

# Uma Aplicação da Meta-heurística Híbrida *Simulated Annealing-Iterated Local Search* ao Problema de Fluxo Multiproduto sob o Espaço Capacitado<sup>1</sup>

C.A. SILVA<sup>2</sup>, S.R. de SOUZA<sup>3</sup>, Programa de Pós-Graduação em Modelagem Matemática e Computacional, CEFET-MG, 30510-000 Belo Horizonte, MG, Brasil.

**Resumo.** Problemas de fluxo multiproduto possuem uma larga variedade de aplicações, sobretudo nas áreas de sistemas de transporte e telecomunicações. Devido à alta complexidade combinatorial dessa classe de problemas, métodos exatos apresentam dificuldades na tentativa de solucioná-los. Este fato motiva a utilização de técnicas heurísticas para o estudo do problema de fluxo multiproduto. Neste trabalho, é proposta uma aplicação das meta-heurísticas *Simulated Annealing* e *Iterated Local Search* para resolver o problema de fluxo multiproduto inteiro capacitado. O objetivo é determinar o fluxo dos produtos pelos arcos da rede ao menor custo possível, respeitando-se as restrições de conservação de fluxo e capacidade. O espaço de restrição de capacidade será utilizado como espaço de busca para a meta-heurística híbrida proposta, penalizando-se, através de uma relaxação, a restrição de conservação de fluxo. Os resultados mostram soluções obtidas em tempo computacional aceitável e de boa qualidade.

**Palavras-chave.** Problema de fluxo multiproduto inteiro, meta-heurísticas, restrição de capacidade.

## 1. Introdução

Os problemas de fluxo multiproduto (PFM) que retratam uma aplicação real, como controle de sistemas de transporte e gerenciamento de fluxo de informações, apresentam, em geral, uma grande quantidade de variáveis envolvidas. A eficácia dos métodos exatos na tentativa de solucioná-los é perdida devido à lenta convergência, associada à alta complexidade do problema tratado. Este fato motiva a introdução de heurísticas e meta-heurísticas na resolução dessa classe de problemas que, apesar de não garantirem a determinação de soluções ótimas globais, levam, por outro lado, a soluções sub-ótimas de boa qualidade em tempo computacional razoável.

Desde o início da década de 60, são relatados na literatura trabalhos que têm por tema problemas de fluxo multiproduto, em especial os trabalhos iniciais de [5] e

---

<sup>1</sup>Trabalho financiado pela CAPES e CEFET-MG

<sup>2</sup>carlos@dppg.cefetmg.br

<sup>3</sup>sergio@dppg.cefetmg.br

[6]. Essa classe de problemas possui uma extensa variedade de aplicações, incluindo o sequenciamento de suprimentos militares [11]; operações em refinarias de petróleo [9]; e o roteamento de tráfego na internet [3].

O problema é modelado através de uma rede identificada por um grafo, cujos nós representam pontos de oferta e demanda de determinados produtos, os quais trafegam pelos arcos da rede. Os arcos são capacitados e possuem um custo associado a cada produto. O problema de fluxo multiproduto surge quando vários produtos compartilham os arcos da rede e competem pela capacidade dos mesmos. O objetivo é determinar, ao menor custo, o fluxo dos produtos na rede, de maneira a atender basicamente a três conjuntos de restrições: restrição de conservação de fluxo; restrição de capacidade e restrição de integralidade.

A restrição de conservação de fluxo desempenha o papel de gerenciar o fluxo dos produtos pelos arcos da rede, que saem de um ponto de oferta e chegam em um ponto de demanda. Esta restrição dita a lei de fluxo válida pelos arcos da rede. A restrição de capacidade limita o fluxo dos produtos, de modo que em nenhum arco trafegue uma quantidade de produtos superior à capacidade suportada por ele. A restrição de integralidade garante que as variáveis de interesse sejam valores positivos e inteiros.

