

Aplicação de Algoritmo Evolucionário Populacional ao Problema de Fluxo Multiproduto Inteiro¹

F.P. MOURÃO², S.R. DE SOUZA³, Programa de Pós-Graduação em Modelagem Matemática e Computacional, CEFET-MG, 30510-000 Belo Horizonte, MG, Brasil.

C.A. SILVA⁴, Universidade de São Paulo, Departamento de Matemática Aplicada e Estatística, 13560-970 São Carlos, SP, Brasil.

Resumo. O objetivo deste trabalho é analisar o comportamento de um algoritmo evolucionário populacional aplicado à solução do Problema de Fluxo Multiproduto Inteiro (PFMI). Este problema pertence à classe de problemas NP-difíceis, tendo forte aplicação de cunho econômico. O uso de técnicas heurísticas é justificada pela elevada dimensão do problema em relação à quantidade de variáveis e restrições. Assim, mesmo sem garantir otimalidade, técnicas heurísticas (locais e populacionais) podem conseguir boas soluções e em menor tempo computacional, quando comparadas a algoritmos exatos. Neste trabalho, é proposta uma abordagem do problema via algoritmos genéticos, associado a uma heurística de busca local, para resolver o PFMI capacitado. Busca-se determinar o fluxo dos produtos pelos arcos da rede ao menor custo possível, respeitando-se as restrições de conservação de fluxo e de capacidade. São desenvolvidas heurísticas aplicadas aos indivíduos da população inicial, em particular utilizando-se o Método da Descida Randômica. Para verificar a eficiência do algoritmo proposto, foram feitos testes computacionais com instâncias de teste da literatura. Os resultados obtidos pelo método proposto foram comparados a resultados obtidos por métodos exatos. Em especial, para uma das instâncias foi encontrada uma solução factível, fato não atingido pelo método exato utilizado na comparação.

Palavras-chave. Fluxo Multiproduto, Algoritmo Genético, Heurística.

1. Introdução

O objetivo deste trabalho é analisar o comportamento de uma metaheurística populacional aplicada ao Problema de Fluxo Multiproduto Inteiro (PFMI). O problema em tela pertence à classe dos problemas NP-difíceis, possuindo grande aplicação de cunho econômico. O uso de técnicas heurísticas se justifica pela elevada dimensão do problema em relação à quantidade de variáveis e restrições. Mesmo sem garantir

¹Trabalho financiado por CAPES, CNPq e CEFET-MG.

²fabimourao@dppg.cefetmg.br

³sergio@dppg.cefetmg.br

⁴calex@icmc.usp.br

soluções ótimas, técnicas heurísticas podem conseguir boas soluções e em menor tempo computacional quando comparados a algoritmos exatos.

Problemas de Fluxo Multiproduto são relatados na literatura desde o início da década de 60, com as contribuições iniciais de [5] e [7], possuindo uma larga variedade de aplicações. De forte importância, na atualidade, são suas aplicações no estudo de problemas de roteamento de tráfego na internet, apresentadas, por exemplo, em [2] e [10].

Em [3], é descrito um problema de otimização aplicado ao roteamento do tráfego na internet, dado um conjunto de demandas (pacotes de dados), com o objetivo de minimizar o congestionamento na rede. É apresentado um algoritmo genético para resolver o problema, sendo os resultados comparados com os resultados mais conhecidos obtidos por heurísticas aplicadas especificamente a este tipo de problema. O algoritmo proposto foi capaz de produzir soluções de boa qualidade para a maioria das instâncias testadas.

Em [9], o autor utiliza o método estocástico *simulated allocation* em problemas de fluxo multiproduto inteiro capacitado e não-capacitado. Os testes incidem em um problema da mochila múltipla e uma rede de telecomunicação. Um algoritmo genético é comparado à heurística estocástica, que, apesar de não superar os resultados do *simulated allocation*, produz soluções bem próximas e de boa qualidade em um tempo computacional razoável.

O problema é modelado por uma rede identificada por um grafo, pela qual 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. Assim, os produtos compartilham os arcos em uma rede e competem pela capacidade dos mesmos. Caso o fluxo de cada produto seja indivisível, o problema passa a ser denominado Problema de Fluxo Multiproduto Inteiro (PFMI). No presente 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, o Algoritmo Genético (AG) foi utilizado juntamente com um método heurístico de busca local, sendo utilizada a heurística Descida Randômica em cada indivíduo da população inicial e nos filhos gerados em cada geração do AG.

