



Um estudo de caso abordando o problema de localização de concentradores com alocação simples em um estado brasileiro

Rodrigo F. Brito¹

Bruno N. Gomes^{1,2}

Aion A. F. Silva¹

contato@rodrigobrito.net

bruno.nonato@ifmg.edu.br

aion.silva@ifmg.edu.br

¹Instituto Federal de Minas Gerais - IFMG

Av. Serra da Piedade, 299 - Morada da Serra. Sabará / MG

²Universidade Federal de Minas Gerais - UFMG

Av. Presidente Antônio Carlos, 6627 - Pampulha. Belo Horizonte / MG

Abstract. *This paper addresses the hubs-and-spoke networks (HS) that are network structures allocating points of origin and destination indirectly by hubs. This model is widely used in carrier services, where the cargos are allocated to a central that concentrates the demand flow and redistribute it to another central. This problem is known NP-Hard, so a heuristic method was proposed for solve it. This method is composed by an efficient construction phase and four neighborhood strctures explored in local searches and pertubations. The results reported in the computational experiments shown that the proposed heuristic outperforms other method, considered to be state-of-the-art of the literature, both in solution quality and CPU time. The implemented algorithm was also applied to a case study considering the cities of a brazilian state. Results showed that the method was able to design networks applicable in pratice.*

Keywords: *Hub-spoke networks, Combinatorial optimization, Heuristic.*

1 INTRODUÇÃO

As redes estão presentes no cotidiano das pessoas, elas são aplicadas de diversas formas buscando interligar entidades para compartilhar recursos, transmitir dados ou até mesmo organizar logicamente sistemas de transporte. Existem uma ampla variedade de setores industriais e tecnológicos onde as redes podem ser aplicadas, mostrando assim sua grande importância de investimento e estudo (Alumur and Kara, 2008).

Por possuir uma ampla aplicação, diversas configurações de redes e topologias foram criadas, através delas que serão definidos como as redes serão interligadas, tanto em aspecto físico como lógico.

Uma possível configuração de rede é a do tipo malha (*mesh*), tal topologia interliga todos os nós definidos formando um grafo completo. Na maioria das vezes tal formato se torna inviável devido ao alto custo de implementação, principalmente em situações que envolvam problemas de grandes dimensões. Desta forma, o modelo de redes eixo-raio (ER) vem sendo amplamente abordado por ser uma alternativa de configuração mais econômica na construção de redes.

De modo geral, a rede ER é uma topologia onde a conexão entre os pares de origem e destino não é feita diretamente, e sim por pontos de transbordo, conhecidos como concentradores. (O’Kelly and Miller, 1994). Com isso, as redes do tipo ER reduzem significativamente o custo de instalação e gerenciamento da rede, além de permitir o usufruto da economia de escala ao consolidar os fluxos nos concentradores.

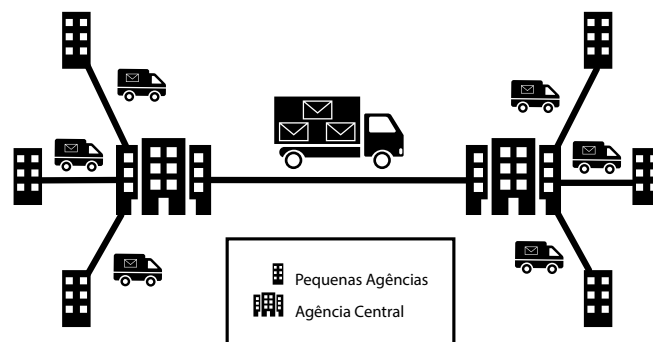


Figura 1: Redes ER - Sistema de correspondências

Um exemplo de aplicação de redes ER é o sistema de entrega de correspondências brasileiro. Uma correspondência não é enviada diretamente entre duas agências, geralmente a demanda de uma certa região é agregada a uma determinada agência central, que se encarrega de distribuir para outras centrais, a qual alocam as agências destinatárias. Essa estratégia de agregação do fluxo de demanda proporciona economia às empresas, uma vez que as mesmas enviam maior fluxo de encomendas em conjunto por meio de um transporte único e mais econômico, assim como mostrado na Fig. 1.

