

Algoritmo memético para o problema de designação de pesos para o protocolo DEFT

Roger Reis¹, Marcus Ritt¹, Luciana S. Buriol¹, Mauricio G.C. Resende²

¹ *Instituto de Informática - Universidade Federal do Rio Grande do Sul - UFRGS*

² *AT&T Labs Research*

{rsreis, mrpritt, buriol}@inf.ufrgs.br, mgcr@research.att.com

Resumo

Ao usar protocolos de roteamento internos como OSPF (*Open Shortest Path First*) e DEFT (*Distributed Exponentially-weighted Flow Splitting*) é necessário ajustar os pesos dos links para permitir o roteamento. O problema de encontrar um conjunto de pesos que minimizem o congestionamento para estes protocolos é conhecido como Problema de Designação de Pesos. Este trabalho apresenta um algoritmo memético que pode ser aplicado tanto ao protocolo OSPF quanto ao protocolo DEFT, utilizando algoritmos dinâmicos para o cálculo do fluxo e dos caminhos mínimos. Para o DEFT, propõe-se um algoritmo dinâmico que precisa em média de 30% menos atualizações do fluxo do grafo, melhorando o desempenho em relação ao nosso trabalho anterior.

PALAVRAS-CHAVE: Protocolo de roteamento, problema de designação de pesos, DEFT.

Redes, Metaheurística.

Abstract

To use interior gateway protocols like OSPF (*OpenShortest Path First*) and DEFT (*Distributed Exponentially-weighted Flow Splitting*) it is necessary to set the link weights to allow data routing. The problem of finding suitable weights for these protocols is known as the weight setting problem. In this paper we present a memetic algorithm which can be applied to both the OSPF and DEFT protocols, using dynamic flow and dynamic shortest paths calculation. For DEFT, we propose a dynamic algorithm that requires about 30% less graph flow updates, improving the performance over our previous work.

KEYWORDS: Routing protocol, weight setting problem, DEFT.

Networks, Metaheuristic.

1 Introdução

A Internet é dividida em Sistemas Autônomos (*Autonomous Systems*, ASes). Cada AS controla seu roteamento interno por meio de um *interior gateway protocol* (IGP). Protocolos IGP como o *open shortest path first* (OSPF) e *intermediate system-intermediate system* (IS-IS) permitem que o operador de rede ajuste as rotas configurando pesos inteiros nos seus links. O problema de encontrar um conjunto de pesos que otimize alguma função objetivo como por exemplo o congestionamento total da rede, a utilização máxima dos links ou a latência, é conhecido por Problema de Designação de Pesos (*Weight Setting Problem*, WSP).

Para aproveitar toda a capacidade da rede, uma função objetivo, as regras de roteamento do protocolo e as expectativas de demandas de dados são considerados para calcular a configuração de pesos com menor custo. O problema de designação de pesos para o OSPF é NP-Difícil segundo Fortz and Thorup (2000). Entre os melhores resultados para o OSPF encontram-se a abordagem por Busca Tabu em Fortz and Thorup (2000) e o algoritmo memético (MA) em Buriol et al. (2005).

O protocolo DEFT é um protocolo de roteamento IGP, do tipo estado de enlace, proposto em Xu et al. (2007), concebido para facilitar a otimização da designação do conjunto de pesos em uma topologia determinada e para uma demanda conhecida. Os resultados aqui apresentados representam uma melhoria de tempo em relação a um estudo anterior, obtidos por uma otimização nos códigos, eliminando estruturas de dados para manter referências aos caminhos mínimos. Outro avanço foi o desenvolvimento de um algoritmo para cálculo dinâmico do fluxo de dados, detalhado na seção 6.

Este trabalho está organizado da seguinte forma. Na Seção 2 é apresentada a descrição do problema geral de roteamento. Na Seção 3 são introduzidos os protocolos de roteamento OSPF e DEFT. A seguir, na Seção 4, é apresentado o algoritmo memético proposto para o OSPF. Na Seção 5, são introduzidas as modificações aplicadas ao MA, para executar o roteamento segundo as regras do DEFT. A Seção 6 apresenta em detalhes o algoritmo de atualização dinâmica de fluxo modificado para DEFT. Os resultados do MA proposto para DEFT são apresentados na Seção 7. Finalizando, a Seção 8 apresenta as principais conclusões do estudo e seu direcionamento futuro.

2 O problema geral de roteamento

Seja $G = (V, E)$ um grafo dirigido, $c_{u,v}$ as capacidades dos links, e D a matriz de demandas onde D_{uv} denota o tráfego do nodo origem u para o nodo destino v , para $1 \leq u, v \leq |V|$. Define-se T como um subconjunto de nodos $T \subseteq V$ com todos os nodos que são destino em pelo menos um par de demanda, i.e. $T = \{v | D_{uv} > 0\}$. O problema geral de roteamento consiste em encontrar os fluxos $f_{u,v}$ que satisfaçam todas as demandas e que minimizem a função objetivo (utilização máxima dos links, por exemplo). Desta forma, temos

$$\text{minimizar} \quad \sum_{(u,v) \in E} \Phi(f_{u,v}, c_{u,v}) \quad (1)$$

onde Φ é a função de custo para um link. Uma escolha típica de Φ é a função linear por partes proposta em Fortz and Thorup (2000, 2004) e apresentada na Figura 1.