A utilização de meta-heurísticas para resolver problemas de alta complexidade combinatorial em espaços finitos e enumeráveis vem se difundindo ao longo dos anos e apresentando bons resultados na tentativa de solucionar esses problemas. Pode-se definir uma meta-heurística como “*uma ferramenta algorítmica geral, que pode ser aplicada a diferentes problemas de otimização, com modificações relativamente pequenas, para torná-las adaptável a um problema específico*” [2]. As meta-heurísticas, de forma contrária às heurísticas convencionais, como o método da descida, conseguem escapar de ótimos locais e, assim como os métodos de busca local tradicionais, diferenciam-se basicamente pelas seguintes características:

- critério de escolha de uma solução inicial;
- definição da vizinhança  $\mathcal{V}(s)$  de uma solução  $s$ ;
- critério de seleção de uma solução vizinha dentro de  $\mathcal{V}(s)$ ;
- critério de término.

Neste trabalho, é proposta uma aplicação de hibridação heurística, envolvendo as meta-heurísticas *Simulated Annealing* (SA) e *Iterated Local Search* (ILS), para resolver o problema de fluxo multiproduto inteiro (PFMI). Será adotada uma vizinhança inteiramente contida no espaço solução gerado pela restrição de capacidade. Desse modo, será necessário relaxar apenas a restrição de conservação de fluxo, já que as soluções produzidas dentro do espaço de vizinhança adotado jamais violarão a capacidade dos arcos. As instâncias utilizadas para os testes foram produzidas por um gerador aleatório (GenMCF), apresentado em [1].

## 2. Modelagem Matemática

Nesta seção, é apresentado o modelo de programação linear inteira para o problema de fluxo multiproduto, considerando uma formulação baseada em arcos, tendo os produtos envolvidos na rede identificados pelo par de nós origem-destino.

Seja  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$  um grafo direcionado representando uma rede com  $|\mathcal{P}|$  produtos,  $|\mathcal{N}|$  nós e  $|\mathcal{A}|$  arcos. A formulação matemática do problema é dada, a partir de [4], por:

$$\min \sum_{i=1}^{|\mathcal{P}|} (c^i)^T x^i \quad (2.1)$$

$$\text{s. a } \begin{bmatrix} N & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & N & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & N \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^{|\mathcal{P}|} \end{bmatrix} = \begin{bmatrix} b^1 \\ b^2 \\ \vdots \\ b^{|\mathcal{P}|} \end{bmatrix} \quad (2.2)$$

$$\sum_{i=1}^{|\mathcal{P}|} x_j^i \leq u^j, \quad j = 1, \dots, |\mathcal{A}| \quad (2.3)$$

$$x^i \in \mathbb{Z}_+^{|\mathcal{A}|}, \quad i = 1, \dots, |\mathcal{P}| \quad (2.4)$$

sendo

- $x^i \in \mathbb{Z}_+^{|\mathcal{A}|}$ : vetor fluxo para cada produto  $i$ ;
- $N \in \mathfrak{R}^{|\mathcal{N}| \times |\mathcal{A}|}$ : matriz de incidência nó-arco;
- $c^i \in \mathfrak{R}^{|\mathcal{A}|}$ : vetor de custo para cada produto  $i$ ;
- $u^j \in \mathfrak{R}^{|\mathcal{A}|}$ : vetor de capacidade para cada arco  $j$ ;
- $b^i \in \mathbb{Z}^{|\mathcal{N}|}$ : vetor de oferta/demanda para cada produto  $i$ .

A expressão (2.1) representa a função objetivo do problema, representada pelo custo do fluxo pela rede, que se deseja minimizar. A equação (2.2) define a regra de fluxo, denominada de restrição de conservação de fluxo. As desigualdades em (2.3) dizem respeito à limitação da quantidade de produtos que trafegam pelos arcos, sendo  $u^j$  a capacidade máxima do arco  $j$ . Por fim, a expressão (2.4) garante, às variáveis de interesse, valores inteiros e positivos.

