

Heurística de Programação Matemática para o Problema de Fluxo Multiproduto Binário

Mayra Carvalho Albuquerque¹, Haroldo Gambini Santos²,
Moacir Felizardo França Filho¹

¹ Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Programa de Pós Graduação em Modelagem Matemática e Computacional
Av. Amazonas, 7675 - CEP: 30510-000, Belo Horizonte - MG, Brasil

² Universidade Federal de Ouro Preto (UFOP)
Departamento de Ciência da Computação
Campus Universitário - CEP 35.400-000, Ouro Preto- MG, Brasil

mayra.albuquerque@gmail.com, haroldo@iceb.ufop.br, franca@des.cefetmg.br

Resumo. *O presente artigo aborda o Problema de Fluxo Multiproduto Binário (PFMB). Este problema é definido por uma rede, em que um conjunto de produtos competem pelos arcos capacitados a um custo mínimo. Além disso, o fluxo de cada produto pode usar apenas um único caminho da origem ao destino. Propusemos uma Heurística de Programação Matemática para resolver o problema de fluxo multiproduto binário. O modelo é baseado numa programação matemática para o problema e utiliza uma decomposição por produtos, em que escolhem-se subconjuntos de produtos que devem ser trafegados na rede. Os resultados computacionais da Programação Matemática proposta obtiveram soluções melhores em relação ao método de Geração de Colunas encontrado na literatura. O algoritmo forneceu a solução inicial para a busca em um resolvedor matemático que comprovou a otimalidade para a maior parte das instâncias, enquanto que no método de Geração de Colunas se conhecia apenas o limite superior; além disso, em uma instância esse valor nem era conhecido.*

PALAVRAS-CHAVE: *Problema de Fluxo Multiproduto Binário, Programação Matemática, Otimização Combinatória*

Abstract. *This paper deals with the Binary Multicommodity Flow Problem (BMFP). This problem is defined over a network when a set of commodities compete for capacitated arcs at minimal cost. Furthermore, the flow of each commodity may use only one path from origin to destination. We propose a mathematical programming based heuristic to solve the binary multicommodity flow problem. The model is based on mathematical programming to problem and it uses a decomposition by commodities, in which subsets of commodities are chosen to be routed on the network. The computational results of the proposed mathematical programming obtained better solutions than a method of Column Generation found in literature. The algorithm provided the initial solution for search in a mathematical solver that proved the optimality for most of the instances, whereas Column Generation method known only to the upper bound, furthermore, in one instance this value was not known.*

KEYWORDS: *Binary Multicommodity Flow Problem, Mathematical Programming, Combinatorial Optimization*

1 INTRODUÇÃO

Os primeiros estudos do Problema de Fluxo Multiproduto (PFM) foram feitos por Ford e Fulkerson (1962); Hu (1963) na década de 60. O PFM consiste em trafegar diversos produtos que competem pelos arcos capacitados numa mesma rede a um custo mínimo. O PFM é considerado um problema de Otimização Combinatória e pertence à classe dos problemas NP-difícil (Garey e Johnson, 1979). Com grande destaque devido a suas inúmeras aplicações, principalmente na área de telecomunicações e transporte, o PFM também se destaca no grande campo de pesquisa em Programação Linear.

Algumas aplicações do PFM podem ser vistas em problemas de projeto de redes de telecomunicações (Frangioni e Gendron, 2009), roteamento e alocação de transportes (Babonneau e Vial, 2008), roteamento de frota de navios (Al-Khayyal e Hwang, 2007), otimização em redes de fibra óptica (Ozdaglar e Bertsekas, 2003), serviço de rede no transporte de carga (Crainic, 2000), dentre outros.

Os problemas de fluxo multiproduto são uma generalização dos conhecidos problemas de fluxo de único produto. Neste problema, os diferentes produtos coexistem na mesma rede, cada um com sua própria conservação de fluxo. Além disso, diferentes fluxos competem pelos recursos da rede, neste caso, a capacidade do arco.

