mutação que altera, em média, o valor de aproximadamente 1,5 vértices do indivíduo e não altera mais de 3 vértices em um único indivíduo. A implementação realizada pode ser analisada na Figura 3. A mesma consiste em o selecionar aleatoriamente 3 genes do indivíduo e alterar cada um deles com probabilidade de 0,5.

6) Função de Aptidão: O papel da função aptidão é representar os requisitos aos quais a população deve se adaptar. Tecnicamente, a função aptidão é uma função ou procedimento que atribui uma medida de qualidade às soluções candidatas [9]. Com o intuito de encontrar o conjunto convergente mínimo, definimos ϕ, o valor da função de aptidão como: $|c_0^{-1}(0)| + n(G)$, ou seja, a soma da quantidade inicial de 0s na imagem de c_0 com a cardinalidade dos vértices de G , caso o processo de conversão irreversível seja convergente. Por outro lado, caso o processo não seja convergente, o valor da função de aptidão será definido como: $|c_0^{-1}(0)|-|c_i^{-1}(0)|$ onde $i = min\{x \in \mathbb{Z}_+^* : c_x = c_{x+1}\}$, ou seja, a diferença entre a quantidade inicial de 0s na imagem de c_0 e a quantidade de 0s na imagem de c_i sendo i o tempo a partir do qual o processo se estabiliza.

Analisando as definições da função de aptidão, podemos observar que a mesma cresce proporcionalmente à quantidade de 0s na imagem de c_0. Além disso, a mesma recebe uma bonificação caso o processo seja convergente ou uma penalização caso o processo não seja convergente. Com esta definição, qualquer solução convergente possuirá um valor de aptidão maior do que o de uma solução não convergente e as soluções com maior quantidade de 0s na imagem de c_0 receberão um maior valor de aptidão quando comparadas a uma solução com menor quantidade de 0s na imagem de c_0.

III. Metodologia Experimental

A. Instâncias de Teste

Para testar a eficiência do algoritmo genético implementado na busca por conjuntos convergentes mínimos, foram criadas instâncias de teste com topologias para as quais se conhece os conjuntos convergentes mínimos. Portanto, para estas instâncias, são conhecidos os valores máximos desejados da função de aptidão.

Dado um grafo direcionado D, o grafo não-direcionado G subjacente a D é o grafo com conjunto de vértices $V(G) = V(D)$ e conjunto de arestas $E(G) = \{\{u, v\}|(u, v) \in E(D) \vee (v, u) \in E(D)$. Uma árvore binária $T = (V, E)$ é um grafo direcionado, cujo grafo subjacente é uma árvore, com um vértice r, denominado a raiz de T, tal que o grau de entrada dos elementos de $V(G)-\{r\}$ é exatamente 1, o grau de entrada de r é zero, e o grau de saída de todos os vértices é no máximo 2. O nível de um nó v em uma árvore binária T com raiz r é a distância de r a v em T. Dizemos que um vértice v é pai de um vértice w se (v, w) é um arco de $E(T)$. Neste caso, dizemos também que w é filho de v. Uma folha é um vértice com grau de saída zero. Um árvore binária é cheia se todas as folhas se encontram no mesmo nível e todos os vértices não folha possuem grau de saída 2.

Neste trabalho, as árvores binárias consideradas serão árvores binárias cheias. Como o problema de conversão apresentado considera apenas grafos não direcionados, utilizaremos também o termo árvores binárias para o subgrafo subjacente a árvores binárias.

O primeiro tipo de instâncias de teste com topologia conhecida consiste em grafos na forma de árvore binária conforme definida anteriormente, forma esta que pode ser observada na Figura 1. Árvores binárias são particularmente interessantes de serem estudadas no caso da função limiar definida como $f : V(G) \to \{2\}$ pois, na solução ótima, para que um vértice converta é necessário que seus vértices filhos estejam convertidos. Deste modo, a convergência do processo de conversão irreversível depende de que todos os vértices folhas da árvore binária iniciem o processo já convertidos, como pode ser observado na Figura 2. Analisando a Figura 3 vê-se que, vértices que são folhas da árvore binária jamais serão convertidos por possuírem um único vizinho.

