

Otimizando o Roteamento do Tráfego na Internet

Luciana Saete Buriol

Universidade Estadual de Campinas – UNICAMP
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Sistemas
Av. Albert Einstein, 400 - C.P. 6101-13083-970 Campinas - SP
buriol@densis.fee.unicamp.br

Paulo Morelato França

Universidade Estadual de Campinas – UNICAMP
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Sistemas
Av. Albert Einstein, 400 - C.P. 6101-13083-970 Campinas - SP
franca@densis.fee.unicamp.br

Mauricio G. C. Resende

AT&T Labs Research
Internet and Network Systems Research Center
Algorithms and Optimization Research Department
Room C241- 180 Park Avenue - P.O. Box 971
Florham Park, NJ 07932-0971 USA
mgcr@research.att.com

Celso Carneiro Ribeiro

Universidade Federal Fluminense
Departamento de Ciência da Computação
Rua Passo da Pátria 156, Niterói, Rio de Janeiro 24210-240
celso@ic.uff.br

Resumo: Devido ao acelerado crescimento do tráfego na rede Internet observado nos últimos anos, o seu gerenciamento ganhou importância fundamental. Se por um lado os investimentos em infraestrutura são imperiosos, por outro busca-se aproveitar ao máximo os recursos existentes, visando uma adequada qualidade de serviço ao mesmo tempo em que se tenta adiar novos investimentos. Grandes provedores de serviços de Internet são responsáveis por rotear o tráfego, tarefa realizada por meio de protocolos que encaminham pacotes de dados desde a sua origem até o seu destino. Um dos protocolos mais utilizados mundialmente é o *Open Shortest Path First* (OSPF) que roteia o tráfego usando trajetórias de pesos mínimos calculadas sobre a rede provedora. A escolha desses caminhos é uma decisão que deve ser tomada pelos provedores que, por sua vez, buscam satisfazer critérios como minimizar a possibilidade de congestão ou a ociosidade da rede. O protocolo OSPF trata esta questão por meio de uma adequada designação de pesos aos arcos da rede, o que influencia diretamente na determinação dos caminhos mínimos a serem usados. Neste artigo aborda-se o problema de rotear a demanda de tráfego na rede com o objetivo de minimizar a congestão da rede usando uma adequada atribuição de pesos aos arcos. Além de se apresentar um modelo não-linear de otimização combinatória para o problema, descrevem-se e comparam-se alguns enfoques que empregam técnicas heurísticas para a sua solução.

Palavras-chave: tráfego de Internet, engenharia de tráfego, protocolo OSPF, otimização combinatória, heurísticas, metaheurísticas.

Abstract: The accelerated growth of Internet traffic in recent years has motivated large investments in infrastructure as well as the use of modern methods of traffic management aiming at a maximum utilization of existing resources. Internet Service Providers are responsible for routing the data packets that are sent along network links from a source to a destination host. A routing protocol is used to perform this traffic engineering task. The most commonly used protocol is OSPF (*Open Shortest Path First*) which flows data packets to the destination IP address along shortest paths connecting Internet routers. OSPF determines those shortest paths by assigning dimensionless weights to each link and calculating the shortest paths connecting a router (taken as the root node) to all other destinations. Providers can select suitable weight settings seeking some technical or economic criterion such as the minimization of traffic congestion. The lower the weight, the greater the possibility that traffic will get routed on that link. This article deals with the OSPF weight setting problem. First, we present a nonlinear combinatorial optimization model that minimizes the traffic congestion. Then, we describe and compare some heuristic approaches used to solve it.

Keywords: Internet traffic, traffic engineering, OSPF protocol, combinatorial optimization, heuristics, metaheuristics.

1. Introdução

O roteamento do tráfego na Internet é um problema da área de engenharia de tráfego que recentemente vem atraindo a atenção da comunidade acadêmica. De forma geral, o roteamento de pacotes de dados na Internet passa por dois problemas: (i) alocação dos recursos necessários para tratar a demanda de dados entre roteadores e (ii) gerenciamento dos recursos disponíveis de forma a homogeneizar o tráfego na rede, evitando assim a sobrecarga ou a ociosidade de parte dela. Este trabalho tem por objetivo conceituar e descrever as abordagens de solução para o segundo problema, na literatura denominado problema de designação de pesos (*weight setting problem*).