Seja $f_{u,v}^t$ o fluxo no link (u, v) destinado ao nodo t . Então qualquer fluxo resultante deve respeitar as restrições de conservação de fluxo nos nodos intermediários $v \neq t$

$$\sum_{(u,v) \in E} f_{u,v}^t - \sum_{(v,w) \in E} f_{v,w}^t = D_{v,t} \quad (2)$$

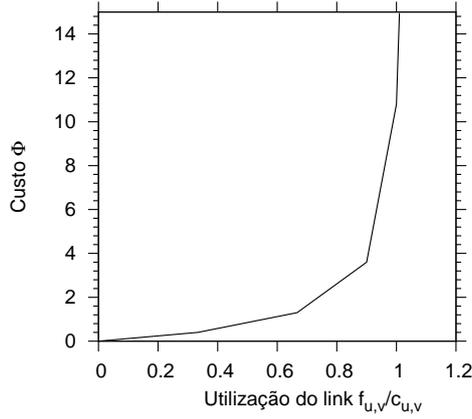


Figura 1: Custo Φ para link em função da sua utilização para $c_{u,v} = 1$.

e a agregação de fluxos individuais

$$f_{u,v} = \sum_{t \in T} f_{u,v}^t. \quad (3)$$

Como a função objetivo e todas as restrições são lineares, pode-se encontrar uma solução ótima resolvendo o programa linear OPT dado pelas equações 1, 2 e 3 junto com $0 \leq f_{u,v}^t, 0 \leq f_{u,v}$. OPT resolve o fluxo multi-produto, i.e., sem considerar quaisquer restrições exceto a capacidade dos links frente a uma demanda determinada. Não é usado na prática, mas serve como um limitante inferior para avaliar protocolos de roteamento. A seguir comparam-se as características dos protocolos OSPF e DEFT quanto ao roteamento.

3 Os protocolos OSPF e DEFT

No protocolo OSPF, o fluxo é determinado por pesos inteiros $w_{u,v} \in [0, 2^{16} - 1]$ atribuídos a cada link. Os roteadores trocam informações sobre os links, por exemplo, sobre seus caminhos mínimos. Cada roteador u usa estes pesos para calcular o grafo de caminhos mínimos para todos os destinos. Então distribuem o tráfego que chega destinado a um nodo t igualmente por todos os links saintes, que fazem parte de um caminho mínimo para t .

O protocolo DEFT relaxa esta restrição e permite pesos reais $w_{u,v} \in \mathbb{R}$. Ele distribui o fluxo entre todos os links saintes desde que se direcionem ao nodo destino. Os links que não fazem parte do caminho mínimo recebem um fluxo decrescente com penalidades exponenciais maiores para caminhos mais longos. Formalmente, seja d_i^t a distância do nodo i ao nodo destino t , e $h_{u,v}^t = d_v^t + w_{u,v} - d_u^t$ a diferença entre o comprimento do caminho mínimo e o comprimento do caminho usando o link (u, v) . Então a fração de tráfego não-normalizado Γ passante no link (u, v) , dirigida a t , é calculada por

$$\Gamma(h_{u,v}^t) = \begin{cases} e^{-h_{u,v}^t} & \text{se } d_u^t > d_v^t \\ 0 & \text{caso contrário} \end{cases} \quad (4)$$

A fração do fluxo total $\Gamma(h_{u,v}^t) / \sum_{v:(u,v) \in E} \Gamma(h_{u,v}^t)$ é calculada para cada link sainte (u, v) de u . De acordo com Xu et al. (2007), em termos de custo para a máxima utilização dos links, sempre há um conjunto de pesos para o DEFT com resultado melhor do que para o OSPF.

Encontrar tais pesos, isto é, resolver o problema de designação de pesos de forma ótima, é uma tarefa difícil. De fato, foi demonstrado em Fortz and Thorup (2000) que calcular tais pesos minimi-

zando a utilização máxima dos links baseado nas regras do OSPF é um problema NP-Difícil. Assim, para o OSPF, vários autores propuseram heurísticas, incluindo algoritmos genéticos em Ericsson et al. (2002), algoritmos meméticos em Buriol et al. (2005) e busca tabu em Fortz and Thorup (2000, 2004). Para o DEFT, Xu et al. (2007) propôs um método iterativo de dois estágios, baseado em otimização não linear.

4 Um algoritmo memético para o OSPF

Salvo raras exceções, os algoritmos meméticos, introduzidos em Moscato (1989), fornecem melhores resultados para problemas resolvidos via algoritmos genéticos. Ainda, são aplicados a problemas nas mais diversas áreas, tais como bioinformática, roteamento de dados, clusterização, entre outras aplicações.

Em Ericsson et al. (2002), um algoritmo genético com população estruturada é apresentado para o roteamento OSPF. Nesta estrutura, os indivíduos são classificados em três classes, de acordo com sua avaliação. Um indivíduo é representado por um vetor onde cada posição corresponde ao peso de um arco do grafo. A classe \mathcal{A} é composta pelos 25% melhores indivíduos, a classe \mathcal{C} é composta por 5% dos indivíduos com avaliações mais baixas e os restantes compõem a classe \mathcal{B} . As soluções da classe \mathcal{A} passam diretamente à próxima geração. As soluções da classe \mathcal{C} são substituídas por novas geradas aleatoriamente. As restantes são substituídas por soluções geradas pela recombinação entre um indivíduo da classe \mathcal{A} e outro do conjunto $\mathcal{B} \cup \mathcal{C}$.