## 3. Metodologia

### 3.1. Representação de uma solução

Uma solução  $s$  para o PFMI, representado por uma rede com  $|\mathcal{N}|$  nós,  $|\mathcal{A}|$  arcos e  $|\mathcal{P}|$  produtos, consiste em uma lista de vetores de fluxo, sendo que o valor contido na posição  $k$ , supondo  $k \neq \alpha |\mathcal{P}|$ ,  $\alpha \in \mathbb{N}$ , representa a quantidade do produto  $k \bmod |\mathcal{P}|$  no arco  $(k \div |\mathcal{P}|) + 1$ ; caso  $k$  seja múltiplo da quantidade de produtos, então o valor contido nessa posição representa a quantidade do produto  $|\mathcal{P}|$  no arco  $(k \div |\mathcal{P}|)$ . A Figura 1 apresenta a estrutura de uma solução  $s$ .

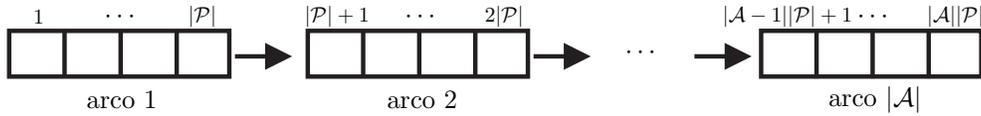


Figura 1: Estrutura de uma solução  $s$ .

### 3.2. Determinação da solução inicial

A solução inicial adotada para o PFMI abordado provém de uma técnica de busca local probabilística, proposta em [7], que, tendo como base a termodinâmica, simula o resfriamento de um conjunto de átomos aquecidos. Partindo de uma solução inicial qualquer, o procedimento gera aleatoriamente, em cada iteração, um único vizinho da solução corrente dentro do espaço gerado pela restrição de capacidade. Este procedimento é representado no algoritmo pela função *GerarVizinhoQualquer*( $\mathcal{V}(s)$ ).

Considere  $\Delta$  a variação de valor da função de avaliação  $f(\cdot)$  ao se mover para uma solução vizinha candidata  $s'$ , ou seja,  $\Delta = f(s') - f(s)$ . O método aceita o movimento e a solução vizinha passa a ser a nova solução corrente se  $\Delta < 0$ ; caso contrário ( $\Delta \geq 0$ ), a solução vizinha candidata poderá ser aceita, mas, neste caso, assumindo uma probabilidade  $e^{-\Delta/T}$ , no qual  $T$  representa um parâmetro do método, denominado de temperatura.

A temperatura  $T$  assume inicialmente um valor  $T_0$ , que, após um número fixo de iterações (até o sistema atingir o equilíbrio térmico em uma dada temperatura), é gradativamente diminuída por uma razão de resfriamento  $\beta$ , tal que  $T_k \leftarrow \beta T_{k-1}$ , sendo  $\beta \in (0, 1)$ . Os parâmetros de controle do procedimento são a razão de resfriamento  $\beta = 0.99$ , o número de iterações para cada temperatura (SMax = 100000) e a temperatura inicial  $T_0$  (autoadaptativa).

A temperatura inicial por simulação (ou autoadaptativa) é gerada a partir de um valor de temperatura baixo, contando-se a quantidade de vizinhos que são aceitos em um determinado número de iterações. Caso o número de vizinhos aceitos seja alto, retorna-se o valor da temperatura corrente como sendo o valor da temperatura inicial; se não, aumenta-se a temperatura e repete-se o processo. O pseudocódigo do método é descrito na Figura 2.

