APLICAÇÃO DA METAHEURÍSTICA ITERATED LOCAL SEARCH À SOLUÇÃO DO PROBLEMA DE FLUXO MULTIPRODUTO INTEIRO

Fábio Pires Mourão, Sérgio Ricardo de Souza
Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675 – Nova Gameleira. Belo Horizonte/MG. CEP: 30510-000
fabiomourao@dppg.cefetmg.br, sergio@dppg.cefetmg.br

Carlos Alexandre Silva
Universidade de São Paulo
Av. do Trabalhador São-Carlense, 400 – Centro. São Carlos/SP. CEP: 13560-970
calex@calex.mat.br

RESUMO

O objetivo deste trabalho é analisar o comportamento de uma metaheurística aplicada ao Problema de Fluxo Multiproduto (PFM), sendo que foi tratado o problema binário, que é uma variação de PFM. O problema em tela pertence à classe dos problemas NP-difícies, possuindo grande aplicação de cunho econômico, como nas áreas de telecomunicação e sistemas de transporte. O uso de técnicas heurísticas se justifica pela quantidade de variáveis do problema. Mesmo sem garantir a otimalidade, técnicas heurísticas podem gerar bons resultados. Neste trabalho é estudado o Método *Iterated Local Search* (ILS), sendo o Método da Descida Randômica usado como heurística de busca local. Foram feitos testes em instâncias geradas aleatoriamente e os resultados comparados com resultados obtidos por métodos exatos. O método proposto encontrou boa solução em uma das instâncias testadas e razoáveis nas demais.

PALAVARAS CHAVE. Problema de Fluxo Multiproduto, Metaheurísticas, Iterated Local Search.

ABSTRACT

This object of this paper is to analyse the application of the Iterated Local Search (ILS) metaheuristic to the Integer Multicommodity Network-Flow Problem (IMFP). This problem has a great economic importance in areas like telecommunications and transport. As this problem belongs to the class of NP-hard problems, metaheuristics are used for solving it, in spite of optimality conditions gap of the found results. The local search method used in this work within ILS is the Random Descent Method. Computational tests are performed with random generated instances, available from cited bibliography, and the found results from the proposed method are compared with the ones obtained with exact methods.

KEYWORDS. Multicommodity Network-Flow Problem, Metaheuristics, Iterated Local Search.

1. Introdução

O objetivo deste trabalho é analisar o comportamento da Metaheurística *Iterated Local Search* (ILS) aplicada à solução do Problema de Fluxo Multiproduto (PFM). Trata-se de um dos mais importantes problemas em otimização em redes, sendo largamente estudado, desde sua proposição inicial, por Ford (1962) e Hu (1963). Seu grande número de aplicações e o forte impacto econômico associado justificam o grande interesse em seu estudo. Em especial, devem ser destacadas as aplicações em roteamento de trafégo na internet, como em Buriol (2003); escala de tripulações (Vaidyanathan (2007)); roteamento de aeronaves (Verweij (1997)); seqüenciamento de operações em refinarias de petróleo (Milidiu (2001)); seqüenciamento de cargas (Shan (1985)), dentre outras

O problema de fluxo multiproduto é modelado por uma rede, identificada por um grafo, de modo que os produtos trafegam pelos arcos capacitados da mesma, a um determinado custo, o qual pode depender somente do arco ou, também, estar associado aos produtos, em função dos arcos pelos quais eles trafegam, i.e., diferentes produtos podem trafegar em um mesmo arco a custos distintos. Os nós representam pontos de oferta e demanda dos produtos. Neste trabalho, para cada produto, existe um par de nós origem-destino especificado. O objetivo é, então, o de determinar o fluxo destes produtos na rede ao menor custo possível, de forma a atender às restrições de conservação de fluxo, de capacidade e de integralidade.

