

UMA ABORDAGEM EVOLUCIONÁRIA PARA O PROJETO DE REDES EIXO-RAIO COM ALOCAÇÃO SIMPLES

Bruno N. Gomes, Alexandre X. Martins

Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais

Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil

brunobng@yahoo.com.br, xmartins@decea.ufop.br

Ricardo S. de Camargo, Jaime A. Ramirez

Universidade Federal de Minas Gerais

Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil

rcamargo@dep.ufmg.br, jramirez@ufmg.br

Resumo – Neste artigo, um eficiente algoritmo genético (AG) é proposto para o projeto de redes eixo-raio com alocação simples. A criação da população inicial do AG é baseada na fase de construção da heurística Greedy Randomized Search Procedure (GRASP), que fornece indivíduos de alta qualidade. Além disso, novos operadores de cruzamento e mutação, específicos para o problema, foram desenvolvidos com o objetivo de melhorar as soluções durante o processo evolucionário. Os resultados no estudo comparativo mostram que o AG proposto claramente supera três outros AGs, considerados estado da arte da literatura, em qualidade de soluções e tempo de processamento.

Palavras-chave – Projeto de redes eixo-raio, algoritmos genéticos, otimização combinatória.

Abstract – In this paper, an efficient genetic algorithm (GA) is proposed for the designing of hub-and-spoke networks with single allocation. The creation of the initial population of the implemented GA is based on construction phase of the Greedy Randomized Search Procedure (GRASP) heuristic, which provides high quality individuals. Furthermore, new crossover and mutation operators, specific to the problem, were developed in order to improve the solutions over the evolutionary process. The results in the comparative study show that the proposed GA clearly outperforms three other GAs, considered to be state-of-the-art of the literature, both in solution quality and CPU time.

Keywords – Hub-and-spoke network design, genetic algorithms, combinatorial optimization.

1. INTRODUÇÃO

Redes estão presentes no cotidiano das pessoas de diversas maneiras, como por exemplo: redes de telecomunicações, transporte, computadores, sociais, entre outras. De uma maneira geral, as redes possuem pontos de origem e destino que comunicam entre si. A topologia de uma rede depende das características da demanda, assim como dos custos fixos de instalação e operação da mesma. Portanto, vários tipos de redes surgiram para atender os requisitos de fluxo de cada configuração.

Uma das topologias existentes é a do tipo malha, ou em inglês *mesh*. Em redes do tipo malha, geralmente, todos os nós são interligados formando um grafo completo, e devido a inviabilidade de custo e operação das mesmas, as redes eixo-raio (E-R), ou em inglês *hub-and-spoke networks*, é uma das alternativas de aplicabilidade mais econômica.

Nas redes E-R a comunicação entre os nós não acontece de forma direta, mas através de nós concentradores. Dependendo da aplicação, os concentradores podem se apresentar de diversas maneiras, como por exemplo, depósitos, *hubs*, pontos *switching*, entre outras. Assim, essas redes são formadas por concentradores que consolidam o fluxo de tráfego dos nós não concentradores, sendo responsáveis pelo roteamento e distribuição desse fluxo entre os diferentes pontos de origem e destino. Tal configuração diminui a quantidade de ligações diretas entre nós, conforme evidenciado pela Figura 1. Nessa figura, (a) mostra a disposição dos nós no espaço, (b) apresenta uma rede do tipo malha e (c) uma rede E-R com alocação simples.

Diversas aplicações possibilitam a utilização das redes E-R, como por exemplo: companhias aéreas [1], redes de comunicação de dados [2], empresas de transporte [3], sistemas de telecomunicações [4], cadeia de lojas [5], entre outras [6, 7]. Assim, as redes E-R se tornaram um importante campo de pesquisa na área de otimização discreta onde concentradores são responsáveis por agregação, roteamento e distribuição de tráfego entre diferentes pontos de origem/destino. A utilização desse tipo de rede objetiva a redução do custo total de transporte e infra-estrutura.

Um benefício trazido pela configuração das redes E-R é o fluxo agregado nos concentradores que permite a utilização de meios de forma mais eficiente, sendo transmitido maior volume de tráfego nas conexões entre concentradores, resultando em menor custo de transmissão por unidade de demanda [8]. Assim, economia de escala é obtida na transferência de grande quantidade de fluxo nas ligações entre os nós concentradores. Essa economia de escala geralmente é representada por um fator constante de desconto ($0 \leq \alpha \leq 1$) no custo de comunicação entre concentradores. Portanto, a redução do custo de transmissão obtida pela

