

COMPLEXIDADE PARAMETRIZADA PARA PROBLEMAS EM GRAFOS E/OU

Uéverton dos Santos Souza
Universidade Federal Fluminense
usouza@ic.uff.br

Fábio Protti
Universidade Federal Fluminense
fabio@ic.uff.br

Maise Dantas da Silva
Universidade Federal Fluminense
maisedantas@id.uff.br

Resumo

Grafos E/Ou e Grafos X-de-Y são estruturas capazes de modelar muitos problemas (teóricos e práticos) nas mais diversas áreas de aplicação. Uma vez modelado um problema com uma dessas estruturas, geralmente sua solução poderá ser obtida através de um subgrafo (subgrafo-solução) deste grafo. Neste trabalho, são estudados os problemas Min-E/Ou e Min-X-de-Y, que consistem, respectivamente, em encontrar um subgrafo-solução de custo mínimo em grafos E/Ou e grafos X-de-Y. Como ambos são problemas NP-difíceis, este trabalho foca em abordagens parametrizadas para Min-E/Ou e Min-X-de-Y, obtendo alguns resultados quanto à tratabilidade parametrizada das versões Min-E/Ou(k, r), Min-X-de-Y(k) e Exato-X-de-Y(k) destes problemas.

Palavras-chave: Complexidade Parametrizada. Grafos E/Ou. Grafos X-de-Y

Abstract

And/Or Graphs and X-of-Y Graphs are structures capable of modeling many theoretical and practical problems in several application areas. Once a problem is modeled using one of these structures, the solution can usually be obtained via a subgraph (*solution subgraph*) of this graph. In this work, it is studied the Min-And/Or and Min-X-of-Y problems, consisting respectively of finding a solution subgraph of Minimum cost in And/Or and X-of-Y graphs. As both problems are NP-hard this paper focuses on a parameterized approach to Min-And/Or and Min-X-of-Y, where some results were obtained on the parameterized tractability of versions Min-And/Or(k, r), Min-X-of-Y(k) and Exactly-X-of-Y(k).

Keywords: Parameterized Complexity. And/Or Graphs. X-of-Y Graphs

1 Introdução

A teoria de complexidade parametrizada (Downey e Fellows 1999), proposta em 1995, surgiu como uma alternativa promissora para se trabalhar com problemas NP-difíceis, que passam a ter um conjunto K de parâmetros fixos adicionais à entrada. O interesse em parametrizar tais problemas se deve ao fato de que, em muitos casos, somente uma pequena faixa de valores de parâmetro é realmente importante na prática. Logo, a intratabilidade (aparente) desses problemas no caso geral é indevidamente pessimista. Analisando-se mais profundamente a estrutura da entrada (considerando-a com um conjunto de parâmetros adicionais), tenta-se limitar a explosão combinatória aparentemente inevitável na solução do problema.

Segundo a teoria de complexidade parametrizada (Flum e Grohe 2006) (Niedermeier 2006), dado um conjunto K de parâmetros (fixados com valor constante), um problema NP-difícil Π é dito tratável por parâmetro fixo (*FPT*) para K se, uma vez adicionado K à sua formulação, for possível desenvolver um algoritmo que solucione Π em tempo $f(K)n^\alpha$, onde n é o tamanho da entrada, $f(K)$ é uma função arbitrária (possivelmente exponencial) em K e α é uma constante independente de n e de K .

Além da classe *FPT*, Downey e Fellows definiram classes apropriadas de problemas parametrizados de acordo com seu nível de intratabilidade parametrizada. Essas classes são organizadas em uma hierarquia (*W-hierarquia*), e baseadas intuitivamente na complexidade dos circuitos necessários para se verificar a validade de uma solução, ou, alternativamente, na profundidade lógica natural do problema.

Muitas vezes, determinar o custo mínimo para a produção de uma empresa ou o lucro máximo que esta poderá obter, por exemplo, são problemas difíceis de serem resolvidos. No entanto, na prática, as empresas geralmente necessitam determinar apenas se é possível efetuar a produção com um determinado orçamento ou verificar se é possível cumprir uma determinada meta de venda/lucro. Sendo assim, é possível adicionar aos problemas de custo mínimo ou lucro máximo destas empresas parâmetros fixos adicionais, como um orçamento ou uma meta, dando origem às suas versões parametrizadas, que podem ser mais fáceis de serem solucionadas (menos complexas) e ao mesmo tempo satisfazer às necessidades requeridas. Perante estes fatos, observa-se que a Teoria da Complexidade Parametrizada modela perfeitamente esta realidade; muitos problemas considerados intratáveis (a menos que $P = NP$) sob o ponto de vista teórico, na prática são tratáveis por parâmetro fixo. Quanto à intratabilidade parametrizada, um problema provado W[1]-difícil tende a não possuir algoritmos muito mais eficientes para a sua solução do que examinar todas as combinações de k elementos.

Grafos E/Ou vem sendo utilizados em inteligência artificial desde o início da década de 1970, para representação de decomposição de problemas (Nilsson 1971)(Hansen e Zilberstein 2001). Desde então, ao longo do tempo, novas aplicações para estes grafos vem surgindo, como em: versionamento de software em engenharia de software (Corandi e Westfechtel 1998), geração de padrões de cortes em placas em pesquisa operacional (Morabito e Pureza 2010)(Arenales e Vianna 2006), representação de hipergrafos em teoria dos grafos (Gallo et al. 1993) e representação de *game trees* em teoria dos jogos (Schaeffer et al. 2001), entre outras. Grafos X-de-Y, por sua vez, são uma generalização de grafos E/Ou, definidos com base nos modelos para detecção de *deadlock* em sistemas distribuídos (Barbosa 2002).

Em geral, a solução de um problema modelado por grafos E/Ou ou grafos X-de-Y pode ser representada por um subgrafo (denotado por *subgrafo-solução*) deste grafo, que satisfaz um determinado conjunto de restrições. Os problemas Min-E/Ou e Min-X-de-Y são problemas NP-difíceis, que consistem basicamente em encontrar um subgrafo-solução de custo mínimo em grafos E/Ou e grafos X-de-Y, respectivamente. Motivou-nos o fato de que os problemas Min-E/Ou

e Min-X-de-Y, apesar de serem problemas naturais e possuírem várias e interessantes aplicações, não vinham sendo muito estudados na literatura especializada. As referências encontradas são poucas e, de uma forma geral, pouco recentes.