A otimização do uso dos recursos físicos disponíveis numa rede Internet tem ganhado importância maior nos últimos anos. Essa preocupação tem um motivo plausível: o tráfego na Internet tem aumentado consideravelmente ano após ano. Vários são os fatores que têm contribuído para esse aumento. Dentre eles, destacam-se o crescente aumento do número de usuários e o uso da Internet para o envio de quantidades de dados cada vez maiores. Até duas décadas atrás a Internet era usada principalmente por pesquisadores e militares. Hoje, ela tornou-se indispensável para uma parcela considerável da população, assim como para a maioria das empresas. A Tabela 1 mostra o crescimento do número de computadores conectados à Internet nas últimas duas décadas. Além disso, a quantidade e o tamanho dos dados que os usuários trocam tem aumentado consideravelmente. O emprego de recursos multimídia, como som, áudio e vídeo, assim como figuras e documentos, têm sido prática comum entre os usuários da Web. O instituto de pesquisas *International Data Corporation*¹ (IDC) prevê que o volume de tráfego gerado pelos usuários domésticos de todo o mundo irá dobrar anualmente pelos próximos cinco anos, partindo de 180 petabits por dia em 2002, para 5175 petabits por dia até o fim de 2007.

Tabela 1 – Crescimento do número de computadores conectados à Internet²

1981	1985	1990	1995	2000	2001	2002
213	1961	376.000	4.852.000	72.398.092	109.574.429	147.344.723

¹ www.idcresearch.com

² Atualmente os EUA possuem 70% dos hosts no mundo, enquanto o Brasil está em 9º lugar com 1,3% dos hosts. Dados divulgados pelo Comitê Gestor da Internet no Brasil (<http://www.cg.org.br/indicadores/>).

Quando se envia ou se recebe algo pela Internet, a informação é acondicionada em pequenos pacotes de dados que contêm um cabeçalho com um conjunto de informações necessárias para a sua correta transmissão ao longo da rede. Uma dessas informações é o endereço IP (*Internet Protocol*) a que se destina. Assim que um pacote chega a um roteador da rede, ele consulta uma tabela de roteamento que contêm o endereço IP do próximo roteador. Como cada pacote é tratado de forma individual, a ordem em que os pacotes chegam pode não ser a mesma em que são enviados. O protocolo IP apenas envia os pacotes, ficando a tarefa de sua reordenação a cargo de outro protocolo chamado TCP (*Transmission Control Protocol*). Além disso, como o roteamento é uma tarefa muito complexa, a rede mundial é dividida em domínios com regras próprias para rotear tráfego em seu interior e outras regras para rotear tráfego entre domínios. Um domínio é chamado de Sistema Autônomo (*Autonomous System – AS*). Há protocolos válidos para atuar dentro de um AS (*Interior Gateway Protocols – IGP*) ou entre unidades de AS (*Exterior Gateway Protocols – EGP*).

Open Shortest Path First (OSPF) é o protocolo IGP mais usado. O conceito básico deste protocolo é realizar o roteamento da carga por meio do caminho de peso mínimo calculado sobre a rede e, no caso de múltiplos caminhos mínimos, a carga é dividida igualmente entre todos os arcos adjacentes a um roteador no grafo de caminho mínimo. Ainda, este protocolo estabelece que roteadores troquem informação de roteamento com todos os outros roteadores dentro do mesmo AS. Cada roteador tem conhecimento da topologia da rede e armazena dados suficientes para determinar qual o caminho que cada pacote deve seguir a partir dele, assim como computar novas rotas, caso haja mudanças na rede.

O cálculo dos caminhos mínimos utilizados pelo OSPF é realizado, primeiramente, por meio de uma adequada designação de pesos às ligações (*links*) que conectam os roteadores de um AS. Fixando cada roteador como a raiz, OSPF calcula, em seguida, os caminhos de peso mínimo que levam a todos os outros destinos possíveis, usando para isso um algoritmo de caminhos mínimos (Dijkstra, por exemplo). Os pesos devem ser escolhidos de forma a satisfazer um critério técnico ou econômico da empresa provedora, por exemplo, a minimização do congestionamento da rede. Quanto maior o peso associado a uma ligação, menor será o tráfego circulando por ela. É bom lembrar que cada ligação física tem uma capacidade máxima de escoamento de tráfego que deve ser respeitada.

Para melhor formalizar o problema de atribuir pesos aos arcos e encontrar os caminhos mínimos utilizados pelo protocolo OSPF, conhecido como *Weight Setting Problem – WSP*, considera-se que os roteadores sejam os nós de um grafo direcionado e que as ligações entre roteadores constituam o conjunto dos arcos. Considera-se, então, o grafo direcionado $G = (N, A)$, onde N é o conjunto de n nós e A o conjunto de m arcos, uma capacidade c_a associada a cada arco $a \in A$, e uma matriz de demandas D que, para cada par $(s, t) \in N \times N$, especifica a demanda d_{st} entre os nós fonte s e destino t . O problema consiste em atribuir um valor positivo $w \in [1, w_{max}]$ para cada arco $a \in A$, de tal maneira que uma função objetivo Φ que avalie o congestionamento na rede seja minimizada. Os pesos não precisam obedecer a padrões pré-estabelecidos, como por exemplo ser proporcionais às distâncias físicas dos arcos ou inversamente proporcionais às suas capacidades, e sim objetivar a minimização do congestionamento da rede. O WSP é um problema de otimização combinatória não linear e prova-se que é *NP-hard* (Fortz e Thorup, 2000).