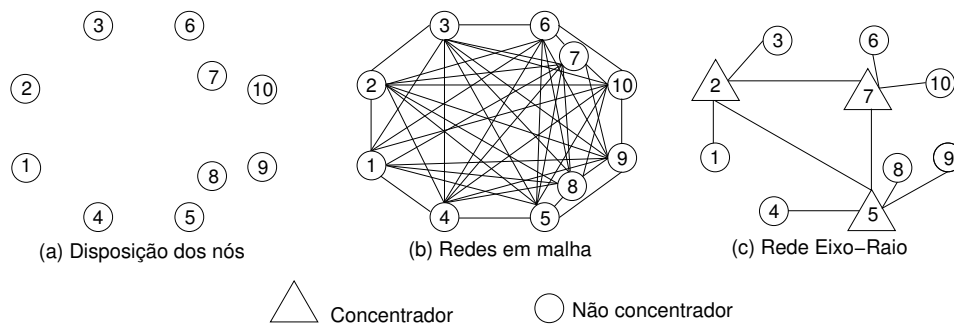


Figura 1: Topologias: Malha e Eixo-Raio.

economia de escala, a diminuição dos custos de instalação, aumento da eficiência logística e do desempenho do sistema são as principais vantagens obtidas pela utilização desse tipo de rede [6].

Geralmente, no projeto de redes E-R é assumido que: todos concentradores são interconectados; nós não concentradores não são diretamente conectados; toda demanda é roteada por um ou no máximo dois concentradores. Além disso, conforme as características consideradas, diferentes suposições devem ser contempladas incluindo:

- Alocação simples ou múltipla dos nós aos concentradores: No problema com alocação simples os nós devem estar alocados a exatamente um concentrador, já o projeto considerando alocação múltipla os nós podem estar alocados a mais de um concentrador;
- Número de concentradores pré-determinado ou variável: Uma das variantes deve possuir p nós concentradores, sendo p um número pré-determinado. Entretanto, na outra variante o número de nós que devem ser concentradores é variável de decisão do problema;
- Permissão ou não de ligação direta entre nós não concentradores: Nessas variantes a ligação direta entre alguns nós não concentradores pode ou não ser permitida. Ou seja, quando essa ligação não é permitida todo fluxo deve ser roteado via concentradores. Porém, quando existe a permissão, o fluxo pode ser trocado diretamente entre os nós não concentradores;
- Projeto com ou sem restrição de capacidade: No projeto com restrição de capacidade, um concentrador possui um valor máximo de tráfego que é capaz de suportar, dessa forma o fluxo agregado à ele é limitado por esse valor. Já no projeto sem restrição de capacidade, a quantidade de tráfego que pode ser acumulada em um concentrador é ilimitada.
- Congestionamento: O projeto considerando os efeitos de congestionamento possui um custo, representado por restrições ou expresso na função objetivo do problema, resultante da sobrecarga de fluxo em um concentrador. Já na outra variante, caso de não consideração desses efeitos, um concentrador opera normalmente inclusive em casos de sobrecarga.

Nesse artigo o projeto de redes E-R com alocação simples é abordado, problema de difícil resolução muito encontrado na área de telecomunicações tais como rede de comunicação de dados, rede de telefonia, entre outras [4]. No projeto de redes E-R o objetivo é encontrar o número de concentradores e a localização dos mesmos na rede, além de atribuir os nós aos concentradores instalados. As seguintes características são encontradas no referido problema: (i) o número de concentradores é variável, mas existe custos fixos para instalá-los; (ii) todos concentradores são interconectados; (iii) impossibilidade de ligação direta entre nós não concentradores; (iv) pelo menos um e no máximo dois concentradores devem ser responsáveis pelo roteamento e transmissão do fluxo; (v) não há restrição de capacidade nem efeitos de congestionamento; (vi) economia de escala é obtida nas conexões entre concentradores.

A primeira formulação quadrática para o projeto de redes E-R foi produzida por O'Kelly [9], que também propôs uma solução heurística para o problema. Já Campbell [10] apresentou uma formulação linear e Abdinnour-Helm e Venkataramanan [11] desenvolveram uma formulação em programação inteira quadrática baseada no fluxo de *multicomodities*. Abdinnour-Helm e Venkataramanan ainda propuseram um método *branch-and-bound* que utiliza a estrutura de fluxo *multicomodities* para obter limites inferiores e um algoritmo genético (AG) para encontrar bons limites superiores.