As restrições de conservação de fluxo têm a função de gerenciar o fluxo dos produtos na rede. As restrições de capacidade limitam a quantidade de produtos que passam pelos arcos, evitando que se trafegue por um arco uma quantidade maior que a por ele suportada. A restrição de integralidade garante que as variáveis envolvidas sejam inteiras e não negativas.

Assim, os vários produtos compartilham os arcos na rede e competem pela capacidade dos mesmos. De acordo com a natureza das variáveis de decisão, três variações deste problema podem ser consideradas: problemas contínuos, problemas inteiros e problemas binários. Problemas de fluxo multiproduto contínuos permitem que cada unidade dos produtos possa ser dividida. Problemas de fluxo multiproduto inteiro (PFMI) surgem quando o fluxo de um determinado produto, identificado pelo par origem-destino, deve utilizar somente um único caminho, sendo indivisíveis as unidades dos produtos. De acordo com Alvelos (2005), *"cada unidade de cada comodidade é indivisível, o fluxo de uma comodidade pode ser dividido por diferentes caminhos, mas o fluxo em cada um deles tem que ser inteiro"*. Já no problema de fluxo multiproduto binário, o fluxo de um determinado produto pode fazer uso apenas de uma rota, diferenciando-se, assim, do caso puramente inteiro.

Neste trabalho, é abordado o Problema de Fluxo Multiproduto Inteiro (PFMI). Este problema pertence à classe dos problemas NP-Completos, de modo que a aplicação de técnicas heurísticas para sua solução se justifica, dada sua complexidade combinatória, que praticamente torna inviável a utilização exclusiva de métodos exatos na solução, em especial quando são analisadas instâncias de maiores dimensões, ou seja, tendo maior número de arcos, nós e produtos envolvidos. Mesmo sem garantir a otimalidade da solução encontrada, as técnicas heurísticas podem gerar resultados sub-ótimos de boa qualidade. É proposta uma resolução utilizando-se a meta-heurística *Iterated Local Search*, tendo o método de descida randômica como método de busca local. Este método requer um menor tempo computacional que o método clássico da Descida, já que este verifica todo o espaço de busca. De acordo com Blum et al (2003), pode-se definir meta-heurística como "*estratégias de alto nível para a exploração do espaço de busca utilizando-se diferentes metodologias*". Justifica-se o uso de metaheurísticas pelo fato de apresentarem estratégias de fuga de ótimos locais. Em particular, o método ILS perturba a solução ótima corrente e aplica a esta nova solução um método de busca local.

Este artigo está organizando da seguinte forma: a próxima seção apresenta o Problema de Fluxo Multiproduto e suas definições; a seção 3 mostra a modelagem matemática do problema; a seção 4 introduz o procedimento de descida local utilizado; a seção 5 apresenta o método *Iterated Local Search* ; a seção 6, 7, 8 e 9 discutem a metodologia de solução utilizada, incluindo as estruturas de vizinhança e a função de avaliação; a geração de uma solução inicial para o problema, no âmbito da metodologia de solução via heurísticas adotada; e a aplicação do método ILS ao PFM; a seção 10 mostra os resultados computacionais alcançados através do uso da metodologia apresentada; a última seção apresenta as conclusões e propostas de trabalhos futuros.

2. O Problema de Fluxo Multiproduto

Os primeiros trabalhos relacionados ao PFM datam do início da década de 60, com contribuições iniciais de Ford (1962) e Hu (1963). Ao longo deste período, diversas aplicações foram realizadas e algumas destas, tendo conexão com o presente trabalho, serão discutidas a seguir.

