
ALGORITMOS PARA O PROBLEMA NÃO CAPACITADO DE FLUXOS COM CUSTOS FIXOS NOS ARCOS: UMA COMPARAÇÃO ESTATÍSTICA

Frederico R.B. Cruz¹

Enrico A. Colosimo¹

Geraldo R. Mateus²

¹ Departamento de Estatística

² Departamento de Ciência da Computação

Universidade Federal de Minas Gerais

Belo Horizonte – MG

Recebido em 10/1999, aceito em 06/2001 após 1 revisão

Resumo

Este trabalho tem como propósito a apresentação de resultados de uma comparação empírica entre algoritmos, sendo este um dos assuntos mais recorrentes na área de desenvolvimento de algoritmos. Os algoritmos sob estudo são para resolver um problema de otimização em redes, importante pelas suas aplicações potenciais em sistemas de telefonia e transporte, o problema não capacitado de fluxos com custos fixos nos arcos (NCFCF), uma generalização do clássico problema de Steiner em grafos. Para tal, são utilizadas ferramentas estatísticas conhecidas tais como planejamento de experimentos, análise de variância e intervalos de confiança, mas não comumente empregadas neste tipo de estudo. O problema NCFCF é apresentado em uma modelagem de programação matemática inteira mista, baseada na qual os algoritmos sob consideração são apresentados. Uma descrição do planejamento de experimentos adequado a este tipo de estudo é apresentada e é ilustrado o uso da técnica estatística baseado em cuja análise foi possível classificar os algoritmos sob consideração.

Palavras-chave: otimização em redes, problemas de Steiner, ANOVA.

Abstract

This paper is concerned about empirical comparisons of algorithms, one of the most recurrent issues in the field of design of algorithms. The problem under consideration is the uncapacitated fixed-charge network flow (UFCNF) problem, a generalization of the classic Steiner problem in graphs. The UFCNF is very important in the practical point of view because of its potential applications for telecommunication and transportation system design. In order to compare the algorithms for the UFCNF problem, we make use of well known and well established statistical tools namely the design of experiments, analysis of variance, and confidence intervals, but rarely applied in such studies. A mixed-integer mathematical programming formulation is presented for the UFCNF problem and the design of experiments suitable for the empirical study is detailed. Finally, the statistical analysis based on which the algorithms could be classified is presented.

Keywords: network design, Steiner problems in graphs, ANOVA.

1. Introdução

Os problemas de otimização em redes representam uma grande classe dentre os problemas de otimização. Como característica fundamental tais problemas são definidos em grafos $G=(N,E)$ nos quais N representa o conjunto de nós e E , o conjunto de arestas, *i.e.*, pares de nós. Um grafo de pequeno tamanho está representado na Figura 1. Neste caso, as arestas estão direcionadas e são denominadas *arcos*. Este tipo de grafo é denominado *dígrafo*. Podemos observar a presença de quatro nós, $N=\{i,j,k,l\}$, e sete arcos, $E=\{(i,j),(j,i),(i,k),(j,k),(k,j),(k,l),(l,k)\}$. No dígrafo $G=(N,E)$, os nós poderiam estar representando quatro regiões de uma cidade e os arcos, as principais vias de acesso entre as localidades.

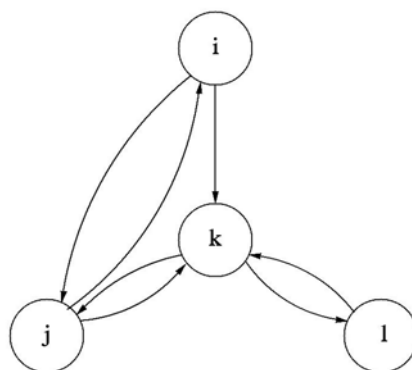


Figura 1 – Grafo $G=(N=\{i,j,k,l\}, E=\{(i,j),(j,i),(i,k),(j,k),(k,j),(k,l),(l,k)\})$

O problema para cuja solução pretendemos apresentar e comparar algoritmos é o chamado problema **não capacitado de fluxos com custos fixos nos arcos**, NCFCF (Nemhauser & Wolsey, 1988). O problema NCFCF pertence à grande classe de problemas de otimização em redes formando ele próprio uma subclasse bem peculiar de modelos com muitas aplicações práticas de interesse. Para citar apenas alguns poucos exemplos, esse modelo genérico tem aplicações em problemas de planejamento de redes de distribuição, redes de transportes e de telecomunicações. Leitores interessados podem encontrar maiores detalhes em Cruz *et al.* (1994, 1998).

No problema NCFCF, estão envolvidos custos de utilização dos arcos e custos dependentes do montante de fluxo suportado pelo arco. O objetivo é determinar uma combinação de arcos com custo total mínimo (custos de utilização mais custos dependentes) que conduza os fluxos dos nós de oferta a todos os nós de demanda possivelmente passando por nós intermediários de transbordo, também conhecidos como nós de *Steiner*. Este é claramente um problema de otimização da classe NP-árdua (do inglês *NP-hard*), uma vez que generaliza entre outros o conhecido problema de Steiner em grafos (Maculan, 1987), que é NP-árduo (Garey & Johnson, 1979). De fato, a prova da sua NP-complexidade pode ser facilmente desenvolvida (Cruz *et al.*, 1994).