Como esse problema pertence à classe dos NP-difíceis [7], a maioria dos algoritmos propostos para sua resolução são baseados em heurísticas ou metaheurísticas. Abdinnour-Helm [12] propuseram um método híbrido composto por AG e busca Tabu. O AG determina o número e localização dos concentradores, enquanto a busca Tabu trabalha na alocação dos nós aos concentradores instalados. Topcuoglu *et al.* [13] desenvolveram um AG que obteve melhor desempenho que o método de Abdinnour-Helm, o AG desenvolvido é responsável por determinar o número e localização dos concentradores, assim como alocar os demais nós aos concentradores instalados. Cunha e Silva [3] desenvolveram uma eficiente combinação das heurísticas AG e *simulated annealing* (SA), e fizeram um estudo de caso com uma empresa de transporte no Brasil. Já o trabalho de Chen [14] apresenta uma heurística híbrida baseada em SA, lista tabu e procedimentos de melhoria, esse método híbrido obteve melhor desempenho que o AG de Topcuoglu *et al.* [13]. Mais recentemente, Silva e Cunha [15] desenvolveram 3 variantes da heurística busca tabu *multi-start* e obtiveram resultados satisfatórios. Já Naeem e Ombuki-Berman [16] propuseram um AG, com novas formas de representação da

solução e novos operadores de cruzamento, obtendo melhores resultados que os métodos propostos em [12], [13], e performance competitiva em comparação aos trabalhos realizados em [14] e [15].

Nesse trabalho foi desenvolvido um eficiente AG para projeto de redes E-R com alocação simples. No algoritmo implementado, os operadores do AG foram adaptados para adequação ao problema, e uma estratégia de construção de soluções extremamente eficaz é apresentada. Além disso, introduz-se o conceito de cruzamento com 3 pais para gerar maior variabilidade genética na população. Na comparação realizada com outros três outros AGs, considerados estado da arte para o problema, essa abordagem evolucionária se mostrou claramente superior, tanto na qualidade das soluções obtidas quanto no tempo de processamento para encontrar um valor alvo. O artigo é organizado da seguinte maneira: Seção 2 fornece as definições e a formulação usadas para o problema. O algoritmo implementado é detalhado na Seção 3, enquanto os testes computacionais e resultados obtidos são apresentados na Seção 4. As conclusões são descritas na Seção 5.

2. NOTAÇÕES E FORMULAÇÕES

A formulação em programação matemática é baseada nas seguintes definições: Seja N o conjunto de nós da rede e w_{ij} o fluxo de demanda originado em i com destino à j ($i, j \in N : i \neq j$) a qual é roteada por um ou dois concentradores instalados. Essa demanda originada em i destinada a j ($i, j \in N : i \neq j$), e roteada pelos concentradores k e m ($k, m \in N : k \neq m$), passa por três segmentos: coleta do nó i para concentrador k ; transferência entre os concentradores k e m ; distribuição do concentrador m para nó j . O custo por unidade de demanda ao longo desse caminho é dado por $c_{ijkm} = c_{ik} + \alpha c_{km} + c_{mj}$. Sendo $0 \leq \alpha \leq 1$ um fator de desconto representa a economia de escala nas conexões entre os concentradores. Se somente um concentrador é utilizado em alguma rota, o fator de desconto não é aplicado. Além disso, o custo fixo para instalação de um concentrador no nó k é dado por f_k . A variável $z_{ik} \in \{0, 1\}$ indica a localização dos concentradores e a alocação dos nós aos concentradores instalados. Assim sendo, se $z_{ik} = 1$ o nó i é atribuído ao concentrador k , e $z_{ik} = 0$ caso contrário. Adicionalmente, se um concentrador é localizado no nó k , $z_{kk} = 1$; senão $z_{kk} = 0$. Assim, o problema pode ser formulado da seguinte forma:

$$\min \sum_k f_k z_{kk} + \sum_i \sum_{j>i} \sum_k \sum_{m \neq k} (w_{ij} c_{ijkm} + w_{ji} c_{ijmk}) z_{ik} z_{jm}, \quad (1)$$

$$\text{Sujeito à: } \sum_k z_{ik} = 1 \quad \forall i \in N, \quad (2)$$

$$z_{ik} \leq z_{kk} \quad \forall i \neq k \in N, \quad (3)$$

$$z_{ik} \in \{0, 1\} \quad \forall i, k \in N. \quad (4)$$

A equação (1) representa a minimização dos custos variáveis de transporte e custos fixos de instalação. As restrições descritas por (2) garantem que cada nó esteja conectado a exatamente um concentrador. As restrições representadas por (3) permitem a alocação de um nó i ao nó k somente se k for um concentrador. As equações 4 representam as restrições de integralidade das variáveis z_{ik} .

3. ALGORITMO GENÉTICO APLICADO NO PROJETO DE REDES EIXO-RAIO

3.1. REPRESENTAÇÃO DA SOLUÇÃO

O indivíduo do AG proposto é representado através de dois vetores que possuem tamanho igual à $|N|$. Sendo um vetor de concentradores utilizando a codificação binária e um vetor de atribuição que utiliza a codificação inteira. No vetor de concentradores, o valor 1 indica que o nó é um concentrador ou não (valor 0). A codificação inteira é responsável por representar o vetor de atribuição dos nós aos concentradores, sendo que se nó i é atribuído ao concentrador k , então a posição i do vetor tem valor k . A Figura 2 exemplifica a representação das soluções.

| | |
|----------------|-------------------|
| Concentradores | 0 1 0 0 0 1 0 0 0 |
| Alocação | 6 2 6 2 2 6 6 2 2 |

Figura 2: Representação dos indivíduos.