O operador de recombinação é implementado como um esquema de chave aleatória que prioriza, com 70% de chances, genes originados do indivíduo de classe \mathcal{A} . A abordagem de busca local é aplicada em cada solução gerada pelo procedimento de *crossover*. Este procedimento examina o efeito de aumentar os pesos de um subconjunto dos arcos. Os arcos candidatos são selecionados entre aqueles com maior custo de roteamento, isto é, que estão sobrecarregados, e que possuam peso menor do que um máximo permitido. Para reduzir o custo de roteamento de um arco candidato, o procedimento busca aumentar seu peso para induzir à redução de sua carga. Neste trabalho são utilizados incrementos unitários. Se esta tentativa levar à diminuição do custo geral de roteamento, ela é aceita e o procedimento é reiniciado. Este procedimento executa avaliações sucessivas de uma única solução, com grande gasto computacional. Para acelerar este processo, os grafos de caminhos mínimos, bem como os fluxos, são atualizados dinamicamente, ao invés de inteiramente recalculados.

O procedimento de busca local visa melhorar a solução candidata corrente avaliando sua vizinhança em busca de soluções com menor custo. Caso uma solução melhor seja encontrada ela é retornada, senão a solução corrente permanece como um mínimo local. A busca local é aplicada às soluções originadas do operador de recombinação em uma vizinhança reduzida para não diminuir a diversidade da população. Este procedimento leva ao aumento da habilidade de encontrar soluções melhores.

5 Um algoritmo memético aplicado às regras do DEFT

Na adaptação do algoritmo memético discutido acima para considerar as regras do DEFT, foram modificados os procedimentos de avaliação das soluções e de busca local.

Segundo as regras do OSPF o fluxo em cada nodo é igualmente repartido entre todos os links saintes nos menores caminhos em direção ao destino t . No DEFT, a carga em cada nodo u é dividida entre *todos* os links saintes (u, v) , mesmo que não estejam em caminhos mínimos, e desde de que aproximem de t , i.e., $d_u^t > d_v^t$. Ainda, a divisão da carga não é igual entre todos os links como

no OSPF. O protocolo DEFT aplica uma penalidade exponencial crescente à medida que a distância para o destino cresce entre pares origem-destino. Assim, os caminhos mínimos são privilegiados, mas caminhos alternativos também são considerados recebendo menos carga quanto maior for a distância para o destino. Distâncias maiores, i.e, com $h_{u,v}^t$ maior que um limite determinado, não recebem qualquer tráfego para evitar uma fragmentação demasiada do fluxo.

```

procedimento CostEvalDEFT( $w = w_1, w_2, \dots, w_{|V|}$ )
1  para  $\forall t \in T$ 
2       $d = \text{DijkstraReverso}(t, w)$ 
3       $G^t = \text{ComputarGrafoCaminhosMínimos}(w, d)$ 
4      [ComputePartialLoads( $d, G^t, D$ )]
5       $H = \text{Nodos ordenados em ordem decrescente de distância}$ 
6      para cada  $u \in H$ 
7           $\Gamma_{total} = 0$ 
8          para cada  $v \in OUT(u)$ 
9              se  $d_u^t > d_v^t$  então
10                  $h_{u,v}^t = d_v^t + w_{u,v} - d_u^t$ 
11                  $\Gamma_{total} += e^{-h_{u,v}^t}$ 
12             fimse
13         fimpara
14          $f = (D_{u,v} + \sum_{(u,v) \in G^t} f_{(u,v)}^t) / \Gamma_{total}$ 
15         para cada  $a \in OUT(u)$ 
16             se  $d_u^t > d_v^t$  então
17                  $h_{u,v}^t = d_v^t + w_{u,v} - d_u^t$ 
18                  $\gamma = e^{-h_{u,v}^t}$ 
19                  $f_{u,v}^t = f * \gamma$ 
20             fimse
21         fimpara
22     fimpara
23     [fim de ComputePartialLoads]
24     para cada  $(u, v) \in A$   $f_{u,v} += f_{u,v}^t$ 
25 fimpara
26  $\Phi = \sum_{(u,v) \in E} \phi(u, v)$ 
fim CostEvalDEFT.

```

Figura 2: Pseudocódigo descrevendo a avaliação de soluções para DEFT.

A Figura 2 descreve, em pseudocódigo, como as regras do DEFT são implementadas. Como no OSPF, para cada nodo destino $t \in T$ computa-se o grafo dos caminhos mínimos reversos G^t (linhas 2–3). As linhas 4–23 apresentam o procedimento `ComputePartialLoads` que implementa as regras do DEFT que permitem que o fluxo seja roteado por caminhos não mínimos. Na linha 5, ordenam-se os nodos em ordem decrescente de distâncias para t . O laço nas linhas 8–13 calcula a soma (Γ_{total}) da função exponencial (Equação 4) para cada link saínte do nodo corrente. Denota-se por $OUT(u) = \{v | (u, v) \in E\}$ o conjunto de links saíntes do nodo u . Na linha 14, é calculada a fração f da demanda (passante e originada do nodo corrente) por cada unidade de Γ .

No laço das linhas 15–21, o fluxo passante de cada link saínte do nodo u é calculado de acordo com sua proporção de Γ . Na linha 24, a carga total de cada arco é atualizada com a soma das parciais calculadas para todos os nodos destinos $t \in T$. Finalmente, na linha 26, calcula-se a avaliação da solução.