Nas seções a seguir, este problema será estudado em maior detalhe. Na Seção 2 uma função de avaliação do congestionamento é apresentada. O pseudocódigo de um algoritmo de avaliação de uma solução é fornecido na Seção 3, enquanto que a Seção 4 apresenta um modelo relaxado para o WSP. A Seção 5 apresenta algumas heurísticas simples para resolver o WSP e na Seção 6 são discutidas três metaheurísticas. Uma comparação entre os métodos de resolução é descrita na Seção 7. Finalizando, a Seção 8 apresenta as conclusões.

2. Função Φ de avaliação de congestionamento

Fortz e Thorup (2000) propuseram uma função Φ que avalia o congestionamento da rede. A cada arco $a \in A$ calcula-se o valor de Φ_a que relaciona o quão próximo da capacidade c_a está a carga l_a que passa por um arco a : quanto maior a proximidade, maior a penalização. O objetivo é distribuir a

carga homogeneamente entre os arcos da rede a fim de minimizar a soma dos custos Φ_a de cada arco a :

$$\Phi = \sum_{a \in A} \Phi_a(l_a).$$

A função Φ_a proposta é convexa e linear por partes, com derivadas definidas como:

$$\Phi'_a(l_a) = \begin{cases} 1 & \text{para } 0 \leq l_a/c_a < 1/3, \\ 3 & \text{para } 1/3 \leq l_a/c_a < 2/3, \\ 10 & \text{para } 2/3 \leq l_a/c_a < 9/10, \\ 70 & \text{para } 9/10 \leq l_a/c_a < 1, \\ 500 & \text{para } 1 \leq l_a/c_a < 11/10, \\ 5000 & \text{para } 11/10 \leq l_a/c_a < \infty. \end{cases}$$

Na Figura 1 pode-se visualizar a função de congestão. Logicamente, se um arco a estiver ocioso, $\Phi_a = 0$. Note-se que, dado que a função penaliza fortemente fluxos que excedem a capacidade, a solução deve ser tal que, se possível, nenhum arco apresente utilização l_a/c_a acima do valor 1. Busca-se assim, um perfil de fluxos o mais homogêneo possível, sem sobrecarregar nenhum arco demasiadamente.

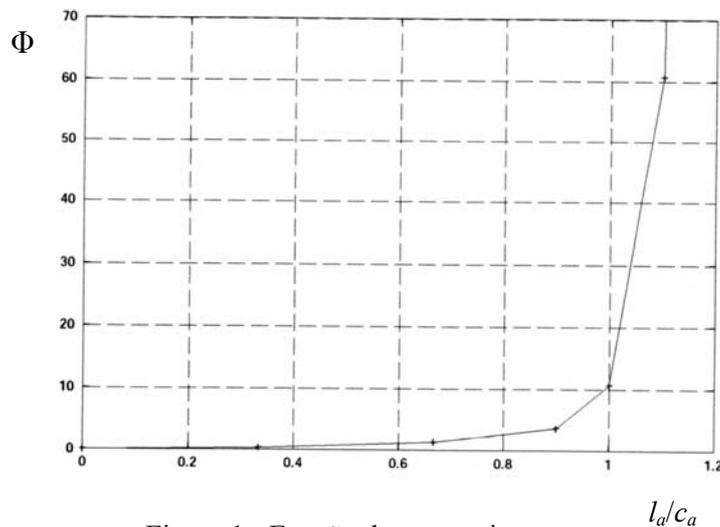


Figura 1 - Função de congestionamento

3. Avaliação de uma solução do WSP

Avaliar o custo de congestão de uma solução do WSP exige um esforço computacional considerável. Uma solução corresponde a um vetor de pesos w , onde cada arco a possui um peso w_a associado. Independentemente de como os pesos sejam atribuídos aos arcos, o procedimento de avaliação é o mesmo. Considera-se a função objetivo Φ mencionada na seção anterior, que calcula Φ_a , o congestionamento de cada arco $a \in A$. A Figura 1 descreve o procedimento básico para avaliar uma solução.

```

1   float avaliação(w)
2   begin
3       para cada nó  $d \in E^* \subseteq E$  faça
4           dist[d] = Dijkstra(d, w);
5           cm[d] = calculeCaminhoMínimo(d, w);
6           delta[d] = conteDelta(d, w);
7           lad[d] = rotearLoad(d, dist[d], cm[d], delta[d], D);
8       fimpara
9        $\Phi = 0$ ;
10      para cada  $a \in A$  faça
11          la = totalLoad(a, lad);
12           $\Phi_a = \text{Phi}(a, lad)$ ;
13           $\Phi = \Phi + \Phi_a$ ;
14      fimpara
15      retorne ( $\Phi$ );
16  end

```

Figura 2 – Procedimento de avaliação de uma solução do problema.