Nesse exemplo, o indivíduo possui 9 nós, sendo que os nós 2 e 6 são concentradores e os demais são nós não concentradores. Lembrando que um concentrador é associado a ele mesmo, a associação dos nós aos concentradores se dá seguinte forma: os nós 1, 3 e 7 são atribuídos ao concentrador 6, e os nós 4, 5, 8 e 9 são associados ao concentrador 2.

3.2. CONSTRUÇÃO DA POPULAÇÃO INICIAL

Na fase de construção da população do AG implementado, cada indivíduo, inicialmente, possui somente um concentrador diferente instalado. Essas soluções, então, são submetidas a um procedimento de melhoria, em que um novo concentrador é adicionado à solução até que o custo total, representado pela equação (1), dessa mudança não resulte em melhoria.

Considere a geração de um conjunto S de soluções viáveis, de forma que $s \in S$ possua uma estrutura de alocação $s(j) = k$, ou seja, nó $j \in N$ é atribuído ao concentrador $k \in N$, e dois conjuntos O_s e C_s representando, respectivamente, os nós concentradores e nós não concentradores da solução. Esse conjunto S de soluções é submetido ao procedimento de melhoria.

Nesse procedimento de melhoria, para cada nó não concentrador $j \in C_s$ de uma solução $s \in S$, calcula-se o custo marginal de redução no custo total β_j quando esse nó é configurado como concentrador e os demais nós são realocados ao concentrador mais próximo. O custo marginal é computado pela diferença de custo da solução depois e antes do procedimento. Percebe-se que caso $\beta_j \geq 0$ a configuração do nó j como concentrador resulta em aumento no custo total, assim o nó j é adicionado à lista NL , que contém os nós descartados como candidato a concentrador da solução em questão.

Após o cálculo de cada β_j , o custo marginal de redução máximo (β_{max}) e mínimo (β_{min}) são determinados entre os nós $j \notin NL$. O próximo passo é a determinação de uma lista restrita L de nós candidatos a concentrador segundo a seguinte regra: $L = \{j \in C_s : \beta_j \leq \beta_{min} + \lambda(\beta_{max} - \beta_{min})\}$, onde $\lambda \in [0, 1]$ representa um parâmetro que controla o nível de aleatoriedade no processo de construção e o tamanho de L . Assim sendo, se existe candidato em L , um nó $l \in L$ é escolhido de forma aleatória para se tornar concentrador ($O_s \leftarrow O_s + \{l\}$ e $C_s \leftarrow C_s - \{l\}$), e os demais nós não concentradores são realocados ao concentrador mais próximo.

Portanto, para cada solução $s \in S$, nós não concentradores são selecionados através de um processo aleatório iterativo até que $L = \emptyset$. Ressaltando que, em cada iteração, são computados, novos valores de β_j para todo nó não concentrador $j \in C_s$, β_{max} e β_{min} , bem como monta-se novas listas L e NL . O Algoritmo 1 apresenta um pseudo-código do procedimento de melhoria.

Algoritmo 1: Procedimento de melhoria do método de construção.

Entrada: Solução (s), Conjunto de concentradores (O_s), Conjunto de não concentradores (C_s), Parâmetro de controle (λ)

Saída: Solução (s)

L a lista de nós candidatos a concentrador

NL a lista de nós não candidatos a concentrador

β_j custo marginal de redução de configurar j como concentrador

β_{max} e β_{min} , respectivamente, os custos marginais máximo e mínimo

d_{ij} a distância entre nós i e j

início

enquanto $L \neq \emptyset$ **faça**

para $j \in C_s$ e $j \notin NL$ **faça**

 Compute β_j

se $\beta_j > 0$ **então**

$NL \leftarrow NL + j$

fim se

fim para

 Determine β_{max}, β_{min} , considerando $j \in C_s, j \notin NL$

$L = \{j \in C_s : \beta_j \leq \beta_{min} + \lambda(\beta_{max} - \beta_{min})\}$

se $L \neq \emptyset$ **então**

 Selecione aleatoriamente $l \in L$

 Faça $O_s \leftarrow O_s + \{l\}$ e $C_s \leftarrow C_s - \{l\}$

para $j \in C_s$ **faça**

$s(j) \leftarrow \operatorname{argmin}_{k \in O_s} \{d_{jk}\}$

fim para

fim se

fim enquanto

 retorne s

fim

3.3. OPERADOR DE SELEÇÃO

O operador de seleção escolhe indivíduos para cruzamento, mutação e reprodução através do torneio binário. Nesse operador dois indivíduos são selecionados para competir entre si, sendo que o mais bem adaptado possui maior probabilidade ($p_{melhor} = 0.8$) de ser escolhido. Além disso, o conceito de elitismo é aplicado durante o processo evolucionário, sendo o melhor indivíduo de cada geração inserido na população da geração seguinte.