Em relação a variável de decisão, três variações podem ser consideradas. Essas variações estão relacionadas às unidades do produto e ao fluxo dos mesmos na rede e caracterizam os seguintes problemas: problemas contínuos, problemas inteiros e problemas binários. Problema de fluxo multiproduto contínuo considera-se cada unidade dos produtos divisível e o fluxo do produto pode ser trafegado por diversos caminhos. Os problemas de fluxo multiproduto inteiro surgem quando o produto possui sua unidade indivisível e o fluxo do produto pode percorrer por diferentes caminhos, contudo o fluxo em cada um deles deve ser inteiro. Finalmente, no problema de fluxo multiproduto binário, ou não-bifurcado, o fluxo do produto deve ter apenas um caminho, não sendo possível se subdividir na rede durante o fluxo. Neste caso, cada produto deve seguir ao longo de um único caminho, tendo assim as unidades do produto indivisíveis. Neste trabalho será abordado o Problema de Fluxo Multiproduto Binário (PFMB).

Como o fluxo dos produtos deve ser feito utilizando apenas um caminho entre a origem e o destino, os produtos se tornam mais competitivos na rede em relação aos arcos capacitados. Segundo Larsson e Yuan (2004) a dificuldade prática de resolver os modelos PFM aumenta rapidamente em relação ao tamanho do problema, especialmente no que diz respeito ao número de produtos e tratar esses casos é computacionalmente difícil.

O artigo está estruturado da seguinte maneira: a Seção 2 apresenta os trabalhos relacionados e o objetivo; na Seção 3 é feita a caracterização do problema e descrita sua formulação matemática; a Seção 4 apresenta a programação matemática e metodologia para a resolução do PFMB; os resultados computacionais são apresentados na Seção 5; finalmente, na Seção 6, têm-se as conclusões e apresentam-se possíveis propostas para trabalhos futuros.

2 TRABALHOS RELACIONADOS E OBJETIVO

Segundo Castro (2003), os estudos sobre o Problema de Fluxo Multiproduto se concentram na resolução através de quatro classes de métodos: baseados no método simplex, métodos

de decomposição, métodos de aproximação e métodos de pontos interiores. A seguir alguns trabalhos que trataram o PFM através de métodos exatos para a sua resolução.

Em Barnhart et al. (2000) é apresentado um algoritmo *branch-and-price-and-cut* para o PFM, que utiliza geração de colunas para encontrar limites na árvore de *branch-and-bound*, e apresenta uma nova regra de ramificação, permitindo que sejam geradas novas colunas de forma eficaz. A adição de cortes diminuiu consideravelmente o número de nós procurados nos casos em que a solução ótima foi encontrada e diminuiu o *gap* de otimalidade para a maioria das outras instâncias. Além disso, concluíram que a redução só funciona para problemas em que o fluxo de produtos é grande em comparação com as capacidades dos arcos.

Hoesel et al. (2002) tratam o PFM não-bifurcado numa rede não-direcionada, de forma a encontrar desigualdades válidas fortes para os politopos do arcos capacitados e determinam condições em que essas restrições definem uma faceta. Concluem que as desigualdades reduzem substancialmente o número de nós necessários para uma abordagem *branch-and-cut*.

Chardaire e Lisser (2002); Castro (2003) utilizam algoritmos especializados em métodos de pontos interiores aplicado ao PFM. Chardaire e Lisser (2002) apresentam uma implementação de Dantzig-Wolfe baseada no método simplex especializado, e é desenvolvido o método de plano de corte de centro analítico ao PFM não-orientado. Castro (2003), por sua vez, faz uma revisão do estado-arte em resolver problemas multiproduto em grande escala e também faz comparações dos resolvidores baseados no método simplex com algoritmos de pontos-interiores. Conclui que os métodos de solução baseados no simplex, ponto-interior, ou métodos de decomposição podem ser mais adequados dependendo do tipo de problema tratado. E mais, se a precisão da solução não é importante, métodos de aproximação devem ser considerados.