Algumas estruturas de dados utilizadas são dados de entrada do programa, como o grafo $G=(N, A)$, o vetor c^m de capacidades e a matriz de demanda $D^{n \times n}$. Outras são criadas especialmente para a avaliação da solução e podemos chamá-las de memória da solução: $cm^{n \times m}$, $delta^{n \times m}$, $dist^{n \times m}$, $lad^{n \times m}$ e la^m . O algoritmo de caminhos mínimos $Dijkstra()$ é aplicado e a distância de cada nó ao nó destino d é armazenada no vetor $dist[d]$ (linha 4). Cada posição do vetor $cm[d]$ armazena 0 ou 1 caso o arco esteja ou não, respectivamente, no grafo de caminho mínimo respectivo a d , calculado pela função $calculeCaminhoMínimo()$. Para cada nó destino d , calcula-se, para cada nó $e \in N$, o valor $delta[d][e]$, que indica o número de arcos adjacentes a e presentes no grafo de caminho mínimo (linha 6). Tendo a informação $dist[d]$, $cm[d]$, $delta[d]$, além da matriz de demandas D , a carga passando por cada arco, referente a cada $cm[d]$, é calculada pela função $rotearLoad$ (linha 7). Finalmente, a carga total passando por cada arco $a \in A$ é somada (linha 11) e a avaliação do tráfego no arco é calculado (linha 12). A função $\text{Phi}()$, além de calcular o valor de Φ_a , normaliza este valor, como proposto em Fortz e Thorup (2000).

Um limitante inferior para o problema pode ser obtido resolvendo-se o problema de *multi-commodity flow* com custos lineares crescentes (Fortz e Thorup, 2000). Este problema é uma relaxação do roteamento OSPF, pois o fluxo passando por um nó i não é necessariamente dividido igualmente entre os caminhos mínimos partindo de i . É o que será apresentado na próxima seção.

4. Modelo relaxado para WSP

É possível obter uma solução relaxada do WSP por intermédio da resolução de um modelo matemático multiproduto de fluxo de custo mínimo (Ahuja et al., 1993). A utilidade de tal modelo é ter-se acesso a um limitante inferior à solução ótima do WSP, que pode ser usado para comparar a qualidade de soluções do WSP obtidas por métodos heurísticos. Considera-se a variável de fluxo x_a^{st} que indica o quanto da demanda entre os nós s e t passa pelo arco a e a variável l_a que indica o fluxo total que passa pelo arco a . As demais definições são as mesmas válidas para o modelo WSP, a menos dos pesos nos arcos, que são desconsiderados aqui. O modelo matemático é formulado como:

$$\text{Min } \Phi = \sum_{a \in A} \Phi_a$$

sujeito a

$$(1) \quad \sum_{u:(u,v) \in A} x_{uv}^{st} - \sum_{u:(v,u) \in A} x_{vu}^{st} = \begin{cases} -d_{st} & \text{se } v = s, \\ d_{st} & \text{se } v = t, \\ 0 & \text{caso contrário} \end{cases} \quad v, s, t \in N,$$

$$(2) \quad l_a = \sum_{st \in N \times N} x_a^{st} \quad a \in A,$$

$$(3) \quad \Phi_a \geq l_a \quad a \in A,$$

$$(4) \quad \Phi_a \geq 3l_a - 2/3c_a \quad a \in A,$$

$$(5) \quad \Phi_a \geq 10l_a - 16/3c_a \quad a \in A,$$

$$(6) \quad \Phi_a \geq 70l_a - 178c_a \quad a \in A,$$

$$(7) \quad \Phi_a \geq 500l_a - 1468c_a \quad a \in A,$$

$$(8) \quad \Phi_a \geq 5000l_a - 19468c_a \quad a \in A,$$

$$(9) \quad x_a^{st} \geq 0 \quad a \in A; \quad s, t \in N.$$

A função objetivo minimiza o custo total de congestão, calculado como a soma dos custos em todos os arcos da rede. As restrições (1) estabelecem o balanço de fluxo em cada nó da rede, ao mesmo tempo que garantem que a demanda de tráfego entre cada par (s,t) seja satisfeita. A carga em cada arco a é composta dos fluxos advindos de cada par (s,t) e está assegurada pelas restrições (2). As restrições (3-8) definem os valores adequados de custo da função linear por partes em função da carga no arco, caracterizando a forma clássica que transforma um modelo linear por partes em um linear. O modelo acima é caracterizado como um problema de fluxos em rede multiproduto não capacitada e pode ser resolvido otimamente em tempo polinomial usando qualquer algoritmo de programação linear ou de fluxos em redes.