3.4. OPERADOR DE CRUZAMENTO

No algoritmo proposto, foi implementado o operador de cruzamento *multi-parent* utilizando três pais. Esse operador foi desenvolvido com intuito de proporcionar maior variabilidade genética na população. Para isso, três pais (P1, P2 e P3) são sele-

cionados para gerar três filhos (C1, C2 e C3). No cruzamento, dois pontos de corte, $r \in \{1, \dots, \lfloor |N|/2 \rfloor\}$ e $t \in \{\lfloor |N|/2 \rfloor, \dots, |N|\}$ gerados aleatoriamente, são aplicados nos vetores de concentradores e alocação. Assim, a construção dos filhos é feita da seguinte forma: C1, C2 e C3 recebem os primeiros r genes de P2, P1 e P3; enquanto os próximos $t - r$ genes são herdados de P1, P3 e P2; e finalmente o restante $|N| - t$ genes são obtidos de P3, P2 e P1, respectivamente. Esse processo é seguido por uma fase de ajuste em que indivíduos inviáveis são corrigidos. Nessa fase de ajuste, se um nó é atribuído a um nó não concentrador, então o mesmo é realocado ao concentrador mais próximo. A Figura 3 ilustra esse operador.

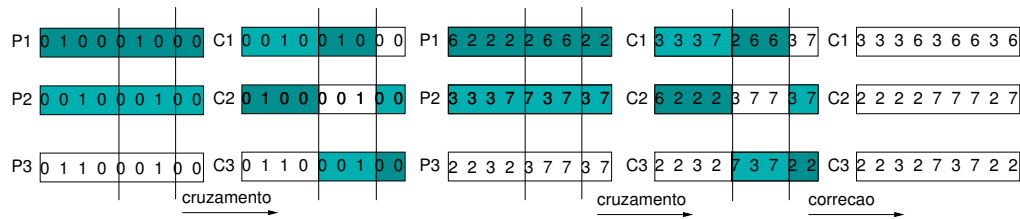


Figura 3: Cruzamento *multi-parent*.

Nessa figura, os pais P1, P2 e P3 possuem como concentradores, respectivamente, os nós $\{2, 6\}$, $\{3, 7\}$ e $\{2, 3, 7\}$. Já os filhos gerados C1, C2 e C3 possuem como concentradores, respectivamente, os nós $\{3, 6\}$, $\{2, 7\}$ e $\{2, 3, 7\}$. Porém, nota-se que os filhos C1 e C2 precisaram passar pelo procedimento de ajuste, pois após o cruzamento possuíam nós associados a nós não concentradores. Como exemplo considere o filho C1, os nós 4, 5 e 9 estão atribuídos aos nós 2 e/ou 7 que não são concentradores da solução em questão. Fato semelhante acontece com o filho C2, em que os nós 1, 5 e 8 estão alocados aos nós 3 e/ou 6 que não são concentradores desse indivíduo. Portanto, foi necessário que esses filhos gerados passassem pelo processo de correção descrito anteriormente.

3.5. OPERADORES DE MUTAÇÃO

Nesse artigo, foram propostos 5 operadores de mutação: deslocamento de alocação, troca de alocação, troca de função, inserir concentrador e remover concentrador. A Figura 4 exemplifica esses operadores.

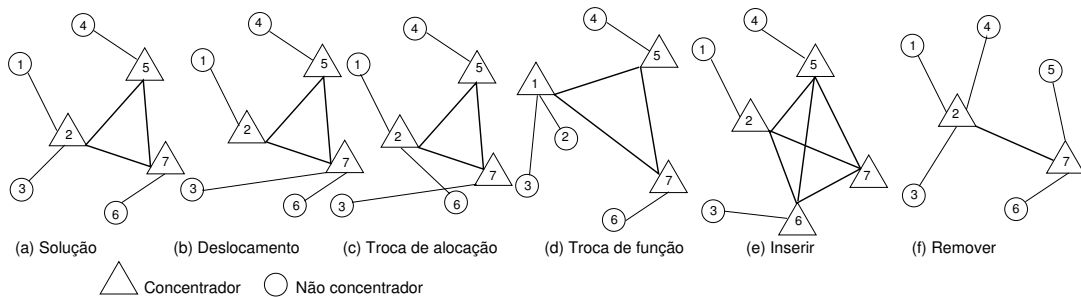


Figura 4: Operadores de mutação.