O objetivo da construção da rede ER é substituir o grande número de ligações diretas por um número menor de ligações indiretas através de nós concentradores. Os concentradores podem ser representados de diversas formas de acordo com o contexto que estiver inserido. Eles podem ser encontrados como *hubs*, depósitos, servidores, centrais, entre outros. Eles são encarregados

de agregar e distribuir a demanda dos demais nós da rede, que podem ser representados em um meio contextualizado como clientes, terminais, destinatários, entre outros.

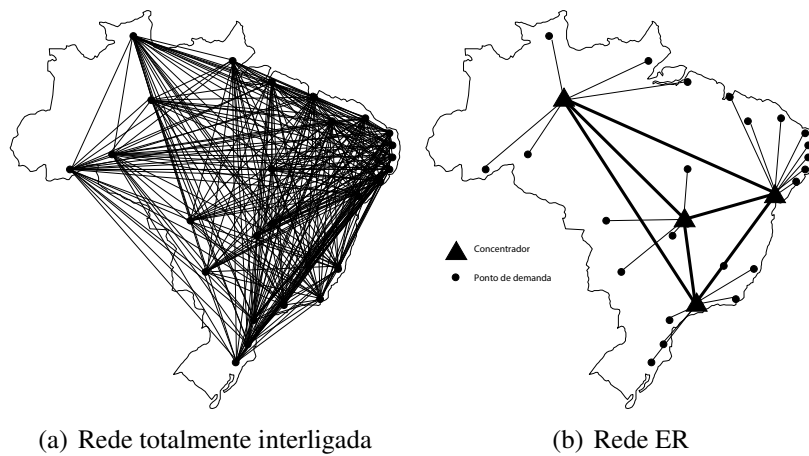


Figura 2: Exemplos de topologias de redes Camargo (2007).

A Fig. 2 ilustra outro exemplo de aplicação das redes do tipo ER, onde (a) representa uma rede malha, na qual todos os nós são interligados entre si, e em (b) tem-se a rede do tipo ER tradicional. Pode-se ver claramente que redes do tipo ER possuem menos conexões em relação a rede malha. Além de reduzir muitas conexões, o concentrador obriga que todo o fluxo passe por ele antes de alcançar o destino. A agregação de fluxo em um nó concentrador proporciona utilização do transporte de maneira mais eficiente, transmitindo maior volume de carga nas conexões entre nós concentradores, consequentemente reduzindo o custo de transporte por unidades.

O restante do artigo está organizado da seguinte maneira: a seção 2 é feita a revisão da literatura, na seção 3 apresenta a modelagem matemática do problema; a heurística proposta é detalhada na seção 4; os resultados computacionais são reportados na seção 5; a seção 6 mostra um estudo de caso considerando municípios de um estado brasileiro; e por fim, as conclusões e considerações finais são descritas na seção 7.

2 REVISÃO DA LITERATURA

As redes ER, também conhecida como Problema de Localização de Concentradores (PLC), foi introduzido por Goldman (1969) e desde então vem sendo amplamente estudada. O PLC possui diversas versões de abordagem, a qual envolvem restrições de fluxo, número de concentradores, custo fixo de instalação, entre outros (Aykin, 1994).

Em problemas onde não existe restrição de fluxo suportado por um concentrador e os nós não-concentradores devem ser alocados em apenas um concentrador, o PLC é conhecido como Problema de Localização de Concentradores Não Capacitado com Alocação Simples (do inglês, *Uncapacitated Single Allocation Hub Location Problem - USAHLP*).

Além da alocação simples, onde é permitido que nós de demanda sejam alocados em apenas um concentrador, também existe a alocação múltipla que permite a alocação de um nó não-concentrador a mais de um concentrador, tal problema é denominado Problema de Localização

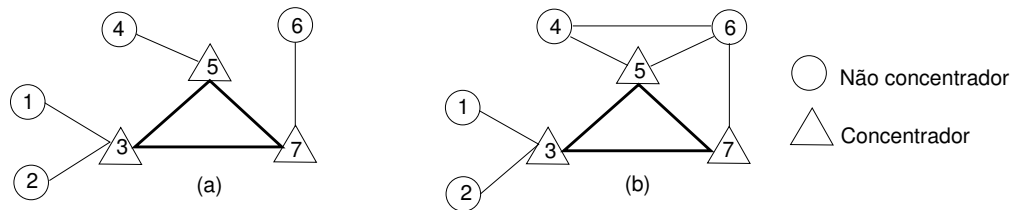


Figura 3: Variantes do PLC.

de Concentradores Não Capacitado com Alocação Múltipla (do inglês, *Uncapacitated Multiple Allocation Hub Location Problem* - UMAHLP).

