

# UMA ABORDAGEM ALTERNATIVA PARA RESOLVER O PROBLEMA DE ATRIBUIÇÃO DE PESOS AOS PROTOCOLOS DE MENOR CAMINHO EM REDES IP

**Vitor Guerra Rolla**

Instituto Militar de Engenharia  
Praça General Tibúrcio 80, Praia Vermelha  
Rio de Janeiro - RJ - CEP 22290-270  
vitor.rolla@ieee.org

**Ronaldo Moreira Salles**

Instituto Militar de Engenharia  
Praça General Tibúrcio 80, Praia Vermelha  
Rio de Janeiro - RJ - CEP 22290-270  
salles@ieee.org

## Resumo

Este trabalho tem por objetivo apresentar uma análise diferente do problema do roteamento ótimo de menor caminho e um novo método heurístico que foi desenvolvido para resolver o problema. Em termos práticos, a solução para o problema do roteamento ótimo de menor caminho determina a configuração de roteamento que otimiza uma rede de computadores que opera através das regras do protocolo OSPF. Baseamos nossa solução nas técnicas de engenharia de tráfego que respeitam o modelo de roteamento OSPF. Levamos em conta alguns dos objetivos principais da engenharia de tráfego: balanceamento de carga, melhoria no uso dos recursos disponíveis e capacidade de absorver demandas crescentes de tráfego.

**Palavras-Chaves:** Engenharia de tráfego; OSPF; Roteamento ótimo.

## Abstract

This work presents a different analysis of /the shortest path optimal routing problem/ and proposes a novel heuristic method to solve it. In practical terms, a solution for the shortest path optimal routing problem determines the routing setting that optimizes a computer network operating under the commonly used OSPF protocol. We based our solution on traffic engineering techniques that respect the OSPF routing model. We also considered some of the objectives of traffic engineering, such as: load balancing, better use of available resources and capacity to support growing traffic demands.

**Keywords:** Traffic Engineering; OSPF; Optimal Routing.

## 1. INTRODUÇÃO

O protocolo Open Shortest Path First (OSPF) é o protocolo de roteamento intradomínio mais utilizado na Internet. O protocolo se tornou tão utilizado por sua confiabilidade, escalabilidade e robustez. O OSPF fundamenta-se no estado do enlace e em algoritmos de menor caminho, tais como Dijkstra e Bellman-Ford. Os nós que fazem parte de um domínio OSPF trocam informações que proporcionam a cada nó a implementação de um banco de dados que represente a topologia da rede. O algoritmo de menor caminho é utilizado para construir uma árvore de menor caminho para todos os destinos, levando em conta o estado do enlace e o peso atribuído a cada enlace. Os pesos de cada enlace são configurados pelos administradores da rede. Os fabricantes de roteadores sugerem que o peso de cada enlace deva ser  $1/C$ , onde  $C$  é a capacidade de comunicação do enlace. Tal assertiva pode ser

observada no artigo [10] de Thomas T. M. publicado pela Cisco Press em 1998.

A maioria das dificuldades do OSPF deve-se a utilização do paradigma: encaminhamento baseado no destino. A origem do tráfego não tem controle e nem pode influenciar sobre o procedimento de seleção do caminho, a decisão de roteamento é efetuada em cada roteador. O algoritmo baseado no menor custo seleciona os caminhos mais curtos para preencher a tabela de roteamento. Caso os caminhos mais curtos estejam sobrecarregados, outros caminhos de maior comprimento não serão utilizados mesmo se estiverem subutilizados. No entanto, é importante ressaltar que as mesmas características que tornam o OSPF inflexível e com baixa capacidade de prover engenharia de tráfego o tornam altamente confiável, escalável e robusto.

O balanceamento de carga, que é um dos objetivos fundamentais da engenharia de tráfego, não é totalmente satisfeito no modelo OSPF. O protocolo promove a utilização da técnica Equal-Cost Multipath (ECMP). Essa técnica permite que um mesmo fluxo de dados de uma demanda origem-destino (OD) seja carregado ao longo de múltiplos caminhos. É importante lembrar que a técnica ECMP continua não levando em conta a quantidade de carga a qual cada enlace está submetido. A técnica ECMP entra em ação nos nós que possuem mais de um caminho mais curto até um determinado destino, dividindo a demanda do fluxo de dados entre esses caminhos mais curtos.

O crescimento das demandas de velocidade e confiabilidade na rede mundial levou ao surgimento de novas tecnologias de engenharia de tráfego e roteamento. Muitas delas com o objetivo de endereçar os problemas do próprio protocolo OSPF. Um aspecto que motivou nosso estudo é a comparação do desempenho dessas novas tecnologias com o modelo OSPF. Principalmente as redes de comunicação MPLS. Essa motivação foi introduzida pelos autores Bernard Fortz e Mikkel Thorup no artigo [1]. O MPLS é uma tecnologia que permite decidir o caminho para cada pacote de dados individualmente. Por esse motivo essa tecnologia permite uma implantação de roteamento semelhante ao da solução do problema geral do roteamento ótimo.