Este trabalho procura investigar variações dos problemas Min-E/Ou e Min-X-de-Y e, em particular, analisá-las do ponto de vista da complexidade parametrizada (algo que, tanto quanto sabemos, ainda não havia sido iniciado). Como ferramenta para prova matemática de tratabilidade parametrizada, será utilizado o método de *redução a um núcleo do problema*. A ideia principal deste método é reduzir, em tempo polinomial, uma instância I do problema a uma instância I' , tal que uma solução para I' possa ser apresentada como uma solução para I , e o tamanho de I' seja limitado por $f(K)$. Dessa forma, a instância I' constitui um núcleo do problema com cardinalidade constante, que pode ser exaustivamente analisada sem que o tempo total do algoritmo ultrapasse $f(K)n^\alpha$. Por outro lado, para a prova de intratabilidade parametrizada, será utilizada a técnica de *redução parametrizada* de problemas. Essa técnica é similar à utilizada na teoria de complexidade computacional clássica para prova de NP-dificuldade, com o diferencial de que a redução parametrizada deve ser computada em tempo $f(K)n^\alpha$, e o conjunto de parâmetros resultantes K' deve ser uma função do conjunto de parâmetros originais K .

O artigo está estruturado da seguinte forma: esta seção possui natureza introdutória, a seção 2 apresenta a definição de grafos E/Ou e grafos X-de-Y, a formulação dos problemas Min-E/Ou e Min-X-de-Y e uma aplicação de Min-E/Ou em geração de padrões de cortes bidimensionais em placas; na seção 3 é formalizada a versão Min-E/Ou(k,r) é provado que esta versão é tratável por parâmetro fixo; na seção 4 são formalizadas as versões Min-X-de-Y(k) e Exato-X-de-Y(k) e provado que ambas são intratáveis por parâmetro fixo. Por fim, a última seção apresenta as reflexões finais.

2 Grafos E/Ou e grafos X-de-Y

Um grafo E/Ou é um digrafo G tal que todo vértice $v \in V(G)$ possui um rótulo $f(v) \in \{E, Ou\}$. Neste grafo, os arcos representam relações de dependência entre os vértices: vértices do tipo E dependem estritamente de todos os seus vizinhos de saída (dependência conjuntiva), enquanto vértices do tipo Ou dependem apenas de um dos seus vizinhos de saída (dependência optativa).

Neste trabalho, O_{v_i} é utilizado para denotar o conjunto dos vizinhos de saída de v_i , consequentemente $|O_{v_i}|$ representa o número de vizinhos de saída de v_i . Também é utilizado n e m para denotar, respectivamente, o número de vértices e o número de arestas/arcos de um grafo/digrafo, e τ para denotar o peso de um vértice ou arco.

Na representação de grafos E/Ou, os vértices do tipo E são ilustrados com um anel entre seus arcos de saída. A Figura 1 ilustra um grafo E/Ou, onde s é um vértice do tipo E e t um vértice do tipo Ou . Na literatura, também são encontradas definições de grafos E/Ou onde os rótulos $\{E, Ou\}$ se encontram nos arcos (Gallo et al. 1993).

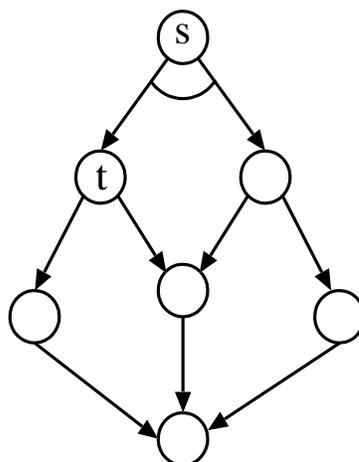


Figura 1: Exemplo de Grafo E/Ou .

Grafos X-de-Y são uma generalização de grafos E/Ou. Nestes grafos, todo vértice v_i possui um rótulo x_i-y_i , tal que v_i depende de x_i dentre os seus y_i vizinhos de saída, $0 \leq x_i \leq y_i$ e $y_i = |O_{v_i}|$.

Dado um grafo E/Ou, denotado por G , é possível construir um grafo X-de-Y, denotado por G' , da seguinte forma:

- para cada vértice $v_i \in V(G)$ do tipo E , cria-se em G' um vértice x_i-y_i , onde $x_i = y_i$, pois um vértice do tipo E depende de todos os seus y_i vizinhos de saída;

- para cada vértice $v_i \in V(G)$ do tipo Ou , cria-se em G' um vértice x_i-y_i onde $x_i = 1$, pois um vértice do tipo Ou depende de apenas um dentre seus y_i vizinhos de saída.

2.1 Min-E/OU e Min-X-de-Y

O problema Min-E/Ou consiste basicamente em, dado um grafo E/Ou ponderado com um nó fonte s , encontrar um subgrafo (subgrafo-solução) que satisfaça a um determinado conjunto de restrições. A seguir, é descrita a sua formulação:

Problema: Min-E/OU

Instância: Um inteiro positivo r , um grafo E/Ou $G=(V,E)$ acíclico, conexo e ponderado com um vértice fonte $s \in V$, onde cada vértice v possui um rótulo $f(v) \in \{E, Ou\}$, cada arco e possui um peso inteiro $\tau(e) \geq 1$, e todo vértice v tal que $f(v) = Ou$ possui no máximo r arcos de saída de mesmo peso.

Questão: Encontrar um subgrafo $H = (V', E')$ de G tal que o somatório dos pesos de seus arcos seja mínimo, e que satisfaça a:

1. $v \in V'$;
2. se $v \in V'$ e $f(v) = E$, então todos os arcos de saída de v devem pertencer a E' ;
3. se $v \in V'$ e $f(v) = Ou$, então exatamente um dos arcos de saída de v deve pertencer a E' .

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

De forma análoga a grafos E/Ou, existe o problema Min-X-de-Y para grafos X-de-Y. A formulação desse problema é definida a seguir:

Problema: Min-X-de-Y

Instância: Um grafo X-de-Y $G=(V,E)$ acíclico, conexo e ponderado com um vértice fonte $s \in V$, onde cada vértice v_i possui um rótulo x_i-y_i , $0 \leq x_i \leq y_i$ e $y_i = |O_{v_i}|$, e cada arco e possui um peso inteiro $\tau(e) \geq 1$.