Um problema NCFCF definido sobre o dígrafo da Figura 1 poderia ter o seguinte enunciado. *Quais seriam as estradas (arcos) a serem recuperadas, para escoar a produção da região i até os centros consumidores representados pelas regiões j e l gastando o mínimo possível?* Supomos existir um custo fixo associado à escolha da estrada, representando obras de infra-estrutura, e

um custo variável de utilização da estrada, representando obras de manutenção. A Figura 2 ilustra uma solução para o problema que pode ser a ótima dependendo dos custos fixos e variáveis considerados.

Um dos problemas mais recorrentes na área de desenvolvimento de algoritmos é o de compará-los e classificá-los adequadamente. Frequentemente, há restrições de custo/tempo e um número infinito de possíveis instâncias de teste. Há também uma variabilidade intrínseca ao processo de aquisição de algumas medidas de desempenho (*e.g.* tempo de processamento). Surge daí a necessidade do uso de alguma ferramenta estatística. O nosso interesse se concentra especificamente na comparação de algoritmos para o problema NCFCF para o qual certamente tal dificuldade não poderia deixar de ocorrer. Entretanto, esta deve ser também a preocupação de desenvolvedores de algoritmos para outras finalidades.

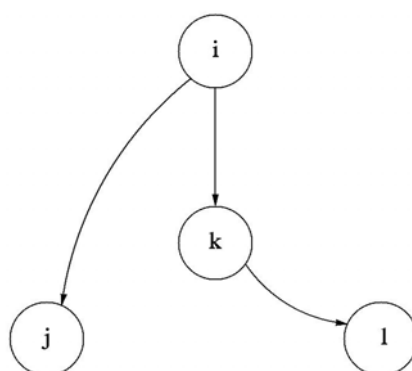


Figura 2 – Uma Solução para o Problema NCFCF

A comparação de vários algoritmos (tratamentos) com relação a uma medida de desempenho (resposta) envolve um estudo experimental e a análise estatística dos dados resultantes, pois há um número infinito de instâncias passíveis de ser consideradas e mesmo em computadores existe variação entre as medidas de desempenho de programas de execução para execução. Experimentar com alocação aleatória dos “tratamentos” às execuções é então útil para gerar as distribuições necessárias às comparações necessariamente feitas com estatística.

Em estudos envolvendo algoritmos, é razoável que se considerem *classes de problemas* e que instâncias destas classes venham a ser resolvidas por todos os *algoritmos* sob estudo. O experimento correspondente consta de duas aleatorizações. Uma é sobre as classes de problemas, ou seja, são escolhidas aleatoriamente instâncias de uma determinada classe. A outra, dada uma instância de determinada classe, é feita selecionando-se a ordem de aplicação dos algoritmos. Este planejamento é descrito por Neter *et al.* (1990) como experimento a dois fatores com medidas repetidas em um dos fatores e vem sendo utilizado com frequência em várias áreas como biologia, engenharia e ciências sociais. Na comparação de algoritmos para o problema NCFCF, será considerado um estudo experimental com dois fatores (*classes de problemas* e *algoritmos*) com medidas repetidas no fator *algoritmos*.

Este artigo está organizado do seguinte modo. Na Seção 2, apresentamos mais detalhadamente o problema NCFCF. Formalizamos o problema NCFCF como um modelo de programação matemática inteira mista e apresentamos os algoritmos para a sua resolução. A Seção 3 é

dedicada à apresentação dos resultados obtidos cuja análise e discussão se encontram na Seção 4. Finalmente, a Seção 5 é dedicada às conclusões finais.

2. Formulação do Problema NCFCF e Algoritmos

2.1 Modelo Matemático

É muito importante formalizar matematicamente o problema NCFCF, pois os algoritmos de resolução são derivados diretamente da sua formulação. Um modelo de programação matemática inteira mista típica para o problema (Rardin & Wolsey, 1993) definida sobre o grafo direcionado $G=(N,E)$, é apresentado a seguir.

Modelo **M**:

$$\min \sum_{(i,j) \in A} (c_{ij}x_{ij} + f_{ij}y_{ij}) \quad (1)$$

sujeito a:

$$\sum_{j \in \delta^-(i)} x_{ji} - \sum_{j \in \delta^+(i)} x_{ij} = \begin{cases} -\sum_{k \in D} d_k, & \forall i \in S, \\ 0, & \forall i \in T, \\ d_i, & \forall i \in D, \end{cases} \quad (2)$$

$$x_{ij} \leq \left(\sum_{k \in D} d_k \right) y_{ij}, \forall (i,j) \in E, \quad (3)$$

$$x_{ij} \geq 0, \forall (i,j) \in E, \quad (4)$$

$$y_{ij} \in \{0,1\}, \forall (i,j) \in E, \quad (5)$$

onde c_{ij} é o custo variável por unidade de fluxo no arco (i,j) , x_{ij} é a quantidade de fluxo no arco, f_{ij} é o custo fixo pelo uso do arco (i,j) para transmissão do fluxo, y_{ij} é uma variável binária de decisão que assume o valor 0 se o arco (i,j) não está sendo usado e 1, caso contrário, $\delta^+(i) = \{j \mid (i,j) \in E\}$, $\delta^-(i) = \{j \mid (j,i) \in E\}$, d_i é a quantidade demandada pelo nó i , S é o conjunto de nós de oferta, T é o conjunto de nós de *Steiner*, transbordo, e D é o conjunto de nós de demanda.

