

Um Estudo Comparativo de Estratégias Evolutivas Aplicadas ao Problema de Roteamento Multicast

Thiago Fialho de Queiroz Lafetá
Universidade Federal de Uberlândia (UFU)
Uberlândia, MG, Brasil
fialhot@gmail.com

Gina Maira Barbosa de Oliveira
Universidade Federal de Uberlândia (UFU)
Uberlândia, MG, Brasil
gina@facom.ufu.br

Christiane Regina Soares Brasil
Universidade Federal de Uberlândia (UFU)
Uberlândia, MG, Brasil
christiane.ufu@gmail.com

Resumo— Para garantir que seja obtido um nível adequado de comunicação fim-a-fim em redes é importante garantir uma Qualidade de Serviço (QoS). O problema do roteamento com QoS envolve múltiplos objetivos a serem otimizados e/ou atendidos simultaneamente. Neste contexto, este trabalho propõe o uso do Algoritmo Evolutivo Multiobjetivo (AEMO) uma vez que este lida com diversos objetivos a fim de obter um conjunto de possíveis soluções para o problema. O AEMO empregado neste trabalho é o SPEA2, um dos algoritmos multiobjetivos mais investigados na literatura. Uma das características importantes dessa modelagem evolutiva reside no fato da solução ser representada por uma árvore. O objetivo deste trabalho é realizar um estudo comparativo das estratégias evolutivas em um AEMO, tais como a geração da população inicial e o cruzamento, aplicado ao problema de roteamento. Um novo método de cruzamento entre os pais (árvores) é proposto aqui chamado de cruzamento por caminho. A partir dos resultados dos experimentos aqui apresentados foi possível concluir: (i) qual a estratégia mais adequada para a geração da população inicial; (ii) que as estratégias de crossover possuem características distintas e que sua adequabilidade pode variar de acordo com o problema; (iii) que uma estratégia de cruzamento mista que mistura o novo crossover por caminho com o crossover adotado em trabalhos anteriores se mostrou mais estável em relação a diferentes instâncias do problema.

Keywords— Algoritmos evolutivos multiobjetivos; roteamento multicast; qualidade de serviço; algoritmos evolutivos.

I. INTRODUÇÃO

Um dos fatores de grande importância para as redes de computadores é a Qualidade de Serviço (QoS), que garante um bom nível de comunicação fim-a-fim no roteamento. Transmissão multicast é a entrega de informação para múltiplos destinatários simultaneamente usando a estratégia mais eficiente, onde as mensagens só passam por um link uma única vez e somente são duplicadas quando o link para os destinatários se divide em duas direções.

Neste contexto, pode-se aplicar algoritmos de otimização para obter a rota desejada para uma transmissão multicast.

Dentre estes, destacam-se os Algoritmos Evolutivos (AEs) por sua facilidade no desenvolvimento, e por possibilitar a geração de um conjunto de possíveis soluções ao final da execução (e não apenas uma solução, como ocorre em alguns algoritmos). Define-se um algoritmo evolutivo como um procedimento iterativo de otimização inspirado na Teoria da Evolução de Darwin. Um exemplo destes métodos é o Algoritmo Genético (AG) que inicialmente gera uma população aleatória de possíveis soluções para o problema. A partir dessa população inicial, as soluções são evoluídas usando operadores de reprodução (cruzamento e mutação) inspirados na Teoria da Evolução, de forma que os indivíduos representem soluções cada vez melhores [5]. Diversos trabalhos anteriores empregaram um algoritmo genético padrão na busca de rotas para transmissão multicast que atendessem a determinados requisitos de QoS [10], [11], [12]. Nesses trabalhos precusores, o algoritmo genético realiza sua busca evolutiva baseado em uma função objetivo única, que compõe em um valor escalar diversas métricas de QoS tais como, custo, delay e largura de banda. Entretanto, pela própria natureza multiobjetiva do problema de roteamento com múltiplas métricas, trabalhos posteriores mostraram que o emprego de técnicas multiobjetivos se mostrou mais adequado ao problema [13], [14], [15], [2].

Especificamente, os Algoritmos Evolutivos Multiobjetivos (AEMOs) são abordagens que trabalham com dois ou mais objetivos simultaneamente, onde estes objetivos não necessitam ter uma relação explícita entre eles. Dentre estes, pode-se destacar o SPEA2 [16], por sua simplicidade e bom desempenho. Em 2010, Bueno e Oliveira [2] aplicou o SPEA 2 para o problema de roteamento obtendo resultados promissores.

