



SPOLM 2008

ISSN 2175-6295

Rio de Janeiro- Brasil, 05 e 06 de agosto de 2008.

UMA APLICAÇÃO DO ALGORITMO GENÉTICO E DA META-HEURÍSTICA ITERATED LOCAL SEARCH A PROBLEMAS DE FLUXO MULTIPRODUTO

Fábio Pires Mourão

Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675 – Nova Gameleira. Belo Horizonte/MG. CEP: 30510-000
fabiomourao@terra.com.br

Sérgio Ricardo de Souza

Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675 – Nova Gameleira. Belo Horizonte/MG. CEP: 30510-000
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 um algoritmo populacional, associado a uma heurística de busca local aplicado ao Problema de Fluxo Multiproduto (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 Algoritmo Genético (AG), sendo o Método da Descida Randômica aplicado a todos os indivíduos da população inicial e também nos filhos gerados. A meta-heurística Iterated Local Search (ILS) é aplicada como processo de pós-otimização. Foram feitos testes em instâncias geradas aleatoriamente e os resultados comparados com resultados obtidos por métodos exatos.

Palavras-Chaves: Fluxo Multiproduto; Algoritmo Genético; Iterated Local Search.

Abstract

Abstract em Inglês. Abstract em Inglês. Abstract em Inglês. Abstract em Inglês. Abstract em Inglês. Abstract em Inglês. Abstract em Inglês. Abstract em Inglês. Abstract em Inglês. Abstract em Inglês.

Keywords: Key-Word; Key-Word; Key-Word; Key-Word; Key-Word; Key-Word; Key-Word; Key-Word;

1. INTRODUÇÃO

Problemas de fluxo multiproduto geralmente apresentam um grande número de

variáveis e restrições. Diante desse fato, a utilização de métodos exatos pode se tornar inviável, motivando a utilização de heurísticas e metaheurísticas na resolução do problema, que podem produzir bons resultados, mesmo sem garantir otimalidade.

Os primeiros relatos na literatura deste tipo de problema são do início da década de 60, em particular com os trabalhos de Hu (1963) e Fulkerson (1962).

Problemas de fluxo multiproduto possuem uma larga variedade de aplicações, principalmente nas áreas de transporte e telecomunicações. Dentre essas aplicações podem ser citadas:

- Roteamento de tráfego na internet (Buriol (2003));
- Roteamento de mensagens em redes de comunicação (Hu (1969));
- Sequenciamento de operações em refinarias de petróleo (Milidiu (2001));
- Sequenciamento de carga (Shan (1985)).

O problema é modelado por uma rede identificada por um grafo, onde os produtos trafegam pelos arcos capacitados da rede 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 de demanda. 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 três conjuntos de restrições: restrições de conservação de fluxo, restrições 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, fazendo não trafegar por ele uma quantidade maior que a suportada. A de integralidade garante que as variáveis envolvidas sejam inteiras e não negativas.

Problemas de fluxo multiproduto (PFM) surgem quando vários produtos compartilham os arcos em uma rede e competem pela capacidade dos arcos. 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*”. Neste trabalho, além das variáveis serem inteiras, foi considerado que cada unidade de cada produto poderá fazer uso apenas de uma rota, o que caracteriza o problema de fluxo multiproduto binário.

Neste trabalho é proposta uma resolução do PFM através do Algoritmo Genético, utilizando uma heurística de busca local em todos os indivíduos da população inicial e também nos filhos gerados em cada geração pelo AG. O método de busca aplicado à população inicial e também nos filhos foi o Método da Descida Randômica, por apresentar um tempo computacional menor quando comparado ao método clássico da descida, já que este verifica todo o espaço de busca. Além disso, como trata-se de um problema de otimização combinatória, testar todo o espaço de busca é um processo inviável.

Como procedimento de pós-otimização, foi utilizada a Meta-heurística *Iterated Local Search* (ILS) à melhor solução encontrada pelo AG. Esta heurística foi capaz de melhorar significativamente a solução final. De acordo com Becceneri (2007) pode-se definir 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*”. 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 solução um método de busca local, como o método da

descida. Mesmo no ILS, a heurística de busca utilizada após a solução sofrer perturbações, foi o método da Descida Randômica, pela mesma justificativa apresentada para o método populacional.