Dentre os trabalhos, cita-se aqui, primeiramente, soluções através de métodos exatos. Ahuja (1993) apresenta a utilização de métodos simplex e técnicas de particionamento primal para resolver o PFM. Goffin (1996) apresenta um algoritmo que aborda técnicas de decomposição usando relaxação lagrangeana. São aplicados os métodos de plano de corte e centro analítico para resolver o problema de maximização de uma função dual não-diferencial. Park (2002) utiliza a técnica de geração de colunas para resolver duas classes de problemas de fluxo multiproduto inteiro. No primeiro problema, são dados o conjunto de produtos, com suas respectivas demandas e custo unitário em cada arco,. e o objetivo é selecionar subconjuntos de produtos para serem roteados, associando-se, a cada um deles, uma única rota, ligando o nó de origem ao nó destino, tendo, como finalidade, maximizar o benefício da distribuição. Para avaliar o benefício, é utilizado um valor constante para cada produto em cada rota. O segundo problema consiste na seleção de todos os produtos necessários, sendo o objetivo determinar o custo mínimo. Em Alvelos (2005), a abordagem é também por meio de geração de colunas para programação linear e inteira, sendo os testes computacionais feitos em instâncias geradas pelo *Generator Multicommodity Flow* (GenMCF), também apresentado em Alvelos (2005).

Goldberg (1998) utiliza um algoritmo heurístico para solucionar o problema de fluxo multiproduto. Quando comparado a métodos exatos, em particular usando o CPLEX 6.5, o algoritmo proposto perde em eficiência. Wille (2005) utilizam as metaheurísticas Busca Tabu e Algoritmo Genético (AG) para gerar soluções para um problema multiproduto topológico de redes IP. O objetivo é minimizar o custo da distribuição de pacotes pela rede, atendendo a um conjunto de restrições, dentre elas a qualidade de serviços. Os resultados computacionais obtidos mostram uma melhor eficiência da abordagem utilizando AG, gerando boas soluções para redes de tamanho médio, em comparação com o método Busca Tabu.

Souza e Silva (2007) apresentam uma metodologia baseada em heurísticas de busca local aplicadas ao problema de fluxo multiproduto, considerando a comodidade divisível. Em Silva e Souza (2008) é proposta uma hibridização envolvendo as Metaheurísticas *Simulated Annealing* e *Iterated Local Search*, no intuito de resolver o problema de fluxo multiproduto sob um espaço gerado por uma das restrições de sua formulação. Os resultados obtidos demonstram a boa eficiência da utilização das técnicas heurísticas.

3. Modelagem Matemática

Nesta seção, apresenta-se uma modelagem do PFM, considerando uma formulação baseada em arcos, tendo cada produto uma associação com um par de nós origem-destino.

Seja $G = (N, A)$ um grafo orientado, composto por um conjunto N de nós e um conjunto A de arestas, representando uma rede com p produtos, a arcos e n nós, sendo $|N| = n$ e $|A| = a$ as cardinalidades dos conjuntos de nós e arestas, respectivamente.

A formulação matemática do problema é posta, então, da seguinte maneira:

$$
\begin{array}{lll}
\min & z = cx & (1) \\
suj.\ a & Nx^i = b^i,\ \ i = 1, \ldots, p & (2) \\
 & I.x \le u & (3) \\
 & x \in \mathbb{Z}_+ & (4)
\end{array}
$$

Nesta formulação, (1) representa a função objetivo a ser minimizada; (2) representa o conjunto de restrições de conservação de fluxo; (3) representa o conjunto de restrições de capacidade e (4) representa a restrição de integralidade, sendo x a matriz de variáveis de decisão, na qual cada posição x^i_j representa o fluxo do produto i em um determinado arco j. Cada uma dessas variáveis deve ser inteira e não negativa. Além disso:

- $N \in Z^{n \times a}$ é a matriz de incidência nó-arco, tendo n linhas, associadas aos n nós, e a colunas, associadas aos a arcos;
- $x^i \in Z^a$ é o vetor de fluxo do produto i;
- $b^i \in Z^n$ é o vetor oferta/demanda para o produto i;
- $I \in Z^{a \times pa}$ é uma matriz, composta de p matrizes identidades $(a \times a)$;
- $x \in Z^{ap}$ é o vetor formado a partir dos p vetores de fluxo de cada produto, tendo, cada um, dimensão a;
- $u \in Z^a$ é o vetor capacidade dos arcos.