Na seção II deste documento explicaremos o problema geral do roteamento ótimo e o problema do roteamento ótimo de caminho único, problemas estes que facilitarão a discussão e o entendimento das restrições impostas pelo problema do roteamento ótimo de menor caminho. Na seção III vamos caracterizar o problema do roteamento ótimo de menor caminho (roteamento OSPF). Em sequência, na seção IV, descreveremos nossa proposta heurística de solução para o problema do roteamento ótimo de menor caminho. Apresentaremos como foram efetuados nossos experimentos e discutiremos nossos resultados nas seções V e VI, respectivamente. Na seção VII apresentaremos um sumário do trabalho efetuado até o momento e descreveremos trabalhos futuros.

## 2. PROBLEMA GERAL DO ROTEAMENTO ÓTIMO / PROBLEMA DO ROTEAMENTO ÓTIMO DE CAMINHO ÚNICO

### 2.1. PROBLEMA GERAL DO ROTEAMENTO ÓTIMO

O problema geral do roteamento ótimo consiste em distribuir o tráfego de cada demanda OD entre os vários caminhos disponíveis na rede, de forma a minimizar uma certa função de custo global. Esse problema também é conhecido na literatura como problema do roteamento ótimo com bifurcação. É importante lembrar que estamos considerando que a matriz de tráfego é estática, isto é, as demandas de cada par OD (origem-destino) são fixas. A formulação matemática desse problema pode ser encontrada em várias bibliografias. No entanto, utilizaremos a formulação encontrada em [9], por ser simplificada:

$$\begin{aligned} \text{Minimizar: } & \sum_{(i,j)} D_{ij}(F_{ij}) \\ \text{Sujeito a: } & \sum_{p \in P_w} x_p = r_w, \text{ para todo } w \in W \end{aligned}$$

$$x_p > 0, \text{ para todo } p \in P_w, w \in W$$

Onde:

- $W$  é o conjunto de todos os pares  $w$  (nó origem – nó destino);
- $P_w$  é o conjunto de todos os caminhos que conectam o nó origem ao nó destino do par  $w \in W$ ;
- $x_p$  fluxo carregado pelo caminho  $p$ ;
- $F_{ij}$  é o somatório dos fluxos  $x_p$  que passam pelo enlace  $(i, j)$ , isto é, o fluxo total (dados/seg) carregado pela aresta  $(i, j)$ :  $F_{ij} = \sum_{\text{caminhos que passam por } (i, j)} x_p$  ;
- $r_w$  é o fluxo de demanda do nó origem ao nó destino do par  $w$ ;
- $D_{ij}(F_{ij})$  é a função custo que deve ser otimizada  $D_{ij}(F_{ij}) = \frac{F_{ij}}{C_{ij} - F_{ij}}$  ;
- $C_{ij}$  é a capacidade de transmissão do enlace  $(i, j)$ .

A idéia básica do problema geral do roteamento ótimo é equilibrar as cargas das demandas. Um fluxo de um par OD pode ser distribuído por um, vários ou todos os caminhos existentes entre o par. Os fluxos nos enlaces são variáveis contínuas. Essa característica permite que o problema seja tratado matematicamente [5].

A função que apresentamos na formulação do problema é não-linear e convexa. Ela é a relação da carga submetida ao enlace ( $F_{ij}$ ) sobre a capacidade ainda disponível no enlace ( $C_{ij} - F_{ij}$ ). Essa função representa o atraso médio sofrido pelos pacotes na fila do roteador. Chamaremos o problema geral do roteamento ótimo utilizando a função de custo  $\min \sum D_{ij}(F_{ij})$  de OPTRouting-Delay.

No nosso experimento também computamos o problema utilizando uma segunda função objetivo. Chamaremos o problema geral do roteamento ótimo utilizando a função de custo  $\min \max(\alpha)$  de OPTRouting-LinkUtilization. A função objetivo  $\max(\alpha)$  a ser minimizada representa o enlace com máxima utilização entre todos os enlaces da rede de comunicação. Essa função caracteriza a otimização do balanceamento de carga em uma rede de computadores.

O problema geral do roteamento ótimo pode ser resolvido em tempo polinomial. As técnicas mais conhecidas para resolução do problema baseiam-se em métodos de projeção do gradiente da função.