Questão: Encontrar um subgrafo $H = (V',E')$ de G tal que o somatório dos pesos de seus arcos seja mínimo, e que satisfaça a:

1. $v \in V'$;
2. para todo $v_i \in V'$, x_i dentre seus y_i arcos de saída devem pertencer a E' .

A Figura 2 ilustra em (a) um grafo X-de-Y e em (b) um possível subgrafo-solução.

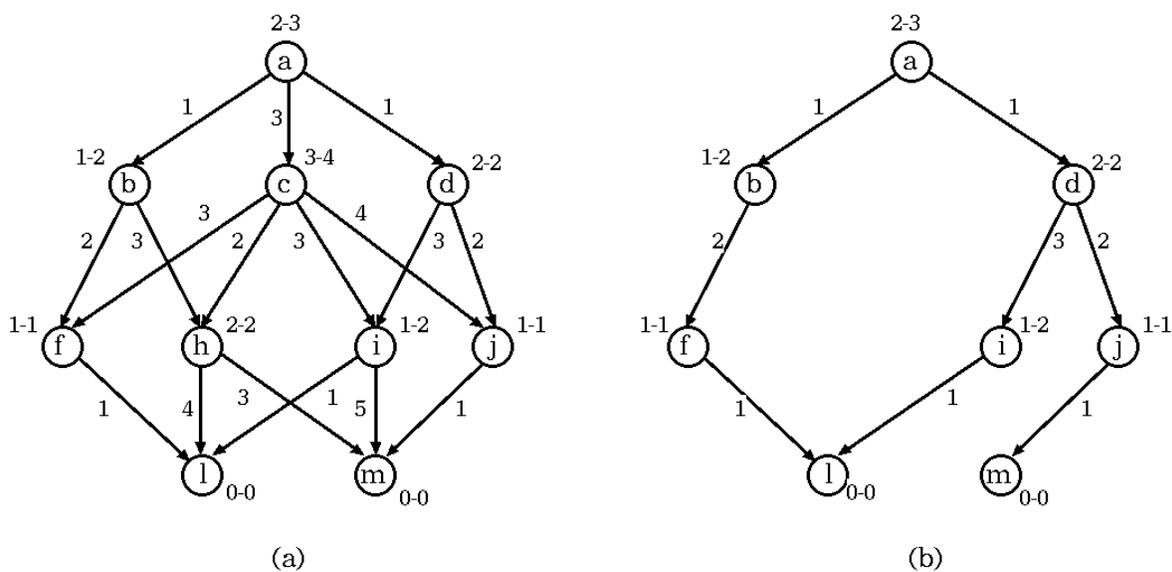


Figura 2: (a) Grafo X-de-Y. (b) Subgrafo-solução do grafo em (a).

Em 1974, Sahni provou, através de uma redução do problema 3-SAT, que o problema Min-E/Ou é NP-difícil (Sahni 1974). Nesta prova, foi exibida uma transformação de uma instância F do problema 3-SAT em um grafo E/Ou, onde todo vértice do tipo Ou possui no máximo 3 vizinhos de saída. Conclui-se então que o problema Min-E/Ou é NP-difícil, mesmo fixado $r = 3$. Consequentemente, Min-X-de-Y também é NP-difícil.

2.2 Min-E/Ou em Geração de Padrões de Cortes

Na área de Pesquisa Operacional, grafos E/Ou e grafos X-de-Y são utilizados para auxiliar na tomada de decisões e na análise de sistemas complexos do mundo real, tipicamente com o objetivo de melhorar ou otimizar a sua performance. Uma das aplicações de grafos E/Ou em Pesquisa Operacional é a sua utilização na geração de padrões de cortes bidimensionais em placas.

Este problema, que é NP-difícil (Morabito e Pureza 2010), aparece em diversos processos industriais, tais como: corte de bobinas de papel e alumínio, barras de aço, chapas metálicas e de madeira, placas de circuito impresso, caixas de papelão, rolos de tecido, entre outros. O modo como as peças são arranjadas na placa é chamado *padrão de corte*.

A Figura 3 ilustra dois exemplos de padrões de corte.

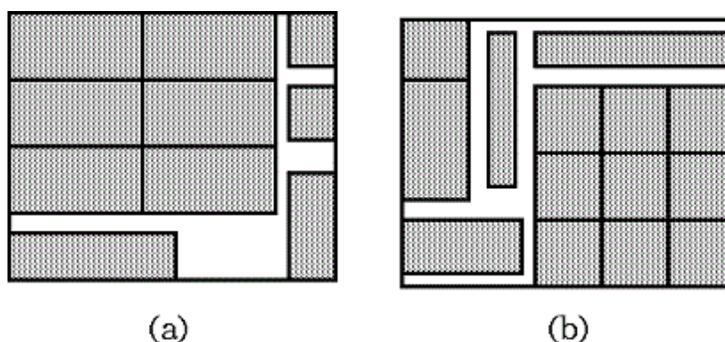


Figura 3: Ilustrações de padrões de cortes.

A seguir, uma descrição do problema de cortes em placas é apresentada:

Problema: *Geração de Padrões de Cortes Bidimensionais em Placas*

Instância: *Uma placa retangular P de dimensões (L, W) , onde L é o comprimento e W a largura; um conjunto de m peças retangulares p_i de dimensões (l_i, w_i) , tais que $l_i \leq L$ e $w_i \leq W$, $1 \leq i \leq m$; e um valor utilidade v_i associado a cada peça p_i .*

Questão: *Encontrar um padrão de corte da placa P que produza um conjunto de peças que maximize o valor de utilidade total, ou seja, maximize o somatório dos valores utilidade v_i das peças obtidas pelo corte. (Se o valor utilidade v_i for a área da peça p_i , então o objetivo é equivalente a minimizar a perda de material da peça P .)*

Dada uma placa P , um grafo E/Ou G pode ser definido para representar todos os possíveis padrões de corte de P . Os vértices do tipo *Ou* representam partes da placa e os vértices do tipo *E*, possíveis cortes na placa. Nesta representação, cada vizinho de saída de um vértice v do tipo *Ou* é um vértice do tipo *E*, que representa uma alternativa de corte na placa representada por v , e cada vértice u do tipo *E* possui dois vizinhos de saída (placas obtidas após o corte representado por u). O vértice fonte do grafo é um vértice do tipo *Ou*, que representa a placa P .