Uma variação do problema que também é bastante estudada é o PLC com restrição de capacidade, onde os concentradores possuem uma quantidade máxima de fluxo que pode ser roteado, variante conhecida como Problema de Localização de Concentradores Capacitado (do inglês, *Capacitated Single Allocation Hub Location Problem* - CSAHLP).

A Fig. 3 apresenta diferentes variantes do PLC: alocação simples ou múltipla; e permissão ou não de ligação direta entre nós não concentradores. Nessa Fig., a parte (a) representa uma rede com alocação simples e sem permissão de ligação direta entre nós não concentradores; já a parte (b) exemplifica a rede com alocação múltipla e com permissão de ligação direta entre nós não concentradores.

Existem diversas variações do problema, um estudo detalhado sobre as versões do PLC pode ser encontrado na revisão de Alumur and Kara (2008), onde são citados mais de 100 trabalhos com diferentes variações do problema.

A primeira abordagem acerca do USAHLP foi realizada por O’Kelly (1987). Nesse trabalho, os autores propuseram uma modelagem matemática com função objetivo quadrática. Outras formulações matemáticas envolvendo o PLC foram trabalhadas por Campbell (1994). O autor classifica o problema de acordo com a função objetivo, englobando o custo total de transporte, o custo fixo de instalação de concentradores e também tempo total de transporte.

Por se tratar de um problema NP-Difícil, abordagens exatas se tornam inviáveis para instâncias com grandes dimensões. Nesses casos, a determinação da solução ótima pode exigir grande esforço computacional. Diante disso, diversas abordagens heurísticas tem sido propostas para resolução do USAHLP.

No trabalho de Topcuoglu et al. (2005) foi proposto um Algoritmo Genético (AG), onde o AG aplicado é responsável por localizar, definir o número de concentradores da solução e também alocar os nós não-concentradores. O método se mostrou muito eficiente, alcançando valores ótimos para algumas instâncias *Civil Aeronautics Board* (CAB) e também apresentando bons resultados para a instância *Australian Post* (AP).

Também está presente na literatura algumas aplicações práticas envolvendo o setor de transporte brasileiro. No trabalho de Silva and Cunha (2004), os autores propuseram uma aplicação de AG para o USAHLP em uma empresa de transporte de cargas fracionadas brasileira. Tal método se mostrou eficaz e viável para aplicação na tomada de decisões da transportadora, possuindo tempo de processamento de 163 segundos no pior caso computado.

No trabalho de Chen (2007), os autores propuseram uma heurística híbrida baseada em *Simulated annealing* (SA) e Busca Tabu (BT). O algoritmo se mostrou eficiente para instâncias

de pequeno porte, superando os resultados do trabalho de Topcuoglu et al. (2005) em tempo computacional e também em qualidade de soluções.

No trabalho de Cunha and Silva (2007) foi proposta uma heurística híbrida composta por AG e SA. O algoritmo foi aplicado à base de dados de uma companhia de caminhões brasileira com rede de 46 nós configurados. O trabalho proporcionou bons resultados e ajudou a empresa no processo de análise do transporte para redução de custos. Tal algoritmo foi considerado um dos métodos de melhor eficiência para o problema de USAHLP de acordo com Alumur and Kara (2008).

Silva and Cunha (2009) desenvolveram 3 variantes da heurística BT com *multistart*. Os autores testaram os algoritmos em instâncias padrão da literatura e obtiveram resultados satisfatórios que superam os demais métodos desenvolvidos até então.

Mais recentemente Gomes et al. (2013) propôs uma abordagem evolucionária composta de um eficiente processo construtivo, novos operadores de cruzamento e mutação, além de um procedimento de busca local. Tal método mostrou-se muito eficaz aplicado às instâncias CAB e AP, superando 4 outros métodos evolucionários considerados estado da arte na literatura.

Uma revisão mais detalhada pode ser encontrada nos trabalhos de Alumur and Kara (2008) e Farahani et al. (2013), onde os autores realizaram estudos sobre o estado da arte dos principais trabalhos desenvolvidos até o momento.