2. REVISÃO BIBLIOGRÁFICA

Os primeiros trabalhos relacionados ao PFM (Problemas de Fluxo Multiproduto) datam do início da década de 60, com contribuições iniciais de Hu (1963) e Fulkerson (1962). Wille (2005) utilizam as metaheurísticas Busca Tabu e 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, entre elas a qualidade de serviços. Os resultados computacionais obtidos mostram uma melhor eficiência da abordagem feita pelo AG, gerando boas soluções para redes de tamanho médio, em comparação com o método Busca Tabu.

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.

Kamath (1993) propõem um código baseado em Métodos de Pontos Interiores aplicado a PFM.

Park (2002) utiliza a técnica de geração de colunas para resolver duas classes de problemas de fluxo multiproduto inteiro, sendo que em um dos problemas são dados o conjunto de produtos com suas respectivas demandas e custo unitário em cada arco. O objetivo desse problema é selecionar um subconjunto de produtos para serem roteados e associar, 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 o de determinar o custo mínimo.

Em Alvelos (2005), a abordagem é por meio de geração de colunas para programação linear e inteira. Os testes computacionais são feitos em instâncias geradas pelo GenMCF (Generator Multicommodity Flow), desenvolvido pelo próprio autor.

Bocanegra (2005) propõem um algoritmo para resolver o problema não linear de fluxo multiproduto, utilizando planos de corte e centros analíticos. O problema é relaxado utilizando a função lagrangeana parcial, construída a partir de hiperplanos suporte. Foi também utilizado o método de pontos interiores. Segundo Bocanegra (2005), o algoritmo proposto se destaca por gerar uma seqüência monótona estritamente crescente de cotas para alcançar a solução, e conseqüentemente segue trajetórias centrais associadas ao máximo da função modelo.

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.

Em Ahuja (1993) são utilizados os métodos simplex e técnicas de particionamento primal para resolver o PFM.

3. MODELAGEM MATEMÁTICA

Nesta seção é apresentada uma modelagem do problema do PFM, considerando uma formulação baseada em arcos, tendo cada produto uma associação com um par de nós origem – destino. A formulação nesta seção é referente ao Problema Binário.

Seja $G = (N,A)$ um grafo orientado representando uma rede com p produtos, a arcos e n nós, sendo as cardinalidades dos conjuntos N e A , iguais a n e a , respectivamente. Dessa forma temos que n e a são definidos como:

$$\#N = n;$$

$$\#A = a.$$

Define-se o conjunto dos produtos como P e sua quantidade de elementos igual a p . A formulação matemática original do problema é a seguinte:

$$\min \sum_{k=1}^p \sum_{(i,j) \in A} c_{ij}^k d^k x_{ij}^k \quad (1)$$

$$s.a. \sum_{(i,j) \in A} x_{ij}^k - \sum_{(i,i) \in A} x_{ji}^k = b_i^k \quad \forall i \in N, 1 \leq k \leq p \quad (2)$$

$$\sum_{k \in P} d^k x_{ij}^k \leq u_{ij} \quad \forall (i,j) \in A \quad (3)$$

$$x_{ij}^k \in \{0,1\}, \forall k \in P, \forall (i,j) \in A \quad (4)$$

sendo então, que (1) representa a função a ser minimizada; (2) representa as restrições de conservação de fluxo; (3) representa as restrições de capacidade; e os parâmetros são tais que:

c_{ij}^k representa o custo unitário do produto no arco (i,j) ;

x_{ij}^k pode assumir 0 ou 1, no caso de 0, significa que o produto k não faz uso do arco (i,j) e se for 1 significa que o produto k o utiliza;

u_{ij} representa a capacidade do arco (i,j) ;

d^k representa a demanda do produto k pertencente a P e que deve sair do nó de origem e chegar ao nó de destino;

b_i^k indica se o nó i é oferta ou demanda para o produto k , de modo que:

$$b_i^k = \begin{cases} 1, & \text{se } i = o^k \\ -1, & \text{se } i = d^k \\ 0, & \text{em caso contrário} \end{cases}$$

de forma que o^k é representa origem do produto k e d^k destino para o produto k .

4. ALGORITMO GENÉTICO

A idéia básica da Computação Evolutiva é aplicar o processo de evolução natural como um paradigma de solução de problema. Segundo Zuben, três algoritmos para computação evolutiva foram desenvolvidos independentemente:

- Algoritmos Genéticos: Holland (1962), Bremerman (1962) e Fraser (1957);
- Programação evolutiva: Fogel (1962);
- eEstratégias evolutivas: Rechenberg (1965) e Schwefel(1965).