4. Método da Descida Randômica

O método da Descida Randômica é uma variação do método clássico de Descida, no qual não é necessário a realização exaustiva do processo de busca por todo o espaço de vizinhança. Neste método, qualquer solução de melhora encontrada é imediatamente adotada. Caso o vizinho não seja melhor do que a solução ótima corrente, que é a solução que gera o melhor valor da função de avaliação, outro vizinho é gerado. O procedimento é interrompido após um determinado número de iterações sem melhora, chamado neste trabalho de *iter_max*. Em Souza (2007), é apresentado o pseudocódigo do procedimento da Descida Randômica mostrado na Figura 2, na qual *f(.)* é a função de avaliação adotada e *N(.)* corresponde à vizinhança da solução *s*.

```
procedimento DescidaRandomica (f(.), N(.), IterMAx, s);
   Iter = 0;
   enquanto (Iter < IterMax) faça
      Iter = Iter+1;
      Selecione aleatoriamente s' ∈ N(s);
      se (f(s') < f(s)) então
         Iter = 0;
         s ← s';
      fim-se
   fim-enquanto
   Retorne s;
fim DescidaRandomica;
```

Figura 2 – Pseudocódigo do método da Descida Randômica.

5. Metaheurística *Iterated Local Search*

O método *Iterated Local Search* (ILS), proposto em Lourenço (2002), é classificado como uma metaheurística, pois apresenta uma estratégia de fuga de ótimos locais, através da idéia de melhorar um processo de busca local. Neste método, são aplicadas perturbações na solução ótima local e, em seguida, realiza-se outra busca local nesta solução gerada pela perturbação. Tal perturbação não pode ser muito fraca, pois neste caso a solução poderá não sair do ótimo local encontrado, tampouco muito forte, para evitar um reinício aleatório. A Figura 3 apresenta o pseudocódigo do método ILS.

```
procedimento ILS;
    s0 ← SolucaoInicial;
    s ← BuscaLocal(s0);
    Iter = 0;
    enquanto (Iter < IterMax) faça
        Iter = Iter+1;
        s' ← perturbacao(s, historico);
        s'' ← BuscaLocal(s');
        s ← CriterioAceitacao(s, s', s'');
    fim-enquanto
    Retorne s;
fim ILS;
```

Figura 3 – Pseudocódigo do Algoritmo ILS.

6. Metodologia

6.1 Representação de uma solução

Uma solução é representada por uma matriz x de dimensão $a \times p$, onde são representados os fluxos dos produtos em cada arco. Cada elemento dessa matriz é uma variável, onde o elemento $x_{i,j}$ representa o fluxo do produto j no arco i. Cada coluna j da matriz solução representa o fluxo do produto j.

6.2 Vizinhança de uma solução

Para explorar o espaço solução do problema, é aplicado um movimento que consiste em trocar o fluxo de um produto escolhido aleatoriamente, de acordo com a factibilidade da solução corrente, i.e., se a solução corrente violar alguma restrição de capacidade, então é selecionado um produto que passe por algum arco violado, sendo que, primeiro, um arco violado é selecionado aleatoriamente e, em seguida, um produto que passe por este arco é também selecionado aleatoriamente. No caso de solução corrente factível, é selecionado um produto qualquer e traçada uma nova rota para esse produto. Tal método de seleção do produto para o qual o fluxo será trocado implica em maior chance de obter factibilidade.

6.3 Função de avaliação

Uma solução x é avaliada com uma função f tal que:

$$f(x) = Tr.(c.x) + \alpha.\upsilon$$

sendo υ correspondente ao somatório das violações nos arcos e $\alpha > 0$ um parâmetro de penalização. Assim, para soluções factíveis, υ é igual a zero. Isso implica que o algoritmo minimiza a função objetivo sem penalização. Não foi utilizado nenhum método para tratar restrições de conservação de fluxo, pois a solução inicial gerada aleatoriamente não viola nenhuma restrição desse conjunto, como será mostrado nas próximas seções. As matrizes c e x já foram definidas anteriormente, sendo que $c.x$ possui dimensão $p \times p$. Considera-se o traço da matriz $c.x$ tendo em vista que o elemento (i,i) dessa matriz, com $1 \leq i \leq p$, representa a soma dos produtos dos elementos da linha i da matriz c (custos do produto i) pela coluna i da matriz x (fluxo do produto i na solução gerada).

