2
Transporte em Redes de Dutos

Neste capítulo, apresentamos uma formalização para o Problema de Transporte por Dutos (PTD), algoritmos e reduções que determinam a complexidade de variações deste problema.

O modelo matemático utilizado pelo PTD é uma simplificação de um modelo mais complexo, que descreve as operações realizadas nas redes de oleodutos da Transpetro. A utilização de um modelo simplificado nesta tese de doutorado tem por objetivo estudar as características essenciais do problema e suas conseqüências.

2.1
Descrição do PTD

O modelo matemático do PTD utiliza as seguintes hipóteses simplificadoras:

1. os líquidos transportados são incompressíveis;
2. os nós da rede têm capacidade de estoque ilimitada;
3. os nós de destino de todas as bateladas são dados;
4. os volumes de todas as bateladas são unitários;
5. os volumes de todos os dutos são inteiros;
6. as bateladas não podem ser divididas durante o transporte;
7. a inserção de qualquer batelada em qualquer duto tem duração unitária.

2.1.1 Rede de Dutos

Seja $G = (N, A)$ um grafo dirigido, onde N é um conjunto de nós e A um conjunto de arcos. Dado um arco $a = (i, j) \in A$, denotamos i por *nó inicial* de a e j por *nó final* de a. Os arcos e nós modelam respectivamente os dutos da rede e as suas áreas operacionais. Cada arco $a \in A$ tem um volume associado $v(a)$. Além disso, definimos um conjunto de posições de duto $A^* = \{(a, l) | a \in A \text{ e } l \in \{1, \ldots, v(a)\}\}$.

Lembre que assumimos que os nós têm capacidades ilimitadas, ou seja, não há nenhuma limitação quanto ao número de bateladas que podem ser armazenadas simultaneamente em um mesmo nó.

2.1.2 Ordens de Transporte

Seja L um conjunto de ordens de transporte, onde cada ordem $k \in L$ determina que $u(k)$ bateladas de peso $w(k)$ sejam transportadas para o nó $d_k \in N$. Além disso, definimos um subconjunto $F \subset L$ de ordens cujo transporte não precisa ser completado. Chamamos as ordens deste subconjunto de *ordens proteláveis*. As demais ordens são chamadas de *não proteláveis*.

Agora, dividimos cada ordem de L em sub-ordens de volume unitário. Para cada ordem $k \in L$, seja $b_{l'}(k)$ a l'-ésima sub-ordem de k, para $l' = 1, 2, \ldots, u(k)$. Neste caso, temos $w(b_{l'}(k)) = w(k)$ e $d_{b_{l'}(k)} = d_k$. Com isso, definimos os seguintes conjuntos de sub-ordens:

$$L^1 = \{b_1(k), \ldots, b_{u(k)}(k) | k \in L\}$$

e

$$F^1 = \{b_1(k), \ldots, b_{u(k)}(k) | k \in F\}.$$

Observe que cada sub-ordem determina o transporte de uma única batelada. Por isso, para cada sub-ordem $b \in L^1$, também denotamos por b a batelada correspondente. As bateladas associadas a ordens (não) proteláveis são igualmente chamadas de bateladas (não) proteláveis.

Vale mencionar que os conjuntos L^1 e F^1 não trazem nenhuma informação adicional em relação aos conjuntos L e F. De fato, os últimos conjuntos denotam uma forma mais compacta para representar os primeiros.

No entanto, para simplificar a descrição dos algoritmos e as demonstrações, utilizamos os primeiros sempre que possível.

2.1.3 Estados da Rede

Agora, apresentamos a descrição de um estado da rede. No PTD, apenas o estado inicial é dado. Os estados seguintes dependem da solução do problema.

Cada duto $a \in A$ tem um volume associado $v(a)$. Como assumimos que todas as bateladas têm volume unitário, cada duto a pode conter exatamente $v(a)$ bateladas em um dado estado da rede. Apesar da inserção de uma batelada em um duto ser um processo gradativo, desconsideramos todos os estados intermediários que podem ser gerados pela inserção parcial de bateladas em dutos. Com isso, uma solução viável para o PTD gera apenas um número finito de estados relevantes. Estes estados são indexados de 0 a q, onde o estado inicial recebe o índice 0. Em cada um destes estados, a posição de cada batelada em cada duto ou nó está bem definida.

Denotamos por $p_t(b) \in A^* \cup N$ a posição da batelada b no estado t, para $t = 0, 1, \ldots, q$. Se $p_t(b) = (a, l) \in A^*$, então a batelada b está localizada na l-ésima posição do duto a, no estado t. Em caso contrário, se $p_t(b) = i \in N$, então a batelada b está armazenada no nó i, neste mesmo estado. Além disso, o conteúdo de um dado duto a em um dado estado t é representado por uma lista da bateladas $[b_1, b_2, \ldots, b_{v(a)}]$, onde b_l é a única batelada tal que $p_t(b_l) = (a, l)$, para $l = 1, 2, \ldots, v(a)$.

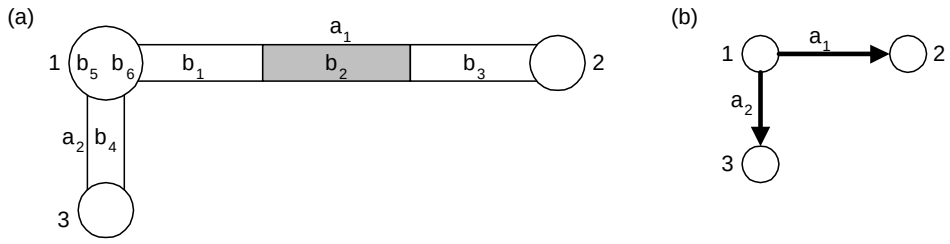

Figura 2.1: (a) O conteúdo de uma rede de dutos em um dado estado; (b) o grafo G correspondente.

Por exemplo, a Figura 2.1.(a) representa os dutos de uma rede e seus respectivos conteúdos em um dado estado t. O grafo G correspondente aparece na Figura 2.1.(b). A rede apresentada tem os dois dutos $a_1 = (1, 2)$ e $a_2 = (1, 3)$. Os respectivos sentidos de fluxo são indicados pelos arcos de G. Os volumes de a_1 e a_2 são respectivamente $v(a_1) = 3$ e $v(a_2) = 1$.

Observe que, pela Figura 2.1.(a), temos $p_t(b_1) = (a_1, 1)$, $p_t(b_2) = (a_1, 2)$, $p_t(b_3) = (a_1, 3)$ e $p_t(b_4) = (a_2, 1)$. Conseqüentemente, os conteúdos de a_1 e a_2 são respectivamente representados pelas listas $[b_1, b_2, b_3]$ e $[b_4]$. Além disso, temos $p_t(b_5) = p_t(b_6) = 1$, indicando que b_5 e b_6 estão armazenados no nó 1.

Para simplificar a notação, assumimos que todas as bateladas que correspondem a uma mesma ordem, têm a mesma posição inicial. Neste caso, observe que as bateladas localizadas em dutos no estado inicial precisam estar associadas a ordens unitárias.

Também utilizamos a seguinte terminologia. Denotamos por nó de *entrada (saída)* de $p_t(b)$:

1. o nó inicial (final) de a, caso $p_t(b) = (a, l) \in A^*$;

2. o próprio nó i, caso $p_t(b) = i \in N$.

2.1.4 Operações

Uma solução para o PTD é representada por um conjunto Q de operações *push-pop* (PP) definidas da seguinte forma. Seja $a = (i, j)$ um arco de G cujo conteúdo em um dado estado t é representado pela lista $[b_1, b_2, \ldots, b_{v(a)}]$. Além disso, seja b uma batelada armazenada no nó i, neste mesmo estado. Uma operação PP é uma tripla (b, a, t) que determina que a batelada b é inserida no duto a, durante o intervalo de tempo $[t, t+1)$. Como resultado desta operação, no estado $t + 1$, o conteúdo de a é representado pela lista $[b, b_1, b_2, \ldots, b_{v(a)-1}]$ e a batelada $b_{v(a)}$ é armazenada no nó j.

Vale mencionar que duas operações PP (b_1, a_1, t_1) e (b_2, a_2, t_2) podem ser executadas simultaneamente $(t_1 = t_2)$ desde que $b_1 \neq b_2$ e $a_1 \neq a_2$. Observe que um dado conjunto Q de operações gera uma seqüência de estados $1, 2, \ldots, q$ para a rede, onde $q = \max\{t + 1 | (b, a, t) \in Q\}$. Tal seqüência é viável se e somente se ela satisfaz as seguintes condições:

1. cada batelada $b \in L^1 - F^1$ está armazenada no nó d_b, no estado q;

2. para cada batelada $b \in F^1$, existe um caminho em G que começa no nó de saída de $p_q(b)$ e termina no nó d_b.

2.1.5 Objetivos

A primeira função objetivo considerada é o custo total das operações PP. Este custo não depende dos intervalos de tempo de execução destas operações. Por isso, quando utilizamos esta função objetivo, assumimos que as operações PP são executadas seqüencialmente, ou seja, a t-ésima operação PP é executada durante o intervalo $[t-1, t)$. Neste caso, se a ordem de execução das operações PP é dada, cada operação pode ser representada apenas pela dupla (b, a),onde b é a batelada a ser inserida no arco a.

Definimos o custo de uma operação PP de forma bastante genérica. Primeiro, associamos um coeficiente não negativo $\alpha_{a,l}$ a cada posição de duto $(a, l) \in A^*$. A cada arco $a \in A$, também associamos um coeficiente $\alpha_{a,0}$, relacionado ao custo de inserção de uma batelada no duto a. Com isto, para um dado arco a cujo conteúdo é representado pela lista $[b_1, b_2, \ldots, b_{v(a)}]$, uma operação PP (b, a, t) tem custo

$$c_t = \alpha_{a,0} \times w(b) + \sum_{l=1}^{v(a)} \alpha_{a,l} \times w(b_l).$$

O custo total de uma dada seqüência de operações PP é dado por

$$c^{\text{tot}} = \sum_{t=1}^{q} c_t.$$

Uma motivação prática para a função de custo anterior é que diferentes posições de um oleoduto podem apresentar diferentes inclinações. Por isto, devido à ação da gravidade, uma batelada pesada posicionada em um trecho de subida pode provocar um custo de bombeamento mais elevado que a mesma batelada quando posicionada em um trecho de descida. Neste caso, as posições de duto que correspondem a trechos de subida podem ter coeficientes $\alpha_{a,l}$ mais elevados que as posições associadas a trechos de descida.

A segunda função objetivo considerada é o *makespan*, que é dado simplesmente pelo valor de q.

2.1.6 Exemplo

Agora, ilustramos nosso modelo com um exemplo. Neste exemplo, o grafo G é representado pela Figura 2.2.

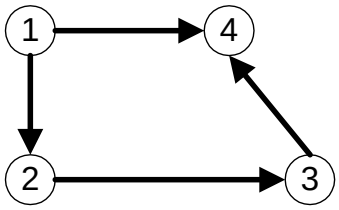

Figura 2.2: Um exemplo de grafo G.

Além disso, temos os seguintes dados de entrada:

1. $L^1 = \{b_1, b_2, \ldots, b_{10}\}$, onde $F^1 = \{b_6, b_7, \ldots, b_{10}\}$;
2. $p_0(b_1) = ((1,4),1)$;
3. $p_0(b_2) = ((1,2),1)$;
4. $p_0(b_3) = ((2,3),1)$ e $p_0(b_4) = ((2,3),2)$;
5. $p_0(b_5) = ((3,4),1)$;
6. $p_0(b_6) = p_0(b_7) = 1$;
7. $p_0(b_8) = p_0(b_9) = 2$;
8. $p_0(b_10) = 3$;
9. $d_{b_1} = 4$;
10. $d_{b_2} = 2$;
11. $d_{b_3} = 3$;
12. $d_{b_4} = d_{b_5} = \cdots = d_{b_{10}} = 4$.

Neste caso, uma solução viável é o conjunto de operações PP dado por

$$\{\ (b_9,(2,3),0),\ (b_4,(3,4),1),\ (b_{10},(3,4),2),\ \ (b_8,(2,3),3),\ (b_6,(1,2),4),\ (b_7,(1,4),4)\ \}.$$

A execução desta solução é representada pela Figura 2.3. Nesta figura, o estado inicial é representado no canto superior esquerdo. Além disso, a seqüência de desenhos que segue da esquerda para a direita, e depois de cima para baixo, representa a seqüência de estados gerada. Nestes desenhos, as bateladas proteláveis são representadas em cinza. Além disso, as setas indicam os dutos que realizaram operações PP entre o estado anterior e o atual.