Neste sentido, o objetivo deste trabalho é realizar um estudo comparativo de algumas estratégias evolutivas, tais como a geração da população inicial e o cruzamento (do inglês, *crossover*), sobre o método de Bueno e Oliveira [2] aplicado ao problema de roteamento.

O restante do trabalho está assim organizado: a próxima seção apresenta alguns conceitos básicos sobre otimização multiobjetivos; a Seção III formaliza o problema do roteamento multicast, apresentando alguns dos principais objetivos que normalmente são otimizados em uma abordagem com QoS; a Seção IV apresenta as principais investigações desse trabalho, detalhando as estratégias evolutivas avaliadas nos experimentos, tanto as utilizadas na geração da população inicial, quanto os métodos de cruzamento; a Seção V detalha e discute os principais resultados desses experimentos enquanto a Seção VI apresenta as principais conclusões desse trabalho e os próximos passos em nossa pesquisa.

II. OTIMIZAÇÃO MULTIOBJETIVOS

Existem diversos problemas de otimização que consideram mais de um objetivo, sendo que eles se tornam mais interessantes quando esses objetivos entram em conflito. Esses problemas são conhecidos como multiobjetivos. Logo abaixo vamos passar alguns conceitos para ajudar no entendimento do artigo:

1. **Otimização Multiobjetivos:** Dado um vetor de variáveis de decisão de dimensão n , $x = \{x_1, \dots, x_n\}$ no espaço de busca X , visamos encontrar um vetor $x^* \in X$ que minimiza (ou maximiza) simultaneamente as r funções objetivo $f(x^*) = \{f_1(x^*), \dots, f_r(x^*)\}$.
2. **Dominância:** Uma solução viável x domina outra solução viável y e somente se $f(x) \leq f(y)$ para $i=1, \dots, r$ e $f(x) < f(y)$ em pelo menos um dos objetivos, considerando um problema de minimização.
3. **Pareto Ótimo, Não-Dominada:** Uma solução viável ótima é dita Pareto ótima ou não-dominada se não for dominada por nenhuma solução viável no espaço de busca.

III. FORMALIZAÇÃO DO PROBLEMA ROTEAMENTO MULTICAST

Considere a topologia da rede modelada como um grafo não direcionado $G = (V, E, W)$, onde V representa o conjunto de nós do grafo, E o conjunto de arestas e W os rótulos de cada aresta. Um nó $\{r\} \subset V$ é dito nó raiz (servidor que envia a mensagem). Um conjunto de nós $F \subset V$ representa o grupo multicast (grupo de nós de destino final f), ou seja, o grupo de nós em que a mensagem deve chegar. Cada aresta $e \in E$ é rotulada com quatro pesos, $c(e)$, $d(e)$, $z(e)$, $t(e)$, que corresponde ao custo $c(e)$ (tempo de envio da mensagem), delay $d(e)$ (tempo de preparo de envio da mensagem), capacidade $z(e)$ (capacidade do tráfego) e tráfego $t(e)$ (tráfego gerado pela mensagem), estes pesos compõem um elemento do conjunto W .

O objetivo do problema é encontrar uma árvore $T = (V, E_T)$ gerada do grafo G , onde o nó r é o nó raiz da árvore, os nós folhas são os nós destinos do conjunto F (não necessariamente todos os nós de F devem ser folhas, porém todos os nós folhas são destinos). Considere ϕ o fluxo de dados da mensagem enviada pelo nó r .

O intuito é gerar árvores que otimizem alguns objetivos relacionados a QoS. Podemos citar como possíveis objetivos a serem otimizados, também investigados em [2]:

- Custo total da árvore: Soma dos custos de todas as arestas;
- Delay fim-a-fim atendido: Soma o delay de cada aresta que forma um caminho do nó raiz até um nó destino e verifica se a soma ultrapassa a tolerância. É avaliado para cada nó destino;
- Delay total da árvore: Soma o delay de todas as arestas da árvore;
- Delay fim-a-fim médio: Soma o delay de cada aresta que forma um caminho do nó raiz até um nó destino. O delay é somado para cada nó destino e tira-se a média das somas;
- Delay fim-a-fim máximo: Soma o delay de cada aresta que forma um caminho do nó raiz até um nó destino. O delay é somado para cada nó destino e retorna-se a maior soma;
- *Hops count*: Retorna o número de arestas da árvore;

