

# Algoritmos para Solução do Problema de Atribuição de Capacidades Discretas em Redes TCP/IP

Eduardo Yabcznski, Emilio C. G. Wille e Heitor Silvério Lopes

**Resumo**—Este artigo trata do dimensionamento de redes IP sujeitas ao tráfego TCP. O problema considerado corresponde à Atribuição de Capacidades (*Capacity Assignment*) discretas onde os valores de capacidades para cada enlace são escolhidos de um conjunto discretos de valores. Para a solução deste problema são propostas três técnicas: Busca Exaustiva (ES - *Exhaustive Search*), Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*) e uma heurística baseada em restrições.

**Palavras-Chave**— Dimensionamento de redes IP, Atribuição de Capacidades discretas, Busca Exaustiva, Otimização por Enxame de Partículas.

**Abstract**—This article is concerned with the dimensioning of IP networks when submitted to TCP traffic. The problem taken into account is the Discrete Capacity Assignment where capacity values for each network link are chosen based on a set of discrete values. Three techniques are proposed to deal with it: Exhaustive Search (ES), Particle Swarm Optimization (PSO) and a restriction based heuristic.

**Keywords**—Dimensioning of IP networks, Discrete Capacity Assignment, Exhaustive Search, Particle Swarm Optimization.

## I. INTRODUÇÃO

Diversos modelos matemáticos para projeto e planejamento, de forma otimizada, redes de comutação de pacotes, foram extensivamente investigados no passado e tiveram início com o trabalho clássico de Kleinrock [1],[2]. Contudo esses modelos consideravam somente a infra-estrutura da camada de rede, ignorando fatores como a qualidade de serviço (QoS - *Quality-of-Service*) para usuários finais e o acordo de nível de serviço (SLA - *Service Level Agreement*). Hoje em dia estes modelos tornaram-se inapropriados, pois pela rede de telecomunicações circula uma variedade de serviços críticos que necessitam que a rede garanta certos níveis de qualidade aos usuários finais. Portanto é necessário criar novos modelos e metodologias que permitam um planejamento adequado dos recursos da rede para a entrega dos inúmeros serviços com a qualidade de serviço desejada e ao mesmo tempo a baixo custo.

Em [3] o autor propõem uma metodologia de projeto de redes IP que considera a dinâmica das redes de pacotes, assim como os efeitos dos protocolos nas diferentes camadas e na QoS experimentada pelos usuários finais. Primeiramente, os índices de desempenho dos usuários finais são mapeadas em índices de desempenho da camada de transporte e dessa, mediante outro mapeamento, para índices de desempenho da camada de rede. Este processo de mapeamento é então usado

em conjunto com um modelo de tráfego TCP/IP (proposto em [4], que produz uma boa estimativa do desempenho da rede sujeita à padrões de tráfego real) para o dimensionamento de uma rede IP.

O processo de dimensionamento pode ser visto como um processo de otimização, onde se busca minimizar uma função custo e ao mesmo tempo satisfazer uma série de restrições associadas ao desempenho da rede.

Porém, em [3], no processo de otimização utilizado, são considerados valores contínuos de capacidades. Geralmente este não é o caso em sistemas realísticos. No presente trabalho, será apresentado o problema de Atribuição de Capacidades (CA - *Capacity Assignment*) discretas, sujeito a restrições de QoS fim-a-fim, onde os valores de capacidades de tráfego para cada enlace, são escolhidos de um conjunto discreto de valores, serão consideradas como restrições os atrasos médios nas transmissões fim-a-fim. Como propostas de solução serão utilizadas a Busca Exaustiva (ES - *Exhaustive Search*), a Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*) e uma nova heurística, chamada de Ajuste do Enlace Crítico (AEC), baseada em restrições.

## II. MODELOS DE TRÁFEGO

Análises recentes do tráfego na Internet mostram que o processo de chegada dos pacotes IP nas interfaces dos roteadores não é um processo de Poisson [5], pois há correlação entre pacotes que chegam em instantes diferentes, exibindo características LRD (*Long Range Dependent*) [6]. Este comportamento pode ser atribuído aos mecanismos de controle do protocolo TCP e faz com que os modelos tradicionais ( $M/M/1$  e  $M/M/1/B$ ) não modelem corretamente a rede. Em [7], os autores pela primeira vez utilizam um modelo de tráfego LRD, mais especificamente, um modelo baseado em Movimento Browniano Fracional (*Fractional Brownian Motion*). Infelizmente, a expansão deste modelo para considerar problemas gerais de análise e projeto, é difícil, pois não é possível obter uma função matemática fechada que expresse as relações entre tráfego, capacidades e atrasos de filas. Para contornar este problema, neste trabalho utiliza-se o modelo de filas markoviano  $M_{[X]}/M/1$  com chegada de pacotes em grupos.