A solução do problema geral do roteamento ótimo é extremamente importante para o nosso estudo por representar um limite aos demais problemas de roteamento. A solução do problema também representa a melhor forma de rotear demandas em uma rede de comunicação. Portanto, será utilizada como parâmetro na avaliação do desempenho da solução proposta para o problema do roteamento ótimo de menor caminho.

## 2.2. PROBLEMA DO ROTEMANTO ÓTIMO DE CAMINHO ÚNICO

O problema do roteamento ótimo de caminho único consiste em determinar, para cada demanda OD, um único caminho pelo qual a demanda passará integralmente, tal que a combinação de todos os caminhos selecionados (um para cada demanda) minimize uma certa função de custo global. Esse problema também é conhecido na literatura como problema do roteamento ótimo sem bifurcação. Ele está estritamente relacionado com o problema geral do roteamento ótimo e com o problema do roteamento ótimo OSPF, representando um meio termo entre esse dois últimos. Segue abaixo a formulação matemática do problema do roteamento ótimo de caminho único:

$$\text{Minimizar: } \sum_{(i,j)} D_{ij}(F_{ij})$$

$$\text{Sujeito a: } \sum_{p \in P_w} x_p \cdot \delta_p = r_w, \text{ para todo } w \in W$$

$$\sum \delta_p = 1, \delta_p = \{0,1\}, \text{ para todo } p \in P_w, w \in W$$

$$x_p > 0, \text{ para todo } p \in P_w, w \in W$$

Onde:

- $W$  é o conjunto de todos os pares  $w$  (nó origem – nó destino);
- $P_w$  é o conjunto de todos os caminhos que conectam o nó origem ao nó destino do par  $w \in W$ ;
- $x_p$  fluxo carregado pelo caminho  $p$ ;
- $\delta_p$  é o parâmetro de seleção do caminho entre cada par  $w$ ;
- $F_{ij}$  é o somatório dos fluxos  $x_p$  que passam pelo enlace  $(i, j)$ , isto é, o fluxo total (dados/seg) carregado pela aresta  $(i, j)$ :  $F_{ij} = \sum_{\text{caminhos que passam por } (i, j)} x_p$  ;

- $r_w$  é o fluxo de demanda do nó origem ao nó destino do par  $w$ ;
- $D_{ij}(F_{ij})$  é a função custo que deve ser otimizada  $D_{ij}(F_{ij}) = \frac{F_{ij}}{C_{ij} - F_{ij}}$  ;
- $C_{ij}$  é a capacidade de transmissão do enlace  $(i, j)$ .

A característica que difere o problema geral do roteamento ótimo e o problema do roteamento ótimo de caminho único é que no último existe a obrigatoriedade do tráfego de cada demanda passar integralmente por um único caminho entre o nó origem e o nó destino. Com essa característica o problema passa a ser combinatório e NP-difícil. No artigo [12] os autores provam a complexidade NP-difícil do problema do roteamento ótimo sem bifurcação. No mesmo artigo eles propoem uma heurística para resolver o problema. Essa heurística basicamente executa a solução do OPTRouting-LinkUtilization e re-roteia as demandas que foram segmentadas (sofreram bifurcação) não permitindo a bifurcação.

O problema do roteamento ótimo de caminho único não constará em nossos experimentos. As soluções conhecidas para esse problema são soluções heurísticas pelo fato da alta complexidade que o problema apresenta.

### 3. PROBLEMA DO ROTEAMENTO ÓTIMO DE MENOR CAMINHO (ROTEAMENTO OSPF)

O problema do roteamento ótimo de menor caminho consiste em encontrar os pesos ótimos para os enlaces de uma rede de comunicação que possibilite ao roteamento OSPF minimizar uma certa função de custo global.

A principal característica do problema do roteamento ótimo de menor caminho (roteamento OSPF) é que as variáveis desse problema são os pesos que devem ser configurados em cada enlace de comunicação.

É interessante destacar que o objetivo do problema geral do roteamento ótimo é encontrar a parcela de tráfego que percorrerá cada caminho do grafo (rede de comunicação). Portanto, de acordo com a formulação matemática apresentada em 2.1, precisamos determinar as variáveis  $x_p$  das demandas de fluxo  $r_w$ . Reparando agora o problema do roteamento ótimo de caminho único, observamos que seu objetivo é determinar qual dos caminhos, entre todos os caminhos existentes entre uma demanda OD, deve ser selecionado para minimizar a função objetivo. Portanto, de acordo com a formulação apresentada em 2.2, precisamos determinar os parâmetros  $\delta_p$  que representam os caminhos selecionados.