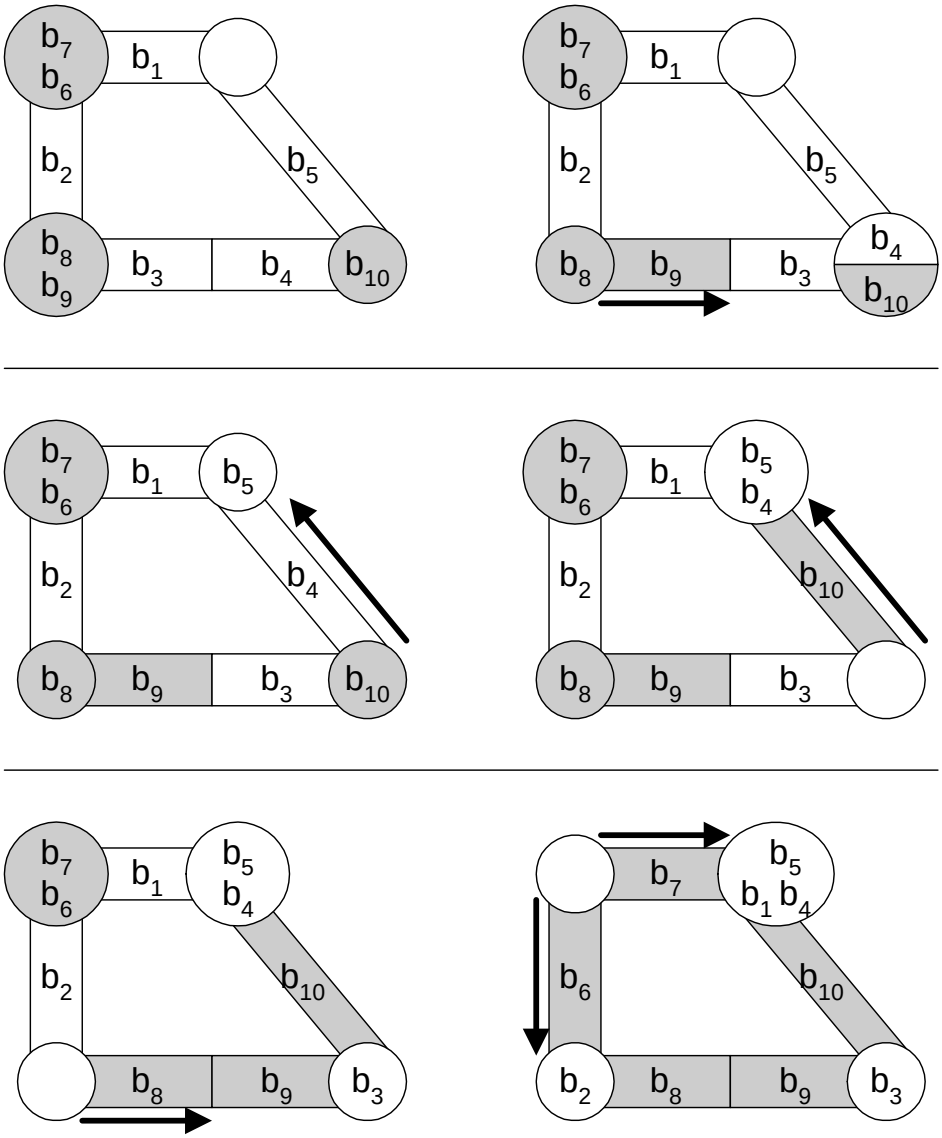

Figura 2.3: Um exemplo de execução de operações PP.

2.2 Complexidade do PTD

Nesta seção, demonstramos que o PTD é um problema $\mathcal{NP}$-difícil. Nesta demonstração, apresentamos uma redução polinomial do problema conhecido como Problema de Cobertura por Vértices [7] ao PTD. Como conseqüência desta redução, a existência de uma algoritmo polinomial para o PTD implicaria em $\mathcal{P} = \mathcal{NP}$ [13].

Primeiro, definimos o Problema de Cobertura por Vértices. Dado um grafo não dirigido $\mathcal{G} = (V, E)$, uma cobertura para $\mathcal{G}$ é definida como um subconjunto $\mathcal{C} \subset V$ tal que, para toda aresta $e = (i, j) \in E$, $i \in \mathcal{C}$ ou $j \in \mathcal{C}$ (ou ambos). Para um dado número inteiro positivo $\mathcal{K}$, o problema consiste

em encontrar uma cobertura de até $\mathcal{K}$ vértices para $\mathcal{G}$. Agora, estabelecemos o seguinte teorema.

Teorema 2.1 *O PTD é $\mathcal{NP}$-difícil. Este resultado ainda vale se assumimos que o grafo G é acíclico.*

Prova. Seja $\mathcal{G} = (V, E)$ um grafo não dirigido e $\mathcal{K}$ um número inteiro positivo. Construímos uma instância I da PTD a partir de $\mathcal{G}$ e $\mathcal{K}$ da seguinte forma:

1. começamos com todos os conjuntos que descrevem I vazios;
2. inserimos os nós 0 e $|V| + |E| + 1$ em N;
3. para cada $l = 1, \ldots, \mathcal{K}$, inserimos uma batelada protelável b_l em F^1, tendo $d_{b_l} = |V| + |E| + 1$ e $p_0(b_l) = 0$;
4. para cada vértice $i \in V = \{1, 2, \ldots, |V|\}$, inserimos:
   (a) um nó i correspondente em N e uma arco $(0, i)$ em A;
   (b) uma batelada protelável b'_i em F^1, tendo $d_{b'_i} = i$ e $p_0(b'_i) = ((0, i), 1)$;
5. para cada aresta $e = (i, j) \in E = \{1, 2, \ldots, |E|\}$, inserimos:
   (a) um nó $|V| + e$ correspondente em N;
   (b) três arcos $(i, |V| + e)$, $(j, |V| + e)$ e $(|V| + e, |V| + |E| + 1)$ em A;
   (c) cinco bateladas proteláveis $b^1_e, \ldots, b^5_e$ em F^1, tendo $d_{b^1_e} = \cdots = d_{b^1_e} = |V| + |E| + 1$, $p_0(b^1_e) = i$, $p_0(b^2_e) = j$, $p_0(b^3_e) = ((i, |V| + e), 1)$, $p_0(b^4_e) = ((j, |V| + e), 1)$ e $p_0(b^5_e) = ((|V| + e, |V| + |E| + 1), 1)$;
   (d) uma batelada não protelável b^*_e em L^1, tendo $d_{b^*_e} = |V| + e$ e $p_0(b^*_e) = 0$;
6. para todo arco $a \in A$, fazemos $v(a) = 1$.

A Figura 2.4 mostra uma representação genérica do grafo G obtido com a redução de uma instância do Problema de Cobertura por Vértices. Nesta figura, representamos os números dos nós dentro dos círculos correspondentes. Além disso, as setas desenhadas dentro dos dutos indicam os respectivos sentidos de fluxo. Observe que o grafo G construído é sempre acíclico.

A seguir, demonstramos que I tem uma solução viável se e somente se $\mathcal{G}$ tem uma cobertura de até $\mathcal{K}$ vértices.

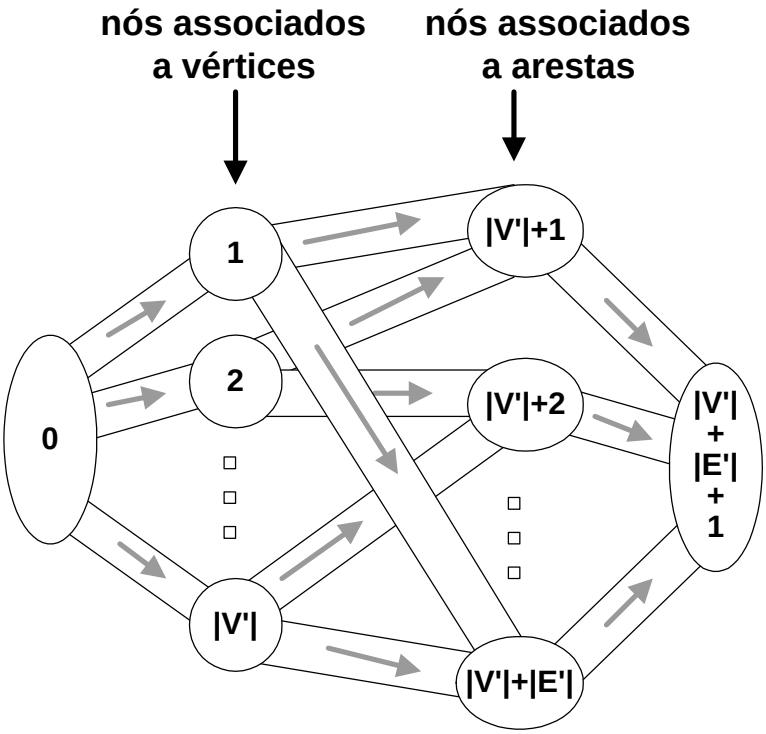

Figura 2.4: Uma representação genérica do grafo G obtido com a redução de uma instância do Problema de Cobertura por Vértices.

Primeiro, consideramos uma solução viável qualquer Q para I. Seja $\mathcal{C}$ um conjunto de vértices contendo todo vértice i tal que pelo menos uma batelada $b \in \{b_1, b_2, \ldots, b_\mathcal{K}\}$ é inserida no arco correspondente $(0, i)$, segundo Q. Por construção, $\mathcal{C}$ não pode ter mais do que $\mathcal{K}$ vértices. Logo, basta provar que $\mathcal{C}$ é uma cobertura para $\mathcal{G}$. Por construção de I, para cada aresta $e = (i, j) \in E$, há uma batelada não protelável $b_e^* \in L^1 - F^1$ destinada ao nó correspondente $|V| + e$. Observe também que b_e^* precisa passar pelo arco $(0, i)$ ou pelo arco $(0, j)$ para chegar ao seu destino. Por isto, pelo menos uma batelada protelável precisa ser inserida no arco $(0, i)$ ou no arco $(0, j)$, para cada $e = (i, j) \in E$. Por outro lado, $b_1, b_2, \ldots, b_\mathcal{K}$ são as únicas bateladas proteláveis armazenadas no nó 0, no estado inicial. Conseqüentemente, para toda aresta $e = (i, j) \in E$, $i \in \mathcal{C}$ ou $j \in \mathcal{C}$, o que mostra que $\mathcal{C}$ é uma cobertura para $\mathcal{G}$.

Agora, a partir de uma cobertura qualquer $\mathcal{C}$ de até $\mathcal{K}$ vértices para $\mathcal{G}$, construímos uma solução viável para I. Para isto, utilizamos uma função arbitrária $f : E \to \mathcal{C}$ tal que e é sempre adjacente ao nó $f(e)$. Como $\mathcal{C}$ é uma cobertura, garantimos que tal função existe. Então construímos um solução viável para I em quatro passos:

1. para cada $e \in E$, inserimos b_e^* no arco $(0, f(e))$ (correspondendo à

operação PP $(b_e^*, (0, f(e)))$;

2. para cada $i \in \mathcal{C}$, escolhemos arbitrariamente uma batelada $b \in \{b_1, b_2, \ldots, b_\mathcal{K}\}$ e a inserimos no arco $(0, i)$;

3. para cada $e \in E$, inserimos b_e^* no arco $(f(e), |V| + e)$;

4. para cada $e \in E$, inserimos no arco $(f(e), |V| + e)$ a única batelada $b \in \{b_e^1, b_e^2\}$ tal que $p_0(b) = f(e)$;

□

Observe que o teorema anterior não prova que o PTD é $\mathcal{NP}$-completo. Para isto, precisaria ser demonstrado que o problema pertence a $\mathcal{NP}$, ou seja, para qualquer instância viável existe um certificado desta viabilidade que pode ser testado em tempo polinomial. Esta demonstração ainda é um problema aberto.

2.3 Definição do PTD Síncrono

Devido ao resultado apresentado pelo teorema 2.1, passamos a estudar casos particulares do PTD que permitem encontrar soluções viáveis de forma eficiente, ou seja, em tempo polinomial. Estes casos particulares são caracterizados pela inclusão da seguinte hipótese no PTD: todas as bateladas proteláveis estão armazenadas em nós no estado inicial. Conseqüentemente, os conteúdos iniciais dos dutos são representados exclusivamente por bateladas não proteláveis. Chamamos esta hipótese de *sincronismo*. Além disso, denominamos o problema resultante PTD Síncrono (PTDS). Por exemplo, a instância definida na subseção 2.1.6 é uma instância do PTDS.

A partir deste ponto até o final do capítulo, apresentamos resultados relativos ao PTDS. Para este problema, consideramos as duas funções objetivo descritas na subseção 2.1.5. O problema de encontrar uma solução viável para o PTDS que minimiza o custo total das operações PP é chamado de PTDSC. Por outro lado, quando desejamos minimizar *makespan*, o problema é chamado de PTDSM.

2.4 Algoritmos para o PTD Síncrono

Nesta seção, consideramos o PTDS, uma variação do PTD definida na seção 2.3. Para este problema, apresentamos uma condição necessária e suficiente para que uma instância I do PTDS seja viável. Em seguida, descrevemos o algoritmo *Batch-to-Pipe Assignment* (BPA). Este algoritmo testa a condição anterior e, caso ela seja satisfeita, encontra uma solução viável para I. Se o grafo G for acíclico, também garantimos que o custo total das operações PP utilizadas pela solução é mínimo.