5. Heurísticas propostas para WSP

Até o presente não há método exato proposto para solucionar o WSP. Na categoria de heurísticas há várias propostas, desde as simples heurísticas construtivas até metaheurísticas sofisticadas. Neste primeiro grupo é possível destacar algumas heurísticas simples para designação dos pesos aos arcos. Logicamente, os resultados obtidos são em geral não satisfatórios principalmente se o tráfego na rede é intenso. A seguir, enumeram-se quatro heurísticas simples:

- pesos com valores proporcionais às distâncias euclidianas entre as extremidades dos arcos;
- pesos designados com valores aleatórios;
- pesos iguais a 1 para todos os arcos; e
- pesos inversamente proporcionais a sua capacidade.

Esta última é a opção da Cisco (Thomas, 1998), que é uma importante fornecedora de equipamentos para roteadores. O peso de um arco a é inversamente proporcional a (c_{ma}/c_a) , onde c_{max} é a maior capacidade da rede.

Técnicas de buscas locais dependem da definição precisa de uma vizinhança. Uma vizinhança $N(w)$ de uma solução w dada é constituída por todas as soluções w' possíveis de serem obtidas por meio de uma perturbação determinística da solução w . As soluções da vizinhança são então

sucessivamente avaliadas em termos da função objetivo e, no caso de haver uma solução w' com melhor avaliação que w , então ela é aceita como a nova solução e uma nova vizinhança $N(w')$ é construída. O algoritmo termina numa solução chamada de um ótimo local, quando nenhuma solução da vizinhança é melhor do que a solução corrente.

No caso do WSP, pode-se definir como uma vizinhança $v1$ de uma solução w todas as soluções w' que possuem um único arco com peso diferente. Mais formalmente, uma solução w' vizinha de w é qualquer solução que possui $w'_i \neq w_i$ para um arco qualquer i , e $w'_a = w_a$ para todo $a \neq i \in A$.

Uma vizinhança $v2$ maior é formada por toda solução w' que possui um conjunto C de arcos com pesos diferentes dos correspondentes em w . Quanto maior o tamanho do conjunto C , maior é a vizinhança. Uma vizinhança reduzida $vr1$ pode restringir a mudança dos pesos a um subconjunto $S1 \subset A$. Uma vizinhança reduzida $vr2$ pode restringir a faixa em que o valor do peso de um arco a , ou um conjunto de arcos, pode diferir entre as soluções. Uma terceira redução $vr3$ pode ser restringir os pesos a somente aumentarem ou diminuírem de valor.

A avaliação de qualquer solução w' pertencente a uma das vizinhanças definidas acima pode ocorrer de forma diferente de como foi descrito no pseudocódigo da Figura 2. Isso porque a memória da solução w pode ser reaproveitada em parte para o cálculo da solução w' . Ramalingam and Reps (1998) propuseram um algoritmo, denominado *RR*, que, ao modificar o valor do peso de um arco, somente a parte afetada do grafo de caminhos mínimos é recalculada. Estes algoritmos são conhecidos como algoritmos para caminhos mínimos dinâmicos (*dynamic shortest paths*) e podem facilmente ser generalizados para atualizar um grafo de caminhos mínimos quando um conjunto de arcos mudam de peso. Usando esta mesma idéia de aproveitar a memória da solução anterior e apenas atualizar a parte realmente modificada, Fortz and Thorup (2000) propuseram um algoritmo que recalcula as cargas apenas dos arcos que tiveram aumento ou diminuição do tráfego (mudança de carga).

A Figura 3 apresenta o pseudo-código do cálculo de uma solução w' , da vizinhança de w , que possui apenas o arco a com valor diferente de peso ($w'_a \neq w_a$ e $w'_e = w_e$ para to $e \in A$).

```

1   float avaliaçãoUpdated( $w'$ ,  $a$ ,  $w'_a$ )
2   begin
3       para cada nó  $d \in E^* \subseteq E$  faça
4            $dist[d] = RR(d, a, w'_a, w')$ ;
5            $lad[d] = update\_lad(d, dist[d], cm[d], delta[d], D)$ ;
6       fimpara
7       para cada  $a \in A$  faça
8            $la = totalLoad(a, lad)$ ;
9            $\Phi_a = Phi(a, lad)$ ;
10           $\Phi = \Phi + \Phi_a$ ;
11      fimpara
12      retorne ( $\Phi$ );
13  end

```

Figura 3 – Procedimento de avaliação rápida da vizinhança.

A função $RR()$ atualiza as memórias $dist$, $delta$ e cm , enquanto que o $update_lad()$ codifica a atualização da carga proposta por Fortz e Thorup (2000). O laço **para** das linhas 7-11 também poderiam ser atualizados por $update_lad()$, mas como é composto por operações muito simples, a diferença entre atualizar e recomputar estes dados é muito pequena.