IV. ESTUDO DE ESTRATÉGIAS EVOLUTIVAS

O primeiro passo neste trabalho foi implementar um AEMO para o roteamento *multicast* baseado no SPEA2 [16] e nas estratégias usadas por Bueno e Oliveira [2]. Após implementar e testar o AEMO, investigamos algumas variações nas estratégias de geração da população inicial e nos métodos de *crossover*. As demais estratégias utilizadas no AEMO, tais como a mutação e a seleção de pais, foram implementadas de acordo com [2].

A. Investigação I: Geração da População Inicial

No trabalho de Bueno e Oliveira [2], os indivíduos são representados na forma de árvores genéricas e a população inicial é gerada de modo aleatório, garantindo que o nó raiz e os nós destinos sejam alcançados. Em [7] foi apresentado um estudo sobre variações nas estratégias aleatórias de geração da população quando os indivíduos são árvores.

Quatro tipos de inicializações foram testados, sendo as duas primeiras citadas em [7]:

- **Inicialização Aleatória (ini_a):** Este método percorre os nós do grafo em busca de novas arestas. Iniciando no nó raiz seleciona de forma aleatória umas das arestas adjacentes, e encontra um novo nó x . Em seguida, adiciona a aresta e o nó à árvore, se possível. O nó x será o nó usado na próxima iteração. Essa é a estratégia de inicialização que mais se assemelha à empregada em [2];
- **Inicialização de Prim (ini_p):** A inicialização *ini_p* é baseada no algoritmo de PRIM [8] para o cálculo da árvore geradora mínima. Esse algoritmo utiliza uma lista de arestas L , que guarda as arestas que podem ser adicionadas na nova árvore A . A lista L é iniciada a partir das arestas adjacentes do nó raiz. Na execução do algoritmo, a cada nó N adicionado na árvore A , a lista L é atualizada com a inserção das arestas adjacentes a N . Entretanto, para que o método gere

uma população inicial com variedade genética, as arestas que serão inseridas na árvore são escolhidas de forma aleatória, enquanto que no algoritmo PRIM essa escolha é baseada no menor peso, para a geração da árvore geradora mínima.

- **Inicialização Aleatória Modificada (ini_am):** Aplica a mesma ideia da ini_a, porém, o nó da próxima iteração é escolhido de forma aleatória;
- **Inicialização Aleatória Modificada usando Torneio (ini_amt):** Usando como base a inicialização ini_am, cada aresta do grafo é inicializada com um peso p antes de começar a busca, durante o processo de criação de um novo indivíduo, sendo que toda vez que uma aresta é inserida na árvore o peso é decrementado. A aresta escolhida é a de maior peso.

O objetivo da *Investigação I* é analisar o quanto cada método de geração da população inicial influencia na convergência do AEMO para soluções não dominadas.

B. Investigação II: Métodos de Crossover

O operador de cruzamento, ou *crossover*, é aquele que imita o processo de reprodução na natureza, em que a partir dos pais obtém-se um novo indivíduo com a combinação genética dos mesmos.

O método de *crossover* utilizado no trabalho de Bueno e Oliveira [2], aqui denominado *crossover por similaridade* (CS), gera uma nova árvore (filho único) a partir de duas árvores na população corrente. Este *crossover* adota os seguintes passos:

1. **Divisão:** Identificar arestas comuns entre os pais gerando um conjunto de subárvores;
2. **Reconexão:** Reconectar as subárvores resultando em uma nova árvore;
3. **Poda:** Podar os vértices extras.

No primeiro passo, o *crossover* separa as arestas que se repetem nos dois pais e delas formam-se várias subárvores, sendo que estas subárvores são armazenadas em uma lista. A cada etapa de reconexão, o método separa duas subárvores, seleciona um nó de cada e encontra um caminho entre estes nós, formando uma nova subárvore. Em [2] a reconexão das subárvores pode ser feita de duas maneiras: (i) reconexão por menor caminho e (ii) reconexão aleatória. A primeira tenta encontrar o menor caminho entre dois nós, enquanto que a segunda gera um caminho aleatório, fornecendo possibilidades iguais a todas as arestas.