Em Alvelos (2005) foi desenvolvido o algoritmo *branch-and-price* aplicado ao PFMB. Neste trabalho duas decomposições foram propostas: decomposição por caminho e decomposição da mochila, para o método de Geração de Colunas (GC). Essas formulações foram baseadas na decomposição de Dantzig-Wolfe e, cada uma depende do tipo de restrição que define o subproblema. Na decomposição por caminho o subproblema é definido pela restrição de conservação de fluxo e, na decomposição da mochila, a restrição de capacidade que definirá o subproblema para o método de Geração de Colunas.

O objetivo do trabalho é tratar o problema de fluxo multiproduto binário através de uma abordagem de programação inteira e comprovar sua eficiência em relação ao método de Geração de Colunas, proposto por Alvelos (2005).

3 CARACTERIZAÇÃO DO PROBLEMA

O fluxo multiproduto é uma generalização do problema de fluxo de único produto. O modelo da rede multiproduto é dado por meio de um grafo direcionado. Nesta rede, os arcos possuem uma capacidade restringindo a quantidade de produtos a trafegar por eles. Além disso, no Problema Fluxo Multiproduto, todo arco pertencente à rede tem um custo associado a cada produto. Assim, o objetivo é encontrar o fluxo desses produtos na rede, de forma a obter o custo mínimo.

Os produtos da rede devem trafegar dos seus respectivos nó-origem ao nó-destino, que por sua vez, são únicos para cada produto. Alguns nós da rede podem não estar associ-

ados a nenhum produto, isto é, não são nó-origem ou nó-destino, sendo então considerados nós de transbordo, utilizados apenas como passagem para o produto.

Portanto, o Problema de Fluxo Multiproduto consiste em determinar o fluxo dos produtos na rede de maneira que o custo seja o menor possível, respeitando a conservação de fluxo e a capacidade dos arcos. Na figura 1 é apresentada uma rede de fluxo multiproduto contendo 2 produtos, P_A e P_B , que devem trafegar na rede. O produto P_A é ofertado pelo nó 1 e o produto P_B pelo nó 5. Os nós de demanda para os produtos P_A e P_B são os nós 4 e 2, respectivamente. Cada aresta possui um limite de quantidade de produtos a trafegar por ela, além de um custo associado representados na figura.

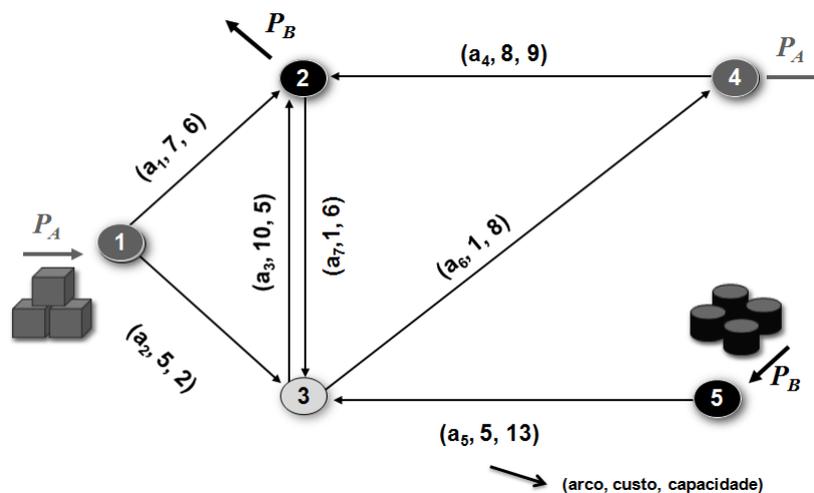


Figura 1. Exemplo de uma rede multiproduto.