Neste grafo G , a busca por um padrão de corte pode ser feita examinando-se todas as possíveis alternativas de cortes a partir de um nó *Ou* e, para cada vértice *E*, as peças obtidas após o corte.

A partir desta representação por grafos E/Ou, algoritmos de busca e programação dinâmica

podem ser aplicados para encontrar uma solução ótima. Em 2006, Arenales (Arenales e Vianna 2006) propôs uma abordagem utilizando grafos E/Ou para geração de padrões de corte em placas defeituosas. Em seguida, em 2010, Morabito (Arenales e Vianna 2006) desenvolveu um método para geração de padrões de cortes restritos utilizando esses grafos.

Nesta representação, um padrão de corte pode ser observado como um subgrafo-solução de G , onde cada vértice Ou deste subgrafo possui um vizinho de saída (corte escolhido) e cada vértice E possui dois vizinhos de saída (placas obtidas após o corte), sendo as peças deste padrão de corte os vértices sumidouros desse subgrafo-solução.

A Figura 4 ilustra em (a) uma placa P , em (b) um conjunto de m peças que podem ser obtidas a partir de P , e em (c) um possível padrão de corte para P .

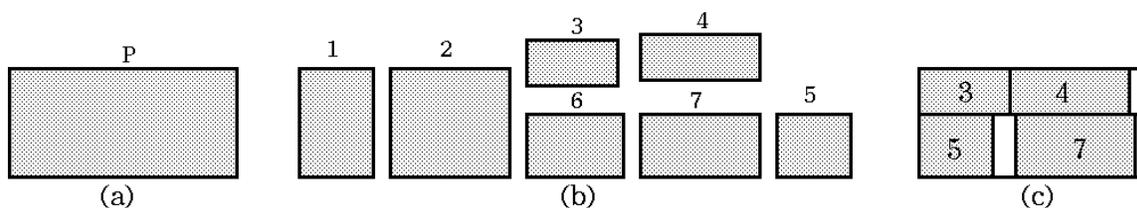


Figura 4: (a) Placa P . (b) Conjunto das m peças. (c) Padrão de corte de P .

A Figura 5 ilustra uma árvore E/Ou T que modela possíveis cortes na placa P , onde as folhas são possíveis peças obtidas pelos cortes. Nesta árvore, os vértices do tipo Ou indicam que cada filho é um possível corte e os vértices E indicam que seus filhos são placas obtidas a partir de um corte. As arestas em destaque formam a subárvore-solução do padrão de corte representado na Figura 4 (c).

Muitos trabalhos limitam seus estudos a padrões guilhotinados, isto é, padrões obtidos por uma sequência de cortes em placas retangulares que produzem sempre dois novos retângulos. Alguns trabalhos adicionam ainda uma restrição para o número máximo de vezes que um tipo de peça poderá ser cortado a partir de uma chapa. A existência de possíveis defeitos nas placas também é considerada por alguns autores. Para mais informações sobre o problema, as referências (Morabito e Pureza 2010) (Arenales e Vianna 2006) (Cui 2011) podem ser consultadas.

Como Min-X-de-Y é uma generalização de Min-E/Ou, além de sua aplicação direta a problemas de escalonamento e na área de sistemas distribuídos (Barbosa 2002), Min-X-de-Y é também capaz de modelar uma grande gama de problemas reais.

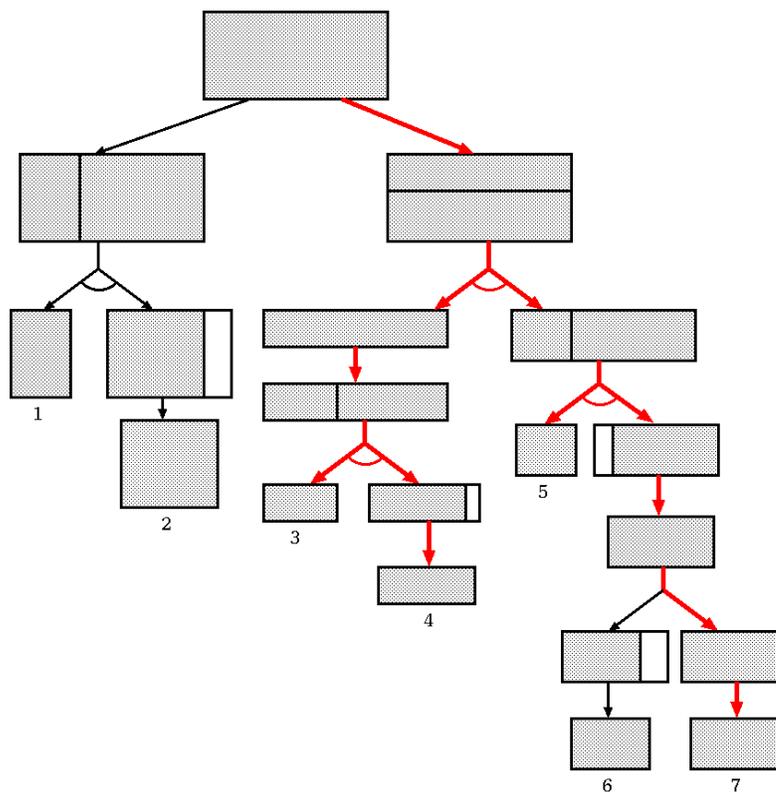


Figura 5: Árvore E/Ou de padrões de corte da Figura 3.

3 A versão Min-E/OU(k, r)

Conforme dito anteriormente, Min-E/Ou permanece intratável (a menos que $P = NP$), mesmo fixando um valor constante para r maior ou igual a 3. Sendo assim, é formulada uma variação parametrizada do problema Min-E/Ou, denotada Min-E/Ou(k, r), onde, além de fixado um valor constante para r , é fixado um valor inteiro positivo k para o custo do subgrafo-solução.

A seguir é apresentada essa nova formulação:

Problema: Min-E/OU(k, r)

Parâmetros: Dois inteiros positivos k e r .

Instância: Um grafo E/Ou $G=(V, E)$ acíclico, conexo e ponderado com um vértice fonte $s \in V(G)$, tal que cada vértice v possui um rótulo $f(v) \in \{E, Ou\}$, cada arco e possui um peso inteiro $\tau(e) \geq 1$, e cada vértice com rótulo *Ou* possui no máximo r arcos de saída de mesmo peso.