Os testes computacionais foram realizados em instâncias geradas aleatoriamente pelo GenMCf, desenvolvido por [1], e os resultados obtidos comparados a resultados encontrados por métodos exatos. Foram obtidos resultados próximos das soluções ótimas para a maior parte dos problemas testes, além de um resultado factível para uma instância não solucionada anteriormente em [1].

2. Descrição do PFMI

O problema de fluxo multiproduto é modelado por meio de uma rede identificada por um grafo $\mathcal{G} = (\mathcal{A}, \mathcal{N})$, 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 se apresenta

quando um conjunto de produtos \mathcal{P} compartilham os arcos da rede e competem pela capacidade dos mesmos, sendo o objetivo o de determinar, ao menor custo, o fluxo de produtos na rede, de maneira a atender quatro conjuntos de restrições: restrição de conservação de fluxo, restrição de capacidade, restrição de linearidade e restrição de integralidade. Neste trabalho, foi estudado o problema de fluxo multiproduto não-bifurcado ou binário, que consiste em não dividir o fluxo de cada produto pelos arcos da rede, fazendo com que cada unidade de cada produto trafegue por uma única rota.

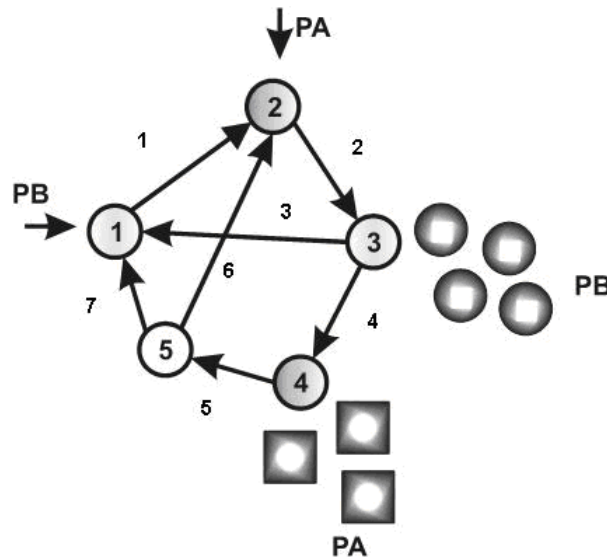


Figura 1: Exemplo de uma rede multiproduto.

A Figura 1 apresenta um exemplo de uma rede de fluxo multiproduto com 5 nós, 7 arcos e 2 produtos, de forma que os nós 4 e 2 são de oferta e demanda, respectivamente, para o produto PA , e os nós 3 e 1 são de oferta e demanda, para o produto PB . No caso do Problema de Fluxo Multiproduto Inteiro, as variáveis de decisão são variáveis inteiras, representando os fluxos pelos arcos. Caso os fluxos façam uso de apenas uma rota, as variáveis de decisão do problema passam a definir a passagem ou não de um dado produto por um certo arco, sendo, portanto, variáveis binárias. Então, deve-se acrescentar, ao conjunto de restrições, a restrição quanto ao tipo de variáveis envolvidas.

As restrições de conservação de fluxo desempenham o papel de gerenciar o fluxo dos produtos pelos arcos da rede que saem de um ponto de oferta e chegam a um ponto de demanda. As restrições de capacidade limitam o fluxo dos produtos, de modo que em nenhum arco trafegue uma quantidade de produtos superior à capacidade suportada por ele. A restrição de linearidade vem da formulação matemática do problema e a restrição de integralidade garante que as variáveis de interesses sejam inteiras e não-negativas.

3. Modelagem Matemática

A formulação matemática do PFMI abordado é dada por:

$$\min \quad z = cx \quad (3.1)$$

$$\text{sujeito a} \quad Nx^i = b^i, \quad i = 1, \dots, p \quad (3.2)$$

$$\mathbf{I}x \leq u \quad (3.3)$$

$$x \in Z_+ \quad (3.4)$$

sendo $a = |\mathcal{A}|$ arcos, $n = |\mathcal{N}|$ nós e $p = |\mathcal{P}|$ produtos.

A expressão (3.1) representa a função objetivo a ser minimizada, considerando $c \in Z^{p \times a}$ a matriz de custo associada. A expressão (3.2) refere-se à restrição de conservação de fluxo, de modo que $N \in Z^{n \times a}$ é a matriz de incidência nó-arco; $x^i \in Z^a$ é o vetor de fluxo do produto i ; $b^i \in Z^n$ é o vetor oferta/demanda para o produto i . A expressão (3.3) refere-se à restrição de capacidade, para $\mathbf{I} \in Z^{a \times pa}$ um vetor composto de p matrizes identidades de ordem a ; $x \in Z^{a \times p}$ é a matriz de fluxo; $u \in Z^a$ o vetor de capacidade dos arcos.