Investigando o método de *crossover* utilizado em [2], observa-se no indivíduo gerado que sua informação genética não advém 100% dos pais, isto é, parte da informação genética é oriunda dos pais e a parte restante é inserida na etapa de reconexão. Por esse motivo esse tipo de *crossover* contribui bastante na diversidade da população.

Neste trabalho foi proposto um novo método de *crossover* que absorve mais informação dos pais do que o utilizado em [2], impactando em uma convergência mais forte. Este método

denominado *crossover por caminho* (CC) difere do anterior na etapa de Divisão, onde é gerada a floresta de subárvores. Nesse novo método, a quantidade de subárvores gerada é igual à quantidade de nós destinos, uma vez que cada subárvore é o caminho do nó raiz até um nó destino. Um caminho de uma árvore é um conjunto de arestas que ligam um nó qualquer da árvore a outro nó da árvore. Sendo assim, dados dois pais, ou seja, duas árvores, cada pai gera um conjunto de subárvores, sendo que cada subárvore representa um caminho entre o nó raiz e cada nó destino. Assim, teremos duas subárvores para cada nó destino, uma advinda de cada pai. Na etapa de Reconexão o método cria uma nova árvore selecionando um caminho para cada nó destino na nova árvore, dentre cada par possível por destino. Para cada nó destino, o método seleciona uma das duas subárvores de forma aleatória, dando a oportunidade igual desta subárvore ser idêntica a qualquer um dos dois pais. A Figura 1 ilustra o processo do crossover CC, através de um exemplo onde o nó 0 é a fonte e os nós 4, 5 e 7 são os destinos. Inicialmente, cada árvore pai é subdividida em 3 caminhos, que levam da raiz até cada nó destino. Esses caminhos serão utilizados como subárvores na geração da nova árvore filho. É escolhida aleatoriamente dentre os dois pais, uma subárvore para cada nó destino. No exemplo, a subárvore que alcança o destino 4 é extraída do Pai 1, enquanto as subárvores que alcançam os destinos 5 e 7 são extraídas do Pai 2, gerando a árvore filho mostrada ao final da figura.

V. RESULTADOS

Para essa investigação foram utilizadas 6 topologias de rede diferentes: elas foram ordenadas por número de modo que quanto maior o número, maior a complexidade da rede. As topologias utilizadas neste trabalho são as mesmas utilizadas por Bueno e Oliveira [2]. A partir desse estudo prévio [2], obtivemos as informações sobre as fronteiras de Pareto para cada cenário (Rede, Par de objetivos). A Tabela I mostra a dimensão de cada topologia, apresentando a quantidade de nós e de arestas das redes.

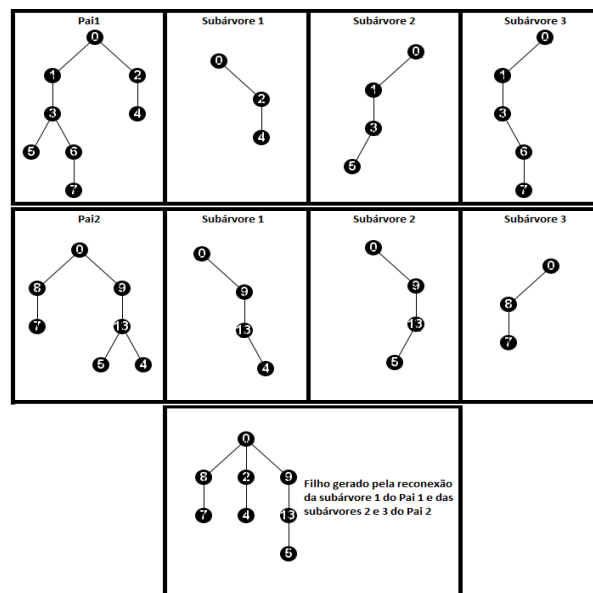


Figura 1: Exemplo de Crossover CC.