O algoritmo BPA executa em tempo polinomial em função do tamanho de I, incluindo o conjunto L^1. No entanto, se representamos as ordens de forma mais compacta, através do conjunto L, então o tempo de execução do BPA passa a ser pseudo-polinomial em função do novo tamanho de I. Na subseção 2.4.8, apresentamos uma variação do algoritmo BPA que executa em tempo polinomial em função do tamanho da representação mais compacta de I.

2.4.1 Rotas de Bateladas

Agora, introduzimos o conceito de rotas de bateladas. Este conceito é fundamental para a apresentação do algoritmo BPA. Dada uma batelada b e um conjunto Q de operações PP, a rota $R(b)$ de b é definida como a seqüência de todos os arcos visitados por b em ordem cronológica, durante a execução de Q. Observe que, se b está inicialmente contida em um arco a, então a é necessariamente o primeiro arco de $R(b)$.

Também utilizamos a seguinte notação. Denotamos por $\bar{v}(b)$ a "distância" entre o nó inicial do primeiro arco de $R(b)$ e a posição inicial de b, e por $\hat{v}(b)$ a "distância" entre a posição final de b e o nó final do último arco de $R(b)$. Formalizando, temos

$$\bar{v}(b) = \begin{cases} \sum_{k=0}^{l-1} \alpha_{a,k}, & \text{se } p_0(b) = (a,l) \in A^*; \\ 0, & \text{se } p_0(b) = i \in N; \end{cases}$$

e

$$\hat{v}(b) = \begin{cases} \sum_{k=l}^{v(a)} \alpha_{a,k}, & \text{se } p_q(b) = (a,l) \in A^*; \\ 0, & \text{se } p_q(b) = i \in N. \end{cases}$$

Com isto, observe que o custo total c^{tot} de Q (definido na subseção 2.1.5) pode ser reescrito como função das rotas de bateladas e das distâncias

definidas anteriormente:

$$c^{\text{tot}} = \sum_{b \in L^1} w(b) \left[\left(\sum_{a \in R(b)} \sum_{l=0}^{v(a)} \alpha_{a,l} \right) - \bar{v}(b) - \hat{v}(b) \right].$$

A seguir, definimos uma terminologia que relaciona as rotas de batelada com as suas respectivas posições finais. Dizemos que $p_q(b)$ é uma *posição final válida* para $b \in F^1$ quando existe dois caminhos em G: um conectando o nó de saída de $p_0(b)$ ao nó de entrada de $p_q(b)$ e outro conectando o nó de saída de $p_q(b)$ ao nó d_b. Se b é uma batelada não protelável, então d_b é a única *posição final válida* para b. Além disso, dada uma posição final válida $p_q(b)$ para $b \in L^1$, $R(b)$ é dita uma *rota válida* para b quando ela satisfaz as quatro condições a seguir:

1. o primeiro arco de $R(b)$ parte do nó de entrada de $p_0(b)$;
2. se $p_0(b) = (a, l) \in A^*$, então a é o primeiro arco de $R(b)$;
3. o último arco de $R(b)$ termina no nó de saída de $p_0(b)$;
4. se $p_q(b) = (a, l) \in A^*$, então a é o último arco de $R(b)$.

2.4.2 Condições de Viabilidade

Agora, apresentamos uma condição necessária para que uma dada instância I do PTDS seja viável. Como os dutos da rede só podem conter bateladas proteláveis no estado final, qualquer solução viável para i associa bateladas proteláveis a posições de duto. Como esta associação é determinada pela posição final de cada batelada protelável, cada posição de duto $(a, l) \in A^*$ deve estar associada a exatamente uma batelada protelável $b \in F^1$. Logo, trata-se de um emparelhamento de bateladas a dutos (daí o nome *Batch-to-Pipe Assignment*). Vale mencionar que pode haver um excesso de bateladas proteláveis. Neste caso, as bateladas excedentes não precisam estar associadas. Esta condição de viabilidade é formalizada pela seguinte proposição.

Proposição 2.2 *Se um dada instância I do PTDS é viável, então existe um emparelhamento entre um subconjunto de F^1 e o conjunto A^* tal que, para toda batelada b associada a um posição de duto (a, l), (a, l) é uma posição final válida para b.*

Um *estado final válido* para uma rede de dutos atribui uma posição final válida $p_q(b)$ a cada batelada $b \in L^1$, satisfazendo a condição dada pela Proposição 2.2.

A seguir, apresentamos uma condição suficiente para a viabilidade de I. Para isto, definimos um grafo de precedências em função de um estado final válido e uma rota válida associada a cada batelada. Este grafo é um grafo dirigido $H = (A, D)$ construído da seguinte forma:

1. utilizamos o conjunto de arcos A do grafo G, como conjunto de nós para o grafo H;

2. criamos um arco $d = (a, a') \in D$ se e somente se existe uma batelada $b \in L^1$ cuja rota $R(b) = [a_1, a_2, \ldots, a_{|R(b)|}]$ é tal que $a_j = a$ e $a_{j+1} = a'$ para algum $j \in \{1, 2, \ldots, |R(b)| - 1\}$.

Por definição, se H tem um arco de a para a', então existe pelo menos uma batelada que passa do arco a para o arco a' durante a execução de qualquer solução viável que use as rotas e o estado final dados. Como conseqüência disto, há pelo menos precedência obrigatória entre as operações PP previstas para o duto a e as previstas para o duto a'. Se tal arco não existe no grafo H, então as operações previstas para o duto a não têm precedência sobre as previstas para o duto a'. Com base nesta observação, estabelecemos o seguinte teorema.

Teorema 2.3 *Dados um estado final válido e uma rota válida para cada batelada, se o grafo de precedência H correspondente é acíclico, então existe uma solução viável para I que transporta as bateladas pelas rotas dadas e que leva a rede ao estado final dado.*

Prova. Demonstramos este teorema mostrando como construir a dita solução viável.

Como H é acíclico, ele tem pelo menos um nó fonte (sem nenhum arco incidindo nele). Seja $a = (i, j)$ um arco de G que corresponde a um nó fonte arbitrário de H. No estado inicial, qualquer batelada b cuja rota contém o arco a, está armazenada no nó i. Em caso contrário, H teria um arco incidente em a. Logo, iniciamos a construção de uma solução viável com uma sequência de operações PP que insere em a todas as bateladas cuja rota passa por a. Vale mencionar que as últimas bateladas inseridas devem corresponder à lista que representa o conteúdo do duto a no estado final. A ordem cronológica de inserção destas bateladas deve ser o reverso de sua ordem na lista. Após a execução destas operações PP, toda batelada

b cuja rota contém o arco $a = (i, j)$ estará armazenada no nó j, incluindo aquelas com $p_q(b) = j$. Além disso, o conteúdo do duto a estará de acordo com o estado final dado. Em seguida, removemos o arco a de todas as rotas de batelada. Também removemos de H o nó a (e todos os arcos que partem deste nó), mantendo o grafo de precedências consistente com as novas rotas. Com isto, podemos escolher outro nó fonte arbitrário $a' = (i', j')$ no grafo H atualizado (que continua acíclico). Como H não tem nenhum arco incidente em a', qualquer batelada b cuja rota contém o arco a', está armazenada no nó i', no estado atual.

Observe que podemos repetir este procedimento até que o grafo H esteja vazio. Quando isto ocorrer, o estado da rede será igual ao estado final dado e as bateladas terão percorrido as rotas dadas. Por isso, a solução dada pela seqüência de operações PP executadas até então será viável. □

2.4.3 Algoritmo BPA

Seja $n = |N|$, $m = |A|$, e $s = |A^*|$. Descrevemos o algoritmo BPA em seis passos resumidos pelo pseudocódigo da Tabela 2.1. Nesta tabela, denotamos por $c()$ a função de custo do PTDSC. Também chamamos o procedimento *Seqüenciar*, que é descrito na subseção 2.4.5.

Aqui, esclarecemos alguns detalhes omitidos no pseudocódigo da Tabela 2.1. No Passo 1, se não existe caminho de i para j em G, então $v(i, j)$ é iniciado com uma valor suficientemente grande (denotado por ∞). No Passo 2, o custo mínimo de transportar b por uma rota válida até a posição de duto (a, l), onde $a = (i, j)$, é dado por $w(b)(v(p_0(b), i) + \sum_{k=0}^{l-1} \alpha_{a,k})$. Observe que o valor de $v(p_0(b), i)$ é calculado no Passo 1. No Passo 4, para cada $b \in F^1$, se $p_q(b)$ recebe o valor $(a, l) \in A^*$, onde $a = (i, j)$, então uma rota válida de custo mínimo para b é dada pela concatenação de $S(p_0(b), i)$ com $[a]$. Por outro lado, para cada $b \in L^1 - F^1$, se $p_0(b) = (a, l) \in A^*$, onde $a = (i, j)$, então uma rota válida de custo mínimo para b é dada pela concatenação de $[a]$ com $S(j, d_b)$. Em caso contrário, se $p_0(b) \in N$, então este rota é dada por $S(p_0(b), d_b)$. Se $c(X) = \infty$, então a instância I é dita inviável porque a condição dada pela Proposição 2.2 é violada. Em caso contrário, estabelecemos a seguinte proposição.

Proposição 2.4 *Para cada $b \in L^1$, seja $R(b)$ a rota escolhida para a batelada b pelo algoritmo BPA. Se $c(X) \neq \infty$, então nenhuma solução viável para I tem custo menor do que*

Algoritmo BPA

Passo 1: **Para** cada $a \in A$,
(comprimento de a) $\leftarrow \sum_{l=0}^{v(a)} \alpha_{a,l}$;
Fim Para;
Para cada par $i, j \in N$,
$S(i,j) \leftarrow$ um caminho de comprimento mínimo de i até j em G;
$v(i,j) \leftarrow$ comprimento de $S(i,j)$;
Fim Para;

Passo 2: construir o seguinte grafo bipartido $B = (N^1 \cup N^2, A^B)$:
Para cada $b \in F^1$, $N^1 \leftarrow N^1 \cup n(b)$;
Para cada $(a,l) \in A^*$, $N^2 \leftarrow N^2 \cup n(a,l)$;
Para $l = 1, 2, \ldots, |F^1| - s$, $N^2 \leftarrow N^2 \cup n(\cdot, l)$;
Para cada par $(n(b), n(a,l)) \in N^1 \times N^2$,
$A^B \leftarrow A^B \cup (n(b), n(a,l))$;
$c(n(b), n(a,l)) \leftarrow$ o custo mínimo de transportar b por uma rota válida tal que $p_q(b) = (a,l)$;
Fim Para;
Para cada par $(n(b), n(\cdot,l)) \in N^1 \times N^2$,
$A^B \leftarrow A^B \cup (n(b), n(\cdot,l))$; $c(n(b), n(\cdot,l)) \leftarrow 0$;
Fim Para;

Passo 3: obter um emparelhamento X de custo mínimo em B;
Se $c(X) = \infty$ **então** I é inviável; terminar;

Passo 4: **Para** cada $b \in F^1$,
$k \leftarrow$ o único nó de B conectado a $n(b)$ por X;
Se $k = n(a,l)$ para alguma $(a,l) \in A^*$ **então**
$p_q(b) \leftarrow (a,l)$;
$R(b) \leftarrow$ uma rota válida de custo mínimo para b tal que $p_q(b) = (a,l)$;
Senão
$p_q(b) \leftarrow p_0(b)$; $R(b) \leftarrow$ uma rota vazia;
Fim Se;
Fim Para;
Para cada $b \in L^1 - F^1$,
$R(b) \leftarrow$ uma rota válida de custo mínimo para b;
Fim Para;

Passo 5: construir o grafo de precedências H correspondente;

Passo 6: *Seqüenciar* uma solução viável a partir de X e H.

Tabela 2.1: Um pseudocódigo para o algoritmo BPA.

$$c(X) + \sum_{b \in L^1 - F^1} w(b) \left(\left(\sum_{a \in R(b)} v(a) \right) - \bar{v}(b) \right). \qquad (2\text{-}1)$$

Prova. Por construção, $c(X)$ é o custo mínimo de transportar as bateladas

proteláveis por rotas válidas até posições na rede que descrevam um estado final válido. Pela Proposição 2.2, qualquer solução viável para I precisa levar a rede até um estado final válido. Além disso, por definição, as bateladas precisam ser transportadas por rotas válidas. Logo, o custo total de transporte das bateladas proteláveis em qualquer solução viável para I não pode ser menor do que $c(X)$. Como o algoritmo BPA escolhe uma rota válida de custo mínimo independentemente para cada batelada não protelável, nenhuma solução viável para I tem custo total menor do que (2-1). Observe que o segundo termo de (2-1) é o somatório dos custos das rotas escolhidas para as bateladas não proteláveis. □

No Passo 4, o algoritmo constrói um grafo de precedências H em função de X. Pelo Teorema 2.3, se H é acíclico, então podemos construir uma solução viável para I cujo custo é igual ao limite inferior dado por (2-1). A seguir, descrevemos o procedimento *Seqüenciar*, que é chamado no Passo 6. Se H é acíclico, então o *Seqüenciar* sempre fornece uma solução de custo mínimo para I. Em caso contrário, a solução obtida não é necessariamente ótima.

2.4.4 Componentes Fonte

Agora, definimos o conceito de componentes fonte de um grafo dirigido.