A função objetivo (1) minimiza o custo total variável associado aos fluxos mais o custo total fixo associado ao uso dos arcos. As restrições (2) garantem a conservação dos fluxos nos diversos nós. Para os nós de oferta, a quantidade de fluxo que chega menos a que sai deve ser igual ao somatório das demandas com o sinal trocado. Essa parte da restrição é redundante, ou seja, a sua presença/retirada não altera o conjunto de soluções viáveis do problema. Conseqüentemente, também não altera a solução ótima. Nos nós de transbordo, a quantidade de fluxo que chega deve ser igual à que sai. Finalmente, nos nós de demanda, a quantidade de fluxo que chega menos a que sai deve ser igual à demanda interna do nó d_i . As restrições (3) asseguram que não haja fluxo através do arco (i,j) a menos que ele tenha sido selecionado na função objetivo. É curioso notar que as restrições (3) são o que diferencia o problema NCFCF, NP-árduo, do problema linear de fluxos, polinomial.

Com relação ao problema NCFCF aqui tratado, assumimos que o custo fixo f_{ij} seja não negativo, pois caso contrário a variável y_{ij} poderia ser feita igual a 1 e simplesmente eliminada do problema. A variável c_{ij} é irrestrita, mas para garantir que a função objetivo seja limitada por baixo assumimos que não existam *circuitos* com custo negativo.

2.2 Estado da Arte

Por se tratar de um problema de otimização NP-árduo, justificam-se tanto abordagens exatas quanto aproximadas (heurísticas). De fato, vários trabalhos têm sido publicados descrevendo algoritmos exatos e aproximados para o problema NCFCF bem como para alguns casos particulares. Rothfarb *et al.* (1970), modelando o problema de coleta de gás natural de estações em alto mar para locais de armazenamento em terra, desenvolveram um algoritmo para resolver um caso particular no qual era considerada uma estrutura de custos mais simples sem custos variáveis nos arcos. Luna *et al.* (1987) estudaram um modelo um pouco mais complexo e usado para modelar o problema de planejamento de redes locais em telefonia tendo apresentado algoritmos heurísticos e algumas técnicas de otimização local. Outro caso especial, sem nós de transbordo, foi tratado por Magnanti *et al.* (1986) e por Hochbaum & Segev (1989) empregando algoritmos do tipo separação e avaliação (do inglês *branch-and-bound*) associados à técnica de decomposição de Benders e à relaxação Lagrangeana. Maiores detalhes sobre a técnica de relaxação Lagrangeana a ser mencionada mais vezes neste artigo podem ser encontrados no clássico trabalho de Fisher (1985).

Considerando o problema geral NCFCF, notamos contribuições tanto em algoritmos exatos quanto aproximados. O leitor interessado poderá encontrar extenso material no livro de Memhauser & Wolsey (1988) ou mais recentemente no trabalho de Rardin & Wolsey (1993) que aborda modernas técnicas entre as quais a teoria de inequações válidas (do inglês *theory of valid inequalities*). Mateus *et al.* (1994) estudaram procedimentos heurísticos do tipo *ADD* e *DROP*, conhecidos na área. Em Cruz *et al.* (1994), foi proposta uma heurística baseada na relaxação Lagrangeana para solução do problema. Em outro artigo de Cruz *et al.* (1998), um algoritmo exato do tipo separação e avaliação acoplado a testes de redução foi desenvolvido com um desempenho prático satisfatório.

2.3 Algoritmos Considerados

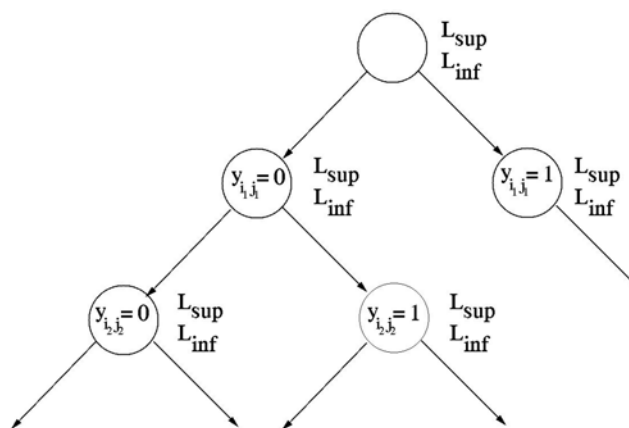


Figura 3 – Algoritmo de Separação e Avaliação

Para questões de conveniência, todos os algoritmos aqui considerados são do tipo separação e avaliação e estão baseados no algoritmo desenvolvido por Cruz *et al.* (1998). Nesta classe de algoritmos, tenta-se resolver o modelo **M** gerando limites superiores (*i.e.* soluções *viáveis*) e limites inferiores. Estes últimos são usualmente determinados por alguma *relaxação* do problema original. Em Cruz *et al.* (1998), é utilizada a relaxação Lagrangeana para esse fim (vide Fisher, 1985, para maiores detalhes). Se os limites superior e inferior são coincidentes significa que o problema está resolvido sendo que a solução viável é também uma solução *ótima*. Caso contrário, emprega-se algum critério para escolher uma variável y_{ij} (*e.g.* o critério desenvolvido por Cruz *et al.*, 1996) que será então fixada nos dois valores possíveis 0 e 1 resultando em subproblemas menores que terão seus limites inferior e superior novamente computados. O algoritmo, ilustrado na Figura 3, fixará variáveis do problema original tantas vezes quanto necessárias até resolvê-lo.