Tabela I. CARACTERÍSTICAS DAS REDES

Redes	Q ^{td} de Nós	Q ^{td} de Arestas
Rede 0	15	44
Rede 1	18	50
Rede 2	33	106
Rede 3	50	126
Rede 4	75	188
Rede 5	75	188

Para as duas primeiras redes, por serem mais simples, foram usadas os seguintes parâmetros no AEMO: 30 indivíduos na população e 50 gerações. As outras redes usam: 60 indivíduos na população e 100 gerações. A taxa de *crossover* é 100% (SPEA2) e a taxa de mutação é 20% [2]. Cada teste foi executado 100 vezes usando os parâmetros definidos. Para avaliar cada execução usamos as seguintes métricas (Considere P* o Ótimo Pareto de uma formulação do problema de acordo com [2] e P o Pareto encontrado pelo método para essa mesma formulação):

- Taxa de Erro (er): Com base no Ótimo Pareto P*, comparamos os elementos do Pareto P com os elementos de P*. A taxa de erro é dada pelo percentual de elementos do Pareto P* não estão em P.
- *Generational Distance* (gd): Calculamos a distancia mínima entre os elementos do Pareto P* e P. Quanto menor essa distância, melhor a convergência.
- *Spread* (sp): Tem o objetivo de avaliar quão uniformemente estão distribuídas as soluções de P ao longo do espaço de objetivos. Quanto menor o valor do *spread* mais distribuído os elementos de P estão.
- Hipervolume (hv): Calcula o quão disperso estão os elementos de P. Quanto maior o valor, mais dispersos estão os elementos de P.

A. Análise Comparativa das Estratégias de Inicialização da População

Na primeira investigação efetuada, foi realizada uma análise comparativa das diferentes estratégias de inicialização da população descritas na Seção III: ini_a, ini_p, ini_am e ini_amt. Para tal, o algoritmo de roteamento foi executado 100 vezes utilizando-se cada estratégia investigada e a média da taxa de erro referente a cada rede foi calculada. Nesses experimentos, foram avaliadas cinco diferentes formulações do problema do roteamento com 2 objetivos também investigadas em [2]. São elas:

P1: *Custo Total* e *Delay fim-a-fim atendido*.

P2: *Custo Total* e *Delay total* da árvore.

P3: *Custo Total* e *Delay fim-a-fim médio*.

P4: *Custo Total* e *Delay fim-a-fim máximo*.

P5: *Custo Total* e *Houps count*.

A taxa de erro média foi computada para cada formulação e a Tabela II apresenta os resultados médios considerando-se as cinco formulações e os desvios correspondentes.

Tabela II. MÉDIA E DESVIO PADRÃO DOS ERROS DE CADA REDE PARA CADA INICIALIZAÇÃO (ERRO +/- DESVIO PADRÃO). SÃO MARCADOS COM CINZA OS MENORES ERROS DE CADA REDE.

Redes	Média dos Erros das Inicializações			
	ini_a	ini_p	ini_am	ini_amt
Rede 0	0	0	0	0
Rede 1	0	0	0	0
Rede 2	5.12 +/- 3.81	4.99 +/- 4.08	4.63 +/- 0.11	4.76 +/- 4.31
Rede 3	1.34 +/- 2.34	1.37 +/- 2.99	2.13 +/- 3.81	1.79 +/- 3.67
Rede 4	0.61 +/- 1.69	0.47 +/- 1.25	0.46 +/- 1.30	0.54 +/- 1.38
Rede 5	18.18 +/- 15.88	16.54 +/- 15.59	16.51 +/- 14.74	15.80 +/- 14.03

De acordo com a Tabela I, pode-se notar que nas duas redes mais simples (Rede 0 e Rede 1), todas as inicializações tiveram o mesmo desempenho. Conforme as redes aumentam a complexidade, pode-se observar diferenças nos resultados de acordo com a estratégia de inicialização utilizada. Neste caso, na Rede 2 e Rede 4, os melhores resultados foram com ini_am, enquanto que na Rede 3 foi com ini_a e na Rede 5, a ini_amt. De forma geral, a inicialização ini_am apresentou melhores resultados na maioria das redes, cerca de 67% das redes.

B. Análise da Similaridade entre os Pais e os Filhos Gerados

Para melhor compreendermos o efeito da modificação proposta no novo método de *crossover*, efetuamos uma investigação preliminar em relação à similaridade entre os filhos gerados e seus pais. No processo desta análise, geram-se diversos pares de pais de maneira aleatória, utilizando-se como base uma rede específica. Para cada par de pais gerado nessa amostra, efetua-se o cruzamento retornando um filho. Ao final, a porcentagem de arestas do filho que existe em comum com os pais é analisada. Em nossos experimentos, utilizamos uma amostra de 100 pares de pais para cada uma das seis redes investigadas. Calculou-se então a média da porcentagem de informação em comum entre os pais e o filho, considerando-se os 100 *crossovers* realizados por rede.