Definição 2.5 *Dado um grafo dirigido $H = (A, D)$, um componente fonte é um componente fortemente conexo $C \subseteq A$ tal que não existe nenhum arco $(i, j) \in D$ onde $j \in C$ e $i \in A - C$.*

Dada a definição anterior, é um fato conhecido que qualquer grafo dirigido tem pelo menos um componente fonte [14].

A seguir, apresentamos algumas definições que são utilizadas associar os componentes fortemente conexos dos grafo G e H. Seja $((i, j), (i', j'))$ um arco de H. Por definição de H, existe uma rota onde o arco (i', j') de G é o sucessor imediato do arco (i, j). Logo, temos $i' = j$. Observe que o nó j funciona como ponto de articulação para o par de arcos (i, j) e (j, j'), em G. Neste caso, dizemos que o nó j de G *articula* o arco $((i, j), (j, j'))$ de H. Vale mencionar que o mesmo nó de G pode articular vários arcos de H. Dado um componente fortemente conexo T de H com $|T| > 1$, definimos um T' associado a T como sendo o subgrafo de G induzido pelos nós que articulam os arcos de T. Com isto, temos a seguinte proposição.

Proposição 2.6 *Dado um componente fortemente conexo T de H com $|T| > 1$, T' também é fortemente conexo.*

Prova. Sejam i e j dois nós quaisquer de T'. Basta provar que existe um caminho de i até j. Sejam também (a_i, a'_i) e (a_j, a'_j) dois arcos de T articulados respectivamente por i e j. Como T é fortemente conexo, existe um caminho P de a_i até a'_j em T. Neste caso, cada arco de P é articulado por um nó de T'. Em particular, o primeiro e o último arcos de P são respectivamente articulados pelos nós i e j. Além disso, para qualquer par de nós i' e j' que articulam arcos consecutivos de P, existe um arco (i', j') em T'. Portanto, os nós articulam os arcos de P definem um caminho (possivelmente contendo ciclos) de i para j em T'. □

2.4.5 O Procedimento Seqüenciar

Aqui, descrevemos o procedimento *Seqüenciar*, que é chamado pelo algoritmo BPA. Vale lembrar que este procedimento utiliza os seguintes dados de entrada:

1. um estado final válido para a rede dado por $p_q(b), \forall b \in L^1$;
2. uma rota válida $R(b)$ para cada batelada $b \in L^1$ consistente com o estado final dado;
3. um grafo de precedências H construído a partir das rotas dadas.

Para cada batelada $b \in L^1$, tanto a rota $R(b)$ quanto a posição final $p_q(b)$ recebidas podem ser alteradas pelo procedimento *Seqüenciar*. Por isto, denotamos respectivamente por $R^*(b)$ e $p^*_q(b)$ a rota e a posição final atribuídas originalmente a b, no Passo 4 do algoritmo BPA.

Vale mencionar que, se H é acíclico, então o procedimento *Seqüenciar* se comporta de forma idêntica ao procedimento utilizado na prova do Teorema 2.3. Lembramos que, nesta prova, selecionamos sucessivamente nós fonte do grafo H. Para cada nó a selecionado, realizamos as operações previstas para o duto a. Depois disto, removemos a de H e não voltamos a este duto. O procedimento *Seqüenciar* é uma generalização do procedimento anterior para grafos de precedência com ciclos. Ao invés de selecionar nós, selecionamos um componente fonte T do grafo H a cada iteração. Seja T' o subgrafo de G associado a T. O procedimento *Seqüenciar* fornece uma seqüência de operações PP para T' que leva esta subrede a uma estado com as seguintes características:

1. se uma batelada b tem $p_q^*(b) = (a, l) \in A^*$, para algum $a \in T'$, então b está contida em algum arco de T' (não necessariamente em a);

2. se uma batelada não protelável b tem um nó de T' como $p_q^*(b)$, então b está armazenada neste nó;

3. se uma batelada b não incluída nos ítens anteriores tem $R^*(b)$ contendo pelo menos um arco de T', então b está armazenada no nó final do último arco de $R^*(b)$ que pertence a T';

Chamamos este estado da subrede T' de *parcial*. Depois disto, T é removido de H e nenhum duto de T é movimentado novamente.

A Tabela 2.2 mostra um pseudocódigo para o procedimento *Seqüenciar*. Neste pseudocódigo, há duas chamadas ao procedimento *Ciclar*, que é descrito em seguida. Dado um componente fonte selecionado, o procedimento *Seqüenciar* testa se $|T| = 1$. Em caso positivo, são executadas todas as operações PP previstas para o único arco associado a T. Neste caso, o comportamento do procedimento é idêntico ao descrito na prova do Teorema 2.3. Por outro lado, se $|T| > 1$, então o procedimento obtém o subgrafo T' de G associado a T. Em seguida, ele transporta cada batelada contida em T para a posição correspondente ao estado parcial de T'. Isto é feito em duas etapas. Primeiro, todos os arcos de T' são preenchidos com bateladas proteláveis cujas posições finais são arcos de T'. Esta etapa é implementada pelo dois primeiros laços de "Para" do pseudocódigo. Na segunda etapa, algumas bateladas que restaram nos nós são transportadas para outros nós conforme o estado parcial de T'. Esta etapa corresponde ao terceiro laço de "Para" do pseudocódigo. Observe que o procedimento *Seqüenciar* utiliza, para cada batelada b, a rota $R^*(b)$ obtida no Passo 4 do BPA. No entanto, ressaltamos que a rota final $R(b)$ percorrida por cada batelada b não é necessariamente igual a $R^*(b)$. Isto ocorre porque o procedimento *Ciclar* não move apenas a batelada referida em sua chamada.

Agora, descrevemos o procedimento *Ciclar*. Dada uma batelada b^*, armazenada em um nó $l \neq i$, com $i, l \in T'$, este procedimento transporta b^* de l até i. Este transporte é realizado sem que nenhum batelada exceto b^* seja retirada de um nó. No entanto, as bateladas contidas nos dutos podem ser deslocadas. Primeiro, o procedimento *Ciclar* obtém um circuito C' de T' que contém os nós l e i. Este circuito é formado pela concatenação dos caminhos mínimos que conectam o nó l ao nó i e o nó i ao nó l, em T'. Tais caminhos sempre existem porque, pela Proposição 2.6, T' é fortemente conexo.

Procedimento Seqüenciar

Enquanto H não está vazio,
  $T \leftarrow$ um componente fonte de H;
  Se $|T| = 1$ **então**
    $a \leftarrow$ o arco de G associado a T;
    realizar as operações previstas para a;
  Senão
    $T' \leftarrow$ o subgrafo de G associado a T;
    Para cada $b \in F^1$ tal que
      $p_q^*(b) =$ uma posição de um duto $(i,j) \in T'$,
      Se b está armazenada num nó $l \neq i$ **então**
        Ciclar para transportar b de l até i, por T';
      Fim Se
    Fim Para
    Para cada $b \in F^1$ tal que
      $p_q^*(b) =$ uma posição de um duto $a \in T'$,
      inserir b em a;
    Fim Para
    Para cada batelada b tal que
      $R^*(b)$ contém pelo menos um arco de T',
      $i \leftarrow$ o nó final do último arco de $R^*(b)$ que está em T';
      Se b está armazenada num nó $l \neq i$ **então**
        Ciclar para transportar b de l até i, por T';
      Fim Se
    Fim Para
    remover T de H;
  Fim Se;
Fim Enquanto;
Para cada $b \in L^1$
  atualizar $R(b)$ e $p_q(b)$ segundo
    a seqüência de operações PP gerada;
Fim Para;

Tabela 2.2: Um pseudocódigo para o procedimento *Seqüenciar*

Procedimento Ciclar

$C' \leftarrow$ um circuito de T' que contém os nós l e i;
$K \leftarrow \{b \in L^1 | b$ está contido em um duto de $C'\}$;
Enquanto b^* não está armazenado em i,
  $b \leftarrow$ a única batelada de $K \cup \{b^*\}$ armazenada em um nó;
  $a \leftarrow$ a próxima aresta de C' a ser visitada por b;
  inserir b no duto a;
Fim Enquanto

Tabela 2.3: Um pseudocódigo para o procedimento *Ciclar*.

A tabela 2.3 um pseudocódigo para o procedimento *Ciclar*.

Seja $k = |C'|$ e $s' = \sum_{j=1}^{k} v(a_j)$. Provamos a seguinte proposição.

Proposição 2.7 *Se o circuito C' é dado, então o procedimento Ciclar executa em tempo $O(ks')$.*

Prova. Claramente, a cada k iterações do procedimento, a batelada b^* é deslocada em uma posição de duto. Como b^* precisa de no máximo $k+s'-2$ deslocamentos para alcançar o nó i, *Ciclar* executa em tempo $O(ks')$. □

2.4.6 Análise do Algoritmo BPA

Aqui, apresentamos uma análise do algoritmo BPA. Como este algoritmo tem uma variação polinomial para uma representação mais compacta da entrada, deixamos a análise de desempenho para a versão polinomial.

O seguinte teorema é uma conseqüência do próprio algoritmo.

Teorema 2.8 *Dada uma instância I do PTDSC, I é viável se e somente se ela satisfaz a condição dada pela Proposição 2.2.*

Prova. Pela própria Proposição 2.2, se I é viável, então a condição é satisfeita. Por outro lado, se esta condição é satisfeita, então o algoritmo BPA sempre obtém uma solução viável para I. □

Agora, analisamos a qualidade da solução obtida pelo BPA. Provamos o seguinte teorema.

Teorema 2.9 *Se o grafo de precedências H obtido pelo BPA é acíclico, então a solução construída pelo algoritmo para o PTDSC é ótima.*

Prova. Pela Proposition 2.4, o custo de uma solução ótima para I não é menor do que (2-1). Além disso, se o grafo de precedências H obtido pelo BPA é acíclico, então o procedimento *Seqüenciar* tem um comportamento idêntico ao procedimento dado pela prova do Teorema 2.3. Como este comportamento gera uma solução onde cada batelada b percorre a rota $R^*(b)$, o custo desta solução é igual a (2-1). Conseqüentemente, a solução obtida é ótima. □

Observe que, se o grafo G é acíclico, então qualquer grafo de precedências construído para esta rede também é acíclico. Por isso, o teorema anterior tem o seguinte corolário.

Corolário 2.10 *Se o grafo G é acíclico, então a solução construída pelo algoritmo para o PTDSC é ótima.*

2.4.7 Operações PP Estendidas

Nesta seção, definimos operações PP Estendidas (PP-E) para PTD. Estas operações não acrescentam nenhuma informação ao PTD. Elas apenas fornecem uma forma mais compacta de representar uma seqüência de operações PP semelhantes. Por isso, o resultado dado pelo Teorema 2.1 permanece válido.

Cada operação PP-E corresponde a uma seqüência de operações PP, onde bateladas da mesma ordem são inseridas em uma mesmo duto. Seja $a = (i, j)$ um arco de G cujo conteúdo em um dado estado t é representado pela lista $[b_1, b_2, \ldots, b_{v(a)}]$. Além disso, Seja $k \in L$ uma ordem tal que as bateladas associadas $b_l(k), b_{l+1}(k), \ldots, b_{l'}(k)$ estão armazenadas no nó i, neste estado, para $l \leq l'$. Uma operação PP-E consiste em inserir as bateladas $b_l(k), b_{l+1}(k), \ldots, b_{l'}(k)$ no duto a, nesta ordem. Denotamos esta operação por $(b_l(k) \ldots b_{l'}(k), a, t)$ (ou apenas $(b_l(k) \ldots b_{l'}(k), a)$). Observe que uma operação PP-E representa $l' - l + 1$ operações PP, utilizando um espaço $O(1)$.

Vale mencionar que, sem as operações PP-E, nem o PTD nem o PTDS pertencem a $\mathcal{NP}$ (considerando a forma mais compacta de representação das ordens). Para mostrar este fato, apresentamos uma instância I do PTDS cujo certificado de viabilidade precisa conter um número de operações PP que é exponencial em relação ao tamanho da entrada. Esta instância I tem $N = \{1, 2\}$, $A = \{(1, 2)\}$ e $L = \{k_1, k_2, k_3\}$, onde $v((1, 2)) = 1$, $u(k_1) = u(k_2) = 1$, $u(k_3) = 2^\beta$, $p_0(k_1) = ((1, 2), 1)$, $p_0(k_2) = p_0(k_3) = 1$ e $d_{k_1} = d_{k_2} = d_{k_3} = 2$. A única ordem protelável é k_2. Qualquer solução viável insere todas as bateladas de k_3 no único duto da rede e depois insere a batelada de k_2. Isto corresponde a $2^\beta + 1$ operações PP. Como I tem tamanho $O(\beta)$ ($u(k_3)$ é representado por β bits), o número de operações PP é exponencial em relação ao tamanho da entrada. Observe que este problema é corrigido pelas operações PP-E descritas anteriormente.