Conforme verificado empiricamente por Cruz *et al.* (1998), os algoritmos de separação e avaliação têm um desempenho satisfatório e resolvem instâncias de interesse prático de tamanho moderado em um tempo relativamente curto. Entretanto, os limites inferiores gerados pela relaxação Lagrangeana obtida diretamente do modelo **M** são *fracos*, *i.e.* são muito distantes do limite superior correspondente. Pretende-se então apresentar resultados para algoritmos derivados de uma modificação na formulação original do modelo **M**. O tipo de modificação é utilizado corriqueiramente em problemas do gênero e consiste em acrescentar ao modelo em estudo uma restrição *redundante*. Assim o modelo **M** ganha a restrição:

$$|Y| \geq \beta, \quad (6)$$

onde Y é o conjunto de arcos presentes na solução, $Y = \{(i,j) \mid y_{ij}=1\}$, $|\bullet|$ é a cardinalidade do conjunto \bullet e Y e β é um limite *inferior* qualquer para esta cardinalidade.

Somente para *pequenos* valores de β a restrição (6) é redundante. Isto pode ser facilmente visto tomando-se como exemplo a solução apresentada na Figura 2 e considerando-se que a mesma seja ótima. A utilização da formulação **M** com a restrição adicional (6) e um β menor ou igual a 3 (o número de arcos presentes na solução ótima) claramente não alteraria o ótimo. Ao contrário, se fosse utilizado um β superior a 3, forçasse-ia a inclusão de um arco adicional. Note-se que o problema com β superior a 7 não teria solução viável.

Um fato a considerar é que a relaxação Lagrangeana derivada do modelo **M** mais a restrição (6) incluirá para sua resolução um procedimento de ordenação para o qual a melhor implementação conhecida tem complexidade de tempo de execução $O(|E|\log|E|)$ no caso médio (Ziviani, 1993). Uma questão importante é avaliar se a restrição (6) traria ganho. Ou seja, deseja-se descobrir se essa restrição adicional mesmo aumentando a complexidade computacional da nova relaxação seria capaz de aumentar os limites inferiores gerados o suficiente para conseguir melhorar globalmente o desempenho do algoritmo de separação e avaliação. Ainda em outras palavras, é de interesse prático saber se este novo algoritmo derivado da inclusão da restrição (6) resolve instâncias maiores do problema NCFCF e mais rapidamente.

3. Planejamento e Execução dos Experimentos

Uma versão dos vários algoritmos foi codificada pelos autores na linguagem C++ e está disponível a pedido. Todos os testes aqui apresentados foram executados no Laboratório de Pesquisa Operacional (LaPO) do Departamento de Ciência da Computação da UFMG,

utilizando-se uma estação de trabalho *Sun Ultra 1 Modelo 140* com 128 MBytes de RAM e sistema operacional *SunOS Release 5.5.1*. Todos os problemas de teste são derivados de grafos euclidianos gerados aleatoriamente segundo procedimento desenvolvido por Aneja (1980) e já consagrado na área. De acordo com tal procedimento, as posições dos nós, as extremidades dos arcos, os seus pesos, o nó de oferta (consideramos apenas problemas com fonte única, por questão de simplicidade) e os nós de demanda são escolhidos aleatoriamente segundo a distribuição uniforme. Leitores interessados em maiores detalhes são encorajados a consultar o trabalho de Aneja (1980).

Inicialmente, foram definidas 5 classes de problemas de interesse. A primeira corresponde àqueles problemas com razão $f_{ij}/c_{ij}=100$. Esses são problemas *próximos* ao problema de *Steiner* em grafos e muito difíceis de ser resolvidos (NP-árduos). Para tais, esperam-se tempos de processamento maiores. Por outro lado, a última classe corresponde àqueles problemas com razão $f_{ij}/c_{ij}=0,01$, *próximos* ao problema linear de fluxos, computacionalmente fáceis (polinomiais). Para estes, esperam-se tempos de processamento menores. A classe de problemas é o correspondente ao fator *A* (Neter *et al.*, 1990).

O algoritmo é o correspondente ao fator *B*, para o qual incluem-se medidas repetidas. Os quatro algoritmos considerados para resolver cada um dos problemas testes são descritos a seguir. O primeiro corresponde a resolver o problema usando o modelo sem a restrição adicional significando que a cardinalidade do conjunto *Y* foi deixada livre, *i.e.* não havia um limite para o número mínimo de arcos presentes na solução do problema. O segundo algoritmo também deixa a cardinalidade do conjunto livre, mas considera a restrição (6). Esses dois casos foram considerados unicamente com o objetivo de verificar empiricamente o impacto na dificuldade computacional da inclusão da restrição (6) no modelo *M* e será mostrado que é significativo. Os demais casos fixam a cardinalidade em dois valores. O terceiro considera uma cardinalidade mínima óbvia e imediata pela consideração de que deve haver pelo menos um arco entrando em cada um dos nós de demanda, o que significa que $|Y|=|D|$. Entretanto, conforme mostraremos na seção de análise dos resultados, esse limite não é eficaz, pois não consegue reduzir substancialmente o número de separações e avaliações nem o tempo de processamento. O quarto utiliza o limite mais justo possível. Sobre a determinação desses limites justos, é importante ressaltar que, infelizmente, no estágio atual da nossa pesquisa, somente conseguimos calculá-los pela resolução completa do problema NCFCF através do primeiro algoritmo.