No presente trabalho, a abordagem é feita pelo algoritmo genético (AG). A motivação do uso do AG está na possibilidade de, junto a alguns métodos de busca local, encontrar boas soluções para o PFM, já que os mesmos possuem um elevado dimensionamento quanto ao número de variáveis envolvidas. É óbvio que não existe nenhuma garantia das soluções encontradas serem ótimas, mas a escolha do método está na lenta convergência de métodos exatos.

No AG um indivíduo representa um *cromossomo* que contém a codificação (genótipo) de uma possível solução (fenótipo). Em particular cada cromossomo representa uma solução x na formulação dada anteriormente, sendo a implementação desta em forma de uma matriz com dimensão já especificada. Os atributos de uma solução tratada neste trabalho são os fluxos correspondentes a cada produto, portanto cada coluna da matriz solução, onde estão os fluxos para cada produto é denominado *gene*. Os possíveis fluxos que podem ser

determinados para cada produto é denominado *alelo*. Trata-se claramente de uma descrição específica para estas definições, porém as mesmas são de caráter genérico, i.e., em outros problemas cada indivíduo pode ser representado de forma diferente e conseqüentemente, cada termo descrito poderá ser visto de outra maneira.

De acordo com Zuben, o processo evolutivo executado por um AG corresponde a um procedimento de busca em um espaço de soluções potenciais para o problema. É necessário discutir alguns parâmetros e operadores do AG, já que os mesmos influenciam diretamente no sucesso do algoritmo.

Operador de crossover: é também chamado de operador recombinação e possui a função de criar novos indivíduos (filhos) a partir da seleção aleatória de mais indivíduos (pais). O operador mais comum e mais usado é o crossover de um ponto, onde dois pais são selecionados e dois novos filhos são gerados, a partir de um ponto de corte escolhido aleatoriamente. Na Figura 1 é apresentado um exemplo clássico deste tipo de operador. Simplesmente o filho 1 (2) “herda” os valores da posição 1 até o ponto de corte do pai 1 (2) e os demais valores do pai 2 (1).

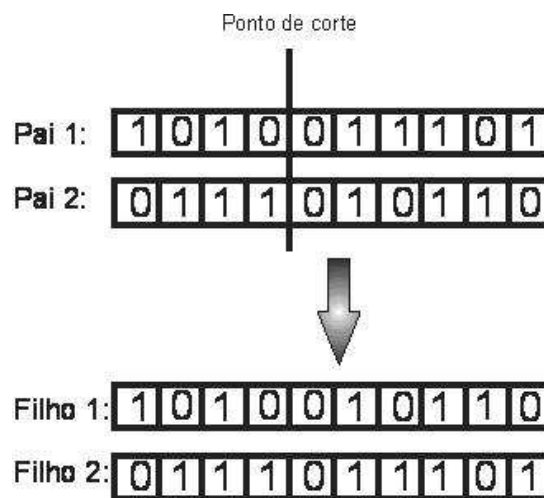


Figura 1: crossover de um ponto

Operador de mutação: é utilizado para modificar um gene do indivíduo gerado. Cada filho normalmente possui uma determinada probabilidade de sofrer mutação, chamada de *taxa de mutação*. O principal objetivo é diversificar o espaço de busca, evitando uma possível convergência prematura e fugindo de ótimos locais. A *taxa de mutação* não pode ser alta, pois assim modificaria muitos novos indivíduos e descaracterizaria o crossover e o processo obtido com a busca.

Seleção de indivíduos: Existem diversas formas de seleção de indivíduos, porém um método mais usado de seleção de indivíduos para a próxima geração é o *roullet wheel* Michalewicz (1996). Neste método, quanto melhor o valor da função de aptidão do indivíduo, mas chances ele tem de ir para a próxima geração. Outro tipo de operador de seleção é a seleção baseada em *rank*, que é baseado na função de aptidão de cada indivíduo, um método bastante utilizado e que de uma forma geral é uma variação do operador baseado em *rank*, é simplesmente conhecido como *elitismo*, que garante que um percentual dos melhores indivíduos não seja perdido. A estrutura geral de um algoritmo evolucionário, encontrada em Michalewicz (1996) é apresentada na Figura 2.

```

procedimento programa evolutivo
início
   $t \leftarrow 0$ 
  inicialize  $P(t)$ 
  avalie  $P(t)$ 
  enquanto (não condição de parada) faça
  início
     $t \leftarrow t + 1$ 
    selecione  $P(t)$  a partir de  $P(t - 1)$ 
    altere  $P(t)$ 
    avalie  $P(t)$ 
  fim
fim