2.4.8 Algoritmo BPA Polinomial

Na subseção 2.4.3, apresentamos o algoritmo BPA para o PTDSC. Da forma como o BPA é descrito nesta seção, ele executa em tempo pseudopolinomial quando as ordens não são unitárias. Nesta seção, apresentamos o algoritmo BPA polinomial (BPA-P) como uma versão aprimorada do BPA. Seja $r = |L|$. O BPA-P executa em tempo $O(r^2 \log r + s^2(rn + \log s))$. Vale

lembrar que estado inicial da rede é dado no PTDSC. Como o estado inicial inclui as listas de bateladas que representam os conteúdos dos dutos, são necessários $\Omega(s)$ bits para representá-lo. Por isso, a complexidade do algoritmo BPA-P é polinomial em relação ao tamanho da entrada. Além disso, tanto o Teorema 2.9 quanto o Corolário 2.10 se aplicam ao algoritmo BPA-P.

Um pseudocódigo para o BPA-P pode ser obtido a partir do pseudocódigo do BPA (ver Tabela 2.1) aplicando-se as seguintes alterações:

1. inserir um passo adicional (Passo $1\frac{1}{2}$) depois do Passo 1;
2. alterar ligeiramente os Passos 2 e 3 para construir um grafo bipartido menor.

No Passo $1\frac{1}{2}$, o BPA-P pré-processa as bateladas de $L^1 - F^1$ que estão inicialmente armazenadas nos nós, gerando operações PP-E. Apresentamos uma pseudocódigo para este passo na Tabela 2.4. Neste pseudocódigo, para $k \in L$, denotamos por $p_0(k)$ (ao invés de $p_0(b_1(k))$) a posição inicial das bateladas de k.

Passo $1\frac{1}{2}$ do BPA-P
Para cada ordem $k \in L - F$ tal que $p_0(k) \in N$, $l \leftarrow u(k)$; $j \leftarrow 1$; **Enquanto** $j \leq \|S(p_0(k), d_k)\|$ e $l \geq 1$, $a \leftarrow$ o j-ésimo arco de $S(p_0(k), d_k)$; inserir $b_1(k), b_2(k), \ldots, b_l(k)$ no duto a; $l \leftarrow l - v(a)$; $j \leftarrow j + 1$; **Fim Enquanto**; **Fim Para**; $t \leftarrow$ o índice do estado atual; remover de L^1 toda b tal que $p_t(b) = d_b$; estado inicial $\leftarrow$ estado atual;

Tabela 2.4: Um pseudocódigo para o Passo $1\frac{1}{2}$ do algoritmo BPA-P

Vale mencionar que, para cada ordem não protelável k com $p_0(k) \in N$, o número de bateladas que não chegam aos seus nós de destino após a execução do Passo $1\frac{1}{2}$ não é maior que o número de posições de duto no caminho $S(p_0(k), d_k)$. Isto ocorre porque apenas as bateladas que ficam nos dutos deste caminho não chegam aos seus nós de destino. Por isto, podemos concluir que temos $|L^1 - F^1| = O(s|L - F|)$ após a execução do Passo $1\frac{1}{2}$.

No Passo 2, o BPA-P constrói um grafo bipartido B' menor do que o grafo B construído pelo BPA. Agora, definimos B', mostrando como

obtê-lo a partir de B. Seja um grafo bipartido B construído pelo BPA. Primeiro, para cada ordem protelável k, *encolher* todos os nós da primeira partição que correspondem a uma batelada de k em um mesmo nó $n(k)$. Após este encolhimento, qualquer arco que partia de um nó $n(b)$, sendo b uma batelada associada a k, passa a partir de $n(k)$. Além disso, atribuímos a cada nó da primeira partição um *excesso* igual ao número de nós que foram encolhidos nele. Com isto, observe que temos $|F|$ nós na primeira partição. Em seguida, encolher os nós $n(\cdot, 1), n(\cdot, 2), \ldots, n(\cdot, |F^1| - s)$ da segunda partição no mesmo nó $n(\cdot, \cdot)$. Atribuir a este nó um excesso igual a $-|F^1| + s$. As demandas dos outros nós da segunda partição são iguais a -1. Com isto, observe que temos $s + 1$ nós na segunda partição. Para quaisquer dois arcos com os mesmos nós inicial e final, remover um deles. Definimos B' como o grafo obtido após estas transformações. Apesar disto, a construção de B' pelo BPA-P deve ser direta, não passando pelo grafo B.

O grafo B' apresentado anteriormente (incluindo os custos associados aos arcos e os excessos associados aos nós) representa uma instância do problema de Fluxo de Custo Mínimo (FCM). Este problema, que é amplamente estudado na literatura [19, 18], consiste em atribuir fluxos não negativos a todos os arcos do grafo de forma que os balanços de fluxo nos nós sejam iguais aos seus respectivos excessos, minimizando a soma total dos fluxos ponderados pelos custos dos respectivos arcos. O problema ainda permite atribuir capacidades (limites superiores para os fluxos) aos arcos. No caso de B', todos os arcos têm capacidade ilimitada. Vale mencionar que, se todas as capacidades e excessos têm valores inteiros, então existem algoritmos polinomiais que fornecem soluções ótimas para o FCM onde todos os fluxos têm valores inteiros.

No Passo 3, o BPA encontra uma solução ótima para a instância do FCM representada por B'. A interpretação desta solução é similar à interpretação do emparelhamento de custo mínimo encontrado para o B pelo BPA. Se uma unidade de fluxo é atribuída a um arco $(n(k), n(a, l))$ de B', então atribuímos uma batelada arbitrária da ordem k à posição de duto (a, l). Neste caso, cuidamos para que uma mesma batelada não seja atribuída a duas posições de duto. Cada unidade de fluxo atribuída a um arco $(n(k), n(\cdot, \cdot))$ de B', corresponde a uma batelada da ordem k que não é inserida em nenhum duto.

2.4.9 Análise do Algoritmo BPA-P

Agora, apresentamos uma análise de desempenho do algoritmo BPA-P. Nesta análise, assumimos que todo duto em G pelo menos uma posição e que todo nó em g tem pelo menos um arco adjacente. Como conseqüência destas hipóteses, temos $s \geq m \geq n/2$.

Vale mencionar que o BPA-P não representa as bateladas de forma explícita. Para cada ordem k, ele só guarda o número de bateladas correspondentes em cada nó. Além disso, para cada batelada contida em um duto, o BPA-P guarda a ordem correspondente mas não identifica o índice da batelada. Isto ocorre porque não é necessário fazer distinção entre duas bateladas associadas à mesma ordem até o Passo 4, quando as rotas e posições finais são atribuídas às bateladas. Neste ponto, como o número de bateladas que restaram é polinomial, o BPA-P passa a guardar informações específicas de cada batelada. Neste caso, provamos o seguinte teorema.

Teorema 2.11 *O algoritmo BPA executa em tempo* $O(r^2 \log r + s^2(rn + \log s))$.

Prova. O Passo 1 pode ser executado em tempo $O(n^3)$, utilizando o algoritmo de Floyd-Warshall [1]. Além disso, utilizando os caminhos mínimos obtidos no Passo 1 e a representação resumida das bateladas descrita anteriormente, o tempo de execução do Passo $1\frac{1}{2}$ é da ordem do número de operações PP-E geradas (ver pseudocódigo da Tabela 2.4). Como uma operação PP-E é gerada para cada ordem não protelável k e para cada nó de um dado caminho mínimo, temos $O(rn)$ operações. Logo o Passo $1\frac{1}{2}$ executa em tempo $O(rn)$.

Como já foi discutido, o grafo B' construído na versão polinomial do Passo 2 tem $|F| = O(r)$ nós na primeira partição e $O(s)$ nós na segunda. Utilizando os caminhos mínimos obtidos no Passo 1 para atribuir custos aos arcos, podemos inserir cada arco no grafo em tempo $O(1)$. Logo, o tempo necessário para a construção de B' é da ordem do número de arcos inseridos, que é $O(rs)$. Com isto, a instância do FCM resolvida no Passo 3 tem $O(r + s)$ nós e $O(rs)$ arcos. Além disso, como B' é bipartido, é possível encontrar um caminho mínimo em tempo $O(1)$, o pois existe no máximo um caminho dirigido para cada par de nós, cada um com apenas um arco. Por isto, a instância correspondente do FCM pode ser resolvida em tempo $O(r^2 \log r + s^2 \log s)$, utilizando o algoritmo "Enhanced Capacity Scaling" [18]. Neste caso, o algoritmo tira proveito do fato de que as capacidades são todas ilimitadas.

Para construir o grafo de precedências H, o BPA-P verifica cada arco da rota de cada batelada que não foi removida no Passo $1\frac{1}{2}$ nem recebeu uma rota vazia. Como restaram $O(rs)$ bateladas não proteláveis após o Passo $1\frac{1}{2}$ e exatamente s bateladas proteláveis não receberam rotas vazias, o Passo 4 executa em tempo $O(rns)$. O fator n na expressão deste tempo é um limite superior para o número máximo de arcos em uma rota. Este limite é válido porque as rotas escolhidas são caminhos mínimos (acrescidos de um arco) e, por isso, não passam duas vezes pelo mesmo nó (exceto pelo arco acrescentado).

Agora, mostramos que o procedimento *Seqüenciar* admite uma implementação que roda em tempo $O(rns^2)$. Nesta implementação, o grafo G e o grafo de precedências H são ambos pré-processados da seguinte forma:

1. obter uma decomposição de H em componentes fortemente conexos;
2. para cada componente obtido T de H, encontrar o subgrafo T' associado em G;
3. para cada subgrafo T' obtido, computar o caminho mínimo que conecta cada par de nós;
4. para cada componente obtido T de H, computar o número de arcos que incidem em T.

O Item 1 roda em tempo $O(m + |D|) = O(m^2)$, utilizando um algoritmo baseado em busca em profundidade (DFS) [3]. Seja m_j o número de nós contidos no j-ésimo componente de H, para $j = 1, \ldots, h$. Neste caso, uma implementação imediata do Item 2 executa em tempo

$$O(m_1) + O(m_2) + \cdots + O(m_h) = O(m) \text{ time.}$$

Seja também n_j o número de nós contidos no j-ésimo subgrafo de G obtido no Item 3, para $j = 1, \ldots, h$. Neste caso, o Item 3 roda em tempo

$$O(n_1^3) + O(n_2^3) + \cdots + O(n_h^3) = O(n^3) \text{ time,}$$

utilizando o algoritmo de Floyd-Warshall [1]. Por último, o Item 4 é calculado em tempo $O(|D|) = O(m^2)$, pesquisando-se cada arco do grafo H.

Utilizando os resultados deste pré-processamento, o procedimento *Seqüenciar* pode encontrar e remover todos os componentes fontes de H, gastando um tempo total de $O(m + |D|) = O(m^2)$ com estas operações.

Para isto, é utilizado um método análogo ao algoritmo de ordenação topológica de grafos descrito em [20]. Também utilizando informações pré-processadas, o procedimento *Ciclar* pode construir cada circuito C' necessário (ver pseudocódigo da Tabela 2.3) em tempo $O(1)$. Sejam $k_{b,j}$ e $u_{b,j}$ respectivamente o número de nós e o volume total do circuito C' utilizado por este procedimento para transportar a batelada b no j-ésimo subgrafo de G obtido, para $j = 1, \ldots, h$. Pela Proposição 2.7, o procedimento *Ciclar* gasta um tempo total dado por

$$\sum_{b \in L^1} \sum_{j=1}^{h} O(k_{b,j} u_{b,j}).$$

Como cada circuito utilizado é composto por dois caminhos mínimos, ele só pode passar no máximo duas vezes por cada nó. Além disso, cada batelada só é transportada em no máximo um circuito de cada subgrafo de G. Por isso, temos

$$\sum_{b \in L^1} \sum_{j=1}^{h} O(k_{b,j} u_{b,j}) \leq \sum_{b \in L^1} \sum_{j=1}^{h} O(n u_{b,j}) \leq \sum_{b \in L^1} O(ns).$$

Como restam apenas $O(rs)$ bateladas no Passo 4 com rotas não vazias, o procedimento *Ciclar* gasta um tempo total $O(rns^2)$.

Portanto, o procedimento *Seqüenciar* roda em tempo $O(m^2) + O(m) + O(m + n^3) + O(mn) + O(mn) + O(rns^2) = O(rns^2)$.

Conseqüentemente, o algoritmo BPA roda em tempo $O(n^3) + O(rn) + O(rns) + O(r^2 \log r + s^2 \log s) + O(rns) + O(rns^2) = O(r^2 \log r + s^2(rn + \log s))$. □

2.5 Aproximabilidade do Makespan para o PTD Síncrono

Nesta seção, apresentamos limites inferior e superior para a aproximabilidade do PTDSM.

Primeiro, demonstramos que, para qualquer $\epsilon > 0$ fixo, o PTDSM não admite nenhum algoritmo $\eta^{1-\epsilon}$-aproximado a menos que $\mathcal{P} = \mathcal{NP}$, onde η é o número total de bits necessário para representar a instância. Neste caso, utilizamos as seguintes hipóteses:

1. o grafo G é acíclico e planar;

2. todas as ordens são unitárias;

3. o número total de bateladas é $O(s)$.