Este modelo foi extensivamente analisado, mostrando-se capaz de prever os indicadores de desempenho de filas alimentadas por tráfego TCP com boa precisão [4].

### A. Modelo de Fila $M_{[X]}/M/1$

Como o modelo clássico de filas  $M/M/1$  não é capaz modelar a correlação existente entre pacotes que chegam nas

interfaces dos roteadores em instantes diferentes, é adotado o modelo de filas  $M_{[X]}/M/1$ , que melhor representa o tráfego em rajadas produzido pelo protocolo TCP. Este modelo representa uma fila Markoviana com *chegadas em grupo*. O tamanho do grupo varia entre 1 e  $W$  com distribuição  $[X]$ , onde  $W$  é o tamanho máximo (em segmentos) da janela TCP. A distribuição  $[X]$  é obtida considerando o número de segmentos que a fonte TCP envia em um  $RTT$ . Dada a distribuição de comprimento dos fluxos presentes na rede, é possível calcular a distribuição de tamanho de pacotes  $[X]$ , através do modelo estocástico do TCP presente em [4]. Considerando os comprimentos dos pacotes exponencialmente distribuídos com média  $1/\mu$  (bits/pacote) [1],[2], taxa de chegada  $\lambda = \mu \cdot f$  (pacotes/segundo), e o fator de utilização do enlace  $\rho = f/C$ , o atraso médio dos pacotes dado pelo modelo de fila  $M_{[X]}/M/1$ , pode ser escrito como (note que os índices  $(i, j)$  foram ignorados para simplificar) [8]:

$$\bar{T} = \frac{K}{\mu} \frac{1}{C - f} \quad \text{com} \quad K = \frac{m'_{[X]} + m''_{[X]}}{2m'_{[X]}} \quad (1)$$

onde  $m'_{[X]}$  e  $m''_{[X]}$  correspondem ao primeiro e segundo momentos da distribuição  $[X]$ .

### III. O PROBLEMA DE ATRIBUIÇÃO DE CAPACIDADES

Neste artigo o problema CA é resolvido considerando como restrições os atrasos (médios) nas transmissões fim-a-fim. Selecionando i) a função custo, ii) o modelo de roteamento, e iii) as restrições de capacidades, é possível obter diferentes formulações para o problema CA. Serão considerados neste trabalho: i) custo linear, ii) roteamento não bifurcado, e iii) capacidades discretas. Seja a função custo  $g(d_{ij}, C_{ij}) = v_{ij} \cdot d_{ij} \cdot C_{ij}$ , onde  $v_{ij}$  é o custo monetário do enlace  $(i, j)$  em \$/Mbps/km/ano<sup>1</sup>. Seja  $\delta_{ij}^{sd}$  uma função indicadora com o valor 1 se o enlace  $(i, j)$  fizer parte do caminho  $(s, d)$  e 0 caso contrário, e  $S$  o conjunto de capacidades discretas com cardinalidade  $N_s$ . Assim, dados: a topologia da rede, o tráfego imposto e o roteamento (fixo), o problema CA discreto é formulado como sendo o seguinte problema de otimização:

$$Z_{CA} = \min \sum_{i,j} g(d_{ij}, C_{ij}) \quad (2)$$

sujeito à:

$$\bar{T}_{sd} = \frac{K}{\mu} \sum_{i,j} \frac{\delta_{ij}^{sd}}{C_{ij} - f_{ij}} \leq \text{Atraso}_{sd} \quad \forall (s, d) \quad (3)$$

$$f_{ij} = \sum_{s,d} \delta_{ij}^{sd} \gamma_{sd} \quad \forall (i, j) \quad (4)$$

$$C_{ij} > f_{ij} > 0 \quad \forall (i, j) \quad (5)$$

$$C_{ij} \in S \quad \forall (i, j) \quad (6)$$

onde  $\text{Atraso}_{sd}$  é o atraso permitido na transmissão fim-a-fim, para o seu cálculo toma-se o  $RTT$  (dividido por dois) subtraído do retardo de propagação do caminho [9]. A função

objetivo (2) representa o custo total da rede, que é a soma das funções custo de cada enlace  $(i, j)$ . A equação (3) é a restrição de atraso fim-a-fim para cada par origem-destino. Isto quer dizer que a soma total do atraso  $\bar{T}_{sd}$  experimentado pelos fluxos roteados pelo caminho  $(s, d)$ , não excederá o  $\text{Atraso}_{sd}$ . A equação (4) define o fluxo médio de dados em um enlace. As restrições (5) são restrições de não-negatividade.