4. Algoritmos Genéticos

Nesta seção, será feita uma breve revisão bibliográfica sobre algoritmos genéticos (AG's). A formulação inicial para os algoritmos genéticos é devida a Holland [6], em 1975. A partir dos anos 80, os algoritmos genéticos passaram a ser utilizados para a solução de problemas diversos de otimização, em especial envolvendo problemas para os quais os algoritmos exatos desenvolvidos não obtiveram sucesso em sua resolução ou, de outro lado, possuem alto custo computacional.

Os algoritmos genéticos mantêm uma população (soluções) que, durante cada geração, passa a ser qualificada por sua efetividade como solução predominante. Uma nova população candidata é formada por operadores genéticos, como o operador de reprodução (seleção), o operador de cruzamento (recombinação) e o operador de mutação. Um fato importante para a eficácia do método incide sobre a escolha dos operadores de cruzamento e mutação.

Os Algoritmos Genéticos reproduzem um modelo simplificado de evolução das espécies através de iterações. Partindo de uma população inicial, é associado, a cada indivíduo desta população, um valor de aptidão, que determina o quanto um indivíduo está adaptado ao ambiente em que vive, determinando suas chances de sobrevivência. Após um processo de seleção, os indivíduos escolhidos para permanecer na população são, então, recombinados, através de cruzamentos (ou recombinações) e mutações. A partir daí, o processo se repete, sendo que, a cada iteração, deseja-se obter um melhor valor de aptidão médio para a população. A Figura 2 mostra o pseudocódigo de um AG básico.

Na operação de recombinação, os genes de dois pais são combinados, de forma a gerar filhos, sendo que, em cada filho, há um conjunto de genes de cada um dos pais. A operação de mutação altera aleatoriamente uma parte dos genes de cada indivíduo-pai. É importante lembrar que as escolhas dos parâmetros do AG têm impacto direto no desempenho do mesmo.

```

procedimento AG
1)  $t \leftarrow 0$ ;
2) Gere a população inicial  $P(t)$ ;
3) Avalie  $P(t)$ ;
4) enquanto (os critérios de parada
   não estiverem satisfeitos) faça
5)    $t \leftarrow t + 1$ ;
6)   Gere  $P(t)$  a partir de  $P(t - 1)$ ;
7)   Avalie  $P(t)$ ;
8)   Defina a população sobrevivente;
9) fim-enquanto;
fm AG;

```

Figura 2: Pseudo código de um Algoritmo Genético básico.

5. Descida Randômica

O método da Descida Randômica [4] 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 de sua versão clássica. Neste método, 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*. A Figura 3 apresenta o pseudo-código da heurística de busca local Busca Randômica.

```

procedimento DescidaRandomica
1)  $iter \leftarrow 0$ ;
2) enquanto ( $iter < itermax$ ) faça
3)    $iter \leftarrow iter + 1$ ;
4)   Selecione aleatoriamente  $x' \in \mathcal{N}(x)$ ;
5)   se ( $f(x') < f(x)$ ) então;
6)      $iter = 0$ ;
7)      $x \leftarrow x'$ ;
8)   fim-se;
9) fim-enquanto;
10) Retorne  $x$ ;
fm AG;

```

Figura 3: Pseudo código do Método da Descida Randômico.

6. Algoritmo Genético Aplicado ao PFMI

Nesta seção, são discutidas características específicas do AG aplicado ao problema de fluxo multiproduto inteiro.