O procedimento de busca local também foi alterado. O algoritmo de atualização dinâmica do grafo de caminhos mínimos foi mantido e um algoritmo para atualização dinâmica da distribuição do fluxo foi desenvolvido. Em média, os grafos testados têm de 60% a 70% de seus links afetados quando o peso de um link muda. A economia alcançada permite um maior número de gerações sendo calculadas

para um determinado limite de tempo estipulado. O princípio aplicado ao OSPF permanece o mesmo, isto é, a redução do congestionamento geral através do aumento do peso de links com maior carga. Observe que estes procedimentos não estão ajustados para pesos reais.

6 O algoritmo de fluxo dinâmico para DEFT

O protocolo DEFT distribui o fluxo por caminhos mínimos e não mínimos para um nodo destino. Uma única mudança de peso em um arco pode afetar a distribuição de fluxo de uma parte significativa da rede. Para minimizar o gasto computacional de numerosas reavaliações da função objetivo pelo procedimento de Busca Local foi desenvolvido um Algoritmo de Atualização Dinâmica de Fluxo para DEFT. Um decremento ou incremento no peso de um arco altera o grafo de distâncias e, conseqüentemente, a distribuição do fluxo. Mesmo mudanças que não afetam o grafo de caminhos mínimos podem representar alterações na distribuição de fluxo.

São três os casos a serem analisados. Seja um nodo destino t e um aumento Δ unitário no peso de um link (u, v) , denotado por $\Delta(u, v)$. Primeiramente, têm-se os nodos que não possuem demanda e não fazem parte da rota de nenhuma carga direcionada à t , logo, não são afetados por Δ .

O segundo caso, diz respeito aos nodos que alcançam t unicamente pelo link (u, v) . Neste caso, qualquer variação Δ afetará igualmente a todo este conjunto de nodos preservando a distribuição de fluxo entre eles.

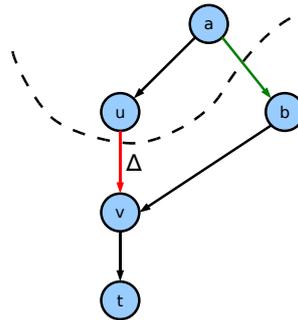


Figura 3: Atualização dinâmica de fluxo.

No terceiro caso, têm-se os nodos, com dois ou mais links saíntes, em que ao menos um link tenha sua distância alterada por $\Delta(u, v)$ e ao menos um link tenha se mantido com a distância antiga. Este caso, caracteriza os nodos com caminhos alternativos para t que não passam pelo link (u, v) . A distribuição do fluxo nestes nodos deve ser reavaliada e em todos os nodos intermediários até o destino t , afetados pela propagação destas mudanças.

A Figura 3 apresenta um grafo com indicação do aumento Δ no arco (u, v) . Supondo inicialmente a não existência do arco (a, b) , as diferenças nas distâncias entre os caminhos possíveis permanecem as mesmas para os nodos a e u pois suas distâncias são igualmente afetadas por Δ . Portanto, a distribuição de fluxo permanece inalterada. Entretanto, na presença do arco (a, b) , uma parcela de fluxo será redirecionada por b , diminuindo o fluxo pelos arcos (a, u) e (u, v) .

A Figura 4 apresenta, em pseudocódigo, o procedimento para avaliação de uma solução utilizando o fluxo dinâmico. Na linha 1, preservam-se as distâncias correntes. Para cada nodo destino $t \in T$ calcula-se o grafo dos caminhos mínimos reversos G^t em 3–4. No laço das linhas 6–13, localizam-se todos os nodos que possuem dois ou mais links saíntes, com ao menos um em que a distância não tenha aumentado de Δ e pelo menos outro que tenha. Estes nodos, com caminhos afetados pela variação de

```

procedimento FlowDistributionDEFT( $d = d_1, d_2, \dots, d_{|V|}, \Delta, w = w_1, w_2, \dots, w_{|V|}$ )
1   $d_{ant} = d$ 
2  para  $\forall t \in T$ 
3       $d = \text{DijkstraReverso}(t, w)$ 
4       $G^t = \text{ComputarGrafoCaminhosMínimos}(w, d)$ 
5       $H_{d, d_{ant}} = \text{Nodos ordenados em ordem decrescente primeiro da}$ 
        distância atual e segundo da distância antiga
6      para cada  $u \in V$ 
7          condição1 = condição2 = falso
8          para cada  $v \in \text{OUT}(u)$ 
9              se  $d_v^t = d_{ant_v}^t + \Delta$  então condição1 = verdade
10             se  $d_v^t = d_{ant_v}^t$  então condição2 = verdade
11             se condição1 && condição2 então  $\text{InsertHeap}(H_{d, d_{ant}}, u)$ 
12         fimpara
13     fimpara
14     para cada  $p \in H_{d, d_{ant}}$ 
15         para cada  $q \in \text{OUT}(p)$ 
16             se  $! \text{IsHeapMember}(H_{d, d_{ant}}, q)$  então  $\text{InsertHeap}(H_{d, d_{ant}}, q)$ 
17         fimpara
18     fimpara
19     para cada  $u \in H_{d, d_{ant}}$ 
20          $\Gamma_{total} = 0$ 
21         para cada  $v \in \text{OUT}(u)$ 
22             se  $d_u^t > d_v^t$  então
23                  $h_{u,v}^t = d_v^t + w_{u,v} - d_u^t$ 
24                  $\Gamma_{total} += e^{-h_{u,v}^t}$ 
25             fimse
26         fimpara
27          $f = (D_{u,v} + \sum_{(u,v) \in G^t} f_{(u,v)}^t) / \Gamma_{total}$ 
28         para cada  $a \in \text{OUT}(u)$ 
29             se  $d_u^t > d_v^t$  então
30                  $h_{u,v}^t = d_v^t + w_{u,v} - d_u^t$ 
31                  $\gamma = e^{-h_{u,v}^t}$ 
32                  $f_{u,v}^t = f * \gamma$ 
33             senão
34                  $f_{u,v}^t = 0$ 
35             fimse
36         fimpara
37     fimpara
38     para cada  $(u, v) \in A$   $f_{u,v} += f_{u,v}^t$ 
39 fimpara
40  $\Phi = \sum_{(u,v) \in E} \phi(u, v)$ 
fim FlowDistributionDEFT.