## IV. PROPOSTAS DE SOLUÇÃO

O problema CA discreto, como apresentado na seção anterior, é um problema de otimização combinatorial, e o ótimo global pode ser encontrado utilizando o método de busca exaustiva (ES). Contudo este algoritmo consome muito tempo computacional. Uma solução sub-ótima, e aceitável, pode ser obtida com menor tempo de processamento, utilizando o método de otimização por enxame de partículas (PSO). Também é apresentada uma heurística simples (AEC) para encontrar soluções admissíveis para o problema CA.

### A. Busca Exaustiva

O algoritmo de busca exaustiva testa todas as combinações possíveis para a solução do problema e armazena aquela que produzir o melhor resultado. No problema de otimização do custo da rede, são testadas todas as combinações possíveis de capacidades para todos os enlaces e armazenada aquela que produzir o menor custo satisfazendo as restrições. Tendo um conjunto  $S$  de capacidades discretas para escolher para cada enlace e sendo  $L$  o número de enlaces, então, para cada enlace  $(i, j) \in L$  são testadas todas as capacidades presentes no conjunto  $S$ . A cada teste, as restrições são verificadas. Tanto no caso de satisfazer ou não as restrições, uma nova combinação de capacidades nos enlaces é testada até que todas as combinações possíveis sejam testadas. Nos casos em que as restrições são satisfeitas e o custo apresentado pela combinação das capacidades nos enlaces for o menor custo até o momento, seu valor é armazenado.

### B. Otimização por Enxame de Partículas

Introduzido em 1995 por Kennedy e Eberhart [10], a otimização por enxame de partículas é uma técnica heurística de computação evolucionária, baseada em população e inspirada em propriedades que surgem do comportamento coletivo de alguns animais: bando de pássaros, enxame de abelhas e cardume de peixes. Por exemplo, em um bando de pássaros, a velocidade de cada elemento é dinamicamente ajustada de acordo com a velocidade dos elementos ao redor. A influência de um pássaro sobre outro, permite que seja mantida uma distância média entre eles. Fazendo uma analogia com o PSO: os pássaros, voando em um espaço tri-dimensional em busca de alimentos, correspondem às partículas percorrendo um espaço  $n$ -dimensional a procura de uma solução para o problema. A cada instante  $(t)$ , a posição e a velocidade das partículas são ajustadas de acordo com a posição e a velocidade no instante  $(t - 1)$ :

$$V_i(t) = V_i(t - 1) + \varphi_1 \cdot r_1 \cdot [X_{blp} - X_i(t - 1)] + \varphi_2 \cdot r_2 \cdot [X_{bgp} - X_i(t - 1)] \quad (7)$$

<sup>1</sup>Neste trabalho será utilizado  $v_{ij} = 1.00$  por simplicidade.

$$X_i(t) = X_i(t-1) + V_i(t) \quad (8)$$

onde  $\varphi_1$  e  $\varphi_2$  são constantes positivas,  $r_1$  e  $r_2$  são variáveis aleatórias normalmente distribuídas entre 0 e 1;  $X_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{id}(t))$  representa a posição atual da  $i$ -ésima partícula;  $d$  é a dimensão da partícula,  $X_{blp}^2$  representa a melhor posição da partícula (dada pelo seu melhor *fitness*);  $X_{bgp}^3$  representa a melhor posição de todo o enxame até o momento;  $V_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{id}(t))$  é a velocidade da  $i$ -ésima partícula. De fato, este último termo não tem a dimensão de velocidade, mas a terminologia usada aqui é a mesma proposta originalmente [10].

A equação (8) descreve como a posição das partículas é dinamicamente ajustada, considerando o deslocamento no espaço  $n$ -dimensional dado pela equação (7). Observe que a equação (7) é formada por três termos. O primeiro termo, chamado de *momentum*, impede que a velocidade mude abruptamente. O segundo termo é a parte *cognitiva*, que permite a partícula aprender com a sua própria experiência de vôo, memorizando a melhor posição encontrada até o momento. O terceiro termo é a parte *social*, que representa a colaboração entre as partículas, e permite que a partícula aprenda com a experiência de vôo do enxame. O ajuste entre estes três termos determina a convergência global ou local.

Quando a soma das três componentes da equação (7) excede o valor  $\pm V_{max}$ , especificado pelo usuário, a velocidade é grampeada para este valor. Isto é necessário para evitar que a velocidade exploda, crescendo cada vez mais a cada iteração. Valores altos de  $V_{max}$  fazem com que as partículas fiquem potencialmente passando sobre as soluções boas, sem nunca atingi-las. Por outro lado, valores baixos de  $V_{max}$  podem levar a uma convergência local (mínimo ou máximo local). Usualmente  $V_{max}$  é fixo, mas a variação dinâmica também pode ser empregada.