Enquanto o primeiro tipo de instâncias de teste possui um conjunto convergente mínimo de tamanho proporcional à ordem do grafo, o segundo tipo de instâncias de teste foi projetado para possuir um conjunto convergente mínimo de tamanho fixo igual a 2, independente da ordem do grafo. A topologia projetada para o segundo tipo de instâncias de teste pode ser observada na Figura 4, na qual nota-se que o padrão que se repete a cada 6 vértices da esquerda para direita pode se estender indefinidamente, formando assim, grafos de ordem elevada sempre com conjunto convergente mínimo de tamanho 2. O conjunto convergente mínimo de tamanho 2, para estas instâncias, estará sempre contido no subconjunto dos 6 vértices mais a esquerda do grafo. Isto acontece pois, a convergência dos vértices se propaga passando sempre de um conjunto de 6 vértices mais a esquerda para um conjunto de 6 vértices mais a direita e nunca de um conjunto mais a direita para um conjunto mais a esquerda. Analisando a Figura 4, podemos observar um conjunto convergente mínimo formado pelo conjunto de vértices pretos e o instante de conversão de cada vértice do grafo durante o processo de conversão irreversível. Já na Figura 5, temos um conjunto não convergente formado pelos vértices pretos. Nesta figura, podemos observar o tempo de convergência de cada vértice no processo e notar que o processo de conversão passa de um subconjunto de 6 vértices mais a esquerda para os subconjuntos mais a direita, mas não o contrário.

B. Ambiente de Execução

Para implementação do algoritmo genético foi utilizada a linguagem JAVA. Todos os testes foram realizados em um computador com processador Intel Ivy Bridge Core i5-3570K, 16GB de memória RAM e Sistema Operacional Ubuntu 15.04.

C. Resultados das Simulações

Por se tratar de um problema NP-difícil, encontrar uma solução ótima por meio de técnicas determinísticas torna-se inviável para grafos de ordem relativamente pequenas. Por exemplo, para um grafo de ordem 50, encontrar o conjunto convergente mínimo por meio do método de força bruta, que consiste em gerar todas as possíveis soluções do problema e testá-las exaustivamente para saber qual satisfaz o enunciado do problema, é inviável analisando-se o tempo que se levaria para analisar todas as 2^{50} possíveis soluções com o intuito de encontrar a solução ótima.

Na Tabela I temos os dados obtidos para 10 execuções do algoritmo genético para cada valor de ordem da árvore binária. Em todos os casos, o algoritmo genético foi executado até encontrar uma solução ótima para o problema, uma vez que o valor de aptidão para o mesmo é conhecido. Pela análise dos dados, podemos notar que o algoritmo genético convergiu em tempo hábil para grafos de tamanhos inconcebíveis para algoritmos determinísticos. Por exemplo, para um grafo de ordem 1023, o algoritmo genético convergiu com tempo médio de aproximadamente 13721 milissegundos, já para um grafo

Tabela I. DADOS DAS EXECUÇÕES EM ÁRVORE BINÁRIA

Ordem do Grafo	Tempo Médio para Convergência (ms)	Quantidade Média de Cálculo de ϕ
63	23,8	2766,2
127	89,7	9854,2
255	440,9	24512,0
511	2655,7	71299,0
1023	13721,7	177420,2
2047	67666,6	428228,8
4095	351296,3	1090642,6

Tabela II. DADOS DAS EXECUÇÕES EM TOPOLOGIA PROJETADA

Ordem do Grafo	Tempo Médio para Convergência (ms)	Quantidade Média de Cálculo de ϕ
60	19,6	1127,9
126	71,3	5313,5
252	253,6	8334,2
510	1440,7	23007,8
1020	8005,6	61956,2
2046	43382,5	170554,0
4092	246260,0	471693,3

de ordem 4095, a convergência se deu em um tempo médio de aproximadamente 351296 milissegundos.

A Tabela II exibe resultados semelhantes aos exibidos na Tabela I, entretanto os dados se referem aos resultados obtidos para 10 execuções do algoritmo genético para cada valor de ordem do grafo na topologia projetada. Em todos os casos, o algoritmo genético foi executado até encontrar uma solução ótima para o problema, assim como aconteceu para as árvores binárias.