```

Figura 4: Pseudocódigo descrevendo o cálculo dinâmico do fluxo.

Δ e com caminhos alternativos, são os que apresentam redistribuição do fluxo na comparação com as distâncias atuais e anteriores. Os nodos selecionados são armazenados em uma estrutura de dados Heap com chave dupla, quais sejam, as distâncias após e antes do cálculo do grafo de caminhos mínimos e denotado por $H_{d, d_{ant}}$. Esta estrutura mantém os nodos ordenados segundo a regra $(d_1, d_2) < (d'_1, d'_2)$ se $d_1 < d'_1$, e $d_2 < d'_2$ se $d_1 = d'_1$. O laço das linhas 14–18, armazena na mesma estrutura todos os nodos intermediários entre o grupo originalmente selecionado e o nodo destino. A distribuição do fluxo ocorre no laço das linhas 19–37, sendo calculado o fluxo passante na linha 27. No laço das linhas 28–36, calcula-se a porção de tráfego para cada link sainte (linhas 30–32), eliminando o fluxo nos links que não se aproximam do destino (linha 34). A soma total do fluxo em cada arco, considerando as parciais de cada destino, é calculada na linha 38. Finalmente, na linha 40, calcula-se a avaliação da

solução.

6.1 Comparação com a distribuição de fluxo OSPF

O fluxo no protocolo DEFT é distribuído baseado em penalidades exponenciais. Isto permite utilizar melhor a capacidade da rede e leva a uma distribuição do fluxo de dados próxima da otimalidade, segundo Xu et al. (2007). Quanto maior o comprimento de um caminho em relação ao caminho mínimo, menor a porção de fluxo roteada por este caminho. No OSPF, todo tráfego é roteado somente por caminhos mínimos. No caso de dois ou mais arcos saintes pertencentes ao grafo de caminhos mínimos, o fluxo é dividido igualmente entre eles. Na Figura 5, é apresentado um grafo em que o arco (b, t) tem seu peso alterado de P_2 (grafo da esquerda) para P_2' (grafo da direita). Os pesos nos arcos (u, a) e (u, b) são os mesmos e, portanto, não afetam a distribuição. Sendo $P_1 > P_2' > P_2$, no OSPF, todo o fluxo do nodo u para o nodo t é roteado por b , por ser o caminho mínimo. Os arcos (u, a) e (a, t) não recebem nenhum fluxo e a mudança de peso em (b, t) não afeta esta situação. Entretanto, no DEFT, além destes arcos receberem uma parcela de carga, quando o peso em (b, t) é aumentado, ocorre a diminuição do fluxo pelo caminho passando pelo nodo b e o aumento do fluxo passante por a . Evitando, assim, a subutilização da rede.

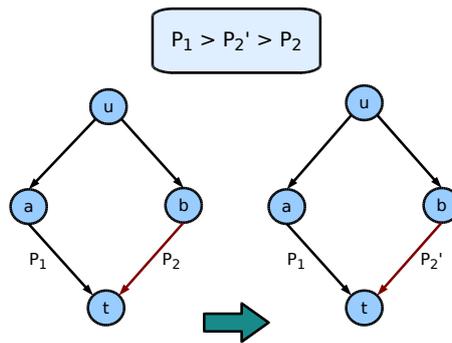


Figura 5: Exemplo de alteração na distribuição de fluxo DEFT sem alteração do caminho mínimo.

7 Resultados

O desempenho do algoritmo memético proposto foi testado em treze redes diferentes, cada qual com sete escalas de demandas. Os resultados representam a média de cinco rodadas para cada instância, usando os seguintes parâmetros:

- Tamanho da população: 50 indivíduos.
- Intervalo de pesos: $[1,20]$.
- Tempo de execução: 60 minutos.
- Probabilidade de herdar um gene do pai de elite: 0.7.
- Limite para $h_{u,v}^t$: 9

Foram testados valores diferentes para alguns parâmetros como o tamanho da população, a porcentagem de seleção do crossover e o tamanho das classes da população sem uma variação expressiva nos resultados. Optou-se, por adotar os mesmos valores usados no algoritmo original para OSPF.

As instâncias usadas são as mesmas usadas em Fortz and Thorup (2004) e Reis et al. (2007) para manter nossos resultados comparáveis. A instância *att* representa dados reais da AT&T, enquanto as demais são sintéticas. Todos os experimentos foram conduzidos em um cluster com 10 computadores de núcleo duplo de 1.23 GHz com 1.0 GB RAM e sistema operacional Linux. A Tabela 1 resume suas características com quantidade de nodos, quantidade e capacidades dos links e número de pares origem-destino.