A Tabela III mostra os resultados desta análise. Considere a sigla CS (*crossover* por similaridade) para o *crossover* utilizado por [2] e CC (*crossover* por caminho), o novo *crossover* proposto neste trabalho.

Tabela III. PERCENTUAL DE SIMILARIDADE APÓS A APLICAÇÃO DOS MÉTODOS DE CROSSOVER CS E CC.

Redes	Média de porcentagem de informação similar	
	Crossover CS	Crossover CC
Rede 0	81.52 %	95.69 %
Rede 1	84.34 %	95.92 %
Rede 2	74.85 %	93.60 %
Rede 3	84.84 %	95.16 %
Rede 4	87.35 %	95.25 %
Rede 5	90.58 %	96.05 %

Observe que o método de *crossover* CC absorve mais informação dos pais que o método CS em todas as redes, ou seja, possui uma maior convergência para soluções que apresentam caminhos fonte-destino idênticos a um de seus pais. Entretanto, essas novas soluções não são necessariamente idênticas por completo a um dos pais individualmente. Uma vez que normalmente existem vários destinos, parte dos caminhos é idêntica a um dos pais e a outra parte é idêntica ao outro.

C. Análise Comparativa dos Métodos de Crossover

Como o problema do roteamento multicast com QoS envolve vários objetivos, um *crossover* com uma convergência muito forte como o CC pode não ser uma boa opção, pois neste tipo de problema a diversidade tem grande impacto e também pode ocorrer convergência prematura. Por outro lado, um *crossover* como o CS, dependendo da rede, pode oferecer diversidade excessiva prejudicando a convergência para o Ótimo de Pareto.

Três métodos foram investigados neste trabalho diferenciando-se apenas na estratégia de crossover. O primeiro método usa o crossover CS, que corresponde ao cruzamento utilizado por Bueno e Oliveira [2]. O segundo usa o crossover CC proposto neste trabalho. O terceiro método denominado MIX é uma mistura dos dois anteriores: a cada geração, o crossover CS ou o crossover CC é aplicado. O método que será usado a cada crossover é definido de acordo com a geração, sendo que os métodos são alternados a cada turno de crossover. Por exemplo, por 5 gerações usa-se somente o crossover CC e nas próximas 5 usa-se somente o crossover CS. Essa alternância prossegue por todas as gerações e a mudança de crossover vai variando de 5 em 5 gerações.

A cada 100 execuções, calculamos a média e o desvio padrão dos resultados de cada métrica. A Tabela IV apresenta os resultados da formulação do problema P1 que envolve os objetivos Custo Total e Delay fim-a-fim atendido.

RESULTADOS DO PROBLEMA P1 USANDO AS REDES 0, 1, 2, 3, 4 E 5 E AS MÉTRICAS ER, GD, SP E HV (FORMATO: MÉDIA +/- DESVIO PADRÃO).

	Método	Tempo (ns)	ER	GD	SP	HV
R E D E 0	CS	106	0.0 ± 0.0	0.0 ± 0.0	0.458 ± 0.0	382.4 ± 20.227
	CC	109	0.0 ± 0.0	0.0 ± 0.0	0.458 ± 0.0	383.33 ± 18.964
	MIX	103	0.0 ± 0.0	0.0 ± 0.0	0.458 ± 0.0	381.95 ± 20.273
R E D E 1	CS	117	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	301.02 ± 15.576
	CC	127	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	304.860 ± 14.583
	MIX	124	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	303.780 ± 16.584
R E D E 2	CS	1503	24.571 ± 12.302	0.754 ± 0.397	0.286 ± 0.092	1478.300 ± 73.356
	CC	1701	18.524 ± 12.493	0.390 ± 0.463	0.185 ± 0.068	1515.140 ± 77.860
	MIX	1598	15.000 ± 14.052	0.396 ± 0.447	0.211 ± 0.075	1499.630 ± 88.064
R E D E 3	CS	2086	2.700 ± 6.458	0.027 ± 0.065	0.494 ± 0.031	488.190 ± 66.417
	CC	2703	0.700 ± 3.441	0.007 ± 0.034	0.480 ± 0.027	482.860 ± 58.640
	MIX	2359	1.533 ± 5.216	0.015 ± 0.052	0.490 ± 0.027	493.580 ± 64.163
R E D E 4	CS	3521	0.143 ± 1.421	0.003 ± 0.028	0.835 ± 0.013	765.330 ± 69.237
	CC	5994	1.483 ± 5.487	0.015 ± 0.055	0.728 ± 0.103	584.790 ± 116.65
	MIX	4773	0.000 ± 0.000	0.000 ± 0.000	0.825 ± 0.043	747.880 ± 92.490
R E D E 5	CS	5658	2.800 ± 6.442	0.046 ± 0.123	0.593 ± 0.043	579.420 ± 55.333
	CC	13188	3.400 ± 8.165	0.087 ± 0.242	0.670 ± 0.039	495.460 ± 56.699
	MIX	9301	3.900 ± 7.747	0.054 ± 0.115	0.616 ± 0.051	551.310 ± 60.185