- **Representação da solução:** Uma solução é representada por uma matriz x de dimensão $a \times p$, na qual são representados os fluxos dos produtos em cada arco. Cada elemento dessa matriz é uma variável, de modo que o elemento x_{ij} representa o fluxo do produto j no arco i ; cada coluna j representa o fluxo do produto j ; e cada linha i representa o fluxo dos produtos j , $j = 1, \dots, p$, no arco i .
- **Representação da população:** A representação da solução é feita na forma de uma estrutura de dados, 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.
- **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. Primeiramente, é escolhido, de forma aleatória, um produto, e, em seguida, é traçada uma nova rota para ele. Esta nova rota atende às restrições de conservação de fluxo associada ao par origem-destino. Obviamente, em caso de melhora da função de avaliação, haverá a troca da solução ótima corrente.
- **Geração da População Inicial:** Para gerar os indivíduos da população inicial, foi implementada uma heurística construtiva, que garante factibilidade quanto às restrições de conservação de fluxo e ainda mantém a aleatoriedade, característica importante 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:** A seleção dos indivíduos para a recombinação é feita de forma aleatória entre todos os indivíduos da população, sendo que cada indivíduo possui probabilidade uniforme.
- **Cruzamento:** Seleciona-se aleatoriamente dois pais. Escolhe-se um ponto de corte menor do que a quantidade de produtos, e, em seguida, gera-se dois filhos, sendo que “Filho 1” terá as colunas correspondentes ao produto 1 até o ponto de corte do pai 1, e as demais colunas do pai 2; “Filho 2” terá as colunas correspondentes ao produto 1 até o ponto de corte do pai 2, e as demais colunas do pai 1. É gerada uma quantidade de filhos igual à metade do tamanho da população.
- **Mutação:** Não foi utilizado nenhum operador de mutação.

- **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 Avaliação:** Uma solução x é avaliada a partir de uma função f , definida como:

$$f(x) = \mathbf{Tr}(cx) + \alpha v \quad (6.1)$$

na qual c é uma matriz de dimensão $p \times a$, em que cada linha corresponde aos custos dos produtos em cada arco. A matriz x é a matriz solução, de dimensão $a \times p$; v é o valor do somatório das violações das restrições de capacidade nos arcos e α é um parâmetro de penalização. Desse modo, a primeira parcela desta função de avaliação é a própria função objetivo do problema, como definida em (3.1); a segunda parcela corresponde ao termo de penalização das violações das restrições.

7. Resultados Computacionais

Para a realização dos testes computacionais, foram usadas as instâncias contidas no pacote *carbin*, geradas aleatoriamente pelo programa GenMCF (*Generator Multi-commodity Flow*), desenvolvido por [1].

As instâncias foram classificadas em grupos, de acordo com suas características, sendo que os parâmetros do AG 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, no Grupo 1, estão as instâncias de *bl01 a bl04*; no Grupo 2, estão as instâncias de *bl05 a bl08* e assim por diante.

A alteração dos parâmetros por grupo se justifica pelas características das instâncias, pois, em instâncias de maiores dimensões, como no caso do *Grupo 04*, utilizar os mesmos parâmetros utilizados no *Grupo 01* poderia tornar o processo inviável computacionalmente.

Tabela 1: Definição dos grupos de instâncias.

Grupo	Instâncias	$\#N$	$\#A$	$\#P$
01	bl01 - bl04	32	96	48
02	bl05 - bl08	32	320	48
03	bl09 - bl12	32	96	192
04	bl13 - bl16	32	320	192

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.

Na Tabela 2, $qmax$ é um parâmetro utilizado na heurística construtiva, utilizada para gerar a população inicial. Esta heurística não foi descrita neste trabalho;

Tabela 2: Parâmetros utilizados nos testes.

Grupo	qmax	maxger	nind	itermax	filhos	alfa
01	70	80	200	100	nind/2	200000
02	70	80	160	90	nind/2	80000
03	70	300	200	30	nind/2	80000
04	70	350	100	25	nind/2	70000

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; *filhos* é o parâmetro que representa a quantidade de filhos gerados em cada geração; e, finalmente, *alfa* é o parâmetro de penalização das soluções não factíveis, descrito anteriormente.

Tabela 3: Resultados dos testes computacionais.

Instância	fo_AG	f*	Tempo(s)	Violação
bl01	1.621.156	1.615.947	451,219	0
bl02	1.816.947	1.816.947	486,062	0
bl03	17.361	17.340	367,829	7
bl04	19.235	21.370	417,937	9
bl05	510.768	474.782	830,844	0
bl06	467.012	411.480	740,14	0
bl07	5.982	5.751	765,047	0
bl08	5.929	5.688	714,219	0
bl09	6.610.068	6.261.671	1.510,625	19
bl10**	7.302.614	-	1.767,484	13
bl11	70.326	69.018	1.940,625	2
bl12	72.400	65.902	1.533,078	0
bl13	3.740.119	3.132.695	2.940,312	0
bl14	3.451.148	2.433.011	2.675,922	0
bl15	46.809	34.274	2.511,672	2
bl16	39.197	28.074	2.754,328	2

Os resultados são apresentados na Tabela 3. Nesta tabela, a coluna *fo_AG* apresenta os resultados obtidos, enquanto que *fo* apresenta os valores ótimos obtidos por [1]. A coluna *Tempo* apresenta o tempo de execução em segundos do algoritmo proposto neste trabalho, enquanto a coluna *Violação* apresenta o somatório das violações das restrições de capacidade em cada arco. No caso da violação ser nula, este valor significa que o algoritmo proposto encontrou uma solução factível. No caso da instância *bl02*, o algoritmo encontrou a solução ótima do problema.