O objetivo é determinar o fluxo para cada produto de forma a obter o menor custo, respeitando as capacidades dos arcos. Para esse exemplo, um fluxo viável é o fluxo do produto P_A pelos arcos: $a_1 - a_7 - a_6$ e o fluxo do produto P_B pelos arcos $a_5 - a_6 - a_4$.

3.1 Formulação Matemática

Considere uma rede direcionada representada por um grafo $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. O grafo \mathcal{G} é dado por um conjunto \mathcal{N} de n nós, um conjunto de a arcos direcionados, denotado por \mathcal{A} . Considere, ainda, um conjunto de h produtos, denotado por \mathcal{K} . Associado a cada arco $a \in \mathcal{A}$, tem-se um par de nós (i, j) e um custo que é diferenciado de acordo com o produto, ou seja, o custo dos produtos não é necessariamente o mesmo para todos.

A variável de decisão x_{ij}^k será 1 se o produto k faz uso do arco (i, j) no seu fluxo e, será 0 em caso contrário. Os arcos da rede possuem um custo de transporte para cada produto, representado por c_{ij}^k , denotando o custo unitário para que o produto k trafegue no arco (i, j) . A quantidade do produto k a ser transportada é denotada por r^k e a capacidade do arco (i, j) por u_{ij} . O parâmetro b_i^k define o nó i em nó-origem ou nó-destino para o produto k :

$$b_i^k = \begin{cases} 1, & \text{se o nó } i \text{ é origem para o produto } k; \\ -1, & \text{se o nó } i \text{ é destino para o produto } k; \\ 0, & \text{se o nó } i \text{ é um nó de transbordo para o produto } k. \end{cases}$$

A formulação nó-arco \mathcal{F}_1 para o Problema de Fluxo Multiproduto, apresentada em Alvelos (2005), é dada por:

$$\min \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k r^k x_{ij}^k \quad (1a)$$

$$\text{sujeito a:} \quad \sum_{j:(i,j) \in \mathcal{A}} x_{ij}^k - \sum_{j:(j,i) \in \mathcal{A}} x_{ji}^k = b_i^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \quad (1b)$$

$$\sum_{k \in \mathcal{K}} r^k x_{ij}^k \leq u_{ij}, \quad \forall (i,j) \in \mathcal{A} \quad (1c)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall k \in \mathcal{K}, \forall (i,j) \in \mathcal{A} \quad (1d)$$

Em (1a) tem-se a função objetivo a ser minimizada referente ao custo da rede, em que objetiva-se encontrar o fluxo da rede de menor custo. A restrição (1b) representa as restrições de conservação de fluxo, sendo responsável pela garantia do fluxo dos produtos pela rede, de forma que cada produto satisfaça o seu respectivo par origem-destino. A restrição (1c) refere-se à restrição de capacidade, limitando a quantidade de produtos no arco. Finalmente, em (1d) têm-se as restrições de integralidade do problema.

4 METODOLOGIA

A primeira estratégia adotada para resolver o PFMB foi através da relaxação do problema na restrição de conservação de fluxo. Posteriormente, foi estudada uma decomposição por produto, obtendo subconjuntos de produtos a serem resolvidos sequencialmente a fim de diminuir o tamanho do problema.

4.1 Relaxação do problema

A formulação nó-arco \mathcal{F}_1 foi alterada acrescentando uma variável de folga t_i^k na restrição de conservação de fluxo. Essa variável permite controlar os produtos que serão roteados. Considere α um parâmetro de valor suficientemente grande, tem-se a seguinte formulação *DPPFM*:

$$\min \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k r^k x_{ij}^k + \alpha t_i^k \quad (2a)$$

$$\text{sujeito a:} \quad \sum_{j:(i,j) \in \mathcal{A}} x_{ij}^k - \sum_{j:(j,i) \in \mathcal{A}} x_{ji}^k - t_i^k \leq b_i^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \quad (2b)$$

$$\sum_{k \in \mathcal{K}} r^k x_{ij}^k \leq u_{ij}, \quad \forall (i,j) \in \mathcal{A} \quad (2c)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall k \in \mathcal{K}, \forall (i,j) \in \mathcal{A} \quad (2d)$$

$$t_i^k \geq 0, \quad \forall i \in \mathcal{N}, \forall (i,j) \in \mathcal{A} \quad (2e)$$