Nesse exemplo, a parte (a) representa uma solução em que os operadores de mutação são aplicados. A parte (b) mostra a mutação deslocamento de alocação que troca as alocações dos nós entre os concentradores. Um nó não concentrador (3) é escolhido aleatoriamente e realocado a outro concentrador (7), também selecionado de forma aleatória. A parte (c) ilustra a mutação troca de alocação que permuta os concentradores (2 e 7) de dois nós não concentradores (3 e 6), selecionados aleatoriamente. A parte (d) exemplifica a mutação troca de função que objetiva mudar quais serão os concentradores da rede. Para isso, escolhe-se de forma aleatória um concentrador (2) para ser transformado em não concentrador e um nó não concentrador (1), alocado a esse concentrador, para se tornar concentrador. Nos dois operadores restantes, inserir e remover concentrador, o objetivo é variar o número de concentradores de uma solução. Na mutação inserir, parte (e), um nó não concentrador (6) é escolhido aleatoriamente e promovido à concentrador, reciprocamente na mutação remover, parte (f), um concentrador (5) é também selecionado de forma aleatória para ser rebaixado à nó não concentrador. É importante ressaltar que, dada a ocorrência da mutação, uma das operações propostas é escolhida, condicionadas a uma probabilidade, para ser aplicada no indivíduo.

O Algoritmo 2 apresenta um pseudo-código do AG implementado.

4. RESULTADOS

Com o objetivo de obter o desempenho do AG proposto, nomeado GGA, realizou-se um estudo comparativo com outros três AGs considerados estado da arte na literatura, Topcuoglu *et al.* [13] (GAI), Cunha e Silva [3] (GAII), e Naeem e Ombuki-Berman [16] (GAIII), em relação a melhor solução encontrada e a fase de construção inicial. Todos os algoritmos foram implementados em C++, e os experimentos foram executados em um computador com processador Core 2 Duo 2.2 GHz, 8 GB de memória RAM e sistema operacional M S Windows 7.

Algoritmo 2: Pseudo-código do GGA.

Entrada: Tamanho da População (μ), taxa de cruzamento (p_{cruz}), taxa de mutação (p_{mut}), probabilidade de mutação deslocamento de alocação (p_s), probabilidade de mutação troca de alocação (p_e), probabilidade de mutação troca de função (p_{troca}), probabilidade de mutação remover (p_r), probabilidade de mutação inserir (p_{ins}), critérios de parada (Q)

Saída: melhor indivíduo (s^*)

início

Inicialize a população com μ indivíduos

enquanto não Q faça

Avalie a fitness da população

Determine $s^* = \max(\text{fitness})$

Selecione μ indivíduos por torneio binário

se $random \leq p_{cruz}$ então

| Aplique cruzamento multi-parent

fim se

se $random \leq p_{mut}$ então

| Aplique mutação (deslocamento de alocação p_s , troca de alocação p_e , troca de função p_{troca} , remover p_r , inserir p_{ins})

fim se

Atualize a população com elitismo

fim enquanto

retorne s^*

fim

Tabela 1: Qualidade da fase de construção.

| Método | Instâncias AP | | | |
|--------|---------------|--------|--------|--------|
| | GAI | GAI | GAI | GGA |
| Score | 87 | 126 | 62 | 0 |
| #Best | 9 | 11 | 10 | 56 |
| DevMin | 0,0895 | 0,2108 | 0,0797 | 0,0000 |
| DevMed | 0,1646 | 0,3806 | 0,1493 | 0,0029 |

Nos experimentos, utilizou-se uma base dados padrão da literatura, Australian Post (AP), introduzida por Ernst *et al.* [17]. Essa base de dados possui instâncias de $|N| = \{10, 20, 30, \dots, 100, 130, 150, 170, 200\}$ e fator de desconto $\alpha = \{0, 2; 0, 4; 0, 6; 0, 8\}$ (economia de escala). Portanto, os 56 problemas teste gerados são representados por $APN - \alpha$, onde N representa o número de nós de demanda e α é o fator de desconto. Cada um dos problemas teste foi executado 30 vezes por cada algoritmo utilizando diferentes sementes geradoras de números aleatórios. Entretanto, em cada execução, todos algoritmos utilizaram a mesma semente.

Após uma série de testes empíricos, os parâmetros do GGA foram configurados com $\mu = 200$ indivíduos, $\lambda = 0,05$, $p_{cruz} = 0,9$, $p_{mut} = 0,4$, $p_{ins} = 0,1$, $p_r = 0,1$, $p_{troca} = 0,3$, $p_e = 0,25$ e $p_s = 0,25$. Os parâmetros dos demais AGs são configurados conforme proposto pelos respectivos autores. Além disso, os critérios de parada utilizados são tempo limite de $|N|$ segundos ou obtenção de uma solução alvo.