Chamamos o PTDSM acrescido destas hipóteses de PTDSM^1. Para uma dada instância J do PTDSM^1, denotamos por $|J|$ o número total de bits necessário para representar J. Observe que as hipóteses anteriores tornam o limite inferior mais abrangente, uma vez que ele também vale para versões mais restritas do problema. Além disso, a terceira hipótese torna dispensável a representação compacta das bateladas introduzida na subseção 2.4.7. Por isto, não usaremos esta representação.

Para demonstrar o limite inferior de aproximabilidade, utilizamos a seguinte abordagem. Inicialmente, consideramos uma certa classe de instâncias do PTDSM^1 e, para cada instância desta classe, duas operações PP π_1 e π_2 que são necessariamente executadas em qualquer solução viável. Para esta classe de instâncias, provamos que encontrar soluções viáveis que não executam π_1 antes de π_2 é um problema $\mathcal{NP}$-completo. Seja I uma instância desta classe. Construímos então uma instância I^α do PTDSM^1 *encadeando* α cópias de I. Este encadeamento é feito de tal forma que a operação π_2 da i-ésima cópia e a operação π_1 da $(i+1)$-ésima cópia são a mesma operação, para $i = 1, 2, \ldots, \alpha - 1$. Por isto, qualquer solução viável para I^α cujo *makespan* é menor do que α, não executa π_1 antes de π_2 em pelo menos uma das cópias de I. Também garantimos por construção que, se existe uma solução viável para I que não executa π_1 antes de π_2, então também existe uma solução viável para I^α cujo *makespan* é $O(|I|)$. Depois disto, basta atribuir um valor apropriado para α em função de $|I|$ para obter o limite inferior mencionado anteriormente.

Por último, mostramos que o algoritmo BPA pode fornecer uma solução m-aproximada para o PTDSM. Para isto, basta ajustar os pesos das bateladas e os coeficientes da função de custo de forma apropriada. Neste caso, a única hipótese utilizada é a de que o grafo G é acíclico. Vale mencionar que, embora o fator de aproximação obtido seja muito alto, o limite inferior demonstrado não permite obter um fator muito melhor do que este a menos que $\mathcal{P} = \mathcal{NP}$.

Ao longo desta seção, utilizamos a seguinte terminologia. Um dado duto $a \in A$ é dito *permitido* para uma dada batelada b quando existe um caminho dirigido em G de $p_0(b)$ até o nó inicial de a e outro do nó final de a até d_b. Observe que qualquer batelada b só pode ser inserida nos dutos permitidos para ela.

2.5.1
Instâncias Justas

Agora, definimos uma classe de instâncias do PTDSM[1] utilizada na demonstração do limite inferior de aproximabilidade deste problema. Chamamos as instâncias desta classe de *instâncias justas.*

Definição 2.12 *Uma instância I do PTDSM[1] é justa quando ela satisfaz as três condições a seguir:*

1. *para cada batelada $b \in L^1$, um único caminho dirigido em G que parte do nó de saída de $p_0(b)$ e termina no nó d_b;*
2. *para cada duto $a = (i, j) \in A$, existem exatamente $v(a)$ bateladas proteláveis associadas a este duto, sendo armazenadas no nó i, no estado inicial, e destinadas ao nó j;*
3. *não há outras bateladas proteláveis.*

Para $l = 1, 2, \ldots, v(a)$, denotamos por b_l^a a l-ésima batelada protelável associada ao duto a. Chamamos estas bateladas de *preenchedoras* do duto a.

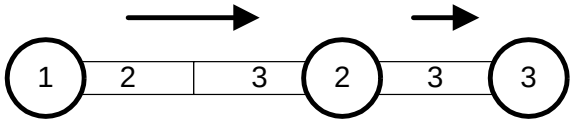

Figura 2.5: Um exemplo de instância justa do PTDSM[1]

Por exemplo, a Figura 2.5 representa uma instância justa do PTDSM[1]. Nesta figura, a representação dos nós, dos dutos e dos seus respectivos conteúdos segue um padrão semelhante ao da Figura 2.1.(a). No entanto, o número de cada nó aparece dentro do círculo correspondente (ao invés de estar ao lado dela). Além disso, cada batelada b contida em um duto é rotulada pelo número do nó de destino correspondente. As setas indicam os sentidos de fluxo nos respectivos dutos. Observe que, nas instâncias justas, o conjunto F^1 é unicamente definido em função dos dutos da rede. Por exemplo, na instância da Figura 2.5, temos $F^1 = \{b_1^{(1,2)}, b_2^{(1,2)}, b_1^{(2,3)}\}$, onde $p_0(b_1^{(1,2)}) = p_0(b_2^{(1,2)}) = 1$, $p_0(b_1^{(2,3)}) = 2$, $d_{b_1^{(1,2)}} = d_{b_2^{(1,2)}} = 2$ e $d_{b_1^{(2,3)}} = 3$. Por isto, este conjunto não é representado explicitamente nesta figura. Outras figuras que representam instâncias justas ao longo desta seção seguem o mesmo padrão, ou seja, omitindo qualquer indicação das preenchedoras dos dutos.

O seguinte lema apresenta uma propriedade das instâncias justas.

Lema 1 *Após a execução de qualquer solução viável para uma dada instância justa do PTDSM*^{1}*, cada duto* $a \in A$ *contém necessariamente todas as suas preenchedoras e apenas elas.*

Prova. Por definição, as preenchedoras de um dado duto $a = (i, j)$ são armazenadas no nó i, no estado inicial, e destinadas ao nó j. Como G é acíclico, a é o único duto permitido para estas bateladas. Como todas as bateladas são preenchedoras em uma instância justa, concluímos que cada duto a contém apenas as suas preenchedoras em qualquer estado final válido. Como cada duto a tem exatamente $v(a)$ preenchedoras, todas elas precisam estar contidas em a no estado final. □

Voltando ao exemplo da Figure 2.5, não é difícil provar a partir do lema anterior que a batelada contida na segunda posição do duto $(1, 2)$, no estado inicial, deve ser inserida no duto $(2, 3)$ antes da preenchedora $b_1^{(2,3)}$. O próximo teorema generaliza esta observação para qualquer instância justa.

Teorema 2.13 *Seja* Q *uma solução viável para uma dada instância justa* I *do PTDSM*^{1}*. Para cada batelada não protelável* b *e cada arco* a *do único caminho dirigido de* G *que parte do nó de saída de* $p_0(b)$ *e termina no nó* d_b*,* b *precisa ser inserido em* a *antes de qualquer preenchedora deste duto.*

Prova. Como b precisa chegar em d_b e a é um arco do único caminho dirigido que pode levar b até este nó, concluímos que b precisa ser inserido em a. Além disso, o conteúdo do duto a no estado final é dado pelas últimas $v(a)$ bateladas inseridas neste duto. Pelo Lema 1, estas bateladas são necessariamente as $v(a)$ preenchedoras a. Como G é acíclico, estas bateladas podem ser inseridas em a uma única vez. Logo, b precisa ser inserido em a antes delas. □

2.5.2 O Problema de Precedência em Dutos

Nesta subseção, provamos que, dada uma instância I do PTDSM^1 e duas operações PP π_1 and π_2, encontrar uma solução viável para I que não executa π_1 antes de π_2 é um problema $\mathcal{NP}$-completo.

Primeiro, definimos este problema de maneira formal. Dados

1. uma instância I do PTDSM^1,
2. duas bateladas $\bar{b}_1, \bar{b}_2 \in L$,

3. dois arcos $\bar{a}_1, \bar{a}_2 \in A$,

o *Problema de Precedência em Dutos* (PPD) consiste em encontrar um conjunto viável Q de operações PP contendo as operações $\pi_1 = (\bar{a}_1, \bar{b}_1, t_1)$ e $\pi_2 = (\bar{a}_2, \bar{b}_2, t_2)$, para qualquer $t_1 \geq t_2$. No Teorema 2.14, provamos que o PPD é um problema $\mathcal{NP}$-completo através de uma redução polinomial do Problema de Cobertura por Vértices (já definido na seção 2.2) para o PPD. Nesta prova, as instâncias do PPD baseadas em instâncias justas do PTDSM[1] também são chamadas de instâncias justas. Aqui, consideramos um caso particular do problema onde o grau (número de arestas adjacentes) de qualquer vértice em $\mathcal{G}$ é no máximo 3. Denotamos este problema por 3-PCV. Vale mencionar que o 3-PCV também é $\mathcal{NP}$-completo [7].

Teorema 2.14 *O PPD é $\mathcal{NP}$-completo.*

Prova. Esta prova é dividida em quatro etapas. Na primeira etapa, provamos que o PPD pertence a $\mathcal{NP}$. Na segunda etapa, apresentamos uma redução polinomial do 3-PCV para o PPD. Na terceira etapa, demonstramos que a existência de um certificado para uma instância do PPD implica na existência de um certificado para a instância correspondente do 3-PCV. Finalmente, na quarta etapa, demonstramos esta implicação no sentido contrário.

Etapa I: o PPD pertence a $\mathcal{NP}$. Seja I uma instância do PTDSM[1]. Como G não tem ciclos, qualquer solução viável Q para I insere cada batelada em no máximo n dutos. Como todas as ordens são unitárias, I tem exatamente r bateladas. Logo, Q não pode ter mais do que rn operações PP. Seja I' uma instância do PPD dada por I, duas bateladas $\bar{b}_1, \bar{b}_2 \in L$ e dois arcos $\bar{a}_1, \bar{a}_2 \in A$. Como qualquer certificado para I' é também uma solução viável para I, tal certificado não pode ter mais do que rn operações PP. Observe que qualquer certificado para I' com um número polinomial de operações PP pode ser facilmente verificado em tempo polinomial. Daí, concluímos que o PPD pertence a $\mathcal{NP}$.

Etapa II: uma redução polinomial do 3-PCV para o PPD. A seguir, apresentamos uma redução polinomial de uma instância do 3-PCV a uma instância justa I' do PPD. Denotamos os vértices de $\mathcal{G}$ por $s_1, s_2, \ldots, s_{|V|}$ e as arestas de $\mathcal{G}$ por $e_1, e_2, \ldots, e_{|E|}$. Para simplificar, a mesma the notação utilizada para os vértices e arestas de $\mathcal{G}$ são também utilizadas para denotar os nós correspondentes em G.

A Figura 2.6.(a) mostra um exemplo de grafo $\mathcal{G}$. Para $\mathcal{K} = 2$, a Figura 2.6.(b) representa a instância justa I' do PPD obtida a partir de

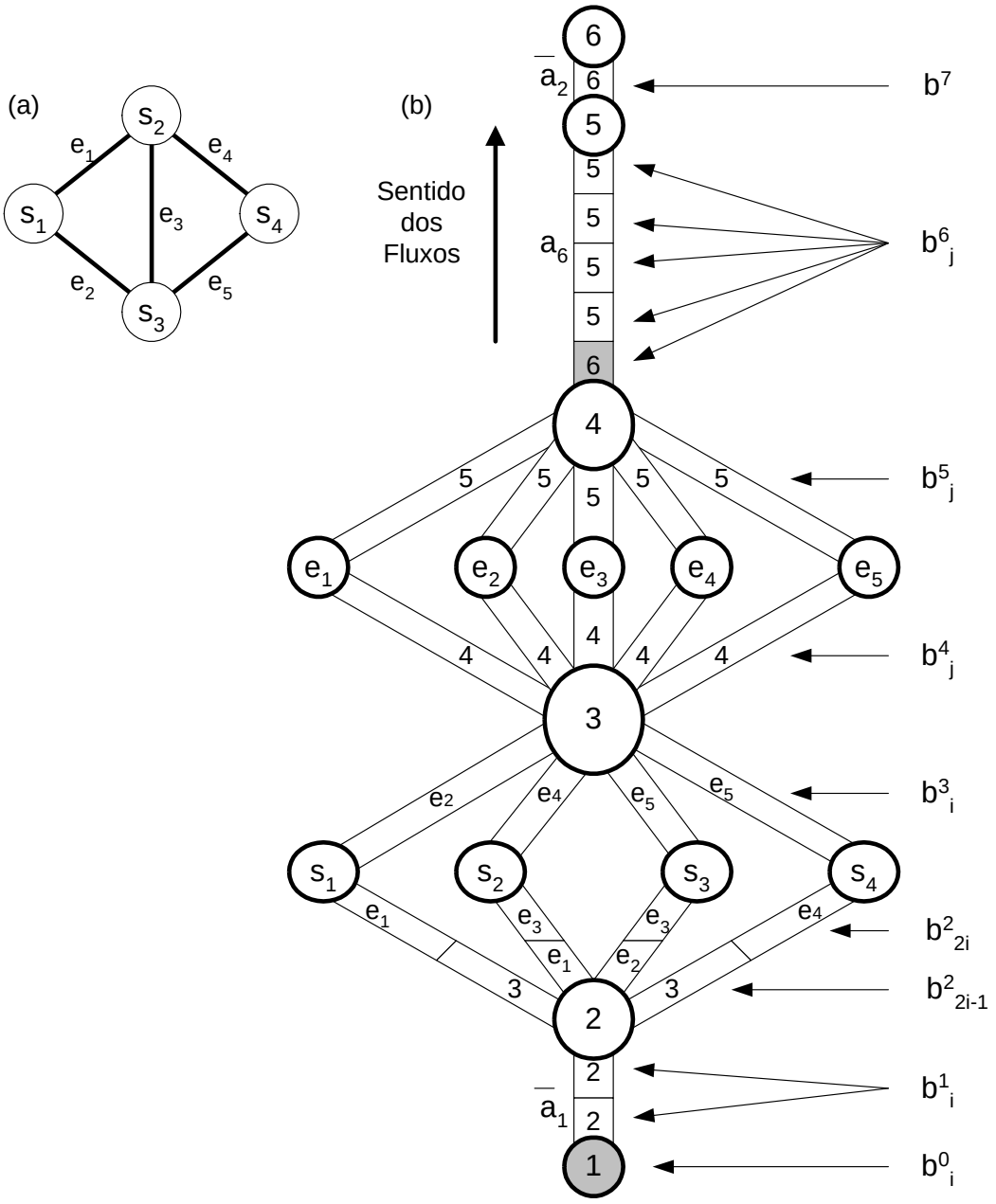

Figura 2.6: (a) Um grafo $\mathcal{G}$; (b) a instância justa I' do PPD obtida a partir de $\mathcal{G}$, para $\mathcal{K} = 2$.