Uma discussão mais aprofundada sobre o tamanho da amostra para cada combinação *classe x algoritmo* foge ao escopo deste trabalho. Entretanto, é importante ressaltar que este número não deve ser muito elevado, pois a consideração de apenas 5 instâncias por combinação *classe x algoritmo* resultou na realização de nada menos que 100 experimentos, mostrados na Tabela 1. Como procuram-se apenas indícios da superioridade do algoritmo proposto, por comodidade optou-se por considerar apenas estas 5 instâncias por combinação. Sendo observado o indício, mais experimentos poderiam ser realizados, possivelmente reduzindo-se o número de níveis nos fatores (*e.g.* apenas duas classes e dois algoritmos).

A Tabela 1 apresenta os resultados obtidos. Para cada uma das vinte e cinco instâncias aleatórias testadas, mostramos a razão f_{ij}/c_{ij} e o número de arcos presentes na solução ótima, $|Y|$. Também apresentamos para cada um dos quatro algoritmos testados o número total de separações e avaliações (coluna *Nós*) e o tempo total (em segundos) gasto para processamento.

Tabela 1 – Resultados Computacionais ($|N|=16$, $|E|=60$ e $|D|=6$)

Classe	f_{ij}/c_{ij}	Instância	$ Y ^*$	Modelo M com a Restrição Adicional (6)							
				Modelo M		$\beta=0$		$\beta= D $		$\beta= Y ^*$	
				Nós	UCP (s)	Nós	UCP (s)	Nós	UCP (s)	Nós	UCP (s)
100	B01	10	3.875	110,00	3.875	170,00	6.053	270,00	2.231	110,00	
	B02	9	201	8,10	201	12,00	215	14,00	65	5,00	
	B03	7	157	6,70	157	11,00	283	17,00	135	7,70	
	B04	7	359	12,00	359	17,00	255	12,00	55	3,40	
	B05	8	683	21,00	683	33,00	831	41,00	519	25,00	
10	B06	10	2.533	71,00	2.533	110,00	4.201	180,00	1.431	69,00	
	B07	9	183	7,00	183	10,00	183	11,00	49	3,40	
	B08	7	97	4,40	97	6,90	263	15,00	131	7,30	
	B09	7	119	4,20	119	6,40	63	3,40	15	1,10	
	B10	8	497	15,00	497	23,00	565	26,00	347	17,00	
1	B11	10	245	7,70	245	12,00	369	18,00	147	7,00	
	B12	9	59	2,50	59	3,90	133	8,30	31	2,20	
	B13	8	25	1,60	25	2,50	45	2,90	3	0,48	
	B14	7	33	1,10	33	1,70	29	1,30	9	0,73	
	B15	8	77	2,90	77	4,50	77	4,20	49	2,40	
0,1	B16	10	7	0,52	7	0,77	11	0,96	5	0,55	
	B17	9	61	1,70	61	2,70	71	3,10	25	1,30	
	B18	8	1	0,20	1	0,29	1	0,28	1	0,27	
	B19	7	9	0,46	9	0,70	9	0,66	9	0,70	
	B20	8	1	0,22	1	0,32	13	0,82	1	0,35	
0,01	B21	10	1	0,20	1	0,28	1	0,28	1	0,31	
	B22	8	25	0,73	25	1,10	25	1,10	1	0,31	
	B23	8	1	0,20	1	0,29	1	0,28	1	0,27	
	B24	7	1	0,19	1	0,26	1	0,26	1	0,26	
	B25	8	1	0,20	1	0,29	1	0,28	1	0,28	

4. Análise dos Resultados

Na análise de variância, ambos os fatores A e B são de interesse, bem como a possível interação entre eles. Por questão de simplicidade, consideramos apenas o tempo de processamento como variável resposta de interesse, mas ressaltamos que poderíamos estar interessados também em comparar os algoritmos por outro critério de desempenho, tal como por exemplo quanto ao número total de separações e avaliações. A transformação logarítmica é uma das mais utilizadas em situações cuja resposta é o tempo até a ocorrência de um evento de interesse (Lawless, 1982) para tornar simétrica a sua distribuição. Esta foi a transformação usada na nossa análise. Em outras palavras, consideramos como variável resposta de interesse o logaritmo dos tempos de UCP. Feito isto, foi ajustado o seguinte modelo linear, adequado ao delineamento experimental anteriormente descrito:

$$Y_{ijk} = \mu_{...} + \rho_{i(j)} + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \varepsilon_{(ijk)}, \quad (7)$$

onde $\mu_{...}$ é o efeito geral da média, $\rho_{i(j)}$ são componentes de variância independentes com distribuição $N(0, \sigma_\rho^2)$, α_j é o efeito do fator A sujeito à restrição $\sum \alpha_j = 0$, β_k é o efeito do fator B sujeito à restrição $\sum \beta_k = 0$, $(\alpha\beta)_{jk}$ é o efeito da interação entre os dois fatores, sujeito à restrição $\sum (\alpha\beta)_{jk} = 0$ tanto em relação a j quanto em relação a k e $\varepsilon_{(ijk)}$ é o erro aleatório com distribuição $N(0, \sigma^2)$ e independente de $\rho_{i(j)}$, para $i=1, \dots, n; j=1, \dots, a; k=1, \dots, b$.