Para avaliar o quão bom é um resultado encontrado por uma partícula é calculado o seu *fitness*. O *fitness* geralmente é uma função baseada na função que se quer maximizar ou minimizar.

O procedimento para implementação do PSO original é mostrado a seguir [11]:

- 1) Inicialize a população de partículas com posições e velocidades distribuídas aleatoriamente no espaço  $n$ -dimensional de busca.
- 2) Calcule o *fitness* de cada partícula, composto de  $n$  variáveis.
- 3) Compare o *fitness* de cada partícula, com o seu próprio  $X_{blp}$ . Se o valor corrente é melhor do que o  $X_{blp}$ , atribua a  $X_{blp}$  o valor corrente  $X_i(t)$ .
- 4) Atribua a  $X_{bgp}$  a partícula que representa o melhor *fitness* encontrado dentre todas as partículas. Em algumas implementações, topologias de diferentes vizinhanças são usadas para encontrar seus  $X_{bgp}$ .
- 5) Mude a velocidade e a posição da partícula, de acordo com as equações (7) e (8) e se necessário grampeie para  $\pm V_{max}$  o valor da velocidade antes de calcular a posição.

- 6) Volte ao passo 2 até que o critério de parada seja satisfeito. Normalmente um *fitness* suficientemente bom ou um número máximo de iterações.

Existem muitas variações do algoritmo descrito. Em geral são usadas várias estratégias para manter a diversidade evitando a convergência rápida (geralmente para ótimos locais). Explosões periódicas do enxame podem ser utilizadas ao ser observada uma aglutinação das partículas, aleatorizando novamente suas posições, mas mantendo  $X_{bgp}$  [12].

Outro incremento muito importante no PSO original foi o uso do *peso de inércia*  $w$ , que multiplica o termo de velocidade na equação (7). Valores altos de  $w$  facilitam a busca global, enquanto valores baixos facilitam a busca local. Geralmente  $w$  começa com valores altos e é decrementado durante a execução do algoritmo [13].

### C. Aplicando PSO no Planejamento de Redes

Como dito anteriormente, a solução do problema CA corresponde a selecionar uma capacidade para cada enlace (dentre um conjunto pré-determinado), sempre respeitando as restrições de atraso.

Um ponto importante quando se trabalha com processos de otimização com restrições, é de como tratar essas restrições. Para trabalhar com as restrições, são propostas duas modificações no PSO original.

- 1) Durante a inicialização, todas as partículas são repetitivamente inicializadas até satisfazerem todas as restrições.
- 2) Após o cálculo e arredondamento de  $X_i(t)$  (equação (8)), se a capacidade  $C_{ij}$  for menor que o respectivo fluxo  $f_{ij}$ , o próximo valor de  $S$  é escolhido até que a restrição (5) seja satisfeita.

O algoritmo PSO proposto penaliza a partícula (se as restrições não forem satisfeitas) usando a seguinte função *fitness*:

$$fitness = \begin{cases} \sum_{(i,j)} g(d_{ij}, C_{ij}), & \text{se } \bar{T}_{sd} \leq Atraso_{sd}, \quad \forall (s, d) \\ Q \cdot \sum_{(i,j)} g(d_{ij}, C_{ij}), & \text{de outro modo} \end{cases} \quad (9)$$

Como o que se quer é um *fitness* baixo e que respeite as restrições, a função *fitness* penaliza as partículas que não respeitarem as restrições, multiplicando o *fitness* dessas pelo fator de penalização  $Q$ . O procedimento a seguir é utilizado na aplicação do PSO para o problema CA discreto: após o cálculo das equações (7) e (8), cada variável  $x_{in}$  é arredondada para um inteiro maior ou igual a ela. Então este valor é mapeado para um elemento do conjunto  $S$  de capacidades.

### D. Heurística

Aqui será mostrada uma heurística simples que permite encontrar soluções admissíveis para o CA discreto. O objetivo desta heurística não é o de minimizar o custo, e sim o de satisfazer as restrições, procurando sempre usar os menores valores de capacidade. Esta heurística é batizada como Ajuste do Enlace Crítico (AEC).

<sup>2</sup>blp - best local position.

<sup>3</sup>bgp - best group position.

Heurística:

- 1) Atribua a cada enlace uma capacidade maior que o fluxo de dados que trafega pelo enlace. Esta capacidade é escolhida como sendo a menor capacidade do conjunto discreto  $S$  de capacidades (para satisfazer a equação (5)).
- 2) Verifique as restrições de atraso (equação (3)):
  - a) Caso sejam satisfeitas, o algoritmo termina
  - b) Caso alguma restrição não seja satisfeita, o algoritmo avança.
- 3) Determine o enlace de maior atraso (enlace crítico) presente no cálculo da restrição de atraso (não respeitado) e atribua um valor maior, escolhido do conjunto discreto  $S$  de capacidades, para a capacidade desse enlace.
- 4) Retorne ao item 2.