7. Geração de uma solução inicial

Foi desenvolvido um algoritmo construtivo aleatório, baseado nas matrizes N e b. A partir da matriz b, foi construída uma matriz esparsa de oferta/demanda. Na matriz esparsa, cada elemento na primeira coluna representa um nó de oferta ou demanda, a segunda coluna apresenta os produtos e a terceira coluna representa a oferta/demanda do produto correspondente à linha. A dimensão da matriz esparsa é $2p \times 3$.

Seja M a matriz esparsa. Esta heurística construtiva aleatória primeiramente atribui o valor contido em $M(i,1)$ a uma variável auxiliar $aux1$. Em seguida, verifica quais são os possíveis arcos, analisando a linha $aux1$ da matriz N. Para isso, o algoritmo armazena, em um vetor *vaux*, todos os índices das colunas cujo valor presente na linha $aux1$ da matriz N seja 1, pois, neste caso, o nó armazenado em $aux1$ será um nó de oferta para o arco. Em seguida, seleciona-se aleatoriamente um elemento do vetor dos índices *vaux* e atribui-se esse valor a uma variável auxiliar $aux2$. Assim, na matriz solução x, o elemento $x(p_j, aux1)$ receberá o valor contido em $M(aux1, 3)$, onde p_j corresponde ao produto j para o qual está sendo traçado o fluxo. Antes de fazer essa atribuição na matriz solução, é verificado se o arco $aux2$ não será violado ao receber tal valor. Se houver violação da capacidade do arco $aux2$, então um novo elemento do vetor *vaux* será selecionado aleatoriamente, gerando um novo valor para $aux2$. Esta nova seleção aleatória em caso de violação do arco $aux2$ é feita até um determinado número de vezes, denominado, neste trabalho, de q_max. Se não houver a possibilidade de fazer uma atribuição a um arco tal que ele não ultrapasse sua capacidade, então a violação será aceita. Em seguida, o índice da linha onde há um valor correspondente a -1 na coluna $aux2$ da matriz N é atribuído a $aux1$ e o processo é feito novamente para decidir outro arco $aux2$ por onde o produto poderá passar. Se a variável $aux1$ receber um valor tal que, na linha $aux1$ da matriz N, não exista nenhum elemento igual a 1, significa que não existe nenhum arco $(aux1, i)$. Neste caso, se o valor de $aux1$ for diferente do nó destino, então o processo é reiniciado desde o nó origem e é traçada aleatoriamente uma nova rota. Se, no entanto, $aux1$ contiver o nó destino, então foi traçada uma rota para este produto. É importante lembrar que, a cada nó visitado, é verificado, pelo algoritmo, se o mesmo é o nó destino: em caso afirmativo, significa que a rota foi traçada e o processo é feito para o próximo produto.

Tal procedimento garante factibilidade quanto às restrições de conservação de fluxo, mas não garante factibilidade quanto às restrições de capacidade.

8. Método da Descida Randômica aplicada ao PFMI

Seja x uma solução do problema e x' pertencente a uma vizinhança de x, definida como um elemento do conjunto $N(x)$ gerado a partir de um movimento n realizado em x.

Um movimento n em x consiste em gerar um nova rota para um produto. A seleção do produto para o qual será gerada uma nova rota é feita da seguinte forma: se a soma das violações no arco não for nula, é selecionado aleatoriamente um dos arcos violados e em seguida é escolhido aleatoriamente um produto que trafega pelo arco selecionado, no caso da violação ser nula, então é escolhido aleatoriamente um produto qualquer. Esta forma de seleção do produto implica em maior chance da solução corrente se tornar factível. A nova rota também é traçada aleatoriamente, de forma que sejam mantidas as restrições de conservação de fluxo, analogamente ao que foi descrito na geração da população inicial.