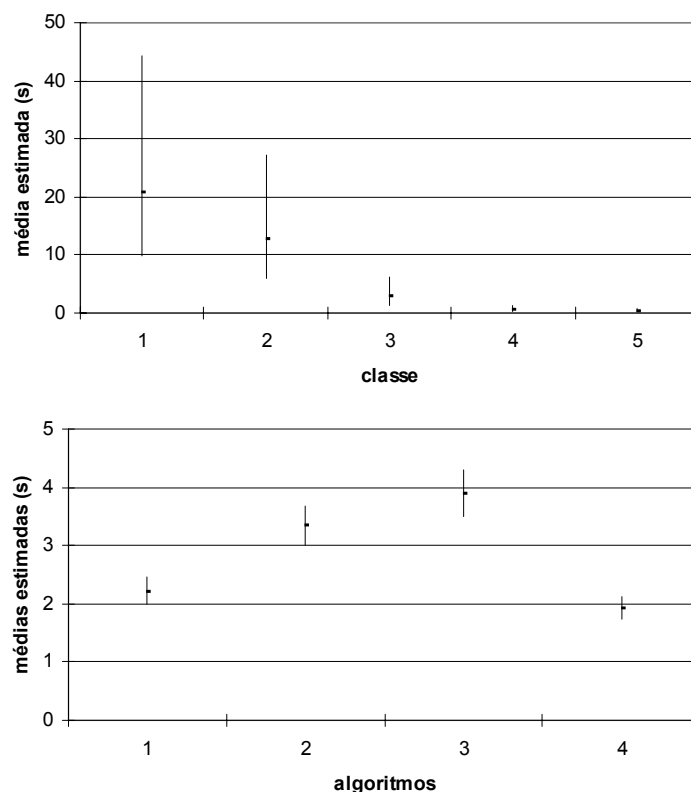
Todas as suposições associadas ao modelo (7) foram verificadas e confirmadas (vide detalhes no Anexo). Na Tabela 2, apresentamos o resultado do ajuste do modelo obtido utilizando o Minitab® (Minitab Inc., 2000). Pela coluna de p -valores, notamos que tanto as classes de problemas quanto os algoritmos têm um efeito significativo na variável resposta. Por outro lado, não há evidência de que exista interação entre estes fatores. Podemos observar uma grande variabilidade dentro das classes de problemas indicando ser alta a heterogeneidade segundo este fator. Em outras palavras, a relação f_{ij}/c_{ij} tem grande influência no tempo de resolução. Se esta relação é alta o problema “tende” mais fortemente para um problema NP-árduo e exige mais esforço computacional.

Tabela 2 – Análise de Variância (planejamento balanceado)

Fonte de Variação	Graus de Liberdade	Soma de Quadrados	Quadrado Médio	F	p -valor
Classe	4	268.195	67.049	17.45	0.000
instância (classe)	20	76.835	3.842	42.37	0.000
Algoritmo	3	8.261	2.754	30.37	0.000
classe x algoritmo	12	1.494	0.125	1.37	0.204
Erro	60	5.440	0.091	–	–
Total	99	360.226	–	–	–

Prosseguindo, formamos intervalos de 90% de confiança para as médias, apresentados na Figura 4. Os dados obtidos para o algoritmo 3 mostram fortes evidências de que a inclusão da restrição (6) não melhora em nada o desempenho médio do algoritmo de Cruz et al. (1998), a não ser que um limite muito forte (correspondente ao algoritmo 4) fosse utilizado. Os dados também confirmam a nossa conjectura de que problemas com uma razão f_{ij}/c_{ij} alta, mais “próximos” do problema de Steiner em grafos, NP-árduo, são significativamente mais difíceis de tratar não importando qual algoritmo esteja sendo usado (pois a análise de variância não indicou interação significativa entre os fatores *classe de problema* e *algoritmo*).

Outro aspecto notável à luz das amplitudes dos intervalos de confiança para as diversas classes de problemas (Figura 4) é a natureza praticamente determinística dos tempos de UCP para a resolução de problemas da classe 5, “próximos” de um problema linear de fluxos. Por outro lado, é difícil prever o tempo de UCP para resolução de um problema dentro da classe 1, que “tende” a um problema de Steiner puro, NP-árduo.