### 3.3. Função de avaliação

A função de avaliação consiste na função objetivo do problema, representada pela expressão (2.1), acrescida das penalizações oriundas das restrições (2.2) e (2.3). Pela proposta deste artigo, a solução corrente percorre o espaço solução gerado pela restrição de capacidade (2.3), não havendo, então, necessidade de se praticar uma relaxação sobre a mesma. No entanto, o fluxo não é conservado, devido às perturbações provocadas pela meta-heurística ILS. É promovida uma relaxação sobre a restrição de conservação de fluxo, dada por (2.2), afim de equilibrar a distribuição dos produtos entre os centros de oferta e demanda da rede. A função de avaliação

```

procedimento SimulatedAnnealing
1  $s^* \leftarrow s$ ; // melhor solução obtida;
2  $T \leftarrow T_0$ ; // Temperatura Inicial;
3  $IterT \leftarrow 0$  // Iterações da temperatura T;
4  $\epsilon \leftarrow 0.01$  // Temperatura próxima a 0;
4 enquanto ( $T > \epsilon$ ) faça
5 enquanto ( $IterT < S\text{Amax}$ ) faça
6    $IterT \leftarrow IterT + 1$ ;
7    $s' \leftarrow \text{GerarVizinhoQualquer}(\mathcal{V}(s))$ ;
8    $\Delta = f(s') - f(s)$ ;
9   se ( $\Delta < 0$ ) então
10     $s \leftarrow s'$ ;
11    se ( $f(s') < f(s^*)$ ) então
12      $s^* \leftarrow s'$ ;
13  senão
14   Tome  $x \in [0, 1]$ ;
15   se ( $x < e^{-\Delta/T}$ ) então  $s \leftarrow s'$ 
16 fim-se;
17 fim-enquanto;
18  $T \leftarrow \beta T$ ;
19  $IterT \leftarrow 0$ ;
20 fim-enquanto;
21  $s \leftarrow s^*$ ;
22 Retorne  $s^*$ ;
fim SimulatedAnnealing.

```

Figura 2: Pseudocódigo da meta-heurística *Simulated Annealing*.

é definida, então, pela expressão (3.1):

$$f(x) = \left( \sum_{i=1}^{|\mathcal{P}|} (c^i)^T x^i \right) - \alpha \max \left\{ 0, \max_{i=1, \dots, |\mathcal{P}|} \{Nx^i - b^i\} \right\}. \quad (3.1)$$

O primeiro termo desta expressão representa a própria função objetivo do PFMI, segundo o modelo apresentado na seção 2. O segundo termo se constitui em um termo de penalização do excesso de fluxo nos nós de oferta/demanda da rede. O parâmetro  $\alpha > 0$  é o peso atribuído à relaxação feita sobre a restrição de conservação de fluxo (2.2). Ele é ajustado conforme as instâncias utilizadas, equivalendo ao somatório dos custos dos produtos pelos arcos da rede, ou seja,  $\alpha = \sum_{i=1}^{|\mathcal{P}|} c^i$ .

Este segundo termo, portanto, caso detecte um excesso de fluxo no arco (situação que corresponde à infactibilidade da solução corrente, pois viola a restrição de conservação de fluxo), assumirá valor positivo, penalizando, em consequência, esta parcela da função de avaliação; caso, por outro lado, não detecte excesso de fluxo no arco (situação que corresponde à factibilidade da solução corrente), assumirá valor nulo. Desta maneira, a formulação do problema deixa de ser do tipo

ODE (origem-destino específico), e passa a admitir mais de um centro de oferta ou demanda por produto, fazendo com que alguns pontos de transbordo tornem-se pontos de estocagem.

### 3.4. Estruturas de vizinhança

O conjunto solução para o PFMI tratado neste artigo pode ser visto como a interseção de dois subconjuntos gerados pelo espaço solução das restrições (2.2) e (2.3). O espaço correspondente à restrição de capacidade (2.3) é representado por um subconjunto **A**, e o espaço referente à restrição de conservação de fluxo (2.2) é representado por um subconjunto **B**. O espaço de busca das soluções correntes, obtidas pelo método, faz uso apenas do subconjunto **A**.

Para exploração do espaço de soluções do PFMI, foi utilizado o movimento de troca entre fluxos de mesmos arcos. Esse movimento de troca corresponde a permutar o fluxo de um determinado produto (sorteado aleatoriamente) com o fluxo de outro (também escolhido de forma aleatória), sendo ambos no mesmo arco. Desta maneira, as novas soluções geradas permanecerão no espaço de solução gerado pela restrição de capacidade, já que as soluções geradas anteriormente não violaram a capacidade do arco objeto do movimento. Note, além disso, que essa troca corresponde à troca de posições em um mesmo vetor de fluxo, lembrando que cada vetor de fluxo corresponde a um arco da rede  $\mathcal{G}$ .

### 3.5. Aplicação da meta-heurística híbrida SA-ILS ao PFMI

A fim de resolver o problema de fluxo multiproduto inteiro, cujo modelo de programação matemática é descrito pelas expressões (2.1) a (2.4), foi utilizado a meta-heurística ILS [8], adaptada especificamente ao problema tratado, junto à meta-heurística SA [7] para a obtenção da solução inicial.