Caso a capacidade do enlace no item 3 já seja a maior capacidade do conjunto  $S$ , escolhe-se o segundo enlace crítico presente no cálculo da restrição, para ter a capacidade aumentada. Caso a capacidade do enlace já seja a maior capacidade do conjunto  $S$ , escolhe-se o terceiro enlace crítico, e assim por diante.

## V. RESULTADOS NUMÉRICOS E DISCUSSÃO

Para testar os métodos propostos para a solução dos problemas CA, três topologias de redes com tamanhos diferentes foram utilizadas (tabela I). Todas as três topologias foram geradas aleatoriamente. Para as três topologias, a matriz de tráfego foi calculada pela geração de valores aleatórios uniformemente distribuídos para cada par origem-destino.

TABELA I  
TOPOLOGIAS UTILIZADAS NOS TESTES.

Topologia	N° de nós	N° de enlaces	N° de restrições	Área (km <sup>2</sup> )
1	5	12	11	215 x 215
2	11	24	15	20 x 20
3	20	48	25	20 x 20

As mesmas restrições de QoS são impostas para todos os pares origem-destino. São elas: (i) latência inferior a 0.5 s para fluxos TCP com menos de 20 pacotes, e (ii) vazão maior que 512 kbps para fluxos TCP com mais de 20 pacotes. Cada pacote possui o tamanho fixo de 1460 bytes. Utilizou-se  $P_{loss} = 0.01$  e  $RTT < 0.07$  s.

Os algoritmos foram implementados em C++, e os experimentos foram executados em um computador com 512 kBytes de memória RAM e com velocidade de processamento de 1GHz.

O problema CA discreto foi resolvido para cada topologia utilizando o PSO, o algoritmo de busca exaustiva (ES) e a heurística proposta. Também foram encontrados os limitantes inferiores utilizando uma metodologia baseada no algoritmo de Chromy [14].

Após vários testes com o PSO, onde o peso de inércia  $w$  inicia em 0.9 e decrementa até 0.4 e várias combinações entre os valores  $\varphi_1 = \{0.3, 0.6, 0.9, 1.5, 2.0\}$  e  $\varphi_2 = \{0.3, 0.6, 0.9, 1.5, 2.0\}$ , optou-se por utilizar  $w = 0.5 +$

$rand()/2$  e  $\varphi_1 = \varphi_2 = 1.49445$  [13], já que nenhuma melhora foi observada utilizando outros valores. Para o fator de penalização do *fitness* utilizou-se  $Q = 4$ . Como o PSO não é determinístico, 500 experimentos foram executados para cada combinação de número de partículas ( $N_p$ ) e número de iterações ( $N_{itr}$ ). Foram observados o número de sucessos em detectar o mínimo global usando valores inteiros, os melhores valores, os valores médios e o desvio padrão.

Primeiramente foi considerada a rede da topologia 1, onde  $S = (4, 6, 10, 20, 50)$  [Mbps],  $d_{ij} = 150$  [km],  $\forall (i, j)$ . Os fluxos para os enlaces (tráfego #1.a), são  $f = (16, 11, 21, 21, 2, 14, 24, 19, 9, 3, 8, 21)$  [Mbps]. Usando o algoritmo de busca exaustiva foi obtido o seguinte conjunto de capacidades  $C = (20, 20, 50, 50, 4, 20, 50, 50, 20, 6, 10, 50)$ , correspondente ao menor custo (melhor resultado) de 52500/ano, o mesmo resultado foi encontrado pela heurística. O limitante inferior encontrado foi de 30400.02/ano. Um segundo experimento foi realizado usando um outro perfil de tráfego (tráfego #x.b) que corresponde a duas vezes o tráfego #x.a. Usando o algoritmo de busca exaustiva foi obtido o seguinte conjunto de capacidades  $C = (50, 50, 50, 50, 6, 50, 50, 50, 50, 10, 20, 50)$ , correspondente ao menor custo (melhor resultado) de 72900/ano. Novamente a heurística obteve o mesmo resultado da busca exaustiva. O limitante inferior, encontrado utilizando o algoritmo de Chromy, foi de 55750.02/ano.

As Tabelas II e III sintetizam o desempenho do PSO para o primeiro teste.

TABELA II  
PSO PARA TOPOLOGIA 1 E TRÁFEGO #1.A.