Podemos observar que para redes simples como a Rede 0 e 1 os três métodos conseguem encontrar o Ótimo Pareto com facilidade. Pelos valores da taxa de erro (ER), é possível observar que o método CC se sobressai ao método CS, nas redes 2 e 3. Porém, nas redes 4 e 5, o método CS apresenta um menor erro. Essa variação mostra ter sido uma boa estratégia propor o método MIX, que mistura as técnicas de CS e CC, pois ele tenta encontrar um equilíbrio adequado para obter uma boa convergência em todas as redes. Apesar de alguns métodos retornarem uma taxa de erro mais alta, a métrica *Generational Distance* (GD) apresenta valores bem promissores para todos os métodos, pois quanto menor o GD mais o Pareto encontrado se aproximou do Ótimo de Pareto. Podemos observar que todos os GDs apresentam valores bem próximos de zero. As métricas *Spread* (SP) e o Hipervolume (HV) mostram o quanto o Pareto encontrado está bem disperso para os três métodos: quanto

menor o valor do SP ou quanto maior o valor do HV, mais disperso está o Pareto.

Com relação ao tempo médio de processamento em cada topologia, observamos na Tabela IV que embora ainda ocorra um aumento significativo se compararmos as redes menores (0 e 1) com as maiores (4 e 5), o algoritmo é viável de ser utilizado em tempo real. Por exemplo, no caso da Rede 5, a maior e de mais difícil convergência, o algoritmo de roteamento empregando o crossover MIX demanda aproximadamente 9,3 ms em média para ser executado. Também é possível perceber que o crossover CC apresenta maior custo computacional que o CS, sendo esse aumento dependente da rede utilizada. O método MIX apresenta um tempo intermediário entre o CS e o CC, justamente por combinar esses dois métodos.

VI. CONCLUSÃO

Por meio deste trabalho, pode-se concluir que, das quatro inicializações testadas, a estratégia de geração da população ini_am obteve melhor desempenho na maioria das redes, considerando que houve um empate nas redes mais simples, e nas mais complexas, ela foi melhor em 50% das redes. Deste modo, este modo de inicialização é o escolhido para o desenvolvimento de trabalhos futuros.

O *crossover* proposto neste trabalho (CC) absorveu mais informações dos pais que o método usado em [2] (CS) em todas as redes, o que impactou em uma convergência mais forte para soluções que herdaram características similares aos seus pais.

Três tipos diferentes de *crossover* foram avaliados: CS, CC e MIX, sendo esse último uma combinação dos métodos CS e CC, alternando-se a aplicação dos mesmos ao longo das gerações. Desta maneira, tanto a diversidade quanto a convergência foram mantidas, tornando-se um método mais equilibrado. Das quatro métricas multiobjetivas avaliadas (erro, GD, Spread e Hipervolume), a taxa de erro apresentou maior diferenciação entre os 3 métodos de crossover. Assim, a taxa de erro mostrou que o método MIX quando não retorna o melhor resultado dentre os três, retorna uma taxa próxima ao melhor valor entre CS e CC. Esse método será adotado nas etapas subsequentes do trabalho

Como trabalhos futuros pretende-se desenvolver um algoritmo capaz de lidar com mais de três objetivos para o problema do roteamento, uma vez que, para casos com muitos objetivos, o trabalho de Bueno e Oliveira [9] mostrou que o desempenho do algoritmo de roteamento evolutivo era prejudicado. Assim, espera-se que a questão da diversidade investigada no presente trabalho torne-se ainda mais crítica em cenários com vários objetivos. Diante desse desafio, serão aplicados algoritmos *many-objectives* [6] para este problema, que trabalham com a combinação dos objetivos [3] e [4], possibilitando que o espaço de busca seja melhor explorado.