Questão: Determinar se existe um subgrafo-solução $H = (V', E')$ de G tal que o somatório dos pesos dos arcos de H não exceda k , atendendo a:

1. $s \in V'$;
2. se $v \in V'$ e $f(v) = E$, então todos os arcos de saída de v devem pertencer a E' ;
3. se $v \in V'$ e $f(v) = Ou$, então exatamente um dos arcos de saída de v deve pertencer a E' .

3.1 Tratabilidade Parametrizada para Min-E/OU(k, r)

Dada a versão parametrizada Min-E/Ou(k, r), deseja-se saber se esta versão é tratável à luz da teoria de complexidade parametrizada, ou seja, se é possível obter a solução para o problema em tempo $f(K)n^\alpha$, onde K é o conjunto formado pelos parâmetros k e r ($K = \{k, r\}$).

Para provar que o problema Min-E/Ou(k, r) é FPT, é utilizado o método de *redução a um núcleo do problema*. A ideia principal deste método é reduzir, em tempo polinomial, uma instância I do problema a uma instância I' , tal que uma solução para I' possa ser apresentada como uma solução para I , e o tamanho de I' seja limitado por $f(K)$. Dessa forma, a instância I' constitui um núcleo do problema com cardinalidade constante, que pode ser exaustivamente analisada sem que o tempo total do algoritmo ultrapasse $f(K)n^\alpha$.

Teorema 1 *Seja G uma instância do problema Min-E/Ou(k, r). É possível obter um núcleo de tamanho $O((kr)^{k+1})$ para este problema em tempo $O(m)$.*

Prova: Como prova, é apresentado um conjunto de regras de redução corretas e suficientes, que podem ser aplicadas à instância de entrada, em tempo $O(m)$, construindo um núcleo de tamanho menor ou igual a $(kr)^{k+1}$:

- Regra 1.** Para todo vértice v_i do tipo E , se $\sum_{v_j \in O_{v_i}} \tau(v_i, v_j) > k$, então remova v_i .
- Regra 2.** Para todo arco $e \in E(G)$, se $\tau(e) > k$, então remova-o.
- Regra 3.** Para todo vértice $v_i \neq s$, se o peso do caminho mínimo de s até v_i for maior que k , então remova-o.
- Regra 4.** Se algum vértice tornou-se inatingível a partir de s por caminhos orientados, então remova-o.
- Regra 5.** Se algum vértice inicialmente não-sumidouro tornou-se um sumidouro, então atribua peso $k + 1$ a todos os seus arcos de entrada.
- Regra 6.** Se um vértice do tipo E teve um sucessor removido, então atribua peso $k + 1$ a um de seus arcos de saída.

Seja G' o grafo resultante da aplicação das regras de redução.

A Regra 1 retira do grafo todos os vértices E tais que o somatório dos pesos de seus arcos de saída seja maior que k . Isto se deve ao fato de que estes vértices não podem fazer parte de um subgrafo-solução de custo máximo k . Assim, todo vértice do tipo E de G' , após a aplicação da Regra 1, terá no máximo k arcos de saída. Em seguida, a Regra 2 é aplicada, retirando do grafo todos os arcos de saída de peso maior do que k , pois estes arcos não poderão fazer parte de um subgrafo-solução de custo máximo k . Devido à aplicação da Regra 1, é fácil ver que a Regra 2 é aplicada apenas a arcos de saída de vértices do tipo Ou . Após a aplicação das Regras 1 e 2, G' terá no máximo k pesos distintos de arcos. Logo, todo vértice Ou terá no máximo kr arcos de saída. A Regra 3, por sua vez, retira do grafo todo vértice v_i tal que o custo de um caminho mínimo de s até v_i seja maior do que k , pois se v_i pertencesse a um subgrafo-solução, então um caminho de s até v_i deveria existir neste subgrafo e, conseqüentemente, este subgrafo teria peso maior que k .

Após a aplicação das Regras 1, 2 e 3, se algum vértice v_i tornou-se inatingível a partir de s , isto significa que todo caminho de s a v_i passava por um vértice ou arco que não poderia fazer parte de um subgrafo-solução de custo máximo k ; logo v_i deverá ser removido, o que é feito pela Regra 4. Se algum vértice v_i inicialmente não-sumidouro tornou-se um sumidouro após a aplicação das Regras 1, 2, 3 e 4, isto significa que nenhuma dependência de v_i pode fazer parte de um

subgrafo-solução de custo máximo k . Assim, v_i também não poderá fazer parte de um subgrafo-solução de G de custo máximo k . A Regra 5 atribui peso $k + 1$ a todo arco de entrada de v_i , impedindo assim que v_i faça parte de um subgrafo-solução de custo máximo k em H (este vértice não é removido, para evitar a criação de novos sumidouros). A Regra 6 efetua processo semelhante ao da Regra 5, para vértices do tipo E .

Conforme pode ser observado, as regras de redução alteraram e/ou removeram somente vértices e arcos que não poderiam fazer parte de um subgrafo-solução de custo máximo k . Desta forma, se o grafo G' possui um subgrafo-solução de custo máximo k , este também será um subgrafo-solução de custo máximo k para G .

A aplicação das regras de redução demanda tempo $O(m)$, o que pode ser verificado observando-se cada regra isoladamente. A Regra 1 pode ser executada em tempo $O(n + m)$, pois será necessário percorrer todos os vértices com rótulo E e seus arcos. A Regra 2 demandará tempo $O(m)$, uma vez que todos os arcos devem ser analisados. A Regra 3 poderá ser executada em tempo $O(n + m)$, a partir do algoritmo *DAG-Shortest-Paths* (Cormen et al. 2002), pois G é um grafo orientado acíclico. A Regra 4 também poderá ser executada em tempo $O(n + m)$ através de um percurso genérico. A Regra 5 demandará tempo $O(n + m)$ pois, no pior caso, para cada sumidouro, será necessário alterar o peso de todos os seus arcos de entrada. A Regra 6 demandará tempo $O(n)$. Como nenhuma regra tem complexidade superior a $O(n + m)$ e o grafo é conexo, as regras de redução podem ser aplicadas em tempo $O(m)$.