Probl.	$N_p$	$N_{itr}$	Taxa de sucesso	Ótimo	Valor médio	Desvio padrão
1	5	50	345/500	52500	53890.2	2218.16
2	10	50	473/500	52500	52734.0	1000.12
3	30	50	500/500	52500	52500.0	0.0
4	5	100	354/500	52500	53731.8	2185.20
5	10	100	478/500	52500	52689.0	903.55
6	30	100	500/500	52500	52500.0	0.0

TABELA III  
PSO PARA TOPOLOGIA 1 E TRÁFEGO #1.B.

Probl.	$N_p$	$N_{itr}$	Taxa de sucesso	Ótimo	Valor médio	Desvio padrão
1	5	50	489/500	72900	72982.2	602.31
2	10	50	498/500	72900	72909.0	201.24
3	30	50	500/500	72900	72900.0	0.0
4	5	100	496/500	72900	72927.0	335.08
5	10	100	500/500	72900	72900.0	0.0
6	30	100	500/500	72900	72900.0	0.0

Note que o espaço de busca para este problema pode chegar a  $N_s^L = 2.4 \times 10^8$  soluções candidatas. O tempo computacional para resolver o problema usando ES foi de uns poucos segundos, enquanto o PSO levou em torno de dez segundos. A diferença entre os resultados das tabelas II e III para um mesmo número de partículas e um mesmo número

de iterações, pode ser justificado pela restrição (5). Neste primeiro teste, dobrando os valores dos fluxos nos enlaces  $(i, j)$  e mantendo o mesmo conjunto de capacidades discretas a escolher  $S$ , reduzimos o espaço de busca, pois  $C_{ij}$  sempre deverá ser maior que  $f_{ij}$ . Por exemplo: para um fluxo de 16 Mbps e pela restrição (5) pode-se escolher 20 Mbps ou 50 Mbps como sendo a capacidade para o respectivo enlace. Já, dobrando o valor do fluxo para 32 Mbps e pela restrição (5), o único valor que se pode escolher para a capacidade do enlace é 50 Mbps.

Observa-se uma redução do desvio padrão relacionado aos valores de custo, tanto em relação ao aumento do número de iterações, quanto em relação ao aumento do número de partículas. Nota-se ainda que o número de partículas é um importante parâmetro associado ao desempenho do PSO. O aumento de  $N_p$  faz reduzir o desvio padrão indicando que um número maior de boas soluções se aproximam da solução ótima.

O segundo teste foi realizado com a rede da topologia 2. Os valores dos fluxos foram distribuídos entre 5 e 55 Mbps, com média igual a 21.4 Mbps (tráfego #2.a). Utilizou-se dois conjuntos de capacidades a escolher:  $S = (6, 8, 34, 51, 155)$ , produzindo um espaço de busca de  $N_s^L = 5.9 \times 10^{16}$ ; e  $S = (6, 8, 10, 15, 20, 34, 40, 51, 100, 155)$ , que produz um espaço de busca  $N_s^L = 10 \times 10^{24}$ . Para o ES e o primeiro conjunto de capacidades, o tempo computacional requerido para o cálculo foi de aproximadamente cinco horas e o resultado encontrado apresentou um custo de 9054.21/ano. O mesmo custo foi encontrado pela heurística. Para o segundo conjunto o resultado encontrado apresentou um custo de 6548.61/ano. Neste caso, para a heurística, o custo encontrado foi de 6595.41/ano. O limitante inferior foi de 5525.76/ano.

As Tabelas IV e V resumem o desempenho do PSO para a topologia 2 e tráfego #2.a. O tempo computacional para o primeiro conjunto de capacidades (tabela IV) gira em torno de dez segundos, já para o segundo conjunto (tabela V) gira em torno de um minuto.

TABELA IV

PSO PARA TOPOLOGIA 2, TRÁFEGO #2.A E CONJUNTO COM 5 CAPACIDADES.

Probl.	$N_p$	$N_{itr}$	Taxa de sucesso	Ótimo	Valor médio	Desvio padrão
1	10	100	62/500	9054.21	9774.55	677.95
2	30	100	252/500	9054.21	9195.75	215.24
3	50	100	333/500	9054.21	9133.09	122.50
4	10	300	69/500	9054.21	9766.46	662.27
5	30	300	280/500	9054.21	9182.24	208.75
6	50	300	350/500	9054.21	9120.71	115.25

Como era esperado, as taxas de sucessos apresentados na tabela V são menores do que as apresentadas na tabela IV, pois o espaço de busca utilizando o conjunto  $S$  com 10 capacidades é bem maior do que o espaço de busca utilizando um conjunto  $S$  com 5 capacidades. Portanto, para um mesmo número de partículas e um mesmo número de iterações, a taxa de sucesso diminui com o aumento do espaço de busca. Observe que apesar da taxa de sucesso ser menor na tabela V, os desvios

TABELA V

PSO PARA TOPOLOGIA 2, TRÁFEGO #2.A E CONJUNTO COM 10 CAPACIDADES.