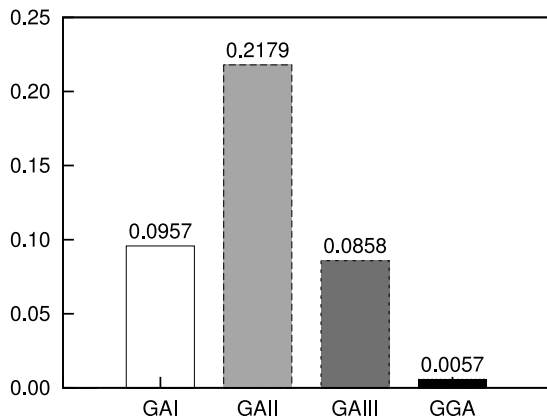
Nos experimentos as seguintes métricas foram computadas: BestValue, DevMed, DevMin, #Best e Score. BestValue é a melhor solução encontrada entre todos os métodos considerados para uma dada instância. Dessa forma, calcula-se, para cada método, a porcentagem relativa de desvio entre a melhor solução obtida pelo mesmo e o BestValue da instância em questão. Considerando todas as execuções de cada método, DevMin e DevMed representam, respectivamente, a média dos mínimos e a média das médias desse desvio. Além disso, #Best denota o número de instâncias em que o BestValue foi encontrado pelo método. Outra estatística reportada é o Score de um algoritmo, que é o número de métodos que obtiveram melhor resultado que o algoritmo em questão. Portanto, o Score máximo que um algoritmo pode obter é o número de métodos com os quais está sendo comparado multiplicado pelo número de problemas teste. Ressaltando que, na Tabela 1 e na Tabela 2, os cálculos foram realizados considerando BestValue, respectivamente, como melhor valor encontrado na fase de construção dos algoritmos e melhor valor encontrado no fim do processo evolucionário.

A Tabela 1 apresenta a qualidade da fase de construção dos AGs. Note que GGA obteve melhor resultado que os demais métodos em todas as métricas, fornecendo o BestValue da fase de construção em todas instâncias. Além disso, o método de construção do GGA obtém DevMed bem menor que DevMin dos outros métodos, comprovando a robustez e qualidade da estratégia de construção proposta. A Figura 5(a) reporta o erro percentual médio da melhor solução inicial de cada método para o BestValue. Percebe-se que, a fase de construção do GGA claramente supera GAI, GAI e GAI fornecendo um erro médio pequeno em relação ao BestValue após o processo evolucionário, chegando, em média, a pouco mais de 5% do BestValue

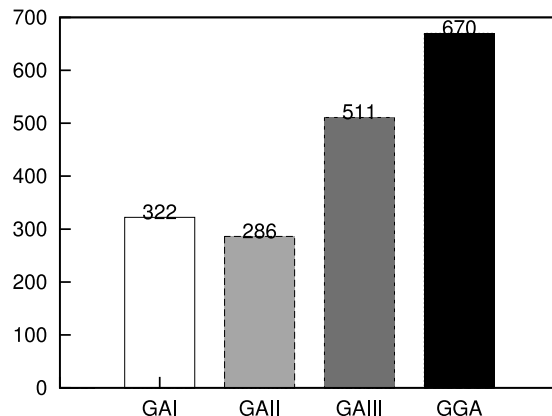
Tabela 2: Eficiência dos métodos.

| Method | AP instances | | | |
|-------------|--------------|--------|--------|--------|
| | GAI | GAI | GAI | GGA |
| Score | 76 | 135 | 32 | 6 |
| #Best | 17 | 11 | 24 | 50 |
| DevMin | 0,0162 | 0,2136 | 0,0011 | 0,0001 |
| DevMed | 0,0514 | 0,3774 | 0,0023 | 0,0013 |
| Ganho Médio | 0,0737 | 0,0039 | 0,0846 | 0,0055 |

encontrado.



(a) Erro percentual médio da melhor solução para o Best-Value.



(b) Número de vezes que o BestValue é encontrado durante os experimentos.

Figura 5: Comparação de performance do GGA com GAI, GAI e GAI.

A Tabela 2 reporta a eficiência de cada método em relação a melhor solução encontrada durante o processo evolucionário. Mais uma vez o GGA obteve soluções superiores, apresentando melhores valores para todas as métricas consideradas, exceto ganho médio. O baixo valor de ganho médio é devido à alta qualidade da fase de construção do GGA que fornece soluções próximas do BestValue. Adicionalmente, o AG implementado é capaz de encontrar o BestValue em 50 de 56 instâncias, tendo também DevMin e DevMed muito menores que os outros AGs. Na Figura 5(b) é apresentado o número de vezes em que o BestValue é encontrado pelo método considerando as 30 execuções das 56 instâncias. Nota-se que o GGA é mais robusto que os demais métodos, sendo capaz de alcançar o BestValue em mais vezes.

Finalmente, a Figura 6 ilustra a distribuição de probabilidade acumulada dos algoritmos GGA e GAI, os 2 melhores entre os métodos comparados, encontrarem uma solução alvo (média das médias das melhores soluções) para a instância $|N| = 100$ com $\alpha = 0, 2$, depois de 200 execuções de cada método. Nessa figura, percebe-se que o tempo requisitado pelo GGA para encontrar o alvo é muito menor que o GAI, mostrando que o algoritmo proposto, além de obter soluções de melhor qualidade, também é mais eficiente em relação a tempo para encontrar uma determinada solução alvo.