Tabela 1: Instâncias usadas nos experimentos computacionais.

| Instância | Nome | Nodos | Links | Capacidades | Pares o-d |
|-------------------------|----------|-------|-------|-------------|-----------|
| AT&T backbone | att | 90 | 274 | - | 272 |
| Hierárquico de 2 níveis | hier50a | 50 | 148 | 200 e 1000 | 2450 |
| | hier50b | 50 | 212 | 200 e 1000 | 2450 |
| | hier100 | 100 | 279 | 200 e 1000 | 9900 |
| | hier100a | 100 | 360 | 200 e 1000 | 9900 |
| Topologia randômica | rand50 | 50 | 228 | todos 1000 | 2450 |
| | rand50a | 50 | 245 | todos 1000 | 2450 |
| | rand100 | 100 | 403 | todos 1000 | 9900 |
| | rand100b | 100 | 503 | todos 1000 | 9900 |
| Waxman | wax50 | 50 | 169 | todos 1000 | 2450 |
| | wax50a | 50 | 230 | todos 1000 | 2450 |
| | wax100 | 100 | 391 | todos 1000 | 9900 |
| | wax100a | 100 | 476 | todos 1000 | 9900 |

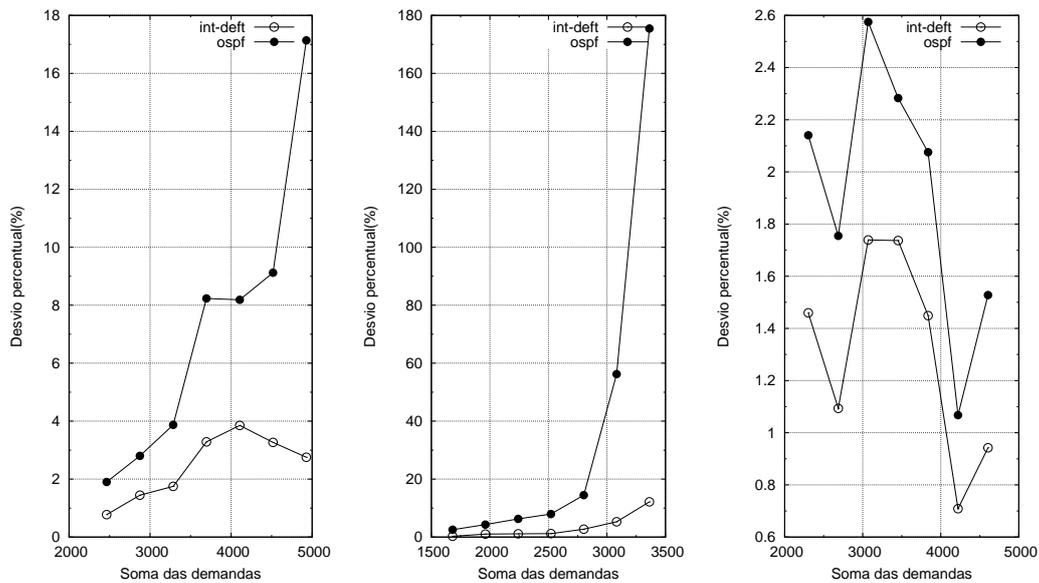


Figura 6: Desvio percentual da melhor solução encontrada pelo MA sobre a solução ótima para as instâncias hier50a, hier50b e hier100.

As Figuras de 6 a 10 apresentam o desvio percentual tanto do DEFT como do OSPF para o roteamento OPT em todas as instâncias testadas. Nas Figuras 6 a 8 observam-se instâncias onde o desempenho do DEFT é substancialmente melhor e outras cujo desempenho permanece nos mesmos níveis do OSPF. Com isto observa-se que para algumas instâncias a alternativa do DEFT de roteamento por caminhos não mínimos permite uma melhor minimização do congestionamento da rede.

Contudo na Figuras 9 e 10 verificam-se resultados onde às vezes o desempenho do DEFT foi ligeiramente pior do que o do OSPF, para algumas demandas. Estes resultados são atribuídos às instâncias onde os pesos inteiros e os intervalos testados não puderam refletir a granularidade que os pesos re-

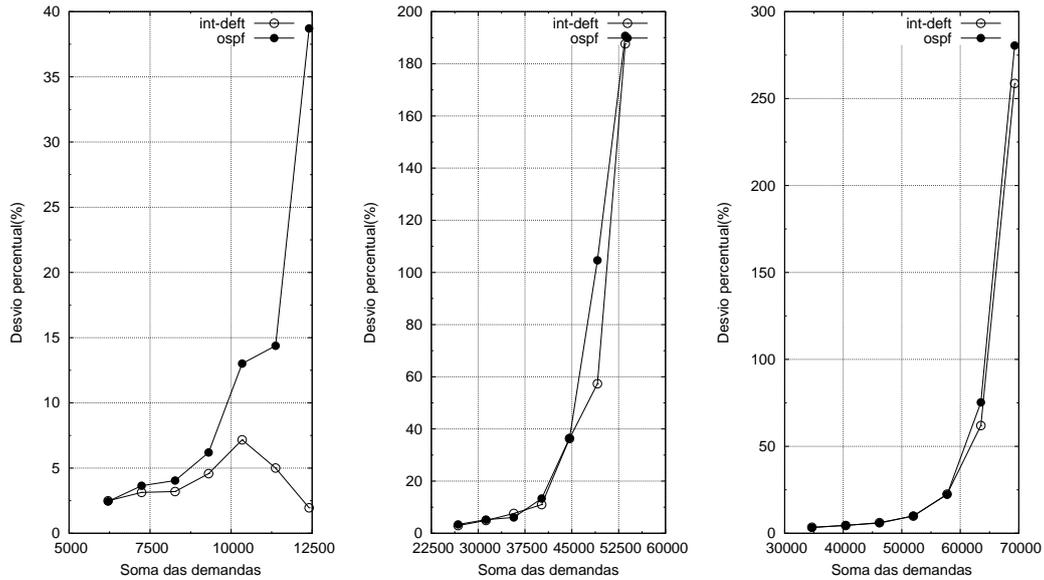


Figura 7: Desvio percentual da melhor solução encontrada pelo MA sobre a solução ótima para as instâncias hier100a, rand50a e rand100.