Se o valor da variável t_i^k for diferente de zero para qualquer nó i , implica que o produto k não terá fluxo na rede. Com a adição dessa variável foi possível fazer uma decomposição que se baseia no fluxo de apenas alguns produtos da rede. Apresentada como uma decomposição por produtos, a proposta é tomar conjuntos menores de produtos e resolvê-los de forma sequencial. O tamanho desse conjunto está associado ao número de variáveis que apresentaram boa solução em instâncias menores testadas.

Sejam P o conjunto de produtos selecionados para serem alocados e p_{opt} o tamanho desse conjunto. O termo alocação de um produto, no presente artigo, significa determinar o seu fluxo na rede. O parâmetro p_{opt} representa o número máximo de produtos a serem alocados na rede e, é dado por:

$$p_{opt} = \left\lceil \frac{max_{var}}{|\mathcal{A}|} \right\rceil, \quad (3)$$

em que $|\mathcal{A}|$ é a quantidade de arcos existentes na rede e max_{var} é o número de variáveis que apresentaram boa solução em problemas menores.

Um modo de alocar os produtos é fixando as variáveis relacionadas a eles. Entende-se por fixar uma variável da seguinte forma: caso a variável relacionada ao produto k for fixada em 1, significa que este produto já foi alocado e, além disso, possui uma rota; caso a variável for fixada em 0, significa que o produto k será alocado na rede sem fluxo, ou seja, não será roteado e, caso libere a variável, o produto estará livre na rede, isto é, está disponível para ser alocado.

4.2 Pseudo-Código para a Decomposição por Produtos

Devido ao tamanho do problema e à sua complexidade, propõe-se dividir a instância em pequenos subproblemas e resolvê-los sequencialmente, integrando-se as soluções obtidas a fim de conseguir a solução completa. Fixar a solução para um subproblema e resolver os demais sequencialmente tem uma desvantagem pois pode-se limitar a solução final em níveis subótimos. Para isso, uma estratégia na escolha dos produtos para o subproblema foi levada em conta.

A escolha dos produtos para os subproblemas foi realizada de acordo com a demanda do produto. Devido ao limite da capacidade dos arcos, os produtos com maior demanda podem se tornar mais competitivos na rede. Assim, os subconjuntos de produtos para os subproblemas se deram por ordem de maior demanda por produto.

A heurística de programação matemática baseada em decomposição por produtos é apresentada no Algoritmo 1.

Na linha 2 inicializa o problema inteiro \mathcal{PI} utilizando a formulação $DPPFM$ com os dados do problema. Cria-se um conjunto R inicialmente vazio (linha 3), de tal forma que R conterá todos os produtos que já foram alocados na rede. Na linha 5 tem-se o conjunto P contendo os produtos escolhidos a serem alocados, para isso, os demais produtos, incluindo os que já possuem uma rota, terão suas variáveis fixadas a zero (linha 6). A solução será usada como solução incumbente e oferecerá um limite superior válido para a busca que se inicia. Assim, o problema será resolvido considerando apenas os produtos do conjunto P na rede (linha 7).

Na linha 8 as variáveis referentes aos produtos que foram alocados serão fixadas em 1. Como os produtos de P foram alocados então serão inseridos no conjunto R . Se o conjunto R não contém todos os elementos da rede, então escolhe-se novamente um outro conjunto P de produtos a serem roteados e segue novamente todo o procedimento. Esse processo ocorrerá até que todos os produtos sejam alocados.