No grafo resultante G' , o maior caminho mínimo de s até um sumidouro possui peso máximo k , e cada vértice possui no máximo kr vizinhos de saída. Desta forma, G' terá número máximo de vértices se: (i) todos os vértices não-sumidouros possuírem grau de saída igual a kr ; (ii) não houver vértices com vizinhos de saída em comum; (iii) o custo de um caminho mínimo de s a qualquer sumidouro for k . Em outras palavras, G' terá número máximo de vértices quando for uma árvore (kr) -ária cheia de altura $k + 1$. Portanto, o número de vértices de G' é $O((kr)^{k+1})$.

Como (i) as regras de redução podem ser aplicadas em tempo polinomial $O(m)$, (ii) o tamanho do grafo resultante G' está em função dos parâmetros k e r , e (iii) um subgrafo-solução de custo máximo k' em G' equivale a um subgrafo-solução de custo máximo k' em G , concluímos que o subgrafo G' é um núcleo para o problema $\text{Min-E/Ou}(k, r)$. \square

Sabendo que o problema $\text{Min-E/Ou}(k, r)$ possui um núcleo de tamanho $f(kr)$, cabe observar o Corolário 2.

Corolário 2 $\text{Min-E/OU}(k, r) \in FPT$.

Prova: O resultado segue diretamente do Teorema 1, pois, uma vez obtido em tempo $O(m)$ um núcleo de tamanho $O((kr)^{k+1})$, este poderá ser exaustivamente analisado sem que o tempo total do algoritmo ultrapasse $f(K)n^\alpha$. \square

3 As versões Exato-X-de-Y(k) e Min-X-de-Y(k)

Nesta seção, é fixado um parâmetro k para o custo do subgrafo-solução, dando origem a duas versão parametrizadas do problema Min-X-de-Y : $\text{Min-X-de-Y}(k)$ e $\text{Exato-X-de-Y}(k)$. As formulações destas versões são apresentadas a seguir:

Problema: *Min-X-de-Y(k)*

Parâmetro: *Um inteiro positivo k.*

Instância: *Um grafo X-de-Y $G=(V, E)$ acíclico, conexo e ponderado com um vértice fonte $s \in V(G)$, onde cada vértice v_i possui um rótulo x_i - y_i , $0 \leq x_i \leq y_i$ e $y_i = |O_{v_i}|$, e cada arco e possui um peso $\tau(e) \geq 1$.*

Questão: *Determinar se existe um subgrafo-solução $H = (V', E')$ de G tal que o somatório dos pesos dos arcos de H não exceda k , atendendo a:*

1. $s \in V'$;
2. para todo $v_i \in V'$, x_i dentre seus y_i arcos de saída devem pertencer a E' .

Problema: *Exato-X-de-Y(k)*

Parâmetro: *Um inteiro positivo k.*

Instância: *Um grafo X-de-Y $G=(V, E)$ acíclico, conexo e ponderado com um vértice fonte $s \in V(G)$, onde cada vértice v_i possui um rótulo x_i - y_i , $0 \leq x_i \leq y_i$ e $y_i = |O_{v_i}|$, e cada arco e possui um peso $\tau(e) \geq 1$.*

Questão: *Determinar se existe um subgrafo-solução $H = (V', E')$ de G tal que o somatório dos pesos dos arcos de H seja exatamente k , atendendo a:*

1. $s \in V'$;
2. para todo $v_i \in V'$, x_i dentre seus y_i arcos de saída devem pertencer a E' .

3.1 Intratabilidade Parametrizada para Exato-X-de-Y(k) e Min-X-de-Y(k)

Dentre as classes pertencentes à W-hierarquia, a classe W[1] é a classe atualmente mais estudada. Diversos problemas já foram provados como W[1]-completos e W[1]-difíceis. Um problema W[1]-difícil pode ser solucionado em tempo $f(K)n^\alpha$ se e somente se $P = NP$. Um exemplo de problema W[1]-completo é o problema Clique(k) (Flum e Grohe 2006), onde é fixado um parâmetro k para o tamanho da clique.

O resultado a seguir diz que o problema Clique(k) pode ser transformado parametricamente no problema Exato-X-de-Y(k). Uma transformação paramétrica é uma redução computada em tempo $f(K)n^\alpha$, onde o conjunto de parâmetros resultantes K' é uma função arbitrária do conjunto de parâmetros K .

Teorema 3 *Clique(k) é transformável parametricamente em Exato-X-de-Y(k).*

Prova: Dada uma instância G de Clique(k), é possível transformá-la em uma instância G' de Exato-X-de-Y(k') da seguinte forma:

Seja G' um grafo inicialmente vazio.

- Cria-se um vértice fonte s ;
- Cria-se um conjunto de vértices $\{u_1, u_2, \dots, u_n\}$, vizinhos de saída de s ;
- Para cada vértice u_i , criam-se dois vizinhos de saída: z_i e w_i ;
- Para cada vértice z_i , cria-se um arco (z_i, w_j) se e somente se $(v_j, v_i) \in E(G)$;

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

- Para cada vértice w_i , cria-se um vizinho de saída sumidouro t_i .
- Atribui-se peso 1 a todos os arcos criados.

A rotulação dos vértices será dada da seguinte maneira:

- s terá rótulo $k-n$, onde $n = |V(G)|$.
- Todo vértice u_i terá rótulo 2-2.
- Todo vértice w_i terá rótulo 1-1.
- Todo vértice t_i terá rótulo 0-0.
- Cada vértice z_i , tal que $d(v_i) \geq k - 1$, é rotulado com $(k - 1) - d(v_i)$.
- Cada vértice z_i , tal que $d(v_i) < k - 1$, é rotulado com $d(v_i) - d(v_i)$.

Considerando que a criação de cada vértice e arco demanda tempo $O(1)$, a construção acima poderá ser executada em tempo $O(n + m)$, uma vez que será necessário, para cada vértice de G , apenas percorrer todas as suas arestas. Logo, a transformação é feita em tempo $f(k)n^\alpha$.

A Figura 6 ilustra um grafo G e seu respectivo grafo X-de-Y G' , obtido através da transformação acima, onde $k = 3$ (os pesos dos arcos do grafo X-de-Y G' foram omitidos para simplificação da figura).