A característica das variáveis serem os pesos nos enlaces torna a formulação matemática do problema do roteamento ótimo de menor caminho muito difícil. Essa característica introduz o conceito de acoplamento de caminhos. O conceito de acoplamento pode ser definido da seguinte forma: se dois caminhos têm interseção em dois nós, o subcaminho entre os dois nós deve ser idêntico [5]. Por exemplo, se no grafo (rede de comunicação) apresentado figura 1 o menor caminho entre os nós  $A$  e  $D$  for  $ABCD$ ,

obrigatoriamente o menor caminho entre os nós  $X$  e  $Z$  deve ser  $XBCDZ$ . Observe que entre os nós  $B$  e  $D$  o subcaminho é o mesmo nos dois caminhos.

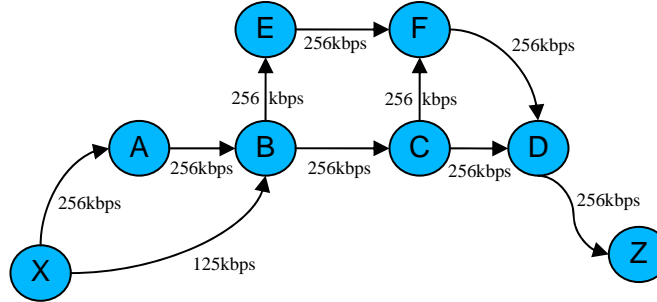


Figura 1: rede de 8 nós.

Em [1] os autores provam que o problema do roteamento ótimo de menor caminho é NP-completo. Caso se leve em conta o conceito ECMP, o problema passa a ser NP-difícil. Por esses motivos os estudos que procuram soluções para o problema normalmente propõem soluções heurísticas.

A heurística proposta em [1] baseia-se na técnica de local search. Os resultados apresentados no artigo demonstram que somente em alguns casos específicos a solução proposta pelos autores chega próximo da solução do problema geral do roteamento ótimo. No mesmo artigo os autores também propõem algoritmos eficientes para atualizar o roteamento e recomputar o custo da solução quando poucos pesos são alterados. O artigo [1] é um texto chave para o estudo do problema do roteamento ótimo de menor caminho. Esse artigo é referenciado com muita frequência nas demais propostas de solução do problema. Outros artigos desses autores também podem ser citados, pois contribuem para o entendimento da evolução do problema. São eles [2], [3] e [4].

No artigo [5] os autores propõem um algoritmo combinatório para resolver o problema do roteamento ótimo de menor caminho considerando o atraso médio dos pacotes como parâmetro a ser otimizado. O problema é caracterizado pelos autores como um caso especial do problema clássico de menor caminho inverso [6]. A solução destes pesquisadores baseia-se na análise da característica de acoplamento que o protocolo OSPF apresenta.

Várias outras propostas abordam o problema. No artigo [7] verificamos uma proposta de algoritmo genético para ajustar o problema demonstrado. Algumas dessas propostas procuram abordar o problema de uma forma diferente. No artigo [8] podemos observar a proposta de um algoritmo genético para desenhar uma rede de comunicação com uma capacidade de enlace total mínima necessária para rotear uma determinada demanda sem sobrecarregar a rede em caso de falha em uma das arestas ou em um dos nós.

#### 4. PROPOSTA DE SOLUÇÃO DO ROTEAMENTO OSPF (HEURCOPLING)

A solução heurística proposta nesse documento difere das demais soluções estudadas pelo fato de dividir o problema do roteamento ótimo de menor caminho em duas etapas distintas. A primeira etapa consiste em estabelecer os caminhos já acoplados, de forma a minimizar o custo de roteamento pelos quais as demandas devem ser roteadas. A segunda etapa consiste em determinar os pesos aos enlaces da rede que reflitam os caminhos estabelecidos na primeira etapa. HeurCopliling é o nome dado ao algoritmo proposto para resolver o problema do roteamento OSPF.

##### 4.1. PRIMEIRA ETAPA: ESTABELECEER CAMINHOS

A idéia básica da primeira etapa do HeurCopliling fundamenta-se em respeitar a quantidade de tráfego a qual cada enlace está submetido durante a confecção dos caminhos acoplados. Com isso introduzimos a solução maior flexibilidade para atingir os objetivos de engenharia de tráfego.

Utilizamos o conceito de quantidade de banda disponível (Available Bandwidth –  $AVBL.BW$ ) para efetuar o roteamento de cada demanda por vez. As demandas são roteadas de

forma decrescente. Esses princípios têm como objetivo forçar que as maiores demandas trafeguem pelos enlaces com maior largura de banda disponível, balanceando a carga entre os recursos disponíveis. A cada interação do nosso algoritmo são utilizadas diferentes configurações de pesos. Os pesos são atualizados de acordo com  $\frac{1}{AVBL.BW}$ .