$\mathcal{G}$. Mais adiante, detalharemos a construção de I'. Na Figura 2.6.(b), a representação dos nós, dos dutos e dos seus respectivos conteúdos segue o mesmo padrão da Figura 2.5. A notação utilizada para cada grupo de bateladas não proteláveis aparece próxima à borda direita da figura. As informações adicionais relativas ao PPD também são indicadas na figura. As posições iniciais das bateladas $\bar{b}_1$ e $\bar{b}_2$ aparecem pintadas de cinza. Os arcos $\bar{a}_1 = (1, 2)$, $\bar{a}_2 = (5, 6)$ e $a_6 = (4, 5)$ são indicados explicitamente. Os sentidos de fluxo de todos os dutos são definidos por uma única seta, do

lado esquerdo do duto $(4, 5)$. Observe que o grafo G associado à rede de dutos representada pela Figura 2.6.(b) é acíclico e planar.

Agora, seja uma instância qualquer do 3-PCV representada por $\mathcal{G}$ e $\mathcal{K}$ (não necessariamente a instância específica da Figura 2.6.(a)). Construímos uma instância justa I' do PPD a partir de $\mathcal{G}$ e $\mathcal{K}$ da seguinte forma:

1. criar o conjunto $N = \{1, 2, \ldots, 6\} \cup \{s_1, s_2, \ldots, s_{|V|}\} \cup \{e_1, e_2, \ldots, e_{|E|}\}$;
2. criar o conjunto A com os seguintes arcos:
   (a) $\bar{a}_1 = (1, 2)$ cujo conteúdo inicial é dado por $[b_1^1, b_2^1, \ldots, b_{|V|-\mathcal{K}}^1]$, onde $d_{b_j^1} = 2$, para $j = 1, \ldots, |V| - \mathcal{K}$;
   (b) $(2, s_i)$ e $(s_i, 3)$, para $i = 1, 2, \ldots, |V|$, onde:
      i. $(2, s_i)$ tem um conteúdo inicial dado por $[b_{2i-1}^2, b_{2i}^2]$;
      ii. $(s_i, 3)$ tem um conteúdo inicial dado por $[b_i^3]$;
      iii. atribuir como nó de destino de cada batelada do conjunto $\{b_{2i-1}^2, b_{2i}^2, b_i^3\}$ um nó de G associado a uma aresta de $\mathcal{G}$ adjacente a i;
      iv. se i tem um grau $\delta < 3$, então destinar as $3 - \delta$ bateladas que sobram do item anterior ao nó 3;
   (c) $(3, e_j)$, para $j = 1, 2, \ldots, |E|$, cujo conteúdo inicial é dado por $[b_j^4]$, onde $d_{b_j^4} = 4$;
   (d) $(e_j, 4)$, para $j = 1, 2, \ldots, |E|$, cujo conteúdo inicial é dado por $[b_j^5]$, onde $d_{b_j^5} = 5$;
   (e) $a_6 = (4, 5)$ cujo conteúdo inicial é dado por $[\bar{b}_2, b_2^6, b_3^6, \ldots, b_{|E|}^6]$, onde $d_{b_2^6} = d_{b_3^6} = \cdots = d_{b_{|E|}^6} = 5$ e $d_{\bar{b}_2} = 6$;
   (f) $\bar{a}_2 = (5, 6)$ cujo conteúdo inicial é dado por $[b^7]$, onde $d_{b^7} = 6$;
3. para cada duto $a \in A$, criar as respectivas preenchedoras;
4. para $i = 1, 2, \ldots, |V|$, criar a batelada não protelável b_i^0, armazenada no nó 1, no estado inicial, e destinada ao nó s_i;
5. criar a batelada não protelável $\bar{b}_1$, armazenada no nó 1, no estado inicial, e destinada ao nó 2.

Claramente, o grafo G obtido com esta construção é sempre acíclico e planar. Além disso, é fácil verificar que $|I'|$ é polinomial em relação a $|V|+|E|$. A seguir, utilizamos o Teorema 2.13 para demonstrar que qualquer certificado para I' induz a uma cobertura para $\mathcal{G}$ com até $\mathcal{K}$ vértices.

Etapa III: um certificado para o PPD implica num certificado para o 3-VCP. Seja Q um certificado para I'. Construiremos uma cobertura para $\mathcal{G}$ com até $\mathcal{K}$ vértices a partir da seguinte informação: quais dentre as bateladas $b_1^0, b_2^0, \ldots, b_{|V|}^0$ saem do duto $\bar{a}_1$ antes de $\bar{b}_2$ ser inserido do duto $\bar{a}_2$, de acordo com Q. Chamamos estas bateladas de *bateladas selecionadas*. Observe que exatamente $|V| - \mathcal{K}$ bateladas precisam permanecer no duto $\bar{a}_1$ preenchê-lo enquanto $\bar{b}_1$ e as preenchedoras de $\bar{a}_1$ ainda não foram inseridas neste duto. Além disso, Pelo Teorema 2.13, as preenchedoras de $\bar{a}_1$ não podem ser inseridas nele antes de $\bar{b}_1$. Como $\bar{b}_1$ não pode ser inserido neste duto antes que $\bar{b}_2$ seja inserido em $\bar{a}_2$, só restam no máximo $\mathcal{K}$ bateladas para deixar o duto $\bar{a}_1$ antes disto. Logo, temos no máximo $\mathcal{K}$ bateladas selecionadas. Chamamos de vértices selecionados os vértices associados aos nós de destino das bateladas selecionadas. Agora, falta provar que o conjunto de vértices selecionados é uma cobertura para $\mathcal{G}$.

Para acompanhar a demonstração a partir deste ponto, recomendamos consultar repetidamente a Figura 2.6, incluindo a notação que está próxima à sua margem direita. Primeiro, observe que pelo menos $|E|$ bateladas precisam ser inseridas no duto a_6 antes que $\bar{b}_2$ alcance o nó inicial de $\bar{a}_2$. Além disso, pelo Teorema 2.13, $b_1^5, b_2^5, \ldots, b_{|E|}^5$ precisam ser inseridas no duto a_6 antes das preenchedoras deste duto. Também é fato que a_6 não é permitido para nenhuma outra batelada. Logo, cada batelada b_j^5 precisa deixar o duto $(e_j, 4)$, para $j = 1, 2, \ldots, |E|$, antes que $\bar{b}_2$ alcance o nó inicial de $\bar{a}_2$. Para isto, pelo menos uma batelada precisam ser inseridas no duto $(e_j, 4)$. No entanto, pelo Teorema 2.13, b_j^4 precisa ser inserida neste duto antes da sua preenchedora. Como nenhuma outra batelada é permitida para este duto, concluímos que b_j^4 precisa deixar o duto $(3, e_j)$, para $j = 1, 2, \ldots, |E|$, antes que $\bar{b}_2$ alcance o nó inicial de $\bar{a}_2$. Também pelo Teorema 2.13, a preenchedora de $(3, e_j)$ não pode ser inserida neste duto antes de uma batelada não protelável b que esteja destinada ao nó e_j. Por construção, para cada $e = (s_i, s_j) \in E$, temos exatamente duas bateladas não proteláveis destinadas ao nó e. Uma delas chega ao nó 3 pelo duto $(s_i, 3)$ e a outra pelo duto $(s_j, 3)$. Como $(3, e)$ só é permitido para esta batelada e para a sua preenchedora, concluímos que, para todo $e = (i, j) \in E$, pelo menos uma batelada precisa ser inserida no duto $(s_i, 3)$ ou no duto $(s_j, 3)$ antes que $\bar{b}_2$ alcance o nó inicial de $\bar{a}_2$. Seja $\mathcal{C} \subset V$ o conjunto de todos os vértices s_i tais que pelo menos uma batelada é inserida em $(s_i, 3)$ antes que $\bar{b}_2$ alcance o nó inicial de $\bar{a}_2$. Pela discussão anterior, $\mathcal{C}$ é uma cobertura para $\mathcal{G}$. Além disso, pelo Teorema 2.13, as duas primeiras bateladas inseridas no duto $(s_i, 3)$ precisam antes sair do duto $(2, s_i)$, para $i = 1, 2, \ldots, |V|$. Pelo

mesmo teorema, b_i^0 deve ser a primeira batelada inserida no duto $(2, s_i)$. Conseqüentemente, para todo i tal que $s_i \in \mathcal{C}$, b_i^0 precisa sair do duto $\bar{a}_1$, antes que $\bar{b}_2$ alcance o nó inicial de $\bar{a}_2$. Logo, todo vértice de $\mathcal{C}$ é um vértice selecionado. Daí, concluímos que o conjunto de vértices selecionados também é uma cobertura para $\mathcal{G}$.

Etapa IV: um certificado para o 3-PCV implica num certificado para o PPD. Seja $\mathcal{C}$ uma cobertura para $\mathcal{G}$ com até $\mathcal{K}$ vértices, Se $\mathcal{C}$ tem menos do que $\mathcal{K}$ vértices, então uma cobertura com exatamente $\mathcal{K}$ vértices pode ser obtida inserindo-se vértices arbitrários de $\mathcal{G}$ em $\mathcal{C}$. Logo, podemos assumir sem perder generalidade que $\mathcal{C} = \{s_1, s_2, \ldots, s_{\mathcal{K}}\}$. Neste caso, construímos um certificado Q para I' da seguinte forma:

1. Para $t = 1, 2, \ldots, |V|$, criar a operação PP $(b_t^0, (1, 2), t - 1)$. Como $v((1, 2)) = |V| - \mathcal{K}$, b_t^0 está armazenado no nó 2, no estado $t + |V| - \mathcal{K}$, para $t = 1, 2, \ldots, \mathcal{K}$.

2. Para $t = 1, 2, \ldots, \mathcal{K}$, criar as operações PP $(b_t^0, (2, s_t), t + |V| - \mathcal{K})$, $(b_1^{(2,s_t)}, (2, s_t), t + |V| - \mathcal{K} + 1)$ e $(b_2^{(2,s_t)}, (2, s_t), t + |V| - \mathcal{K} + 2)$.

3. Para $t = 1, 2, \ldots, \mathcal{K}$, criar as operações PP $(b_{2t}^2, (s_t, 3), t + |V| - \mathcal{K} + 1)$, $(b_{2t-1}^2, (s_t, 3), t + |V| - \mathcal{K} + 2)$ e $(b_1^{(s_t,3)}, (s_t, 3), t + |V| - \mathcal{K} + 3)$.

4. Para $t = 1, 2, \ldots, \mathcal{K}$, criar as operações PP $(b_t^3, (3, d_{b_t^3}), t + |V| - \mathcal{K} + 2)$, $(b_{2t}^2, (3, d_{b_{2t}^2}), t + |V| + 2)$ e $(b_{2t-1}^2, (3, d_{b_{2t-1}^2}), t + |V| + \mathcal{K} + 2)$. Como $\mathcal{C}$ é uma cobertura para $\mathcal{G}$, pelo menos uma batelada é inserida no duto $(3, e_j)$, para $j = 1, 2, \ldots, |E|$. Logo, b_j^4 está armazenado no nó e_j, no estado $|V| + 2k + 3$.

5. Para $j = 1, 2, \ldots, |E|$, criar as operações PP $(b_j^4, (e_j, 4), |V| + 2k + 3)$ e $(b_1^{(e_j,4)}, (e_j, 4), |V| + 2k + 4)$.

6. Para $j = 1, 2, \ldots, |E|$, criar as operações PP $(b_j^5, a_6, |V| + 2k + 3 + j)$ e $(b_j^{a_6}, a_6, |E| + |V| + 2k + 3 + j)$.

7. Criar as operações PP $\pi_1 = (\bar{b}_1, \bar{a}_1, |E| + |V| + 2k + 4)$, $\pi_2 = (\bar{b}_2, \bar{a}_2, |E| + |V| + 2k + 4)$ e $(b_1^{\bar{a}_2}, \bar{a}_2, |E| + |V| + 2k + 5)$.