```

Figura 2: pseudo-código de um AG

5. DESCIDA RANDÔMICA

O método da Descida Randômica é uma variação do método clássico de Descida, porém neste método a busca de todo o espaço de vizinhança não precisa ser feita, como é o caso do clássico método da Descida. Na Descida Randômica qualquer solução de melhora é aceita. Caso o vizinho não seja melhor do que a solução ótima corrente, 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 pseudo-código do procedimento de Descida Randômica, mostrado na Figura 3.

```

procedimento RandomicoDescida( $f(\cdot), N(\cdot), IterMax, s$ );
1   $Iter = 0;$       {Contador de iterações sem melhora }
2  enquanto ( $Iter < IterMax$ ) faça
3     $Iter = Iter + 1;$ 
4    Seleccione aleatoriamente  $s' \in N(s);$ 
5    se ( $f(s') < f(s)$ ) então
6       $Iter = 0;$ 
7       $s \leftarrow s';$ 
8    fim-se;
9  fim-enquanto;
10 Retorne  $s;$ 
fim RandomicoDescida;

```

Figura 3: Pseudo-código do método da Descida Randômica

6. ITERATED LOCAL SEARCH - ILS

O método ILS é classificado como uma Metaheurística, pois apresenta uma estratégia de fuga de ótimos locais, através da idéia de melhorar um processo de busca local. No ILS a idéia é aplicar perturbações na solução ótima local e em seguida aplicar um outro método de busca 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 4, encontrada em Souza (2007) mostra o pseudo-código do método ILS.

```

procedimento ILS
 $s_0 \leftarrow \text{SolucaoInicial}$ 
 $s \leftarrow \text{BuscaLocal}(s_0)$ 
 $\text{iter} \leftarrow 0$ 
enquanto ( $\text{iter} < \text{iter}_{\max}$ )
     $\text{iter} \leftarrow \text{iter} + 1$ 
     $s' \leftarrow \text{perturbacao}(s, \text{histórico})$ 
     $s'' \leftarrow \text{BuscaLocal}(s')$ 
     $s \leftarrow \text{CritérioAceitacao}(s, s', s'')$ 
fim-enquanto
retorne  $s$ 

```

Figura 4: Pseudo-código do método ILS

7. METODOLOGIA

Nesta seção são definidas características utilizadas no método da descida randômica e no ILS.

7.1. 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 . Cada unidade de cada produto possui uma única rota, mesmo essa matriz não possuindo uma representação binária.

7.2. 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 ótima 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 ótima 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.

7.3. FUNÇÃO DE AVALIAÇÃO

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

$$f(x) = \text{Tr.}(c.x) + \alpha.v$$

onde v corresponde ao somatório das violações nos arcos e $\alpha > 0$ é um parâmetro de penalização, portanto em soluções factíveis, v é igual a zero. Isso leva ao algoritmo minimizar a função objetivo sem a penalização. Não foi utilizado nenhum método para tratar restrições de conservação de fluxo, porque 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 $p \times p$. O fato de considerar o traço da matriz $c.x$ se deve ao fato de 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).

8. AG APLICADO AO PROBLEMA

Nesta seção são discutidas as características específicas do AG aplicado ao PFM.

Representação da solução: a matriz solução, como já mencionada anteriormente, é uma matriz x de dimensão $a \times p$, onde são representados os fluxos dos produtos em cada arco. Em vez de ser atribuído 0 ou 1 a cada posição dessa matriz, é atribuído o valor que representa a quantidade do produto que em cada posição, mesmo assim não há divisão do fluxo do produto, i.e., todas as unidades seguirão uma mesma rota, o que não descaracteriza o problema binário.

Representação da população: feita por meio de uma estrutura, onde cada registro contém três campos, um para alocar a matriz solução, outro para o valor da função objetivo associada ao indivíduo e outro para armazenar o valor da função de aptidão.