Os pesos que estamos nos referindo nessa etapa não são os pesos finais para os enlaces da rede, pois a cada demanda roteada eles são alterados pelo algoritmo. Determinaremos os pesos que resultarão em um roteamento OSPF na segunda etapa.

Na recomendação padrão  $1/c$  os pesos são estáticos. Não se leva em consideração a carga submetida a cada enlace para o roteamento das demandas.

Segue abaixo a primeira etapa do algoritmo HeurCoplmg:

1. Os pesos dos enlaces são configurados  $1/c$ .
2. Procurar a maior demanda ainda não roteada.
3. Rotear a maior demanda utilizando um algoritmo de menor caminho (algoritmo de Dijkstra).
4. Verificar se o caminho retornado pelo passo 3 está acoplado. Caso não esteja acoplado, executar função de acoplamento.
5. Atualizar os pesos dos enlaces utilizados pelo caminho já acoplado com a heurística  $\frac{1}{AVBL.BW}$ . Retornar para o passo 2. Parar caso todas as demandas já tenham caminhos definidos.

Vamos analisar cada passo do algoritmo utilizando o exemplo da tabela 1 juntamente com a figura 1:

Origem/ Destino	A	B	C	D	E	F	X	Z
A	0	0	0	100	0	0	0	0
B	0	0	60	0	0	0	0	0
C	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0
X	0	0	0	0	0	0	0	50
Z	0	0	0	0	0	0	0	0

Tabela 1: matriz de tráfego para a rede de 8 nós (Kbps).

No passo 1 os pesos dos enlaces são configurados como  $1/c$ , pois nenhuma demanda ainda foi roteada, isto é, os enlaces não estão submetidos a nenhuma carga. Considerando que o enlace  $XB$  tem capacidade 125Kbps e todos os outros enlaces da rede têm capacidade 256Kbps. Se normalizarmos a heurística recomendada pelos fabricantes para  $1000/c$ , o peso do enlace  $XB$  será 8.0 e dos outros enlaces será 3.9.

Nos passos 2 e 3 devemos observar que a cada demanda roteada, mais de um menor caminho pode ser determinado. Por exemplo, observe a tabela 1. A maior demanda no nosso exemplo é entre os pares  $A$  e  $D$ . O algoritmo verifica se já existe um menor caminho determinado para esse par  $OD$ . Como essa é a primeira demanda que está sendo roteada, nenhum caminho ainda foi definido. Utilizando o algoritmo de Dijkstra é fácil verificar que o caminho retornado é  $ABCD$ . Com isso, automaticamente derivamos os menores caminhos entre os pares  $A$  e  $C$ ,  $A$  e  $B$ ,  $B$  e  $D$ ,  $B$  e  $C$  e  $C$  e  $D$  como sendo  $ABC$ ,  $AB$ ,  $BCD$ ,  $BC$  e  $CD$  respectivamente.

O passo 4 serve para verificar se o caminho retornado pelo algoritmo de Dijkstra respeita a regra de acoplamento em relação aos caminhos já definidos anteriormente (por demandas anteriores). Como essa é a primeira demanda roteada, não existem caminhos já definidos. Voltaremos a explicar esse passo com mais detalhes no roteamento da segunda maior demanda.

No passo 5 atualizamos os pesos para o roteamento da próxima demanda. Portanto, os enlaces  $AB$ ,  $BC$  e  $CD$ , que são os enlaces utilizados pelo caminho definido  $ABCD$ , recebem os pesos 6.4, 10.4 e 6.4 respectivamente. Repare que a demanda do nó  $B$  para o nó  $C$  já foi considerada para atualização do peso do enlace  $BC$ , pois o menor caminho para essa demanda também já está definido como  $BC$ . Como ainda existem mais demandas a serem roteadas, retornamos ao passo 2.

Ao retornar para o passo 2, procuramos de novo a maior demanda ainda não roteada, no nosso exemplo é a demanda entre os pares  $X$  e  $Z$ . Como já mencionamos, o menor caminho entre os pares  $B$  e  $C$  já foi determinado na primeira interação do algoritmo.

No passo 3 roteamos a demanda entre  $X$  e  $Z$  com os pesos já atualizados pela interação anterior. É fácil perceber que o caminho retornado pelo algoritmo de Dijkstra é  $XBEFDZ$ .

No passo 4 verificamos a condição de acoplamento. Nesse caso o caminho não está acoplado, pois já existe um caminho definido entre os nós  $B$  e  $D$  e esse caminho não é  $BEFD$ . O algoritmo executa uma função de acoplamento para corrigir o caminho para  $XBCDZ$ .

No passo 5 os pesos são atualizados. Como não existem mais demandas a serem roteadas o algoritmo para.