O Método Randômico de Descida (MRD) é o procedimento responsável pela busca local, refinando a solução perturbada pela meta-heurística. No algoritmo desenvolvido, são utilizados os procedimentos *DescidaRandomicaComTroca*, o qual representa uma versão do MRD baseada em movimentos de troca entre o fluxo dos produtos pelos arcos da rede durante a busca, e o procedimento *perturbação*, que também é constituído destes movimentos de troca e que produzirá soluções dentro do espaço solução gerado pela restrição de capacidade, tornando o fluxo pelos arcos viável quanto à capacidade dos mesmos. A motivação para a utilização do MRD baseia-se no baixo custo computacional, quando comparado ao Método de Descida Clássico. Foram utilizados 10 níveis de perturbação, que se diferenciam quanto ao tipo de troca aplicada nas variáveis de fluxo, fazendo uso da aleatoriedade. No primeiro nível, é realizada uma troca aleatória envolvendo uma variável de fluxo; no segundo nível, a troca atinge duas variáveis de fluxo, e assim por diante, até atingir os dez níveis. Inicialmente, é gerada uma solução pelo método descrito na seção 3.2. Em seguida, essa solução sofre uma perturbação dentro do espaço **A**, sendo submetida a um refinamento pelo Movimento Randômico de Descida. Depois de avaliada pelo critério de aceitação, a solução é atualizada, assim como o número de iterações e o nível de perturbação. O pseudocódigo é apresentado na Figura 3, na qual tem-se

```

procedimento SA – ILS
1   $T \leftarrow 0$ ; // tempo de processamento corrente;
2   $ci \leftarrow 0$ ; // número de ciclos realizados;
3  CM: número máximo de ciclos;
4  nível  $\leftarrow 1$ ; // nível de perturbação corrente;
5  NM: máximo de níveis de perturbação;
6   $s \leftarrow \text{GereSolucaoInicial}$ ;
7   $s^* \leftarrow s$ ; // melhor solução obtida;
8  enquanto ( $ci < CM$ ) e ( $T < TM$ ) faça
9     $s' \leftarrow \text{perturbacao}(s^*, \text{nível})$ ;
10    $s' \leftarrow \text{DescidaRandomicaComTroca}(s')$ ;
11   Se ( $f(s') < f(s^*)$ ) então
12      $s^* \leftarrow s'$ ;
13      $ci \leftarrow 0$ ;
14     nível  $\leftarrow 1$ ;
15   fim-se;
16    $ci \leftarrow ci + 1$ ;
17   Se ( $\text{nível} > NM$ ) então
18     nível  $\leftarrow \text{nível} + 1$ ;
19   fim-se;
20 fim-enquanto;
21 Retorne  $s^*$ ;
fim SA – ILS.

```

Figura 3: Pseudocódigo da meta-heurística híbrida SA-ILS.

que  $TM = 1800$  (tempo máximo permitido) e  $CM = 2000$  (número máximo de ciclos permitido).

## 4. Análise dos Resultados Computacionais

Para a realização dos testes computacionais, foram utilizadas redes com 32 nós, tendo a quantidade de arcos variando entre 96 e 320 e a quantidade de produtos especificada em 48, 192 e 320, respectivamente, conforme as instâncias presentes em [1]. Para cada produto, há um custo associado, dependente do arco pelo qual ele trafegue.

O algoritmo SA-ILS foi desenvolvido em linguagem C, usando o compilador Borland C++ Builder 5.0 e testado em um microcomputador com processador Pentium IV, 2.4 GHz, com 256 MB de memória RAM, sob sistema operacional Windows XP.