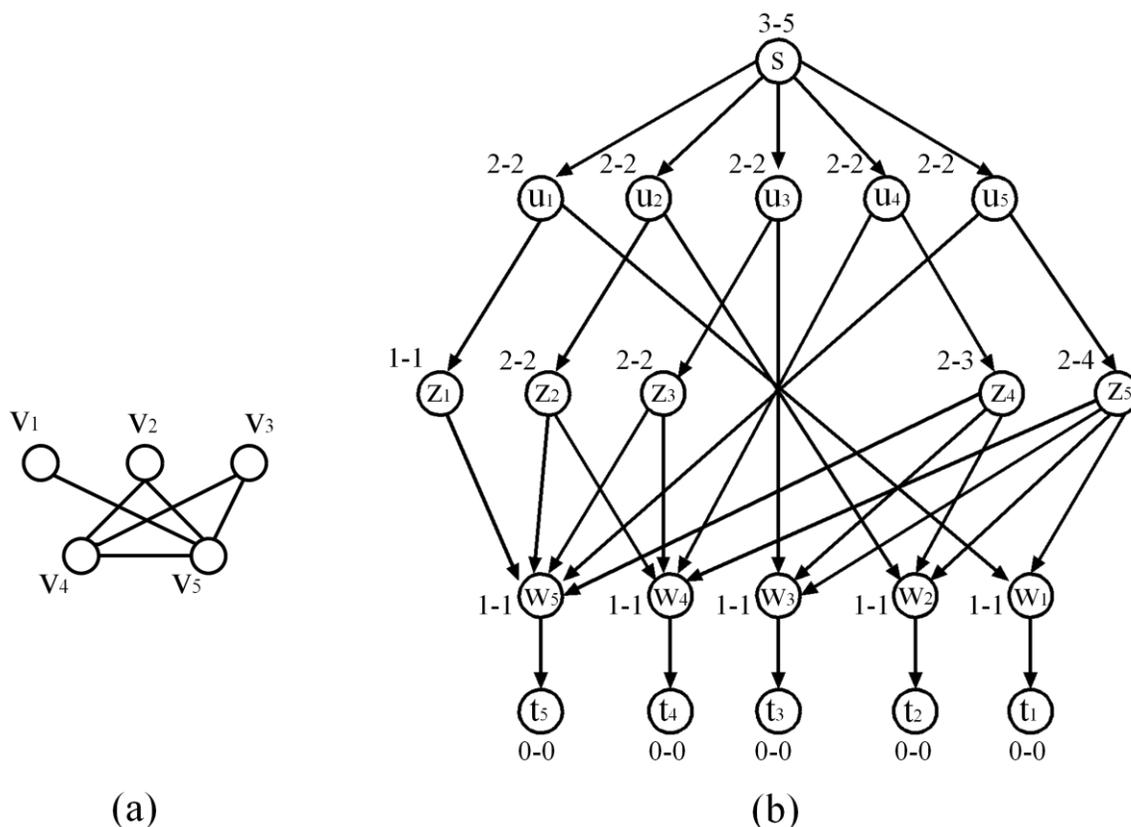


Figura 6: Transformação paramétrica de G em G' .

Como pode ser observado, o grafo G foi transformado em um grafo G' , onde $n' = 4n + 1$. Será demonstrado a seguir que G possui uma clique de tamanho k se e somente se G' possui um subgrafo-solução de custo $k' = k^2 + 3k$.

Se G tem um conjunto de vértices $\{v_1, v_2, \dots, v_k\}$ que formam uma clique C , um subgrafo-solução H de G' pode ser construído da seguinte forma: como s é um vértice de rótulo k - n , escolhe-se $\{u_1, u_2, \dots, u_k\}$ como os vizinhos de saída de s em H . Como cada um desses vértices tem rótulo 2-2, os vértices $\{w_1, w_2, \dots, w_k\}$ e $\{z_1, z_2, \dots, z_k\}$ também farão parte do subgrafo-solução H . Como cada vértice w_i é um vértice 1-1, os vértices $\{t_1, t_2, \dots, t_k\}$ também farão parte do subgrafo H . Neste ponto, H já tem custo $4k$, pois cada um de seus arcos tem peso 1. Como cada vértice z_i depende de $k - 1$ vizinhos de saída, é escolhido para z_i um vértice w_j se e somente se v_j pertence à clique C em G (os arcos de saída dos vértices z_1, z_2, \dots, z_k adicionam custo $k(k - 1)$ a H). Como, para cada vértice z_i , foram escolhidos somente vértices w_j que já se encontravam em H , o custo de H será $k(k - 1) + 4k = k^2 + 3k$.

Suponha, por contradição, que G' tem um subgrafo-solução H de custo $k' = k^2 + 3k$, e G não possui uma clique de tamanho k . Então, o subgrafo H terá pelo menos k vértices w_j , pois cada um dos k vértices u_i terá um vizinho de saída w_i . Como G não possui uma clique de tamanho k , então, pelo menos para um vértice z_i , foi escolhido um vértice w_j novo, pois caso contrário G teria tal clique. No entanto, se H possui $k + 1$ vértices w_j , em H temos que: (i) o vértice s possui k vizinhos, e portanto os arcos de saída de s adicionam custo k a H ; (ii) cada um dos k vértices u_i possui 2 vizinhos de saída, o que adiciona custo $2k$ a H ; (iii) cada um dos k vértices z_i possui $k - 1$ vizinhos de saída, adicionando portanto custo $k(k - 1)$ a H ; (iv) cada um dos $k + 1$ vértices w_j possui 1 vizinho de saída, o que adiciona custo $k + 1$ a H . Sendo assim, tem-se no total $\tau(H) = k + 2k + k(k - 1) + (k + 1) = k^2 + 3k + 1$ (uma contradição, pois foi assumido que H tem custo exatamente $k' = k^2 + 3k$).

Logo, G possui uma clique de tamanho k se e somente se G' possui um subgrafo-solução de custo exatamente k' . \square

Corolário 4 Exato-X-de-Y(k) é W[1]-difícil.

Prova: Segue diretamente do Teorema 3. \square

A Figura 7 ilustra uma clique de tamanho $k = 3$ e o subgrafo-solução de custo $k' = 18$, para os grafos G e G' , respectivamente, apresentados na Figura 6.

Corolário 5 Min-X-de-Y(k) é W[1]-difícil.