O exemplo que utilizamos simplesmente ilustra o funcionamento do algoritmo. No entanto, podemos verificar uma pequena diferença no desempenho do algoritmo HeurCoplmg em relação à recomendação  $1/c$ . Se calcularmos o custo  $\sum D_{ij}(F_{ij})$  dos caminhos computados pelo algoritmo HeurCoplmg e dos caminhos computados pela recomendação  $1/c$  teremos 8.69 e 9.03 respectivamente.

Como já mencionamos, os pesos que estamos nos referindo nessa etapa não são os pesos finais para os enlaces da rede, pois a cada demanda roteada eles são alterados pelo algoritmo. Repare que os pesos utilizados nessa etapa frequentemente geram caminhos desacoplados, isto é, que não respeitam as regras do roteamento OSPF. Determinaremos os pesos que resultarão em um roteamento OSPF na segunda etapa.

**4.2. SEGUNDA ETAPA: PROGRAMAÇÃO LINEAR**

A segunda etapa consiste em determinar os pesos aos enlaces da rede que reflitam os caminhos estabelecidos na primeira etapa. Isto é, dado o conjunto de caminhos, conjunto este que respeite a regra do acoplamento, nosso problema consiste em atribuir pesos às arestas do grafo (enlaces da rede), tal que o roteamento OSPF reflita o conjunto de caminhos estabelecidos na primeira etapa.

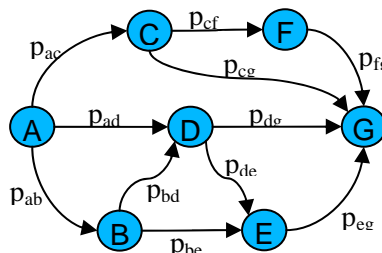


Figura 2: rede de 7 nós.

A solução clássica para o problema é modelar um sistema de restrições linear que garanta que o peso total do caminho selecionado deva ser menor que os pesos dos outros caminhos (não selecionados) entre o par OD. Vamos considerar o mesmo exemplo proposto no artigo [13] onde  $p_{od}$  representa o peso do enlace com origem em O e destino em D. Na figura 2, se assumirmos que o menor caminho para o par com origem em A e destino em G seja  $ADG$ . Obrigatoriamente teremos o conjunto de restrições abaixo para garantir que  $ADG$  seja o menor caminho:

- 1)  $p_{ad} + p_{dg} < p_{ab} + p_{be} + p_{eg}$
- 2)  $p_{ad} + p_{dg} < p_{ac} + p_{cg}$
- 3)  $p_{ad} + p_{dg} < p_{ab} + p_{bd} + p_{dg}$

$$4) p_{ad} + p_{dg} < p_{ab} + p_{bd} + p_{de} + p_{eg}$$

$$5) p_{ad} + p_{dg} < p_{ac} + p_{cf} + p_{fg}$$

$$6) p_{ad} + p_{dg} < p_{ad} + p_{de} + p_{eg}$$

A solução clássica produz um número exponencial de restrições, muitas delas redundantes. Para cada caminho não selecionado teremos uma restrição. Porém, no artigo [13], os autores propõem um modelo simples e eficaz para reduzir o número de restrições, que vem a ser o seguinte:

Assumindo-se  $S_{od}$  como o caminho selecionado entre os vértices O e D (caminho que pertence ao conjunto dos caminhos já acoplados gerados na primeira etapa) e como  $P(S_{od})$  o peso para esse caminho. Sendo  $A_{oi}:A_{id}$  um caminho alternativo que passa no vértice intermediário I, logo  $P(A_{oi}:A_{id})$  é o peso desse caminho alternativo. O critério criado pelos autores do artigo [13] afirma que a restrição  $P(S_{od}) < P(A_{oi}:A_{id})$  é redundante se pelo menos uma das condições abaixo, para qualquer vértice intermediário, for verdadeira:

i)  $A_{oi}$  não é o caminho selecionado entre os vértices O e I;

ii)  $A_{id}$  não é o caminho selecionado entre os vértices I e D.

Adotando esse simples critério o sistema de restrições lineares passa a ter um número polinomial de restrições no número de vértices. Para o exemplo da figura 2, é fácil de observar que as restrições 3, 4, 5 e 6 passam a ser redundante se aplicarmos o critério e assumirmos que o caminho selecionado entre B e G é BEG.

## 5. EXPERIMENTOS

Nos experimentos executados comparamos os resultados de atraso médio e enlace com máxima utilização entre o problema geral do roteamento ótimo, a recomendação padrão feita pelos fabricantes  $1/c$  e a HeurCoplign.

Utilizamos seis diferentes topologias de rede e matrizes de tráfego crescentes onde todos os nós produziam tráfego para os demais nós. As topologias variaram entre 5 e 47 nós.