Várias heurísticas baseadas em buscas locais simples foram propostas (Rodrigues e Ramakhishnan 1994, Lin e Wang 1993 e Bley et al. 1998). Os dois primeiros usam vizinhanças para orientar uma busca local e o último lança mão de relaxação Lagrangiana. Uma característica comum a todos esses métodos é que eles alocam pesos aos arcos de forma que as demandas sejam roteadas por um único caminho de peso mínimo. Além disso, apenas redes de pequena dimensão foram avaliadas

nos testes computacionais realizados. Atualmente, metaheurísticas mais sofisticadas têm sido aplicadas com sucesso para tratar redes de porte realístico, como será visto na próxima seção.

6. Metaheurísticas propostas

A literatura especializada de otimização combinatória consagrou o uso da palavra metaheurística para designar o conjunto de técnicas que se utilizam de uma estratégia mestre que se superpõe a uma heurística (em geral uma busca local) de forma a guiá-la e modificá-la com o fim de produzir soluções capazes de transcender a otimalidade local.

6.1 Busca tabu

Fortz e Thorup (2000) propuseram um algoritmo de busca tabu (BT) para resolver o WSP, que permite achar soluções que dividem igualmente o fluxo caso o conjunto de pesos alocado aos arcos consagrem mais de um caminho mínimo entre um par origem-destino. Este é um preceito básico do protocolo OSPF que é relaxado pelos métodos heurísticos relacionados na seção anterior. A implementação de BT usa funções e tabelas *hash* (Woodruff e Zemel, 1993) que podem ser empregadas em substituição aos aparatos necessários para compor uma estrutura de memória atributiva, o mais comum dos mecanismos usados para evitar ciclagens em BT. Para evitar visitar soluções já avaliadas numa vizinhança dada, usa-se uma função *hash* que mapeia uma atribuição de pesos nos arcos em um número inteiro. É criada uma tabela *hash* onde os inteiros são guardados por um certo número de iterações e usados para evitar que se avaliem soluções já visitadas. Este procedimento é considerado prático, pois para muitos problemas é simples guardar e buscar numa lista de inteiros, comparandp-se com a avaliação da vizinhança.

O método usa uma tabela *hash* com $2^{16} \approx 64.000$ entradas, usada para evitar ciclagem, e uma segunda tabela *hash*, com $(2^{16})/20$ entradas, que é usada para reaproveitar o cálculo de soluções já computadas e também como estratégia de diversificação. A entrada numa tabela obtida por uma solução é calculada através de uma operação XOR dos valores h_a calculados para cada arco. Para cada arco $a \in A$ tem-se associado um valor inteiro p_a de $(l+m-1)$ bits, tendo este sido gerado aleatoriamente. O cálculo de $p_a * w_a$ gera um inteiro de $(2m+l-1)$ bits, do qual extraem-se os m bits iniciais e os $m-1$ bits finais. O valor de h_a é o inteiro de l bits restante do cálculo de $p_a * w_a$.

Uma vizinhança para busca local dessa BT é qualquer solução caracterizada por νl , definida na Seção 5. Uma redução de νl é usada, selecionando-se, aleatoriamente, apenas 20% dos arcos. O algoritmo executa por 5000 iterações. Se a solução incumbente não melhorar por 300 gerações, é somado ao peso de 10% dos arcos um valor $\nu \in [-2, -1, 1, 2]$, escolhido aleatoriamente para cada arco.

6.2 Algoritmos populacionais

Até o momento dois métodos populacionais foram propostos para o WSP. Em ambos é respeitado o preceito do protocolo OSPF que determina que o fluxo deva ser igualmente direcionado por caminhos de mesmo peso mínimo.

Ericsson, Resende e Pardalos (2001) propuseram um algoritmo genético (AG) com população de 100 soluções ordenadas pelo valor de solução. A população é estruturada em 3 classes:

- α 20% melhores soluções;
- δ 10% piores;
- β 70% soluções restantes.

O operador de *crossover* é simples e elitista. Basicamente, para cada arco ele seleciona o peso do pai A com 70% de chance e o peso do pai B com 30%. O pai A é sempre melhor, visto que este é aleatoriamente selecionado entre as soluções componentes da classe α , enquanto que o pai B é selecionado dentre os elementos das classes $\beta \cup \delta$. O algoritmo roda por 500 gerações para instâncias

com 100 nós e por 700 gerações para as instâncias menores. O tamanho da população é composta por 100 soluções para instâncias com menos de 100 nós e por 200 soluções para instâncias maiores.

O algoritmo genético híbrido (AGH) proposto por Buriol, Resende, Ribeiro e Thorup (2003) mantém as características básicas do AG, mudando o tamanho da população (50) e a percentagem de soluções em cada classe da população ($\alpha=25\%$, $\delta=5\%$ e $\beta=70\%$). Para cada solução nova criada após *crossover*, uma busca local é aplicada. A vizinhança é igual a νl , usando três reduções: o peso pode apenas aumentar de valor, o aumento é unitário e apenas alguns arcos são selecionados. A busca local consiste basicamente em ordenar os arcos em ordem decrescente por avaliação Φ_a e testar o incremento do valor do peso w_a de k arcos. O incremento é feito unitariamente dentro de uma faixa $(w_{max} - w)/4$. Como a busca local é aplicada a 70% das soluções a cada geração, fez-se uma considerável redução de vizinhança usando um valor de $k=5$. O algoritmo executa por 500 gerações.