8. Para $t = 1, 2, \ldots, |V| - \mathcal{K}$, criar a operação PP $(b_t^{\bar{a}_1}, \bar{a}_1, t + |E| + |V| + 2k + 4)$. Como resultado desta operação, b_t^0 está armazenado no nó 2, no estado $t + |E| + |V| + \mathcal{K} + 5$, para $t = \mathcal{K} + 1, \mathcal{K} + 2, \ldots, |V|$.

9. Para $t = \mathcal{K} + 1, \mathcal{K} + 2, \ldots, |V|$, criar as operações PP $(b_t^0, (2, s_t), t + |E| + |V| + \mathcal{K} + 5)$, $(b_1^{(2,s_t)}, (2, s_t), t + |E| + |V| + \mathcal{K} + 6)$ e $(b_2^{(2,s_t)}, (2, s_t), t + |E| + |V| + \mathcal{K} + 7)$.

10. Para $t = \mathcal{K}+1, \mathcal{K}+2, \ldots, |V|$, criar as operações PP $(b^2_{2t}, (s_t, 3), t + |E|+|V|+\mathcal{K}+6)$, $(b^2_{2t-1}, (s_t, 3), t+|E|+|V|+\mathcal{K}+7)$ e $(b_1^{(s_t,3)}, (s_t, 3), t+ |E| + |V| + \mathcal{K} + 8)$.

11. Para $t = \mathcal{K}+1, \mathcal{K}+2, \ldots, |V|$, criar as operações PP $(b^3_t, (3, d_{b^3_t}), t + |E|+|V|+\mathcal{K}+7)$, $(b^2_{2t}, (3, d_{b^2_{2t}}), t+|E|+2|V|+7)$ e $(b^2_{2t-1}, (3, d_{b^2_{2t-1}}), t+ |E| + 3|V| - \mathcal{K} + 7)$.

12. Para $j = 1, 2, \ldots, |E|$, criar a operação PP $(b_1^{(3,e_j)}, (3, e_j), |E| + 4|V| - \mathcal{K} + 8)$.

Através de um minucioso acompanhamento dos ítens enumerados anteriormente, é possível verificar que Q é um certificado para I'. □

2.5.3
Limite Inferior de Aproximabilidade

Nesta subseção, provamos o limite inferior para a aproximabilidade do PTDSM^1 mencionado anteriormente. Para isto, utilizamos o Teorema 2.14 da seguinte forma. Dada uma instância do 3-PCV representada por $\mathcal{G}$ e $\mathcal{K}$ e a instância I do PTDSM^1 construída segundo a redução apresentada na prova do Teorema 2.14, construímos uma instância I^α do PTDSM^1, *encadeando* α cópias de I. Mais adiante, detalharemos este encadeamento. Depois disto, provamos que, se $\mathcal{G}$ tem uma cobertura com até $\mathcal{K}$ vértices, então I^α tem uma solução viável cujo *makespan* é dado por $t(I) = O(|I|)$. Em caso contrário, se tal cobertura não existe, então garantimos que I^α não tem nenhuma solução viável cujo *makespan* é menor do que α. Também demonstramos que $|I^\alpha| = O(\alpha|I|)$. Agora, assuma que estes resultados são verdadeiros Assuma também que existe um algoritmo $\mathcal{A}$ que é $|J|^{1-\epsilon}$-aproximado para qualquer instância J do PTDSM^1 e que roda em tempo $O(|J|^c)$ time, onde c é uma constante qualquer. Neste caso, para $\alpha = |I|^{(3/\epsilon)-1}$, concluímos que $\mathcal{A}$ obtém uma solução $O(|I|^{(3/\epsilon)-3})$-aproximada para I^α em tempo $O(|I|^{3c/\epsilon})$. Como $\alpha/t(I) = \Omega(|I|^{(3/\epsilon)-2})$, se $|I|$ é suficientemente grande, então o algoritmo $\mathcal{A}$ pode ser utilizado para decidir se $\mathcal{G}$ tem uma cobertura com até $\mathcal{K}$ vértices. Como $|I|$ é polinomial em relação a $|V|+|E|$ por construção, a existência do algoritmo $\mathcal{A}$ implicaria em $\mathcal{P} = \mathcal{NP}$.

Com isto, provamos o seguinte teorema.

Teorema 2.15 *Para qualquer $\epsilon > 0$ fixo, não existe nenhum algoritmo $\eta^{1-\epsilon}$-aproximado para o PTDSM^1 a menos que $\mathcal{P} = \mathcal{NP}$, onde η é o número total de bits necessário para representar a instância.*

Prova. Pela discussão anterior, basta apresentar um esquema para a construção de uma instância I^α com as seguintes propriedades:

1. se $\mathcal{G}$ tem uma cobertura com até $\mathcal{K}$ vértices, então I^α tem uma solução viável cujo *makespan* é $O(|I|)$;
2. se qualquer cobertura para $\mathcal{G}$ tem mais do que $\mathcal{K}$ vértices, então I^α não tem nenhuma solução viável cujo *makespan* é menor do que α;
3. $|I^\alpha| = O(\alpha|I|)$.

Sejam $I^{(1)}, I^{(2)}, \ldots, I^{(\alpha)}$ cópias de I. Construímos I^α, para $j = 1, 2, \ldots, \alpha - 1$, em cinco passos enumerados a seguir:

1. retirar de $I^{(j)}$
   (a) o nó 6,
   (b) o duto $\bar{a}_2$,
   (c) as bateladas b^7 e $b_1^{\bar{a}_2}$;
2. interligar as redes de dutos $I^{(j)}$ e $I^{(j+1)}$ através de união do nó 5 de $I^{(j)}$ com o nó 1 de $I^{(j+1)}$, resultando em um único nó;
3. considerar o duto $\bar{a}_1$ de $I^{(j+1)}$ como sendo também o duto $\bar{a}_2$ para $I^{(j)}$;
4. retirar a batelada $\bar{b}_1$ de $I^{(j+1)}$;
5. considerar a batelada $\bar{b}_2$ de $I^{(j)}$ como sendo também a batelada $\bar{b}_1$ para $I^{(j+1)}$;
6. destinar esta batelada ao nó 2 de $I^{(j+1)}$.

Claramente, I^α tem a propriedade do Item 3. Denotamos por $b^{(j)}$ e $a^{(j)}$, respectivamente, a batelada $\bar{b}_1$ e o duto $\bar{a}_1$ de $I^{(j)}$, para $j = 1, 2, \ldots, \alpha$. Além disso, a batelada $\bar{b}_2$ e o duto $\bar{a}_2$ de $I^{(\alpha)}$ são respectivamente denotados por $b^{(\alpha+1)}$ e $a^{(\alpha+1)}$. A Figura 2.7 representa os dutos que interligam as cópias $I^{(j-1)}$, $I^{(j)}$ e $I^{(j+1)}$, em I^α. Nesta figura, círculos representam nós, retângulos representam dutos e cada nuvem representa os nós e os dutos omitidos numa cópia de I. Além disso, as batelada $b^{(j)}$, $b^{(j+1)}$ e $b^{(j+2)}$ estão pintadas de cinza.

Observe que $b^{(j+1)}$ representa ao mesmo tempo a batelada $\bar{b}_2$ para $I^{(j)}$ e a batelada $\bar{b}_1$ para $I^{(j+1)}$, para $j = 1, 2, \ldots, \alpha - 1$. Da mesma forma, $a^{(j+1)}$ representa ao mesmo tempo o duto $\bar{a}_2$ para $I^{(j)}$ e o duto $\bar{a}_1$ para $I^{(j+1)}$. Além disso, $b^{(j+1)}$ é a única batelada que passa por dutos de $I^{(j)}$ e de $I^{(j+1)}$.

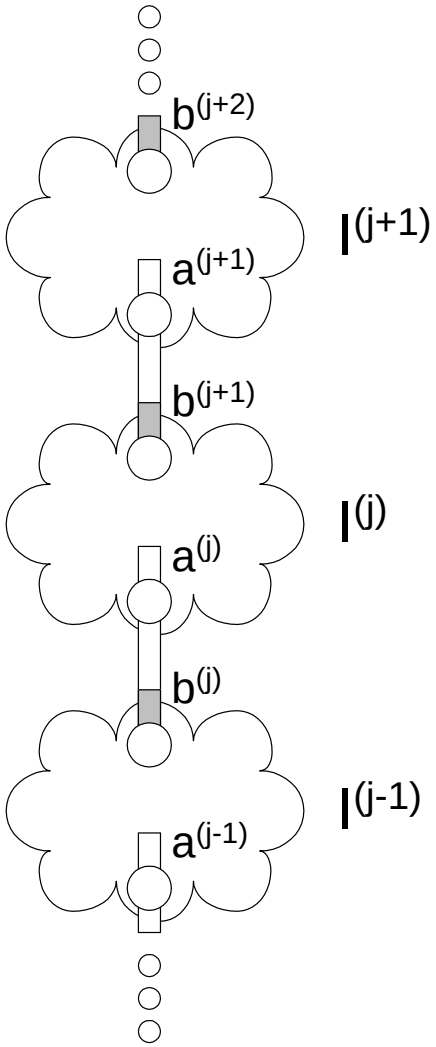

Figura 2.7: Dutos que interligam as subredes $I^{(j-1)}$, $I^{(j)}$ e $I^{(j+1)}$, em I^{α}.

Por isto, uma solução viável para I que não insere a batelada $\bar{b}_1$ no duto $\bar{a}_1$ antes de inserir $\bar{b}_2$ no duto $\bar{a}_2$, pode ser executada nas α cópias de I de I^{α} simultaneamente.

Agora, assuma que $\mathcal{G}$ tem uma cobertura com até $\mathcal{K}$ vértices. Neste caso, consideramos a solução apresentada na Etapa IV da prova do Teorema 2.14. Denotamos por $Q^{(j)}$ uma cópia desta solução executada na cópia $I^{(j)}$ de I antes da construção de I^{α}, para $j = 1, 2, \ldots, \alpha$. Obtemos uma solução viável Q para I^{α} da seguinte forma:

1. para $j = 1, 2, \ldots, \alpha$, retirar de $Q^{(j)}$ a operação PP $(b_1^{\bar{a}_2}, \bar{a}_2, |E|+|V|+2k+5)$;

2. para $j = 1, 2, \ldots, \alpha-1$, unir as operações PP $(\bar{b}_2, \bar{a}_2, |E|+|V|+2k+4)$ de $Q^{(j)}$ e $(\bar{b}_1, \bar{a}_1, |E|+|V|+2k+4)$ de $Q^{(j+1)}$, resultando em uma única operação PP $(b^{(j)}, a^{(j)}, |E|+|V|+2k+4)$;

3. $Q \leftarrow Q^{(1)} \cup Q^{(2)} \cup \cdots \cup Q^{(\alpha)}$.

Claramente, Q tem um *makespan* igual ao *makespan* de $Q^{(j)}$ que é $O(|I|)$. Logo, I^{α} tem a propriedade do Item 1.

Finalmente, se qualquer cobertura para $\mathcal{G}$ tem mais do que $\mathcal{K}$ vértices, então, pelo Teorema 2.14, qualquer solução viável para I insere a batelada $\bar{b}_1$ no duto $\bar{a}_1$ antes de inserir $\bar{b}_2$ no duto $\bar{a}_2$. Conseqüentemente, qualquer solução viável para I^{α} só insere $b^{(j+1)}$ no duto $a^{(j+1)}$ depois de inserir $b^{(j)}$ no duto $a^{(j)}$, para $j = 1, 2, \ldots, \alpha - 1$. Logo, estas soluções não podem ter *makespans* menores do que α. Isto mostra que I^{α} também tem a propriedade do Item 2 e completa esta prova. □

2.5.4
Limite Superior de Aproximabilidade

Aqui, mostramos como utilizar o algoritmo BPA-P, descrito na subseção 2.4.8 para obter soluções m-aproximadas as instâncias do PTDSM cujo grafo G é acíclico.

Vale lembrar que todas operações PP são executadas seqüencialmente nas soluções obtidas pelo BPA-P. Logo, o *makespan* de uma destas soluções é sempre igual ao número de operações PP correspondentes. Por isto, nossa estratégia é ajustar a função de custo de modo a minimizar o número de operações PP executadas. Para isto, basta fazer $\alpha_{a,0} = 1$ para todo duto $a \in A$, $\alpha_{a,l} = 0$ para toda posição de duto $(a, l) \in A^*$ e $w(k) = 1$ para toda ordem $k \in L$. Neste caso, seja I uma instância do PTDSM cujo o número mínimo de operações PP em qualquer solução viável é denotado por $\hat{q}$. Como cada duto de I pode executar no máximo uma operação PP por unidade de tempo, concluímos que o *makespan* de qualquer solução viável para I não pode ser menor do que $\hat{q}/m$. Como o algoritmo BPA-P, com a função de custo definida anteriormente, fornece uma solução viável para I cujo *makespan* é exatamente $\hat{q}$, esta solução é m-aproximada. Isto prova o seguinte teorema.

Teorema 2.16 *O PTDSM admite um algoritmo polinomial m-aproximado.*