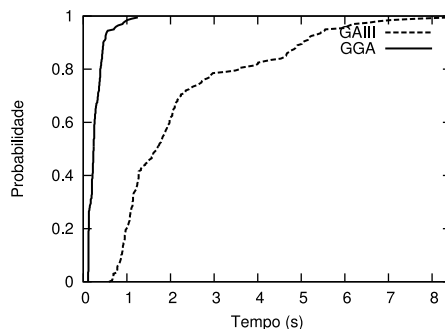


Figura 6: Probabilidade acumulada de encontrar solução alvo em determinado tempo.

5. CONCLUSÃO

Nesse artigo foi proposto um eficiente AG para projeto de redes E-R com alocação simples. A estratégia de inicialização da população implementada se mostrou extremamente eficiente, e claramente supera a construção dos demais AGs. Além disso, os operadores desenvolvidos se mostraram capazes de melhorar as soluções iniciais durante o processo evolucionário. Os resultados mostram que o AG implementado claramente supera os outros AGs da literatura, tanto em qualidade de soluções quanto em tempo de processamento para obtenção de uma solução alvo.

6. AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo suporte financeiro aos projetos: 133550/2010-9, 306910/2006-3 e 305446/2010-0; a Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG), pelo suporte financeiro ao projeto: Pronex APQ 01075/09; e ao apoio financeiro da CAPES - Brasil.

REFERÊNCIAS

- [1] T. Aykin. "Network policies for hub-and-spoke systems with applications to the air transportation system." *Transportation Science*, vol. 29, pp. 201–221, 1995.
- [2] S. Chung, Y. Myung and D. Tcha. "Optimal design of a distributed network with a two-level hierarchical structure." *European Journal of Operational Research*, vol. 62, pp. 105–115, 1992.
- [3] C. B. Cunha and M. R. Silva. "A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil." *European Journal of Operational Research*, vol. 179, pp. 747–758, 2007.
- [4] J. G. Klincewicz. "Hub location in backbone/tributary network design: a review." *Location Science*, vol. 6, pp. 3–7–335, 1998.
- [5] S. Abdinour-Helm. "Using simulated annealing to solve the p-hub median problem." *International Journal of Physical Distribution & Logistic Management*, vol. 31 (3), pp. 203–220, 2001.
- [6] S. Alumur and B. Y. Kara. "Network hub location problems: The state of the art." *European Journal of Operational Research*, vol. 190, pp. 01–21, 2008.
- [7] J. F. Campbell, A. T. Ernst and M. Krishnamoorthy. *Facility location: Applications and theory*. Drezner, Z., Hammacher, H., 2002.
- [8] M. E. O'Kelly. "A geographer's analysis of hub-and-spoke networks." *Journal of Transport Geography*, vol. 3 (6), pp. 171–186, 1998.
- [9] M. E. O'Kelly. "A quadratic integer program for the location of interacting hub facilities". *European Journal of Operational Research*, vol. 32, pp. 393–404, 1987.
- [10] J. F. Campbell. "Integer programming formulations of discrete hub location problems." *European Journal of Operational Research*, vol. 72, pp. 387–405, 1994.
- [11] S. Abdinour-Helm and M. A. Venkataramanan. "Solution Approaches to hub location problems." *Annals of Operations Research*, vol. 78, pp. 31–50, 1998.
- [12] S. Abdinour-Helm. "A hybrid heuristic for the uncapacitated hub location problem." *European Journal of Operational Research*, vol. 2-3(106), pp. 489–499, 1998.
- [13] H. Topcuoglu, F. Corut, M. Ermis and G. Yilmaz. "Solving the uncapacitated hub location problem using genetic algorithms." *Computers and Operations Research*, vol. 32 (4), pp. 967–984, 2005.
- [14] J. F. Chen. "A hybrid heuristic for the uncapacitated single allocation hub location problem". *Omega*, vol. 35, pp. 211–220, 2007.
- [15] M. R. Silva and C. B. Cunha. "New simple and efficient heuristics for the uncapacitated single allocation hub location problem." *Computers and Operations Research*, vol. 36 (12), pp. 3152–3165, 2009.
- [16] M. Naeem and B. Ombuki-Berman. "An Efficient Genetic Algorithm for the Uncapacitated Single Allocation Hub Location Problem". In *IEEE Congress on Evolutionary Computation*, 2010.
- [17] A. T. Ernst and M. Krishnamoorthy. "Efficient algorithms for the uncapacitated single allocation p-hub median problem." *Location Science*, vol. 4, pp. 139–154, 1996.