É de se esperar que alguns produtos alocados na rede não possuam rota, ou seja, não terão fluxo na rede. Isso se deve ao fato de que, a medida em que produtos são roteados

Algoritmo 1: DPPFM

Entrada: Instância do PFMB
Saída: Vetor de solução x

```

1 início
2   Inicialize o programa inteiro  $\mathcal{PI}$  com os dados do problema, utilizando a formulação
    $DPPFM$ ;
3    $R \leftarrow \emptyset$ ;
4   enquanto  $|R| \leq |\mathcal{K}|$  faça
5      $P$  recebe os  $p_{opt}$  elementos  $k \in \mathcal{K} \setminus R$ ;
6      $x_{ij}^k = 0, \forall k \notin P$  e  $k \notin R$ ;
7     Resolva  $\mathcal{PI}$ ;
8     Fixe  $x_{ij}^k$  em 1,  $\forall k \in P, ij \in \mathcal{A} : x_{ij}^k = 1$ ;
9      $R = R \cup P$ ;
10  fim
11   $Q \leftarrow k : k$  foi roteado;
12  para cada  $k \in K$  faça
13    se  $k \in Q$  então
14       $t_i^k = 0$ ;
15    senão
16      Libere as variáveis  $t_i^k$ ;
17    fim
18  fim
19  Libere as variáveis  $x_{ij}^k$  tal que  $k \in K$ ;
20  Resolva  $\mathcal{PI}$ ;
21  Retorne  $x$ ;
22 fim
  
```

os arcos vão atingindo o limite da sua capacidade. Desta forma, podem haver casos que a variável de folga t_i^k seja ativa, representando que o produto não foi roteado, assim será alocado sem fluxo na rede. O conjunto Q reúne todos os produtos roteados (linha 11).

A partir daí, para todos os produtos pertencentes ao conjunto Q , ou seja, aqueles que possuem rota na rede, terão sua variável de folga t_i^k fixada em zero (linha 14). Isto implica que esses produtos terão prioridade na rota fazendo com que certamente sejam roteados. Enquanto que, todas as variáveis restantes do problema serão liberadas (linhas 16 - 19). Posteriormente na linha 20, segue a resolução do problema e, ao final da execução do procedimento, retorna-se a melhor solução inteira encontrada para o problema.

O algoritmo permite que as variáveis sejam fixadas, iterativamente, a cada resolução dos subproblemas. Essa fixação oferecerá uma solução inicial para o subproblema seguinte. Finalmente, com a resolução de todos os subproblemas, algumas variáveis de folga serão fixadas e o restante das variáveis ficarão livres. Isto disponibilizará ao algoritmo todo o espaço de busca, caracterizando-o em um método exato.

5 Resultados Computacionais

Experimentos computacionais foram realizados para avaliação do modelo. Os códigos foram implementados em C++, utilizando o compilador GCC 4 e a biblioteca do resolvidor de programação linear CPLEX 12. Os testes foram executados em um processador *Core(tm)2Duo 3.0Ghz* com 2GB de memória RAM, sob o sistema operacional *Windows 7*.

Foram utilizadas as instâncias contidas no pacote *carbin* gerado aleatoriamente pelo

Tabela 1. Características das instâncias.

Instância	Nós	Arcos	Produtos	Arcos/Nós	Produtos/Nós
<i>bx01 – bx04</i>	32	96	48	3	1.5
<i>bx05 – bx08</i>	32	320	48	10	1.5
<i>bx09 – bx12</i>	32	96	192	3	6
<i>bx13 – bx16</i>	32	320	192	10	6
<i>bx17 – bx20</i>	32	96	320	3	10
<i>bx21 – bx24</i>	32	320	320	10	10

algoritmo *GenMCF*, apresentado em Alvelos (2005). São 48 instâncias tais que, para os casos com um *s* em seu nome a razão média capacidade/demanda é de 1.5, para as instâncias com um *l*, essa relação é de 10. Na Tabela 1 as características destes 48 casos são apresentados (em substituição de *x* por *s* ou *l*).