**Figura 4** – Intervalos de 90% de Confiança

5. Conclusões e Observações Finais

Apresentamos e discutimos a importância de um problema de otimização em redes, o problema não capacitado de fluxos com custos fixos nos arcos (NCFCF). Formulamos o problema NCFCF como um modelo de programação matemática inteira mista para uma apresentação conveniente dos algoritmos que estamos comparando estatisticamente. Este é um problema recorrente na área de desenvolvimento de algoritmos. Utilizamos as técnicas de planejamento de experimentos, análise de variância de um planejamento com medidas repetidas em um fator e construção de intervalos de confiança para comparar os quatro algoritmos de interesse. Os algoritmos propostos se basearam na conhecida idéia de utilização de restrições redundantes e pelos experimentos realizados foi possível demonstrar que se pode alcançar um desempenho superior com a estratégia. Entretanto, os resultados também mostram que o desempenho global do algoritmo pode ser colocado em risco se a restrição não produz a cada iteração melhorias suficientes para compensarem o incremento que induz na complexidade computacional. O dilema parece ser insolúvel para o caso geral e a experimentação com tratamento estatístico dos dados parece ser uma ferramenta eficaz em casos particulares. Neste sentido, a metodologia estatística aqui ilustrada pode ser útil nesta tomada de decisão, pois por ser geral pode ser aplicada a qualquer outro experimento que envolva comparações múltiplas de natureza semelhante.

Trabalhos futuros poderiam envolver a comparação dos algoritmos descritos segundo outros critérios de desempenho (*e.g.* o espaço de memória máximo exigido). Poderiam incluir a investigação de outros tipos de algoritmos. Em particular, algoritmos paralelos apresentam peculiaridades para a tomada de suas medidas de desempenho (Cruz & Mateus, 2001) pela dificuldade de isolamento das máquinas na realização dos experimentos. Mais este fator (carga do sistema) teria que ser levado em consideração modificando um pouco o delineamento experimental. Adicionalmente, outros tipos de planejamentos poderiam ser considerados para verificação da sua adequação ao problema de comparação múltipla de algoritmos (*e.g.* o delineamento em parcelas subdivididas). Em Neter *et al.* (1990), o tema é apresentado em abrangência sendo portanto uma fonte de consulta recomendável para outras situações envolvendo outros tipos de planejamentos.

Agradecimentos

Várias pessoas contribuíram para a realização deste trabalho em diversas etapas do seu desenvolvimento. Em particular, gostaríamos de deixar expressos os nossos agradecimentos aos Professores Emilio Suyama e Rosangela H. Loschi, do ICEX-UFMG, e Rinaldo Artes, do IME-USP, pelo estímulo e pelas valiosas discussões, críticas e sugestões. Agradecemos também ao *Conselho Nacional de Desenvolvimento Científico e Tecnológico*, CNPq, processos 301.809/96-8 e 201.046/94-6, à *Fundação de Amparo à Pesquisa de Minas Gerais*, FAPEMIG, processo 855/98, e à *Pró-Reitoria de Pesquisa da UFMG*, PRPq-UFMG, pelo apoio financeiro à nossa pesquisa.

Referências Bibliográficas

- (1) Aneja, Y.P. (1980). An Integer Linear Programming Approach to Steiner Problem in Graphs. *Networks*, **10**, 167-178.
- (2) Cruz, F.R.B.; Almeida, J.A. & Mateus, G.R. (1994). Análise do Problema de Planejamento de Redes Telefônicas de Alimentação. **In:** *Anais do 10º Congresso Brasileiro de Automática*, SBA, Rio de Janeiro, 572-577.
- (3) Cruz, F.R.B.; MacGregor Smith, J. & Mateus, G.R. (1998). Solving to Optimality the Uncapacitated Fixed-Charge Network Flow Problem. *Computers and Operations Research*, **25**(1), 67-81.
- (4) Cruz, F.R.B. & Mateus, G.R. (2001). Parallel Algorithms for a Multi-level Network Optimization Problem. *Parallel Algorithms and Applications* (em impressão).
- (5) Cruz, F.R.B.; Mateus, G.R. & MacGregor Smith, J. (1996). A Branching Strategy for Uncapacitated Fixed-charge Network Flow Problems. **In:** *Annals of the 11th Brazilian Automatic Control Conference*, SBA, São Paulo, 359-364.
- (6) Fisher, M.L. (1985). An Application Oriented Guide to Lagrangean Relaxation. *Interfaces*, **15**, 10-21.
- (7) Garey, M.R. & Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.
- (8) Hochbaum, D.S. & Segev, A. (1989). Analysis of a Flow Problem with Fixed Charges. *Networks*, **19**, 291-312.

- (9) Lawless, J.F. (1982). *Statistical Models and Methods for Lifetime Data*, John Wiley & Sons, New York.
- (10) Luna, H.P.L.; Zivianni, N. & Cabral, R.M.B. (1987). The Telephonic Switching Centre Network Problem: Formalization and Computational Experience. *Discrete Applied Mathematics*, **18**, 199-210.
- (11) Maculan, N. (1987). The Steiner Problem in Graphs. *Annals of Discrete Mathematics*, **31**, 185-212.
- (12) Magnanti, T.L.; Mireault, P. & Wong, R.T. (1986). Tailoring Benders Decomposition for Uncapacitated Network Design. *Mathematical Programming Study*, **26**, 112-154.
- (13) Mateus, G.R.; Cruz, F.R.B. & Luna, H.P.L. (1994). An Algorithm for Hierarchical Network Design. *Location Science*, **2**(3), 149-164.
- (14) Minitab Inc. (2000). *Minitab User's Guide 1 and 2*. url: <http://www.minitab.com>.
- (15) Nemhauser, G.L. & Wolsey, L.A. (1988). *Integer and Combinatorial Optimization*, John Wiley & Sons, New York.
- (16) Neter, J.; Wasserman, W. & Kutner, M.H. (1990). *Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Designs*. 3th ed., Homewood, Illinois.
- (17) Rardin, R.L. & Wolsey, L.A. (1993). Valid Inequalities and Projecting the Multicommodity Extended Formulation for Uncapacitated Fixed Charge Network Flow Problems. *European Journal of Operational Research*, **71**, 95-109.
- (18) Rothfarb, B.; Frank, H.; Rosembaun, D.M. & Steiglitz, K. (1970). Optimal Design of Offshore Natural-Gas Pipeline Systems. *Operations Research*, **18**, 992-1020.
- (19) Ziviani, N. (1993). *Projetos de Algoritmos – Com Implementações em Pascal e C*. Pioneira, São Paulo, Brasil.