VII. AGRADECIMENTOS

GMBO agradece ao CNPq, à CAPES e à Fapemig pelo suporte financeiro.

REFERENCES

- [1] Assunção, W. K. G., Colanzi, T. E., Vergilio, S. R., e Pozo, A. T. R., Estabelecendo seqüências de teste de integração de classes: Um estudo comparativo da aplicação de três algoritmos evolutivos multiobjetivos. In Workshop de Teste e Tolerância a Falhas, WTF'2011.
- [2] Bueno, M.L.P. and Oliveira, G.M.B. [Multicast flow routing: Evaluation of heuristics and multiobjective evolutionary algorithms](#). Evolutionary Computation (CEC), 2010 IEEE Congress on, 1-8.
- [3] Brasil C. R. S, Delbem A. C. B. and Barroso da Silva F. L., Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction (pages 1719–1734), Journal of Chemistry Computational, 2013.
- [4] Delbem, A.C.B. Restabelecimento de Energia em Sistemas de Distribuição por Algoritmo Evolucionário Associado a Cadeia de Grafos. Phd thesis, EESC/USP, São Carlo, SP, 2002.
- [5] Goldberg, D., Genetic Algorithms in Search, Optimization and Machine Learning. Addison - Wesley, 1989.
- [6] Ishibuchi H., Akedo N., Ohyanagi H. and Nojima Y., Behavior of EMO algorithms on many-objective optimization problems with correlated objectives. IEEE Congress on Evolutionary Computation 2011: 1465-1472.
- [7] Lima, T.W. Delbem, A.C.B., Estrutura de Dados Eficientes para Algoritmos Evolutivos Aplicados ao Projeto de Redes. Instituto de Ciências Matemáticas e de Computação, ISSN - 0103-2569. 2007.
- [8] Cormen, Thomas; Stein, Clifford. Introduction to Algorithms (em inglês). 2 ed. [S.l.]: MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7.
- [9] Bueno, M.L.P. and Oliveira, G.M.B. [Four-objective formulations of multicast flows via evolutionary algorithms with quality demands](#). Telecommunication Systems 55 (3), 435-448, 2014.
- [10] Ravikumar, C. P; Bajpai, R.. "Source-based delaybounded multicasting in multimedia networks". Computer Communications. 1998. Vol. 21, pp. 126-132.
- [11] Zhengying, W., Bingxin, S., and Erdun, Z., Bandwidth-delay-constrained least-cost multicast routing based on heuristic ga.Computer Communications, 24(7-8):685–692, 2001.
- [12] Oliveira, G.M.B.; Araújo, P.T., Determining Multicast Routes with QoS and Traffic Engineering Requirements Based on Genetic Algorithm. In: 2004 IEEE Conference on Cybernetics and Intelligent Systems - CIS2004, 2004, Cingapura. Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems, December, 2004. p. 666-670.
- [13] C. Garrozi and A. G. R. Araújo, "Multiobjective Genetic Algorithm for Multicast Routing," in 2006 IEEE Congress on Evolutionary Computation (CEC'2006). Vancouver, BC, Canada: IEEE, July 2006, pp. 2513-2520.
- [14] J. Prieto, B. Barán, and J. Crichigno, "Multitree- multiobjective multicast routing for traffic engineering." in IFIP AI, ser. IFIP, M. Bramer, Ed., vol. 217. Springer, 2006, pp. 247-256.
- [15] Oliveira, GMB and Vita, S. [A multi-objective evolutionary algorithm with \$\epsilon\$ -dominance to calculate multicast routes with QoS requirements](#). 2009 IEEE Congress on Evolutionary Computation (CEC'2009), Norway, 1-9.
- [16] Zitzler E., Laumanns M., and Thiele L.. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, editors, Evolutionary Methods for Design, Optimisation, and Control, pages 19–26, Barcelona, Spain, 2002. CIMNE.