7. Desempenho de heurísticas e metaheurísticas

Nas comparações realizadas entre os algoritmos, as mesmas instâncias são consideradas: uma instância real e três classes de instâncias sintéticas. Os pesos foram escolhidos na faixa [1,20], visto que $w_{max} > 20$ (Fortz and Thorup, 2000) não beneficia o desempenho dos algoritmos. A primeira classe (*att*) é uma rede real com 90 roteadores e 274 arcos. A segunda classe (*hier*) é constituída de instâncias geradas com dois níveis hierárquicos com diferentes capacidades. A terceira classe (*rand*) contém redes geradas de forma aleatória. A quarta classe (*wax*) tem problemas gerados de forma que os nós são uniformemente distribuídos num quadrado unitário e a existência de um arco depende de uma probabilidade controlada por um parâmetro. As demandas são geradas de tal maneira que elas sejam maiores entre pares de nós mais próximos um do outro e também para forçar que alguns nós sejam caracterizados como sendo preponderantemente geradores ou recebedores de tráfego. De modo geral, as instâncias usadas neste artigo representam a média das soluções obtidas por 12 matrizes de demanda para cada instância. A matriz de demandas da rede original é multiplicada por um fator ρ para que seja obtida uma segunda matriz, que por sua vez é multiplicada por ρ para que seja obtida uma terceira. Este procedimento se repete até que se tenham 12 redes. Devido a este procedimento, uma rede r_i terá menor congestionamento que uma rede r_{i+1} gerada posteriormente. Detalhes podem ser encontrados em [3,4].

Na Tabela 2 é mostrado o resultado da comparação dos métodos, a saber:

- InvCap = Procedimento heurístico recomendado pela Cisco e descrito na Seção 5,
- AG = Algoritmo genético de Ericsson et al. (2001) descrito na Seção 6.2,
- AGH = Algoritmo genético híbrido de Buriol et al. (2003) descrito na Seção 6.2,
- BT = Busca tabu de Fortz e Thorup (2000) descrito na Seção 6.1,
- LI = Limitante inferior obtido pela resolução do modelo multiproducto descrito na Seção 4.

Os valores da tabela correspondem a uma média de 10 rodadas para cada instância e cada método é comparado pelo valor do congestionamento (sol.) e pela sua diferença percentual em relação ao limitante inferior (gap%). Os resultados das metaheurísticas AG, AGH e BT foram alcançados após 1 hora de CPU para cada uma delas. O número associado aos nomes das instâncias correspondem ao número de nós da rede.

As metaheurísticas apresentam resultados de congestionamento muito melhores do que a heurística InvCap, às custas de tempos computacionais bem maiores. Da comparação entre AG e AGH, nota-se que AGH é superior em todas as instâncias. Quando se comparam AGH e BT, as conclusões não são tão claras. Enquanto AGH apresenta melhor performance para as classes *att* e *hier*, BT apresenta melhores resultados para as classes *rand* e *wax*. É relevante salientar que BT apresenta uma média geral mais baixa do que o AGH, mas que isso é devido aos resultados observados para as instâncias *rand100* e *rand100b*, onde houve uma discrepância acentuada, o que influenciou demasiadamente na média geral.

Tabela 2 - Comparações entre métodos heurísticos e metaheurísticas

Problema	Demanda	InvCap		AG		AGH		BT		LI
		sol	gap%	sol	gap%	sol	gap%	sol	gap%	Sol
att90	45134,1	652,7	1842,6	40,8	21,4	35,1	4,5	35,3	5,1	33,6
hier50a	4927,6	626,1	1973,2	39,5	30,8	31,5	4,3	33,7	11,6	30,2
hier50b	3362,7	530,7	3461,7	27,8	86,6	18,5	24,2	18,3	22,8	14,9
hier100	45134,1	281,6	537,1	45,5	2,9	44,6	0,9	44,7	1,1	44,2
hier100a	12406,5	392	1228,8	44,8	51,9	33,5	13,6	36,4	23,4	29,5
Média			1800,2		43,0		10,7		14,7	
rand50	42281,1	681	2041,5	50,4	58,5	33,8	6,3	34,4	8,2	13,75
rand50a	53561,5	1118,4	2627,8	201,5	391,5	77,6	89,3	80,4	96,1	19,65
rand100	69296,8	652,8	2713,8	179,6	674,1	47,6	105,2	29,7	28,0	5,79
rand100b	100594,3	335,9	1447,9	290,4	1238,2	34,9	60,8	23,7	9,2	5,87
Média			2207,8		590,6		65,4		35,4	
wax50	25411,4	302,4	1360,9	33,4	61,4	32,1	55,1	32,1	55,1	302,4
wax50a	39446,6	141,7	641,9	28,1	47,1	27,3	42,9	27,2	42,4	141,7
wax100	48473,7	173,8	414,2	37,2	10,1	34,7	2,7	34	0,6	173,8
wax100a	63493,1	241,2	478,4	54,5	30,7	43,7	4,8	43,1	3,4	241,2
Média			723,8		37,3		26,4		25,4	
Média Geral			1597,7		208,1		31,9		23,6	