O critério de parada do método consiste no número máximo de iterações sem melhora. Neste trabalho este parâmetro foi chamado de *IterMax,* como já mencionado anteriormente.

9. Método ILS aplicado ao PFM

Nesta seção são discutidas características específicas do método ILS aplicado ao Problema de Fluxo Multiproduto.

Uma perturbação consiste em trocar o fluxo de mais um produto, ou seja, na perturbação de nível 1, são trocados os fluxos de dois produtos, na perturbação de nível 2, são trocados os fluxos de três produtos, e assim por diante. O valor da perturbação é implementado após um número máximo de tentativas dentro de um mesmo nível. Este número máximo de tentativas num mesmo nível foi chamado neste trabalho de *vezesnível* e a quantidade máxima de perturbações de *ilsmax*.

Após alterar os fluxos na solução corrente, o método da Descida Randômica é executado e caso a solução retornada pelo método de busca seja melhor do que a solução ótima corrente, essa solução é trocada.

10. Resultados

O algoritmo proposto foi desenvolvido na linguagem C usando-se o compilador *Borland C++ Builder 5.0*. Para testá-lo, foram usadas as instâncias contidas no pacote *carbin*, geradas aleatoriamente pelo algoritmo *GenMCF*, apresentado em Alvelos (2005), que é um gerador aleatório de instâncias para o problema.

As instâncias foram classificadas em grupos, de acordo com a quantidade de nós, arcos e produtos. O *Grupo 1* corresponde às instâncias *bl01* até *bl04*, o *Grupo 2* de *bl05* até *bl08*, o *Grupo 3* de *bl09* até *bl12*, e assim por diante.

Os testes foram realizados em um computador Intel Celeron 1.83 GHz, com 1GB de memória RAM DDR 2, sob o sistema operacional Windows XP Professional Edition SP2.

A Tabela 1 apresenta os parâmetros utilizados pelo algoritmo proposto, para cada grupo de instâncias. Tais parâmetros foram obtidos empiricamente e de acordo com as dimensões das instâncias testadas, já que o aumento dessas dimensões poderia tornar o algoritmo inviável quanto ao tempo computacional.

Tabela 1 – Parâmetros do algoritmo proposto

Grupo	qmax	vezesnivel	ilsmax	itermax	alfa
01	70	200	10	400	80000
02	70	50	10	200	80000
03	70	50	10	200	50000
04	70	50	10	100	20000
05	70	60	10	200	80000
06	70	40	10	50	10000

Na Tabela 1, q_max é o valor máximo de seleção de um arco, no caso de violação, presente no método construtivo aleatório, *vezesnivel* é o número máximo de iterações sem melhora dentro de um nível de perturbação no método ILS, *ilsmax* é o número máximo de perturbações feitas pelo ILS e *alfa* é o parâmetro relacionado ao método de penalização escolhido para o tratamento das restrições de capacidade, i.e., *alfa* armazena o valor que é multiplicado pelo somatório das violações nos arcos da solução encontrada.