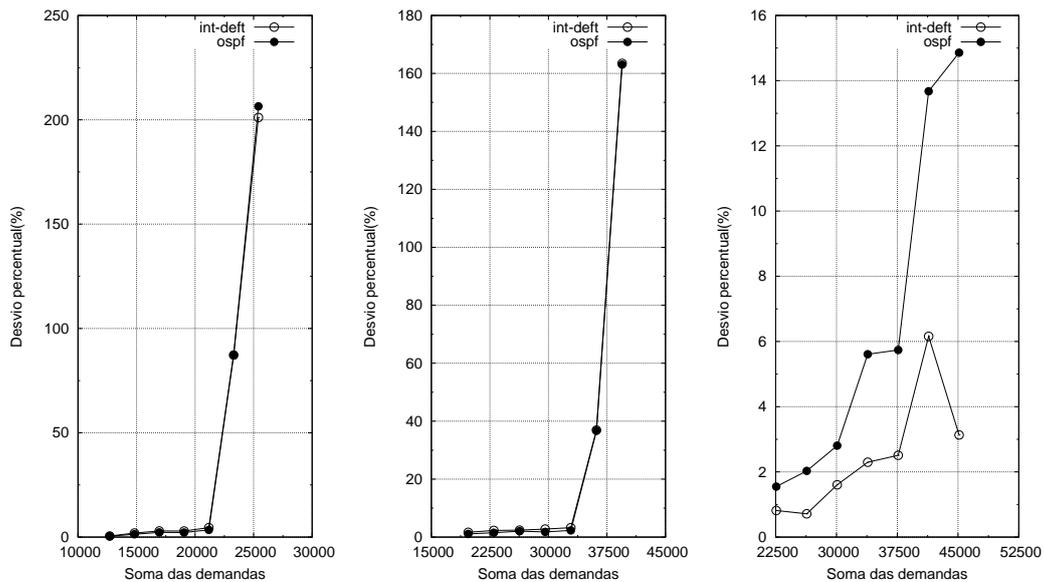


Figura 8: Desvio percentual da melhor solução encontrada pelo MA sobre a solução ótima para as instâncias wax50, wax50a e att.

ais permitem alcançar. Em uma distribuição de fluxo utilizando pesos reais os incrementos de peso nos arcos mais congestionados são calculados baseados em um mínimo local em uma busca linear. Utilizando pesos inteiros têm-se um espaço de busca discreto.

A Figura 11 mostra a comparação dos tempos, em horas, de três implementações. A primeira versão em Reis et al. (2007) foi adaptada diretamente da implementação do algoritmo memético para OSPF. A segunda, é uma otimização da primeira que elimina estruturas de dados responsáveis por manter referências aos caminhos mínimos. A terceira agrega o algoritmo de fluxo dinâmico descrito na Seção 6.

O algoritmo apresentou uma melhora entre 63% e 73% sobre a versão inicial. O cálculo dinâmico do fluxo contribuiu com os maiores ganhos. Quanto mais próximo do fluxo multi-produto maior é a

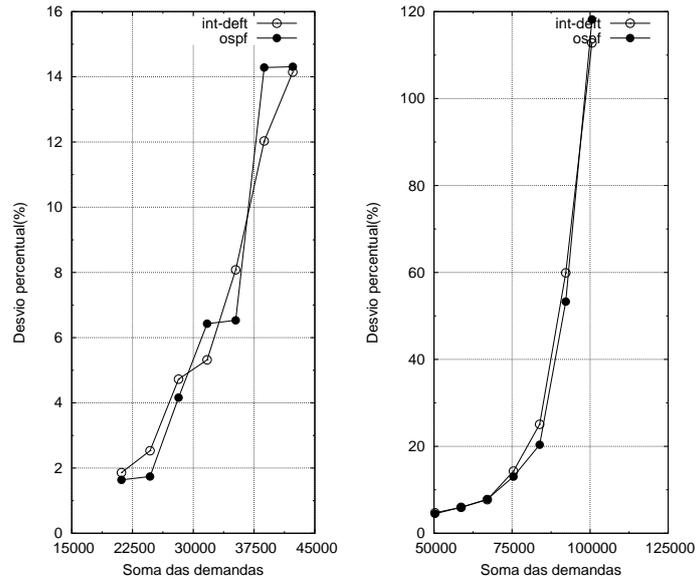


Figura 9: Desvio percentual da melhor solução encontrada pelo MA sobre a solução ótima para as instâncias rand50 e rand100b.