Na Tabela 2 o resultado da programação matemática proposta é apresentada, além disso, apresenta uma comparação com os resultados obtidos na decomposição para o método de Geração de Colunas (GC) desenvolvido em Alvelos (2005). O tempo de execução dos experimentos foi medido utilizando-se o tempo de CPU em segundos. Admitiu-se um tempo máximo de execução de 3600 segundos.

Na Tabela 2, (*) significa que o algoritmo não encontrou uma solução inteira para a instância no tempo limite. A coluna Valor ótimo representa o valor da função objetivo da solução ótima encontrada pelo resolvidor. Na coluna GC tem-se os melhores resultados dentre as duas abordagens (decomposição da mochila e decomposição por caminhos) feitas pela Geração de Colunas em Alvelos (2005). Como o método de Geração de Colunas trabalha com o problema relaxado esses resultados podem ser considerados limitantes superiores. Enquanto que, na formulação *DPPFM* são limitantes inferiores.

Para medir a proximidade da solução obtida (*x*) em relação a solução ótima (*x**) foi feito o cálculo do *gap*, dado por:

$$gap = 100 \times \frac{|x - x^*|}{x^*}. \quad (4)$$

A formulação *DPPFM* quando comparada à formulação usando Geração de Colunas, mostra-se mais eficiente, uma vez que foi capaz de comprovar a otimalidade para várias instâncias ou obteve resultados mais próximos do valor ótimo. Além disso, a instância *bl10* na qual não conhecia o limite superior, o algoritmo *DPPFM* retorna um limite inferior para essa instância. As soluções das instâncias *bl10*, *bs11* e *bs22* foram melhores na nova abordagem, contudo, ainda não se comprova a otimalidade dessas soluções, pois o algoritmo foi interrompido por alcançar o tempo limite estabelecido.

6 Conclusões e trabalhos futuros

Este trabalho teve seu foco na resolução do problema de fluxo multiproduto binário, considerando indivisíveis as unidades dos produtos na rede.

Foi proposto um modelo de programação inteira para o PFMB, utilizando uma decomposição por produtos. O algoritmo foi testado em um conjunto de problemas-teste

Tabela 2. Resultados: GC Alvelos (2005) × DPPFM

Instância	Valor ótimo	gap(%)		Instância	Valor ótimo	gap(%)	
		GC	DPPFM			GC	DPPFM
<i>bl01</i>	1615947	0	0	<i>bs01</i>	1639862	0	0
<i>bl02</i>	1816947	0	0	<i>bs02</i>	1702368	0	0
<i>bl03</i>	17340	0	0	<i>bs03</i>	16828	0	0
<i>bl04</i>	21370	0	0	<i>bs04</i>	20213	0	0
<i>bl05</i>	474782	0	0	<i>bs05</i>	500870	0	0
<i>bl06</i>	411480	0	0	<i>bs06</i>	502151	0	0
<i>bl07</i>	5751	0	0	<i>bs07</i>	7223	0	0
<i>bl08</i>	5688	0	0	<i>bs08</i>	6471	0	0
<i>bl09</i>	6261671	1,10	0	<i>bs09</i>	6287195	0,60	0
<i>bl10</i>	6338276	*	0,29	<i>bs10</i>	7072735	0,90	0
<i>bl11</i>	69018	0,10	0	<i>bs11</i>	64747	2,16	0,01
<i>bl12</i>	65902	0,40	0	<i>bs12</i>	71483	1,10	0
<i>bl13</i>	3132695	0	0	<i>bs13</i>	3605397	0,30	0
<i>bl14</i>	2433011	0,10	0	<i>bs14</i>	2857891	1,52	0
<i>bl15</i>	34274	0	0	<i>bs15</i>	38533	0,10	0
<i>bl16</i>	28074	0,10	0	<i>bs16</i>	31124	0	0
<i>bl17</i>	13190922	0,30	0	<i>bs17</i>	11424499	0,91	0
<i>bl18</i>	10496120	0,30	0	<i>bs18</i>	10486796	0,30	0
<i>bl19</i>	109035	1,38	0	<i>bs19</i>	106142	0,20	0
<i>bl20</i>	111234	1,24	0	<i>bs20</i>	107712	0,50	0
<i>bl21</i>	5796809	0,56	0	<i>bs21</i>	5560585	0,73	0
<i>bl22</i>	4209266	0,30	0	<i>bs22</i>	4455161	2,13	0,18
<i>bl23</i>	56856	0,60	0	<i>bs23</i>	57548	0,10	0
<i>bl24</i>	47964	0,20	0	<i>bs24</i>	50980	0,30	0