Nas Tabelas 2, 3, 4 e 5 são apresentados os resultados referentes aos grupos mencionados. Nessas tabelas, a coluna *N* indica a quantidade de nós, a coluna *A* representa a quantidade de arcos e a coluna *P*, a quantidade de produtos. A coluna *fo* apresenta a solução encontrada pelo algoritmo descrito, enquanto a coluna f^* contém a melhor solução (ótima ou não) encontrada por Alvelos (2005). O tempo total, em segundos, de execução do algoritmo também foi apresentado, além da violação total da solução encontrada, i.e., foi exibido o somatório das violações nos arcos. No caso de violação nula, significa que a solução retornada foi factível. Alvelos (2005) não encontrou solução factível para a instância bl10 no tempo máximo de execução de 1 hora. O algoritmo proposto por Alvelos (2005) não retorna nenhuma solução no caso de não encontrar solução factível dentro do tempo máximo de uma hora. Por isso, a instância bl10 está precedida de **, enquanto algumas instâncias estão precedidas de *, significando que o algoritmo proposto por Alvelos (2005) foi interrompido no tempo de execução de uma hora e retornou uma solução a qual não se tem garantias de otimalidade. As instâncias que não apresentam nenhuma marcação apresentam as suas soluções ótimas na coluna f^*. Os valores apresentados na coluna f^* foram obtidos por um método de geração de coluna, apresentado em Alvelos (2005).

Tabela 2 – Resultados para o Grupo 1.

Instância	N	A	P	fo	f*	Tempo (s)	Violação
bl01	32	96	48	1631696	1615947	393,469	0
bl02	32	96	48	1821368	1816947	527,266	0
bl03	32	96	48	17588	17340	581,843	0
bl04	32	96	48	19160	21340	536,141	10

Tabela 3 – Resultados para o Grupo 2.

Instância	N	A	P	fo	f*	Tempo (s)	Violação
bl05	32	320	48	504932	474782	579,969	0
bl06	32	320	48	470738	411480	362,594	0
bl07	32	320	48	5906	5751	711,781	0
bl08	32	320	48	6053	5688	175,703	0

Tabela 4 – Resultados para o Grupo 3.

Instância	N	A	P	fo	f*	Tempo (s)	Violação
bl09	32	96	192	6663281	6261671	1265,234	6
**bl10	32	96	192	6719166		1484,422	0
bl11	32	96	192	69658	69018	1879,984	0
bl12	32	96	192	68221	65902	1393,984	0

Tabela 5 – Resultados para o Grupo 4.

Instância	N	A	P	fo	f*	Tempo (s)	Violação
bl13	32	320	192	3453719	3132695	983,453	0
bl14	32	320	192	2951235	2433011	1664,516	0
bl15	32	320	192	37058	34274	1140,734	0
bl16	32	320	192	31828	28074	1578,391	0

Tabela 6 – Resultados para o Grupo 5.

Instância	N	A	P	fo	f*	Tempo (s)	Violação
bl17	32	96	320	13435957	13190922	1734,203	0
bl18	32	96	320	10901265	10496120	1348,703	0
*bl19	32	96	320	-	109556	-	-
*bl20	32	96	320	115931	111604	2943,359	0

Tabela 7 – Resultados para o Grupo 6.

Instância	N	A	P	fo	f*	Tempo (s)	Violação
*bl21	32	320	320	6589150	5800149	4601,500	1
bl22	32	320	320	5002865	4209266	5116,328	0
bl23	32	320	320	60534	56856	2965,516	0
bl24	32	320	320	55867	47964	4907,500	0

11. Conclusões

Este trabalho apresentou um algoritmo, baseado na Metaheurística *Iterated Local Search*, para resolver o Problema de Fluxo Multiproduto Inteiro.

O algoritmo proposto se mostrou adequado, obtendo resultados próximos aos ótimos e encontrando uma solução factível para a instância *bl10* em um tempo menor que o testado por Alvelos (2005). Como mencionado anteriormente, Alvelos (2005) não encontrou solução factível para essa instância em um tempo de 1 hora de execução do algoritmo proposto em seu trabalho.

As instâncias dos Grupos 1 e 2 possuem pequeno número de produtos, sendo consideradas de mais fácil solução via métodos exatos; porém as instâncias dos Grupos 3, 4, 5 e 6 são possuem maior número de produtos e, nestes casos, o comportamento do algoritmo proposto foi bom, com soluções próximas às soluções ótimas.

Como continuidade deste trabalho, propõe-se sua aplicação a instâncias de maior porte, envolvendo maior número de arcos, nós e produtos, além da melhoria do desempenho da metaheurística apresentada.