Para modelar o problema geral do roteamento ótimo utilizamos a linguagem de modelagem para programação matemática (AMPL), tanto para a formulação com a função objetivo  $\min \sum D_{ij}(F_{ij})$  como para a função objetivo  $\min \max(\alpha)$ . Para resolver os modelos utilizamos o software de otimização KNITRO no NEOS Server [14]. O KNITRO é uma ferramenta de otimização desenvolvida para resolver problemas de grande escala. Essa ferramenta utiliza informações da segunda derivada da função objetivo.

Desenvolvemos em linguagem C um programa para resolver o problema de acordo com a recomendação dos fabricantes  $1/c$ .

A HeurCoplign foi desenvolvida em linguagem C + LP\_Solve [15].

## 6. RESULTADOS

Em todas as topologias testadas os resultados da HeurComplign foram bem superiores aos resultados da solução recomendada pelos fabricantes.

Os resultados da HeurCoplign acompanharam de perto os resultados obtidos para os problemas OPTRountig-Delay e OPTRouting-LinkUtilization em todas as redes testadas. A figura 3 demonstra os dados coletados em função do atraso médio (custo). A figura 4 demonstra os resultados em função do enlace com maior utilização.

Como mencionado em [1], a complexidade de se afirmar que existe uma configuração de pesos que tenha um desempenho ainda mais perto do OPTRouting é NP-difícil. Portanto, não podemos afirmar se os resultados apresentados pela HeurCoplign estão próximos do ótimo para o problema do roteamento ótimo de menor caminho ou a diferença dos resultados da HeurCoplign em relação ao OPTRouting é devido à heurística proposta ser não-polinomial.

É importante observar que nos experimentos executados o tempo completo de execução do nosso algoritmo para as redes de 20, 26 e 47 nós não passou de 3 segundos. Esse



é um ponto importante a ser analisado. Tal resultado implica na possibilidade de utilizar o algoritmo de forma on-line, isto é, com matrizes de tráfego dinâmicas. Segue abaixo os resultados obtidos para a rede de 26 nós:

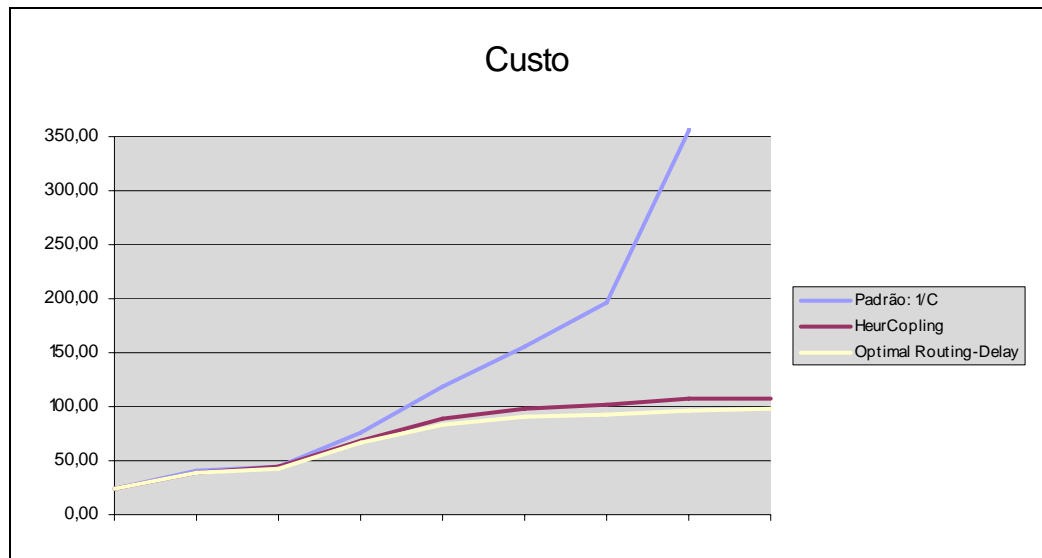


Figura 3: Atraso médio -  $\min \sum D_{ij}(F_{ij})$ .