Probl.	$N_p$	$N_{itr}$	Taxa de sucesso	Ótimo	Valor médio	Desvio padrão
1	10	100	10/500	6548.61	7071.26	518.22
2	30	100	139/500	6548.61	6661.29	201.07
3	50	100	165/500	6548.61	6615.49	105.87
4	10	300	22/500	6548.61	7041.58	505.59
5	30	300	158/500	6548.61	6652.92	195.79
6	50	300	177/500	6548.61	6607.39	101.74

padrões geralmente são menores do que aqueles apresentados na tabela IV. Isto se dá porque as diferenças entre os valores que formam o conjunto  $S$  com 10 capacidades é menor do que as diferenças entre os valores que formam o conjunto  $S$  com 5 capacidades.

Por último foi realizado um terceiro teste com a rede da topologia 3. Os valores dos fluxos foram distribuídos entre 5 e 90 Mbps, com média igual a 33.33 Mbps (tráfego #3.a). Utilizou-se dois conjuntos de capacidades a escolher:  $S = (8, 34, 44, 51, 155)$ , produzindo um espaço de busca de  $N_s^L = 3.5 \times 10^{33}$ ; e  $S = (10, 25, 34, 44, 51, 70, 90, 120, 155, 190)$ , que produz um espaço de busca  $N_s^L = 10 \times 10^{48}$ . Para o algoritmo ES não foi possível obter o tempo computacional por ser muito longo. No entanto o custo mínimo para o primeiro e segundo conjuntos de capacidades a escolher é de 18908.03/ano e 12605.89/ano respectivamente. Utilizando Chromy, o limitante inferior encontrado foi de 11239.95/ano enquanto que para a heurística foi encontrado um custo de 19289.9/ano para o primeiro conjunto de capacidades e 12887.4/ano para o segundo conjunto.

Observa-se que a heurística deve ser empregada com cautela, pois embora sendo fácil e rápida, ela visa satisfazer as restrições e não minimizar o custo.

As Tabelas VI e VII resumem o desempenho do PSO para a topologia 3 e tráfego #3.a. O maior tempo computacional para o primeiro conjunto de capacidades (tabela VI) é o do problema 5 e gira em torno de vinte segundos, já para o segundo conjunto (tabela VII) o tempo computacional do problema 2 gira em torno de cinquenta segundos e do problema 3 em torno de oito minutos.

TABELA VI

PSO PARA TOPOLOGIA 3, TRÁFEGO #3.A E CONJUNTO COM 5 CAPACIDADES.

Probl.	$N_p$	$N_{itr}$	Taxa de sucesso	Ótimo	Valor médio	Desvio padrão
1	200	500	125/500	18908.03	19185.2	162.56
2	300	500	132/500	18908.03	19179.8	164.36
3	200	800	131/500	18908.03	19180.0	164.61
4	300	800	148/500	18908.03	19168.9	169.84
5	800	1000	234/500	18908.03	19089.0	185.16

Observe que os resultados obtidos com o PSO e o conjunto  $S$  com 5 capacidades (tabela VI), mostram-se instáveis. Além de não haver nenhuma melhora nos resultados com o aumento

TABELA VII  
 PSO PARA TOPOLOGIA 3, TRÁFEGO #3.A E CONJUNTO COM 10  
 CAPACIDADES.

Probl.	$N_p$	$N_{itr}$	Taxa de sucesso	Ótimo	Valor médio	Desvio padrão
1	800	1000	53/500	12605.89	12702.1	149.07
2	1000	2000	75/500	12605.89	12673.1	123.35
3	3000	4000	247/500	12605.89	12612.6	13.98

do número de iterações  $N_{itr}$ , e com o aumento do número de partículas  $N_p$ , há uma piora no desvio padrão. Este problema ocorre devido ao número pequeno de partículas e/ou iterações em relação ao tamanho do espaço de busca, causando uma espécie de saturação do PSO.

Observe também que, apesar do número de sucessos em encontrar o valor ótimo ser maior no problema 5 da tabela VI do que no problema 1 da tabela VII, o desvio padrão é maior por ser maior a diferença entre os valores que compõe o conjunto  $S$  com 5 capacidades.