Referências

Ahuja, R. K., Magnanti, T. L. e Orlin, J. B, *Network flows*, Prentice Hall, New York, 1993.

Alvelos, F. P., Branch-and-price and multicommodity flows, *PhD thesis*, Departamento de Produção e Sistemas da Escola de Engenharia, Universidade do Minho, Portugal, 2005.

Blum, C. e Roli, A. (2003), Metaheuristics in Combinatorial Optimization. Overview and Conceptual Comparison, *ACM Computing Surveys*, 35, 268-308.

Buriol, L. S., Roteamento do tráfego na internet: algoritmos para projeto e operação de redes com protocolo OSPF, *Tese de Doutorado em Engenharia Elétrica*, FEEC/UNICAMP, 2003.

Ford, L. R. e Fulkerson, D. R., *Flows in Network*, Princeton University Press, 1962.

Goffin, J. L., Gondzio, J., Sarkissian, R. e Vial, J. P. (1996), Solving nonlinear multicommodity flow problems by the analytic center cutting plane method, *Mathematical Programming*, 76, 131–154.

Goldberg, A. V., Oldham, J. D., Plotkin, S. e Stein, C., An implementation of a combinatorial approximation algorithm for minimum-cost multicommodity flow, em Bixby, R. E., Boyd, E. A. e Rioz-Mercado, R. Z. (Eds), *Proceedings of the 6th International Integer Programming and Combinatorial Optimization Conference*, Lecture Notes in Computer Sciences, Springer-Verlag, Berlin, German, vol. 1412, 1998.

Hu, T.C. (1963), Multicommodity Network Flows, *Operations Research*, 11, 344–360.

Lourenço, H. R., Martin, O. e Stuetzle, T., Iterated Local Search, em Glover, F. e Kochenberger, G. (Eds), *Handbook of Metaheuristics*, Kluwer Academic Publishers, Norwell, MA, EUA, 7, 321-353, 2002.

Milidiu, R. L., Pessoa, A. A., Braconi, V., Laber, E. S. e Rey, P.A. (2001), Um algoritmo grasp para o problema de transporte de derivados de petróleo em oleodutos, *Anais do XXXIII Simpósio Brasileiro de Pesquisa Operacional*, 237–246.

Park, S., Kim, D. e Lee, K. (2002), An integer programming approach to the path selection problems, *Proceedings of INOC2005 - International Network Optimization Conference*, Faculdade de Ciências, Universidade de Lisboa, Lisboa, Portugal.

Shan, Y.S., A Dynamic Multicommodity Network Flow Model for Real Time Optimal Rail Freight Car Management, *PhD thesis*, Princeton University, 1985.

Silva, C.A. e Souza, S.R. (2008), Uma aplicação da meta-heurística híbrida simulated annealing-iterated local search ao problema de fluxo multiproduto sob o espaço capacitado, *TEMA – Tendências em Matemática Aplicada e Computacional* (aceito para publicação).

Souza, S.R. e Silva, C.A. (2007) Uma metodologia heurística de busca local para a resolução de problemas de fluxo multiproduto inteiro, em *Proceedings: of VII Congreso Chileno de Investigación Operativa*, Puerto Montt.

Vaidyanathan, B., Jha, K. C., Ahuja, R. K., (2007) Multicommodity network flow approach to the railroad crew-scheduling problem, IBM Journal Research. & Development, vol. 51, n. 3 / 4.

Verweij, B., Aardal, K. e Kant, G., On an integer multicommodity flow problem from the airplane industry, *Technical Report UU-CS-1997-38*, Department of Computer Science, Utrecht University, Utrecht, 1997.

Wille, E. C. G., Mellia, M., Leonardi, E. e Marsan, M. A. (2005),Topological design of survivable ip networks using metaheuristic approaches, em *Proceedings of Third Internation Workshop on QoS in Multiservice IP Networks*, 191–206, 2005.