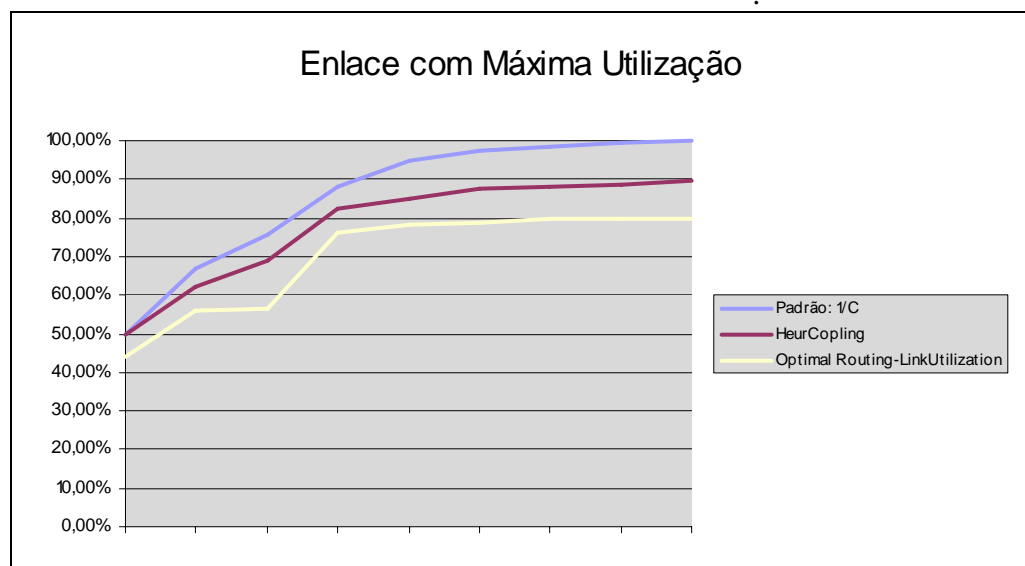


Figura 4: Utilização máxima - minmax (a).

## 7. SUMÁRIO

A maior contribuição de nosso trabalho é a proposta de um algoritmo para resolver o problema do roteamento ótimo de menor caminho. Essa proposta procura utilizar técnicas de engenharia de tráfego para melhorar o desempenho das redes IPs.

Nós modelamos matematicamente o problema geral do roteamento ótimo e o problema do roteamento ótimo de caminho único com o objetivo de introduzir o problema do roteamento ótimo de menor caminho e as técnicas de engenharia de tráfego que possibilitaram o desenvolvimento de nossa solução. Como mencionamos anteriormente, temos a intenção de pesquisar a possibilidade de se implementar o algoritmo de forma on-line, isto é, utilizando matrizes de tráfego dinâmicas.

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Fortz, B.; Thorup M. **Internet Traffic Engineering by Optimizing OSPF weights**. Proc. 19th IEEE Conf. on Computer Communications INFOCOM, 2000.
- [2] Fortz, B.; Thorup M. **Optimizing OSPF/IS-IS weights in a changing world**. IEEE Journal on Selected Areas in Communications, v.20, n.4, p. 756–767, 2002.
- [3] Fortz, B.; Thorup M.; Rexford J. **Traffic engineering with traditional IP routing protocols**. IEEE Communications Magazine, v.40, n.10, p.118–124, 2002.
- [4] Fortz, B.; Thorup M. **Robust optimization of OSPF/IS-IS weights**. W. Ben-Ameur and A. Petrowski, editors, Proc. INOC 2003, p. 225-230, October 2003.
- [5] Ramakrishnan K. G.; Rodrigues M. A. **Optimal routing in shortest-path data networks**. Lucent Bell Labs Technical Journal, v.6, n. 1, 2001.
- [6] Burton D.; Toint P. L. **On an Instance of the Inverse Shortest Paths Problem**. Math. Programming, Series A, v.53, n. 1–6, p. 45–61, Jan. 1992.
- [7] Ericsson M.; Resende M. G. C; Pardalos P. M. **A genetic algorithm for the weight setting problem in OSPF routing**. Journal of Combinatorial Optimization, 2002.
- [8] Buriol L.S.; França P.M.; Resende M.G.C.; Thorup M. **Network design for OSPF routing**. Proceedings of Mathematical Programming in Rio, Rio de Janeiro, pp. 40-44, 2003.
- [9] Bertsekas D.; Gallager R. **Data Networks**. 2<sup>nd</sup> ed. Prentice-Hall, 1992.
- [10] Thomas II T.M. **OSPF Network Design Solutions**. Cisco Press, 1998.
- [11] Buriol L.S.; Resende M.G.C.; Thorup M. **Speeding up dynamic shortest path algorithms**. Technical Report AT&T Labs Research, 2003.
- [12] Wang Y.; Wang Z. **Explicit routing algorithms for internet traffic engineering**. Proceedings of the 8th Computer Communications and Networks, Boston, 1999.
- [13] Mahajan R.; Spring N.; Wetherall D.; Anderson T. **Inferring Link Weights using End-to-End Measurements**. Proceedings of ACM SIGCOMM Internet Measurement Workshop, Nov. 2002.
- [14] Czyzyk J.; Mesnier M. P.; Moré J. J. **The NEOS Server**. IEEE Computational Science & engineering, v.5, n.3, pp. 68-75, Jul/Set. 1998.
- [15] Murty K.G. **Linear Programming**. John Wiley & Sons, 1983.