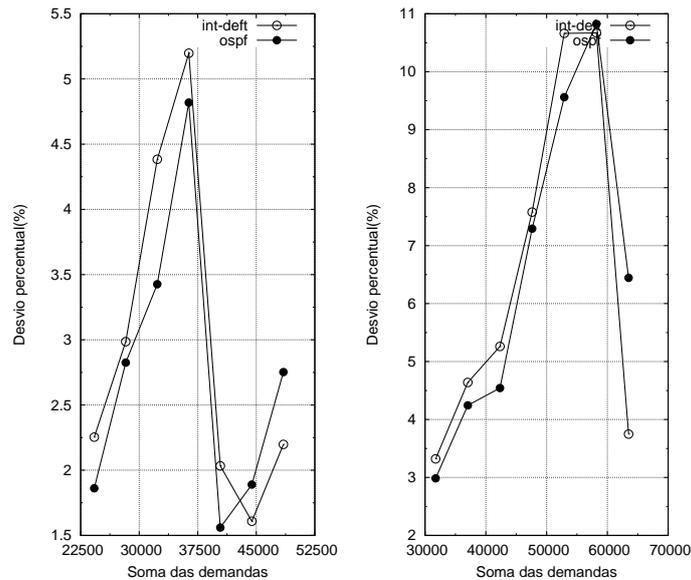


Figura 10: Desvio percentual da melhor solução encontrada pelo MA sobre a solução ótima para as instâncias wax100 e wax100a.

região do grafo afetada por mudanças nos pesos. No roteamento OPT qualquer mudança representa uma reestruturação da distribuição dos fluxos. A dinâmica do protocolo DEFT o deixa mais sensível às mudanças nos pesos do que o OSPF, logo, a importância da atualização dinâmica do fluxo. Somente este algoritmo foi responsável por uma melhora de 51% a 68%, dependendo da instância.

8 Conclusões e trabalhos futuros

Baseado nos resultados apresentados na seção anterior, pode-se concluir que a MA obtém melhores resultados para o DEFT do que para o OSPF. No entanto, a mudança de um peso no roteamento DEFT altera uma porção maior do grafo do que no roteamento OSPF, o que pode não ser desejado. Ainda, a

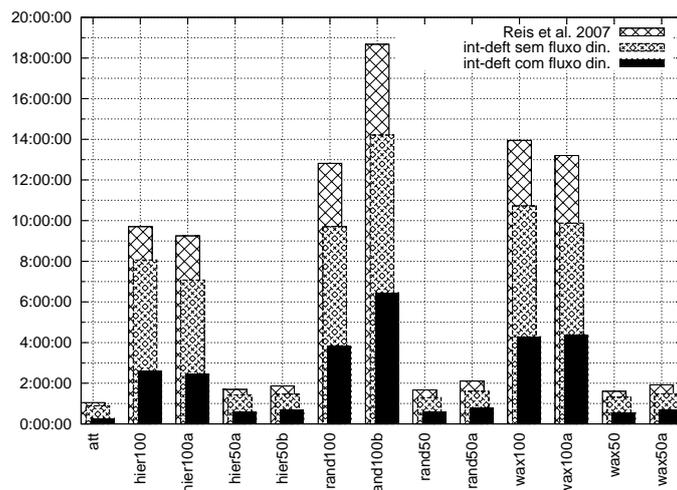


Figura 11: Desempenho de três implementações do roteamento DEFT.

versão atual é cerca de 70% mais rápida do que a inicial em Reis et al. (2007).

Note, que estes resultados são para pesos inteiros. Pretendemos estudar o roteamento DEFT utilizando pesos reais. Neste caso, será utilizada uma nova busca local baseada em *line search* (busca linear). Atualmente, os incrementos usados na busca local são unitários. Com o mecanismo de *line search* podemos variar este incremento e espera-se alcançar resultados ainda melhores.

Outra linha de pesquisa interessante, onde métodos de busca local podem ser vantajosos, é o projeto de redes resistentes a falhas, i.e., redes projetadas para manter o congestionamento em níveis aceitáveis mesmo na presença de falhas de links ou de roteadores. Neste contexto, o uso de otimização por *solvers* não pode ser usada diretamente, pois trata-se de um problema eminentemente combinatório.

Referências

- Buriol, L., Resende, M., Ribeiro, C. and Thorup, M. (2005). A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing, *Networks* **46**(1): 36–56.
- Ericsson, M., Resende, M. and Pardalos, P. (2002). A genetic algorithm for the weight setting problem in OSPF routing, *J. of Combinatorial Optimization* **6**: 299–333.
- Fong, J. H., Gilbert, A. C., Kannan, S. and Strauss, M. J. (2005). Better alternatives to OSPF routing., *Algorithmica* **43**(1-2): 113–131.
- Fortz, B. and Thorup, M. (2000). Internet traffic engineering by optimizing OSPF weights, *INFOCOM 2000*, pp. 519–528.
- Fortz, B. and Thorup, M. (2004). Increasing internet capacity using local search, *Computational Optimization and Applications* **29**(1): 13–48.
- Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, *Technical Report C3P Report 826*, Caltech Concurrent Computation Program.
- Reis, R., Buriol, L., Ritt, M. and Resende, M. (2007). A memetic algorithm for the weight setting problem in DEFT, *Congreso Internacional de Computación Evolutiva, Aguascalientes, México*, pp. 135–140.
- Xu, D., Chiang, M. and Rexford, J. (2007). DEFT: Distributed exponentially-weighted flow splitting, *Proc. 26th IEEE Conf. on Computer Communicatios (INFOCOM)*, pp. 71–79.