da literatura e os resultados obtidos foram comparados com aqueles produzidos por uma abordagem com Geração de Colunas.

Com os resultados obtidos na resolução da formulação baseada em decomposição por produtos pode-se ver claramente que supera a abordagem por Geração de Colunas apresentada em Alvelos (2005). Essa comparação entre as abordagens de decomposição por Geração de Colunas e a decomposição por produtos para o PFM binário mostrou-se bastante surpreendente, visto que as abordagens feitas por geração de colunas são tidas como promissoras, o que não aconteceu para este problema em questão.

Como trabalhos futuros, propõe-se outra estratégia para a escolha dos produtos no conjunto P , além da adição de uma busca local, a fim de evitar que o algoritmo atinja o tempo limite, caso ocorrido em algumas instâncias.

Agradecimentos

Os autores agradecem ao CNPq e ao CEFET-MG pelo apoio ao desenvolvimento do presente trabalho.

Referências

- Al-Khayyal, F. e Hwang, S. (2007). Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part i: Applications and model. *European Journal of Operational Research*, v. 176, n. 1, p. 106–130.
- Alvelos, F. P. *Branch-And-Price and Multicommodity Flows*. PhD thesis, Universidade do Minho, (2005).
- Babonneau, Frederic e Vial, Jean-Philippe. (2008). An efficient method to compute traffic assignment problems with elastic demands. *Transportation Science*, v. 42, n. 2, p. 249–260.
- Barnhart, C.; Hane, C. A. e Vance, P. H. (2000). Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, v. 48, p. 318–326.
- Castro, J. (2003). Solving difficult multicommodity problems with a specialized interior-point algorithm. *Annals of Operations Research*, v. 124, n. 1-4, p. 35–48.
- Chardaire, P. e Lissier, A. (2002). Simplex and interior point specialized algorithms for solving nonoriented multicommodity flow problems. *Operations Research*, v. 50, p. 260–276.
- Crainic, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research*, v. 122, n. 2, p. 272–288.
- Ford, L.R. e Fulkerson, D.R. (1962). *Flows in Network*. Princeton University Press, Princeton.
- Frangioni, A. e Gendron, B. (2009). 0-1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics*, v. 157, p. 1229–1241.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.
- Hoesel, S.P.M.; Koster, A.M.C.A.; van deLeensel, R.L.M.J. e Savelsbergh, M.W.P. (2002). Polyhedral results for the edge capacity polytope. *Mathematical Programming*, v. 92, p. 335–358.
- Hu, T.C. (1963). Multicommodity network flows. *Operations Research*, v. 11, p. 344–360.
- Larsson, T. e Yuan, D. (2004). An augmented lagrangian algorithm for large scale multicommodity routing. *Computational Optimization and Applications*, v. 27, p. 187–215.
- Ozdaglar, A. E. e Bertsekas, D. P. (2003). Routing and wavelength assignment in optical networks. *IEEE/ACM Trans. Netw.*, v. 11, p. 259–272.