Com relação à quantidade média de cálculos da função de aptidão para encontrar a solução ótima das instâncias propostas, pode-se observar pela análise dos dados das Tabelas I e II que a quantidade cresce polinomialmente com o aumento da quantidade de vértices do grafo. Tal comportamento pode ser observado na Figura 7 que exibe a quantidade de execuções realizadas da função de aptidão pela quantidade de vértices do grafo. Observa-se que a quantidade de cálculos é maior para topologia de árvore binária, mas para ambas as topologias a quantidade de cálculos cresce proporcionalmente à quantidade de vértices no grafo.

IV. CONCLUSÕES

Este trabalho pretende resolver o problema da conversão irreversível de grafos por meio de algoritmos genéticos. Estes problemas, considerados da classe dos NP-difíceis, buscam encontrar o número mínimo de vértices em um grafo que necessitam possuir determinada característica de modo que, após um certo período de tempo, todos os vértices do grafo também possuam tal característica.

O algoritmo genético implementado foi capaz de encontrar resultados ótimos e em um tempo computacional viável para o problema de encontrar conjuntos convergentes mínimos de processos de conversão irreversível em topologias conhecidas de grafos. Os resultados obtidos mostram que tais algoritmos possuem potencial para serem utilizados na resolução de problemas relacionados a processos de conversão irreversível.

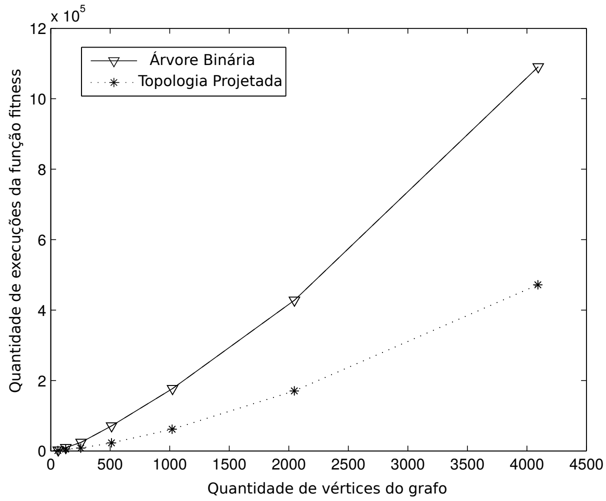

Figura 7. Quantidade média de cálculo da função de aptidão em 10 execuções do algoritmo genético para encontrar solução ótima por ordem do grafo.

Este foi um estudo inicial da capacidade de algoritmos genéticos em resolver a classe de problema proposto. Como trabalho futuro, sugere-se a realização de um estudo mais aprofundado de generalização. Será necessário verificar a capacidade dos algoritmos genéticos de encontrar soluções ótimas ou próximas do ótimo e em um tempo computacional viável para grafos com topologias desconhecidas. De mesmo modo, sugere-se um estudo dos parâmetros ideais a serem utilizados no algoritmo genético a fim de aumentar sua eficiência na resolução de tal problema.

REFERÊNCIAS

[1] J. L. Gersting, *Fundamentos Matemáticos para Ciência da Computação*, 5th ed. Rio de Janeiro: LTC, 2008.

[2] P. Flocchini, R. Královič, P. Ružička, A. Roncato, and N. Santoro, "On time versus size for monotone dynamic monopolies in regular topologies," *Journal of Discrete Algorithms*, vol. 1, pp. 129–150, 2003.

[3] P. A. Dreyer and F. S. Roberts, "Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion," *Discrete Applied Mathematics*, pp. 1615–1627, 2009.

[4] C. Asavathiratham, S. Roy, B. Lesieutre, and G. Verghese, "The Influence Model," *Control Systems, IEEE*, vol. 21, no. December 2001, pp. 52–64, 2001.

[5] D. Rautenbach, V. F. D. Santos, and P. M. Schäfer, "Irreversible conversion processes with deadlines," *Journal of Discrete Algorithms*, vol. 26, pp. 69–76, 2014.

[6] C. C. Centeno, M. C. Dourado, L. D. Penso, D. Rautenbach, and J. L. Szwarcfiter, "Irreversible conversion of graphs," *Theoretical Computer Science*, pp. 3693–3700, 2011.

[7] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA, USA: MIT Press, 1975.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[9] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Springer, 2008.

[10] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," *Foundations of genetic algorithms*, vol. 1, pp. 69–93, 1991.

[11] L. Rutkowski, *Computational Intelligence*. Springer, 2008.