Anexo A

A Figura 5 apresenta a implementação da macro **tfanova** escrita para o Minitab[®] versão 13 (Minitab Inc., 2000) e utilizada para análise dos dados. O arquivo com os dados experimentais da Tabela 1, necessários para a utilização de **tfanova**, está apresentado na Figura 6. A macro **tfanova** gera as saídas gráficas ilustradas na Figura 7, para uma análise exploratória dos dados e verificação da adequação do modelo. Gera também o resultado apresentado na Tabela 2.

```
# Macro:
#   tfanova.MAC
# Author:
#   Frederico R. B. Cruz
#   Departamento de Estatística - UFMG
#   E-mail: fcruz@est.ufmg.br - (c) 2001
# Function:
#   analysis of two-factor experiments with repeated measures on one factor
# Utilization:
#   MTB> %drive:\path\tfanova facA facRand facB resp
# Note:
#   This macro has been designed for Minitab for Windows, release 13 and above.
#   Do not attempt to run this macro in earlier versions of Minitab.
Macro
# template
tfanova facA facRand facB resp
#
# declarations
mcolumn facA facRand facB resp
mcolumn logresp resid
# body
Boxplot resp*facA resp*facB;
    Box;
    Symbol;
    Outlier;
    ScFrame;
    ScAnnotation.
let logresp=loge(resp)
ANOVA logresp = facA facRand(facA) facB facA*facB;
    Random facRand;
    Means facA facB;
    Restrict;
    Residuals resid;
    GHistogram.
%NormPlot resid.
# end
endmacro
```

Figura 5 – Arquivo **tfanova.mac**

algr	clss	inst	bb	cpu
1	1	1	3875	110.00
1	1	2	201	8.10
1	1	3	157	6.70
1	1	4	359	12.00
1	1	5	683	21.00
1	2	1	2533	71.00
1	2	2	183	7.00
1	2	3	97	4.40
.				
.				
.				

Figura 6 – Arquivo **tfnova.dat**

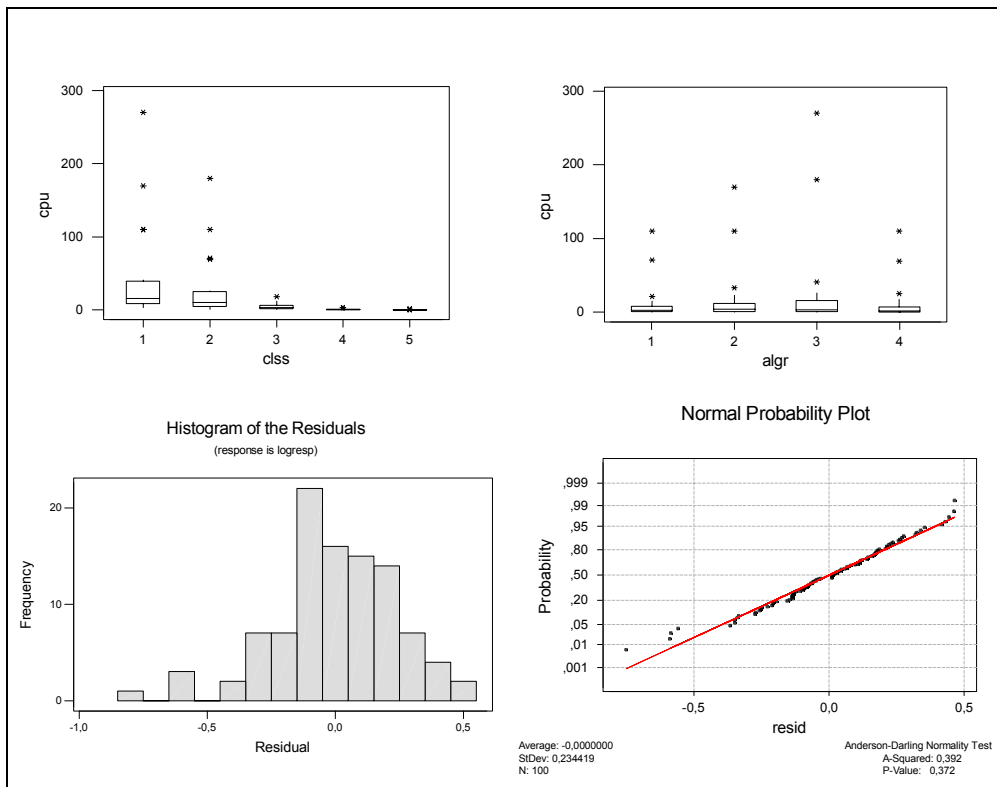


Figura 7 – Saída Gráfica da Macro **tfnova**