Prova: A prova deste corolário é baseada na redução do Teorema 3 Clique(k), no entanto efetuando a seguinte ponderação ao grafo:

- se $d(v_i) < k - 1$, então $\tau(s, u_i) = k^2 + 3k + 1$;
- se $d(v_i) \geq k - 1$, então $\tau(s, u_i) = 1$;
- todas os demais arcos terão peso 1. \square

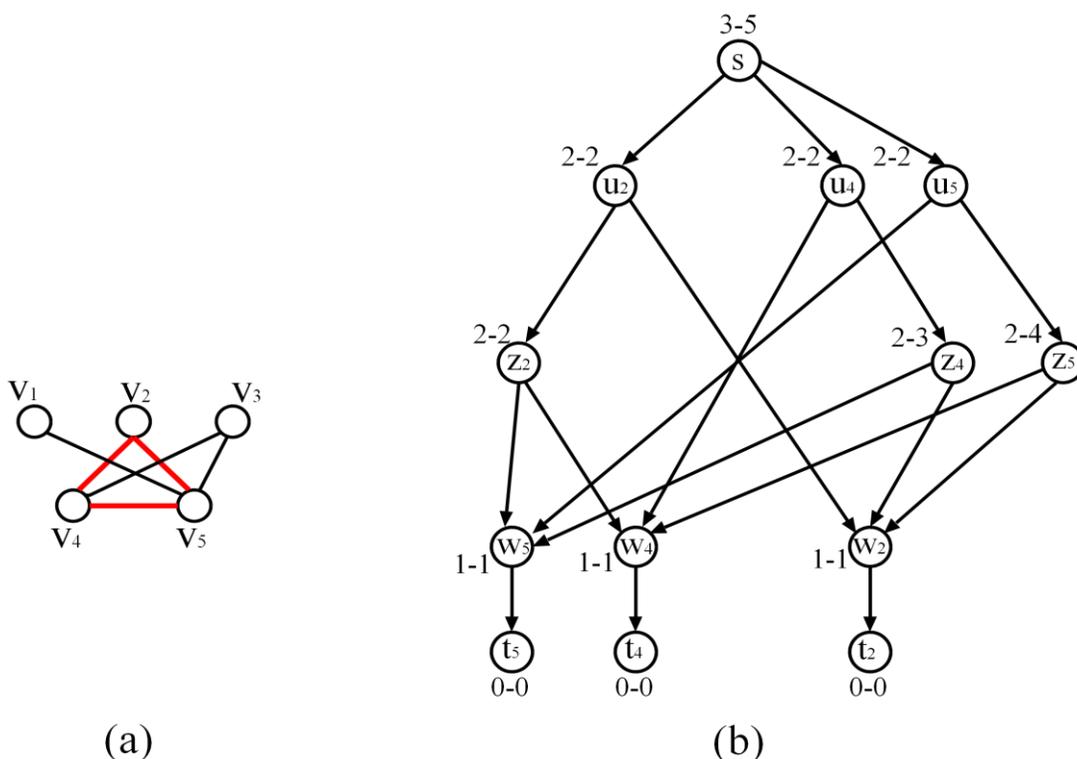


Figura 7: (a) Clique de tamanho 3. (b) Subgrafo-solução de custo 18 em G' .

4 Considerações Finais

Grafos E/Ou são estruturas de dados que possibilitam a modelagem de diversos problemas reais, de forma que soluções para estes problemas possam ser buscadas computacionalmente. Grafos X-de-Y, por sua vez, são estruturas que generalizam os grafos E/Ou, capazes de modelar mais facilmente uma gama maior de problemas.

Neste trabalho foram abordados dois problemas relativos a esses grafos, Min-E/Ou e Min-X-de-Y, pois subgrafos-solução geralmente são interpretados como um roteiro a ser seguido para a solução de um problema.

Estudos sobre a complexidade do problema Min-E/Ou têm sido feitos desde a década de 1970, quando Sahni apresentou uma prova de NP-completude para Min-E/Ou. No entanto, poucos avanços haviam sido feitos no que diz respeito a novas abordagens para esses problemas.

Sob o ponto de vista da complexidade parametrizada, uma teoria recente e promissora para o tratamento de problemas NP-difíceis, foi provado que o problema Min-E/Ou é tratável por parâmetro fixo, quando fixados um parâmetro k para o custo máximo do subgrafo-solução e um parâmetro r para o número de arcos de saída de mesmo peso para os vértices do tipo Ou . Também foi mostrado que as versões Min-X-de-Y(k) e Exato-X-de-Y(k) do problema Min-X-de-Y são W[1]-difíceis, o que significa que o problema Min-X-de-Y permanece intratável mesmo fixado um parâmetro k para o custo.

Referências

- Arenales, M. N. & Vianna, A. C. G.** (2006). O problema de corte de placas defeituosas. *Pesquisa Operacional*, 26(2):185–202.
- Barbosa, V. C.** (2002). The combinatorics of resource sharing. In Fialles, M. & Gomes, F., editors, *Models for Parallel and Distributed Computation: Theory, Algorithmic Techniques and Applications*, pages 27–50. Kluwer Academic Publishers, The Netherlands.
- Corandi, R. & Westfechtel, B.** (1998). Version models for software configuration management. *ACM Computing Surveys*, 30(2):332–282.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C.** (2002). Algoritmos de grafos. In *Algoritmos: Teoria e Prática*, pages 468–470. Campus, Rio de Janeiro.
- Cui, Y.** (2011). Fast heuristic for constrained homogenous t-shape cutting patterns. *Applied Mathematical Modelling*, doi: 10.1016/j.apm.2011.11.005. <http://www.sciencedirect.com/science/article/pii/S0307904X11006937>.
- Downey, R. & Fellows, M.** (1999). *Parameterized Complexity*. Springer.
- Flum, J. & Grohe, M.** (2006). *Parameterized Complexity Theory*. Springer.
- Gallo, G., Longo, G., Nguyen, S., & Pallottino, S.** (1993). Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42:177–201.
- Hansen, E. & Zilberstein, S.** (2001). Lao*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129:35–62.
- Morabito, R. & Pureza, V.** (2010). A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. *Annals of Operations Research*, 179:297–315. 10.1007/s10479-008-0457-4.
- Niedermeier, R.** (2006). *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications, Oxford University Press.
- Nilsson, N. J.** (1971). Problem-reduction representations. In Dojny, R. F. & Eakins, M., editors, *Problem Solving Methods in Artificial Intelligence*, pages 80–112. McGraw-Hill, United States of America.
- Sahni, S.** (1974). Computationally related problems. *Siam Journal on Computing*, 3(4):262–279.
- Schaeffer, J., Plaat, A., & Junghanns, A.** (2001). Unifying single-agent and two-player search. *Information Sciences*, 135(3-4):151–175.