O algoritmo SA-ILS adaptado, proposto neste artigo, é comparado ao algoritmo ILS aplicado ao PFMI sob o espaço de restrição de capacidade [12]. Nas Tabelas 1 e 2, são apresentados os resultados computacionais obtidos pelos dois algoritmos, aplicados em instâncias que variam de acordo com a quantidade de arcos e produtos. A notação associada às instâncias mostra que, considerando a instância  $Bx_{y-z}$ , tem-

Tabela 1: Resultados Computacionais - larga densidade

Instâncias	$f_{ILS}$	$f_{SA-ILS}$	$t_{ILS}$	$t_{SA-ILS}$	$f_{PD}$	$t_{PD}$
Bl <sub>96-48</sub>	4957670	4838400	26.53	37.74	2.40	29.70
Bl <sub>320-48</sub>	11793351	11170531	92.23	129.28	5.28	28.66
Bl <sub>96-192</sub>	5371126	4032304	21.89	33.35	24.92	34.36
Bl <sub>320-192</sub>	16189378	14790524	59.93	105.15	8.64	43.05
Bl <sub>320-320</sub>	18609984	13385493	114.70	163.74	28.07	29.95
<b>Média</b>					<b>13.86</b>	<b>33.14</b>

Tabela 2: Resultados Computacionais - baixa densidade

Instâncias	$f_{ILS}$	$f_{SA-ILS}$	$t_{ILS}$	$t_{SA-ILS}$	$f_{PD}$	$t_{PD}$
Bs <sub>96-48</sub>	3279462	3265437	26.05	36.85	0.43	29.31
Bs <sub>320-48</sub>	10517523	10017484	34.75	64.80	4.75	46.37
Bs <sub>96-192</sub>	5625599	4441846	21.70	32.07	21.04	32.33
Bs <sub>320-192</sub>	13029738	12509966	94.22	138.76	3.98	32.09
Bs <sub>320-320</sub>	19144988	13504288	100.17	126.91	29.46	21.07
<b>Média</b>					<b>11.93</b>	<b>32.23</b>

se  $x = l$ , para densidade alta;  $x = s$ , para densidade baixa; sendo  $y$  a quantidade de arcos e  $z$  a quantidade de produtos na rede em análise.

A densidade é representada pela razão entre a capacidade (média da capacidade dos arcos) e a demanda (média da demanda dos produtos) de cada instância. A qualificação de alta densidade implica numa variação da razão em até 1:10; já a baixa densidade implica em uma variação em até 1:3.

Nas Tabelas 1 e 2, as colunas " $f_{ILS}$ " e " $f_{SA-ILS}$ " representam o valor ótimo da função de avaliação utilizada, considerando como tal a função dada pela expressão (3.1), utilizando respectivamente o método ILS e SA-ILS. As colunas " $t_{ILS}$ " e " $t_{SA-ILS}$ " representam, respectivamente, o tempo, em segundos, despendido pelo processamento computacional dos algoritmos ILS e SA-ILS. A coluna " $f_{PD}$ " representa a diferença, em percentagem, sobre os métodos em relação à função de avaliação; por fim, a coluna " $t_{PD}$ " indica a diferença, em percentagem, dos tempos computacionais obtidos pelos algoritmos. Note que a diferença percentual da função de avaliação e dos tempos computacionais dos métodos são dadas por (4.1):

$$100 - \frac{100 \times \min\{ILS, SA-ILS\}}{\max\{ILS, SA-ILS\}} \quad (4.1)$$

Pelas Tabelas 1 e 2 conclui-se que o método SA-ILS, apesar de apresentar maior tempo computacional, em virtude do acréscimo proporcionado pela metaheurística *Simulated Annealing* para obter uma solução inicial, apresenta resultados em média 11.93% melhores para redes de baixa densidade e 13.86% para redes de alta densi-

dade. Esse valor varia de acordo com a proporção arcos/produtos da rede utilizada, girando em torno de 3% para razões menores que um, e cerca de 25% para razões cuja quantidade de arcos equivale ou supera a quantidade de produtos.