Geração da população inicial: para gerar os indivíduos da população inicial, foi implementada uma elegante heurística construtiva, que garante factibilidade quanto às restrições de conservação de fluxo e ainda mantém a aleatoriedade, característica marcante em algoritmos genéticos. Em cada indivíduo da população inicial foi aplicado o Método de Descida Randômica, por apresentar um baixo custo computacional quando comparado a outras heurísticas de busca local, a fim de melhorar a solução, reduzindo ou até mesmo, em alguns casos, eliminando as violações quanto às restrições de capacidade.

Seleção de indivíduos: A seleção dos indivíduos para a recombinação é feita de forma aleatória entre todos os indivíduos da população, com probabilidade uniforme.

Cruzamento: foi utilizado o cruzamento de um ponto.

Mutação: não foi utilizado nenhum parâmetro de mutação, pelo fato de existir um método de busca local em cada filho.

Eliminação: Sendo $nind$ o número total de indivíduos, são trocados os $nind/2$ piores indivíduos, sendo esses substituídos pelos filhos gerados. Os $nind/2$ melhores indivíduos da população anterior são mantidos e a magnitude da população também continua a mesma.

Função objetivo: A função objetivo é definida como $tr(cx)$, onde c é uma matriz $p \times a$, onde c_{ij} representa o custo unitário do produto i no arco j . Obviamente $c \in Z^{ap}$.

Função de aptidão: A função de aptidão é definida como sendo a função objetivo somada ao produto de uma constante pelo total de violações presentes nos arcos, ou seja, para o cálculo da função de aptidão é preciso o somatório das violações em todos os arcos. Sendo assim, a função é definida como $tr(cx) + \alpha v$, onde v corresponde ao somatório das violações nos arcos. Esta função também caracteriza a função de avaliação nas heurísticas de busca local. A escolha desta função se deve ao fato de se tratar de um problema de otimização restrita e portanto foi escolhido o método da penalização para tratar tais restrições de capacidade. Não foi escolhido nenhum método para tratar as restrições de conservação de fluxo porque as soluções não violam estas restrições mesmo passando pelos operadores do AG, devido à forma como são geradas inicialmente.

9. DESCIDA RANDÔMICA APLICADA AO PROBLEMA

Primeiramente são discutidas as características básicas do método da descida aplicado ao PFM, em seguida serão discutidas as características do ILS.

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

Na Descida Randômica aplicada à população inicial e aos filhos gerados nas gerações pelo crossover 1 ponto, um movimento m em x é definido como gerar uma nova rota para um produto escolhido de forma totalmente aleatória, sendo esta nova rota gerada de forma a manter as restrições de conservação de fluxo.

Porém no método da Descida Randômica utilizado como busca local para o ILS, utilizado como procedimento de pós-otimização, um movimento m em x é definido como gerar uma nova rota para um produto escolhido aleatoriamente. 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 selecionada 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 arco qualquer. Esta forma de seleção do produto implica em maior chance da solução corrente se tornar factível. Esta nova rota também é traçada aleatoriamente, de forma que sejam mantidas as restrições de conservação de fluxo, de forma análoga à geração dos indivíduos da população inicial.

Em caso de melhora, então a solução ótima corrente é atualizada, caso contrário gera-se um novo vizinho.

O critério de parada do método consiste no número máximo de iterações sem melhora, ou seja, no número máximo de vizinhos da solução ótima corrente que são testados, neste trabalho este parâmetro foi chamado de *itermax* para o método da Descida aplicado aos indivíduos iniciais e aos filhos e *itermax2* para o método da Descida utilizado no procedimento ILS.

10. ILS APLICADO AO PROBLEMA

Nesta seção são discutidas características específicas do método ILS aplicado ao problema.

Uma perturbação consiste em aumentar em mais um o número de produtos cuja rota é alterada, sendo que o nível 1 consiste em alterar o fluxo de dois produtos, o nível 2 em trocar o fluxo de 3 produtos e assim por diante.

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.

Para cada nível de perturbação é considerado o critério de parada por nível como sendo um número máximo de iterações sem melhora, chamado neste trabalho de *vezes_nível*.

O critério de parada do ILS é o número máximo de perturbações feitas, denominado no trabalho de *ilsmax*.