O método obteve soluções próximas das ótimas para a maior parte das instâncias. O melhor resultado obtido foi para a instância *bl10*, para a qual nenhuma solução factível foi encontrada em [1].

Observou-se, nos testes realizados, que o parâmetro *itermax*, correspondente ao número máximo de iterações sem melhora na Descida Randômica, tem papel significativo no custo computacional, já que esta heurística é executada em todos os indivíduos no AG, sejam eles da população inicial ou filhos.

8. Conclusões

As características das instâncias influenciaram na solução final, além dos parâmetros escolhidos, pois o algoritmo encontrou dificuldade para chegar a uma solução factível em instâncias cuja relação $\frac{\text{total arcos}}{\text{total produtos}}$ é menor, como no caso das instâncias representadas pelo Grupo 02 na Tabela 1. Neste caso, além desse problema, houve convergência prematura para valores pequenos de *nind*. Na tentativa de resolver este último problema, o valor de *nind* foi aumentado, na busca de maior diversidade das soluções. Em instâncias com maiores quantidades de arcos, o algoritmo encontrou maior facilidade na busca de uma solução boa. Os melhores resultados foram encontrados nas instâncias representadas pelo Grupo 02 da Tabela 2, por apresentarem maior dimensionamento quanto à quantidade de arcos, sendo a mesma quantidade de nós e produtos das instâncias apresentadas na Tabela 1.

Por fim, conclui-se que o método proposto obteve soluções próximas das ótimas para a maior parte das instâncias, tendo seu melhor resultado para a instância *bl10*, para a qual nenhuma solução factível foi encontrada em [1].

Abstract. This paper analyses the behavior of a population evolutionary algorithm applied to the solution of Multicommodity flow integer problems. This problems belongs to the NP-Hard class problems, being the use of heuristic techniques justified by the high dimension of the problem when compared with the amount of variables and restrictions. In this work, it is proposed an approach via genetic algorithms, associated with a local search heuristic, in order to solve the capacitated MFIP. It is sought to determine the flow of products by the arcs of the network at the lowest possible cost, subject to flow conservation and capacity constraints. Heuristic descent methods are applied to initial population of the genetic method. The results obtained from the application of the method are compared with the ones obtained from the application of exact methods. In special, it was found feasible solution for one instance no-solved via exact method.

Keywords. Multicommodity Flow Problem, Genetic Algorithms, Heuristic.

Referências

- [1] F.P. Alvelos, “Branch-and-Price and Multicommodity Flows”, Tese de Doutorado, EPS, Universidade do Minho, Portugal, 2005.

- [2] 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.
- [3] M. Erickson, M.G.C. Resende, P.M. Pardalos, P.M. (2002), A genetic algorithm for the weight setting problem in OSPF routing, *Combinatorial Optimization*, **6**, No. 3 (2002), 299-333.
- [4] A.T. Ernst, M. Krishnamoorthy, Solution algorithms for the capacitated single allocation hub location problem, *Annals of Operations Research*, **86** (1999), 141-159.
- [5] D.R. Fulkerson, L.R. Ford, “Flows in Networks”, Princeton University Press, NJ, 1962.
- [6] J. Holland, “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, (1975).
- [7] T.C. Hu, Multicommodity network flows, *Operations Research*, **11** (1963), 344-360.
- [8] R. Milidui, Um algoritmo grasp para o problema de transporte de derivados de petróleo em oleodutos, em “Anais do XXXIII Simpósio Brasileiro de Pesquisa Operacional”, 237-246, 2001.
- [9] M. Pioró, P. Gajowniczek, Solving multicommodity integral flow problem by simulated allocation, *Telecommunication Systems*, **7**, 17-28, (1997).
- [10] M.G.C. Resende, C.C. Ribeiro, A GRASP with path-relinking for private virtual circuit routing, *Networks*, **41**, No. 2 (2003), 104-114.
- [11] R.R. Schultz, G.L. Meyer, An interior point method for block angular optimization, *SIAM Journal on Optimization*, **42**, (1991).
- [12] E. Tailard, G. Agazzi, L.M. Gambardela, A multiple ant colony system for vehicle routing problems with time windows, em “New Ideas in Optimization” (D. Corne et al., ed.),