## 5. Conclusões

Este trabalho apresenta uma aplicação da meta-heurística híbrida *Simulated Annealing-Iterated Local Search* (SA-ILS) para resolver o Problema de Fluxo Multiproduto Inteiro (PFMI). A estrutura de vizinhança adotada nesta metodologia garante que as soluções produzidas pertençam ao conjunto solução gerado pela restrição de capacidade, tornando o fluxo dos produtos pelos arcos da rede factíveis em relação a restrição de capacidade. A busca local é feita pelo Método Randômico de Descida, fazendo uso de movimentos de trocas entre o fluxo dos produtos nos arcos correntes. A solução inicial é obtida pela meta-heurística *Simulated Annealing*, baseada na termodinâmica. Os resultados obtidos, presentes nas Tabelas 1 e 2, mostram que o método proposto é capaz de produzir boas soluções em tempo computacional aceitável, tanto para redes esparsas e densas. O método SA-ILS, que apesar do acréscimo próximo a 32% no tempo computacional despendido para solucionar o problema, gera soluções melhores que o método ILS puro adaptado, com um nível de melhora em torno de 13%.

Uma das propostas para a continuidade deste trabalho envolve a comparação do método híbrido proposto com meta-heurísticas que façam uso de uma busca populacional ao invés de uma busca local, na tentativa de uma abordagem mais completa sobre o espaço corrente da solução.

**Abstract.** Multicommodity flow problems have a great variety of applications, mainly in areas like transport and telecommunication systems. Due to the high combinatorial complexity of this class of problems, exact methods have difficulties for solving them. This fact address the motivation for studying the application of heuristics to the multicommodity flow problems. In this work, it is proposed an application of Simulated Annealing and Iterated Local Search metaheuristics for solving the capacitated multicommodity flow problem. The objective is to determine the flow of the products through the network arcs at the lowest possible cost, guaranteeing flow conservation and capacity constraints. The space of capacity constraints is used as a search space for the proposed hybrid metaheuristic, being the flow conservation constraints penalized through relaxation. The results show good quality solutions in a reasonable computing time.

## Referências

- [1] F.P. Alvelos, “Branch-and-Price and Multicommodity Flows”, Tese de Doutorado, EPS, Universidade do Minho, Portugal, 2005.
- [2] J.C. Becceneri, “Meta-heurísticas e otimização - curso de verão”, Grupo de Pesquisa Operacional, Laboratório Associado de Computação e Matemática Aplicada (LAC), pp. 8, INPE, 2007.

- [3] L.S. Buriol, “Roteamento do Tráfego na Internet: Algoritmos para Projeto e Operação de Redes com Protocolo OSPF”, Tese de Doutorado, FEEC/UNICAMP, SP, 2003.
- [4] J. Castro, Solving difficult multicommodity problems with a specialized interior-point algorithm, *Annals of Operations Research*, **124** (2003), 35-48.
- [5] D.R. Fulkerson, L.R. Ford, “Flows in Networks”, Princeton University Press, NJ, 1962.
- [6] T.C. Hu, Multicommodity network flows, *Operations Research*, **11** (1963), 344-360.
- [7] Kirkpatrick Jr., C.S. Gelatt, M. Vecchi, Optimization by simulated annealing, *Decision Science*, **220**, No. 4598 (1983), 498-516.
- [8] H.R. Lourenço, O. Martin, T. Stuetzle, Iterated local search. In “The Handbook in Metaheuristics” (F. Glover, G. Kochenberger, eds.), pp. 321–353, Kluwer Academic Publishers, 2002.
- [9] R.L. Milidiu, A.A. Pessoa, V. Braconi, E.S. Laber, P.A. Rey, Um algoritmo grasp para o problema de transporte de derivados de petróleo em oleodutos, em “XXXIII Simpósio Brasileiro de Pesquisa Operacional”, pp. 237–246, SOBRAPO, 2001.
- [10] M.A. Santos, S. Bocanegra, F. Campos, Uma proposta de solução para o problema não linear de fluxo multiproduto utilizando pontos interiores, em “Semana de Ciência da Computação”, Lavras, MG, Vol. 1, pp. 69–73, UFL, 2000.
- [11] G.L. Schultz, R.R. Meyer, An interior point method for block angular optimization, *SIAM Journal on Optimization*, **42**, No. 4 (1991), 583-602.
- [12] C.A. Silva, S.R. de Souza, Uma aplicação da metaheurística *iterated local search* ao problema de fluxo multiproduto inteiro sob o espaço de restrição de capacidade, em “XXXIX Simpósio Brasileiro de Pesquisa Operacional”, SOBRAPO, 2007.