## VI. CONCLUSÃO E TRABALHOS FUTUROS

Neste artigo abordou-se a solução dos problemas CA discreto, seguindo a metodologia de projeto de redes IP que considera a dinâmica das redes de pacotes, assim como os efeitos dos protocolos nas diferentes camadas e na QoS experimentada pelos usuários finais. Ao utilizar a busca exaustiva para resolver o problema CA discreto, foi possível encontrar o menor custo para o projeto. Contudo, aumentando o número de capacidades discretas a escolher e/ou aumentando o número de enlaces que formam a rede, o tempo gasto no processo de otimização torna-se muito elevado, pois o algoritmo testa todas as combinações entre os valores possíveis de capacidades para os enlaces. Observa-se que para redes pequenas (topologia 1), é melhor utilizar o método de busca exaustiva para encontrar o resultado do problema CA discreto, pois o cálculo não consome muito tempo e o resultado encontrado é o melhor existente (ótimo). No entanto, a medida que o conjunto de capacidades discretas a escolher e/ou o número de enlaces da rede aumentam, o tempo que a busca exaustiva consome, para encontrar o resultado para o problema CA, torna-se muito elevado, o que justifica o emprego dos outros métodos utilizados (PSO e heurística).

Também foi possível ver, através dos testes, que a medida que o espaço de busca do problema aumenta, com o aumento da rede ou com o aumento do conjunto de capacidades discretas a escolher, é necessário aumentar o número de partículas ( $N_p$ ) e/ou o número de iterações ( $N_{itr}$ ) do PSO. Observa-se também que a performance do PSO está mais relacionada ao número de partículas do que ao número de iterações. Isto ocorre devido as adaptações empregadas no PSO para a resolução do problema.

Ao se comparar a heurística proposta com os outros dois métodos (ES e PSO), aquela mostrou ser extremamente rápida em encontrar uma solução viável para o problema CA. Vale a pena lembrar que esta heurística não garante o resultado ótimo, mas sim um resultado viável, já que tem por objetivo

satisfazer as restrições e não minimizar o custo da rede. O que faz com que ela deva ser empregada com cautela.

Como proposta para trabalhos futuros seria interessante fazer uma comparação do desempenho do PSO com uma outra meta-heurística conhecida, como por exemplo, os Algoritmos Genéticos ([15]) e também efetuar testes/simulações com a utilização de topologias maiores.

## REFERÊNCIAS

- [1] L. Kleinrock, *Queueing Systems, Volume II: Computer Applications*. Wiley Interscience, 1976.
- [2] M. Gerla e L. Kleinrock, "On the topological design of distributed computer networks," *IEEE Transactions on Communications*, v. 25, pp. 48–60, 1977.
- [3] E. C. G. Wille, "Design and Planning of IP Networks Under End-to-End QoS Constraints," *Politecnico di Torino*, Janeiro 2004.
- [4] M. Garetto e D. F. Towsley, "Modeling, simulation and measurements of queuing delay under long-tail internet traffic," *Proc. ACM Sigmetrics 2003*, pp. 47–57, San Diego - USA, 2003.
- [5] V. Paxson e S. Floyd, "Wide-area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, v. 3, pp. 226–244, 1995
- [6] K. Park e W. Willinger, *Self-Similar Network Traffic and Performance Evaluation*. John Wiley, 2000.
- [7] C. Fraleigh, F. Tobagi e C. Diot, "Provisioning IP Backbone Networks to Support Latency Sensitive Traffic," *IEEE Infocom 03*, São Francisco - USA, Maio 2003.
- [8] X. Chao, M. Miyazawa e M. Pinedo, *Queueing Networks - Customers, Signals and Product Form Solutions*. John Wiley, 1999.
- [9] E. C. G. Wille, M. Mellia, E. Leonardi, M. Ajmone-Marsan e M. Garetto, "Considering End-to-End QoS in IP Network Design," *11<sup>th</sup> Intern. Telecommunications Network Strategy and Planning Symposium (Networks 2004)*, v. 3, pp. 69–74, Vienna - Áustria, 2004.
- [10] J. Kennedy e R. C. Eberhart, "A new optimizer using particle swarm theory," *Proc. 6<sup>th</sup> Int. Symp. on Micro Machine and Human Science*, pp. 39–43, Nagoya - Japão, 1995.
- [11] J. Kennedy e R. C. Eberhart, *Swarm Intelligence*. Morgan Kaufmann, 2001.
- [12] H. S. Lopes e L. S. Coelho, "Particle swarm optimization with fast local search for the blind traveling salesman problem," *Proc. of 5<sup>th</sup> Int. Conf. on Hybrid Intelligent Systems*, pp. 245–250, Rio de Janeiro - Brasil, 2005.
- [13] Y. Shi, "Particle Swarm Optimization," *IEEE Neural Networks Society*, Fevereiro 2004.
- [14] J. R. Chromy, "Design Optimization with Multiple Objectives," *American Statistical Association Proceedings of Surveys Research Methods Section*, pp. 194–199, 1987.
- [15] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.