O ILS é utilizado como um procedimento de pós-otimização, sendo aplicado somente à melhor solução retornada pelo AG. Foi utilizado dessa forma porque seria inviável computacionalmente aplicá-lo aos filhos ou à toda população inicial.

11. RESULTADOS

Nos testes computacionais foram usadas as instâncias contidas no pacote *carbin*, geradas aleatoriamente pelo GenMCF (*Generator Multicommodity Flow*), desenvolvido por Alvelos (2005). Foi utilizado um computador *Intel Celeron 1.87 GHz*, com *1 GB de RAM DDR2*, sob o sistema operacional *Windows XP*.

As instâncias foram classificadas em grupos, de acordo com suas características, sendo que os parâmetros do AG, do ILS e do Método da Descida Randômica foram ajustados de acordo com cada grupo. A classificação se deu de acordo com a Tabela 1, sendo que o **Grupo 1** são as instâncias de *bl01* a *bl04*, do **Grupo 2**, são as instâncias de *bl05* a *bl08* e assim por diante.

| Grupo | Instâncias | # N | # A | # P |
|-------|-------------|-----|-----|-----|
| 1 | bl01 - bl04 | 32 | 96 | 48 |
| 2 | bl05 - bl08 | 32 | 320 | 48 |
| 3 | bl09 - bl12 | 32 | 96 | 192 |
| 4 | bl13 - bl16 | 32 | 320 | 192 |
| 5 | bl17 - bl20 | 32 | 96 | 320 |
| 6 | bl21 - bl24 | 32 | 320 | 320 |

Tabela 1: características de cada grupo

Na Tabela 1, # N, # A e # P, representam as cardinalidades dos conjuntos de nós, arcos e produtos, respectivamente.

Na Tabela 2 são apresentados os parâmetros utilizados em cada grupo, de acordo com as

características mostradas na Tabela 1.

| Grupo | qmax | maxger | nind | itermax | vezesnível | ilsmax | itermax2 | filhos | alfa |
|-------|------|--------|------|---------|------------|--------|----------|--------|--------|
| 1 | 70 | 150 | 300 | 50 | 200 | 15 | 200 | nind/2 | 100000 |
| 2 | 70 | 80 | 100 | 40 | 60 | 15 | 200 | nind/2 | 80000 |
| 3 | 70 | 80 | 80 | 40 | 60 | 15 | 200 | nind/2 | 40000 |
| 4 | 70 | 80 | 60 | 30 | 30 | 15 | 100 | nind/2 | 20000 |
| 5 | 70 | 80 | 60 | 30 | 50 | 15 | 150 | nind/2 | 50000 |
| 6 | 70 | 60 | 40 | 15 | 30 | 15 | 100 | nind/2 | 40000 |

Tabela 2: parâmetros utilizados em cada grupo

Na Tabela 2, *qmax* é um parâmetro utilizado na heurística construtiva, utilizada para gerar a população inicial. Esta heurística não foi descrita neste trabalho, *maxger* é o número máximo de gerações do AG, *nind* é o tamanho da população, *itermax* corresponde ao número máximo de iterações sem melhora utilizado no Método da Descida Randômica aplicado à população inicial e aos filhos gerados em cada geração do AG, *vezesnível* é a quantidade de iterações sem melhora em cada nível de perturbação no método ILS, *ilsmax* é o número máximo de perturbações no ILS, *itermax2* corresponde ao número máximo de iterações sem melhora no método de busca local utilizado no ILS, *filhos* corresponde ao número de filhos gerados em cada geração e *alfa* é o parâmetro utilizado para penalizar uma solução que viole as restrições de capacidade.

Nas onde são exibidos os resultados obtidos pelo método proposto, a coluna *fo* apresenta as soluções encontradas neste trabalho, enquanto que a coluna *f** apresenta as soluções obtidas por Alvelos (2005) através de métodos exatos, em particular usando um método de geração de coluna. O tempo máximo de execução do algoritmo proposto por Alvelos (2005) foi de uma hora, sendo que para a instância bl10, precedida de um ** não foi encontrada nenhuma solução factível em Alvelos (2005) neste tempo máximo de execução. Nas instâncias precedidas por *, Alvelos (2005) encontrou uma solução factível ao fim de uma hora. As instâncias que não possuem nenhuma indicação possuem valores ótimos encontrados por Alvelos (2005) e tais resultados são obtidos em *f**. A coluna *tempo* se refere ao tempo total de execução do algoritmo proposto neste trabalho, sendo este tempo o somatório dos tempos para leitura das instâncias, geração da soluções iniciais, AG e o tempo de pós-otimização (ILS). A coluna violação apresenta a violação da solução final, ou seja, a soma das violações de todos os arcos, como mencionado anteriormente. Obviamente se uma solução apresenta violação nula, a mesma é factível quanto às restrições de capacidade. Todas as soluções obtidas são factíveis quanto às restrições de conservação de fluxo, devido à própria geração de soluções iniciais e pela característica do crossover. Em todos os casos foi utilizado o crossover de um ponto.