Tanto AGH como BT fizeram uso da avaliação rápida da vizinhança mencionada na Seção 5. A avaliação de uma solução ficou, aproximadamente, 15 vezes mais rápida para BT e 30 vezes mais rápida para AGH. Especificamente nos casos em que o peso muda unitariamente, o algoritmo de Ramalingam e Reys (1998) pode ser especializado de modo a evitar o uso de pilhas (*heaps*), o que reduz em 40 vezes o tempo computacional gasto pelo algoritmo *Dijkstra*(). Como a busca local de AGH é baseada em incrementos unitários, esta pode se beneficiar da rapidez do algoritmo especializado.

8. Conclusões

Um dos problemas que emergiram do rápido aumento do uso da Internet foi o risco de congestionamentos na rede. Protocolos e políticas de gerenciamento da rede procuram ordenar e otimizar o tráfego de modo a minimizar os riscos de congestionamento. Um dos problemas de otimização relacionados ao protocolo OSPF é o *Weight Setting Problem* - WSP, que procura adequadamente designar pesos aos arcos da rede entre roteadores de modo que o tráfego possa escoar pelos caminhos de peso mínimo. Embora este problema possua aplicação atual e sua otimização represente forte impacto no desempenho das redes provedoras, somente nos 3 últimos anos foi feito um esforço maior da área de otimização para resolvê-lo. Neste artigo se apresentam algumas das características matemáticas deste problema e dois modelos matemáticos para a sua solução, e se comparam algumas heurísticas que buscam a minimização do custo de congestionamento.

São também mostrados alguns procedimentos especializados que buscam melhorar o desempenho de certas tarefas utilizadas pelos métodos, como o cálculo rápido das soluções vizinhas. Os resultados computacionais sobre quatro conjuntos diferentes de instâncias mostram que as

implementações de metaheurísticas são muito superiores às heurísticas simples. Da comparação entre AG e AGH, nota-se que AGH é superior em todas as instâncias. Quando se comparam AGH e BT, as conclusões são menos claras. Enquanto AGH apresenta-se melhor nas classes *att* e *hier*, BT é superior nas classes *rand* e *wax*.

Devido a WSP ser um problema ainda pouco estudado, abre possibilidades para técnicas mais elaboradas e mais eficientes, principalmente quando se considera que existem redes reais de grande porte e que as demandas tendem a crescer no futuro. Além disso, há uma lacuna a ser preenchida em termos de métodos exatos.

Referências bibliográficas

1. Buriol, L.S, Resende, M.G.C., Ribeiro, C.C. e Thorup, M. "A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing", Technical Report, AT&T Labs. Research, 180 Park Avenue, Florham Park, NJ, 07932, USA, 2003.
2. Ahuja R.K., Magnanti, T.L. e Orlin, J.B., *Network Flows*, Prentice Hall, 1993.
3. Bley, A., Grötchel, M. e R. Wessläy. "Design of broadband virtual private networks: Model and heuristics for the B-WiN". Technical Report SC 98-13, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1998. A ser publicado em *Proceedings of the DIMACS Workshop on Robust Communication Network and Survivability, AMS-DIMACS Series*.
4. Ericsson, M., Resende, M.G.C. e Pardalos, P.M., "A genetic algorithm for the weight setting problem in OSPF routing". *Journal of Combinatorial Optimization* 6 (2002), 299-333.
5. Fortz, B. e Thorup, M., "Internet traffic engineering by optimizing OSPF weights", *Proceedings of INFOCOM*, 2000.
6. Lin, F. e Wang J., "Minmax open shortest path first routing algorithms in networks supporting the smds services". *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 2, pp.666-670, 1993.
7. Moy, J. T. *OSPF, Anatomy of an Internet Routing Protocol*, Addison-Wesley, 1998.
8. Rodrigues, M. e Ramakrishnan, K.G., "Optical routing in data networks", Presentation at the *International Telecommunications Symposium (ITS)*, Rio de Janeiro, Brasil, 1994.
9. Ramalingam, G. e Reps, T., "An incremental algorithm for a generalization of the shortest-path problem". *Journal of Algorithms* 21 (1996), 267-305.
10. Thomas, T.M., *OSPF Network Design Solutions*. Cisco Press, 1998.
11. Woodruff, D.L e Zemel, E., "Hashing vectors for tabu search", *Annals of Operations Research* 41 (1993), 123-137.