A Tabela 3 apresenta os resultados obtidos para as instâncias do **Grupo 1**, a Tabela 4, para o **Grupo 2** e assim por diante.

| Instância | fo | f* | Tempo (s) | Violação |
|-----------|---------|---------|-----------|----------|
| bl01 | 1634724 | 1615947 | 777,422 | 0 |
| bl02 | 1830095 | 1816947 | 1459,281 | 0 |
| bl03 | 17985 | 17340 | 706,89 | 0 |
| bl04 | 19183 | 21370 | 978,031 | 10 |

Tabela 3: resultados para o Grupo 1

| Instância | fo | f* | Tempo (s) | Violação |
|-----------|--------|--------|-----------|----------|
| bl05 | 517367 | 474782 | 888,828 | 0 |
| bl06 | 446544 | 411480 | 1262,906 | 0 |
| bl07 | 5999 | 5751 | 676 | 0 |

bl08 5908 5688 853,281 0

Tabela 4: resultados para o Grupo 2

| Instância | fo | f* | Tempo (s) | Violação |
|------------------|-----------|-----------|------------------|-----------------|
| bl09 | 6396156 | 6261671 | 1302,594 | 2 |
| *bl10 | 6808027 | ** | 2681,171 | 0 |
| bl11 | 69754 | 69018 | 1883,563 | 0 |
| bl12 | 71041 | 65902 | 2109,031 | 0 |

Tabela 5: resultados para o Grupo 3

| Instância | fo | f* | Tempo (s) | Violação |
|------------------|-----------|-----------|------------------|-----------------|
| bl13 | 3320682 | 3132695 | 2137,047 | 0 |
| bl14 | 2929588 | 2433011 | 3210,875 | 0 |
| bl15 | 37333 | 34274 | 2363,891 | 0 |
| bl16 | 31508 | 28074 | 2292,343 | 0 |

Tabela 6: resultados para o Grupo 4

| Instância | fo | f* | Tempo (s) | Violação |
|------------------|-----------|-----------|------------------|-----------------|
| bl17 | 13464612 | 13190922 | 1494,093 | 0 |
| bl18 | 10829826 | 10496120 | 1051,844 | 0 |
| *bl19 | 114141 | 109556 | 1768,515 | 0 |
| *bl20 | 216409 | 111604 | 2344,484 | 2 |

Tabela 7: resultados para o Grupo 5

| Instância | fo | f* | Tempo (s) | Violação |
|------------------|-----------|-----------|------------------|-----------------|
| *bl21 | 6169916 | 5800149 | 3947,922 | 0 |
| bl22 | 4795459 | 4209266 | 6395,453 | 0 |
| bl23 | 61585 | 56856 | 3375,875 | 0 |
| bl24 | 56186 | 47964 | 3731,719 | 0 |

Tabela 8: resultados para o Grupo 6

12. CONCLUSÕES

O algoritmo proposto obteve soluções factíveis para 87,5 % das instâncias testadas, sendo que em alguns casos a solução ficou próxima aos valores ótimo obtidos por Alvelos (2005). No caso da instância bl10, Alvelos (2005) não encontrou solução factível num tempo máximo de um hora de execução, porém neste trabalho foi encontrada uma solução para tal instância em um tempo menor que uma hora. Como mencionado anteriormente o método proposto neste trabalho não garante otimalidade. Aumentando os valores dos parâmetros citados na Tabela 2, podem ser obtidos resultados ainda melhores, porém com um custo computacional mais alto, o que inviabiliza um aumento muito grande nesses parâmetros.

13. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Ahuja, R.K., Magnanti, T.L. & Orlin, J.B. *Network flows*. Prentice Hall, 1993.
- [2] 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).
- [3] Becceneri, J. C., Meta-heurísticas e otimização - curso de verão. Grupo de Pesquisa Operacional, Laboratório Associado de Computação e Matemática Aplicada (LAC), INPE, 2007.

- [4] Bocanegra, S., Filho, F.F.C. & Santos, M.A. Uma proposta de solução ao para o problema não linear de fluxo multiproduto utilizando pontos interiores, *Semana de Ciência da Computação, UFL, Lavras/MG*, 69–73, 2000
- [5] Bremermann, H.J., Optimization through evolution and recombination, M.C. Yovits, G.T. Jacobi & G.D. Goldstine (eds.) *Self-Organizing Systems*, 93-106, Spartan Books, 1962.
- [6] Buriol, L.S. Roteamento do tráfego na internet: algoritmos para projeto e operação de redes com protocolo OSPF, Tese (Doutorado em Engenharia Elétrica), FEEC/UNICAMP, 2003.
- [7] Fogel, L.J., Autonomous automata, *Industrial Research*, vol. 4, 14-19, 1962.
- [8] Fraser, A.S. Simulation of genetic systems by automatic digital computers: I. Introduction, *Austral. J. Biol. Sci.*, vol. 10, pp. 484-491, 1957.
- [9] Fulkerson, L. R. e Ford, D. R., Flows in networks, Technical report, Princeton University, USA, 1962.
- [10] Gondzio, J., Sarkissian, R., Vial, J. P. Goffin, J. L., Solving nonlinear multicommodity flow problems by the analytic center cutting plane method. *Mathematical Programming*, 76:131–154, 1996.
- [11] Holland, J.H., Outline for a logical theory of adaptive systems, *J. Assoc. Comput. Mach.*, vol. 3, pp. 297-314, 1962.
- [12] Hu, T. C., Multicommodity network flows, *Operations Research*, 11:344–360, 1963.
- [13] Hu, T.C., *Integer Programming and Network Flows*, Addison-Wesley, 1969.
- [14] Karmarkar, N. K., Ramakrishnan, K. G. e Kamath, A. P., Computational and complexity results for an interior point algorithm on multicommodity flow problems. Technical report, Dip. Di Informática, Univ. de Pisa, Italy, 1993.
- [15] Michalewicz, Z., *Genetic algorithms + Data Structures = Evolution Programs*, 3 ed., Springer-Verlag, 1996.
- [16] Milidiu, R.L., Pessoa, A.A., Braconi, V., Laber, E.S. & Rey, P.A. Um algoritmo grasp para o problema de transporte de derivados de petróleo em oleodutos, *Anais do XXXIII Simpósio Brasileiro de Pesquisa Operacional*, 237–246, 2001.
- [17] Oldham J. D., Plotkin, S. e Stein, C. Goldberg, A. V, 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 Internacional Integer Programming and Combinatorial Optimization Conference. Lecture Notes in Computer Sciences*, volume 1412, Berlin, German, 1998. Springer-Verlag.
- [18] Park, S., Kim, D. e Lee, K., An integer programming approach to the path selection problems. In *Proceedings of INOC2005 - International Network Optimization Conference*, Faculdade de Ciências, Universidade de Lisboa, Lisboa, Portugal, 2002.
- [19] Rechenberg, I., Cybernetic solution path of an experimental problem, Royal Aircraft Establishment, Library Translation no. 1122, 1965.
- [20] Schwefel, H.-P., *Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik*, Diplomarbeit, Hermann Föttinger Institut für Strömungstechnik, Technische Universität, Berlin, 1965.
- [21] Shan, Y.S., A Dynamic Multicommodity Network Flow Model for Real Time Optimal Rail Freight Car Management, PhD thesis, Princeton University, Canada, 1985.

- [22] Souza, M. J. F. Inteligência Computacional para Otimização, Notas de aula, Departamento de Computação, Universidade Federal de Ouro Preto, disponível em <http://www.decom.ufop.br/prof/marcone>, 2007.
- [23] Zuben, F. J. V., Computação Evolutiva: Uma abordagem Pragmática, UNICAMP.
- [24] Wille, E.C.G., Mellia, M., Leonardi, E. & Marsan, M.A. Topological design of survivable ip networks using metaheuristic approaches, Third Internation Workshop on QoS in Multiservice IP Networks, 191–206, 2005.