O presente artigo restringi-se ao estudo do USAHLP com custo fixo de instalação dos nós concentradores. Motivado pela dificuldade de resolução do problema, propõe-se uma heurística composta por uma fase de construção baseada na metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) e 4 estruturas de vizinhanças exploradas em buscas locais e perturbações. O método implementado foi testado em instâncias padrão da literatura e comparado com a heurística de Gomes et al. (2013). Resultados mostraram a superioridade do algoritmo desenvolvido em qualidade de soluções e tempo de processamento. Além disso, o método implementado foi submetido à instâncias que simulam uma aplicação real com os municípios do estado de Minas Gerais. Nesse caso, a heurística proposta obteve soluções boas na prática, uma vez que distribui os concentradores igualitariamente por todo espaço territorial do estado.

3 FORMULAÇÃO MATEMÁTICA

O modelo matemático proposto por Skorin-Kapov et al. (1996) é baseado nas seguintes definições: N é o conjunto de n nós de uma rede, K é o conjunto de nós concentradores configurados, onde o parâmetro O_i é o fluxo total de demanda originado em $i \in N$ e D_i é o fluxo total de demanda destinado a $i \in N$. O custo de transporte por unidade de demanda com origem em $i \in N$ com destino a $j \in N$ é dado por c_{ij} . A variável x_{ijkm} representa a existência de um caminho entre um nó i e j , passando pelos concentradores $k \in K$ e $m \in K$. O custo total entre dois concentradores $k, m \in K$ de um caminho definido por x_{ijkm} são submetidos a um fator de desconto, $0 \leq \alpha \leq 1$, que representa a economia de escala nas conexões entre os concentradores. O custo fixo da instalação de um concentrador no nó k é dado pelo parâmetro f_k . A variável de decisão $z_{ik} \in \{0, 1\}$ indica a localização dos concentradores e a alocação dos nós aos concentradores instalados. Assim, 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 $z_{kk} = 0$.

$$\text{Minimize } \sum_k f_k z_{kk} + \sum_i \sum_k (O_i + D_i) c_{ik} z_{ik} + \sum_i \sum_j \sum_k \sum_m \alpha w_{ij} c_{km} x_{ijkm} \quad (1)$$

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

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

$$\sum_m x_{ijkm} = z_{ik}, \quad \forall i, j \in N, k \in N \quad (4)$$

$$\sum_m x_{ijkm} = z_{jm}, \quad \forall i, j \in N, m \in N \quad (5)$$

$$z_{ik} \in \{0, 1\}, \quad \forall i, k \in N \quad (6)$$

$$x_{ijkm} \geq 0, \quad \forall i, j \in N, k, m \in K \quad (7)$$

A equação (1) tem o objetivo de minimizar o custo total de operação da rede, composto pelo custo de transporte do fluxo de demanda e custo fixo de instalação dos concentradores. As restrições em (2) garantem que cada nó possa estar alocado a somente um concentrador. As restrições representadas por (3) permitem a alocação do nó $i \in N$ ao nó $k \in K$, somente se k for um concentrador. A restrição (4) garante que as rotas originadas em i passando primeiramente pelo concentrador $k \in K$, somente existirão se o nó i estiver alocado a k . Da mesma forma, a restrição (5) garante que as rotas com destino final em $j \in N$ passando por último pelo concentrador $m \in K$, somente existam se j estiver alocado a m . Por fim a equação (6) define a integralidade da variável z_{ik} .

O USAHLP pode ser exemplificado pela rede da Fig. 4. Essa rede é constituída de 7 nós, $N = \{1, 2, 3, 4, 5, 6, 7\}$, sendo que 2 deles (2 e 5) foram escolhidos como concentradores. A atribuição dos nós se deu da seguinte maneira: os nós 1 e 3 são associados ao concentrador instalado em 2, e o restante dos nós (4, 6 e 7) estão alocados ao concentrador instalado em 5. Dessa forma, o fluxo de demanda dos nós dessa rede, representado por w_{ij} , deve ser roteado

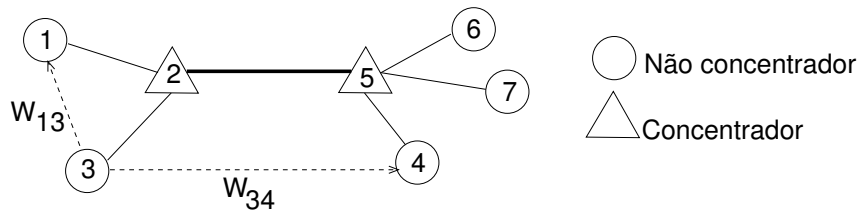


Figura 4: Comunicação nas redes eixo-raio.

pelos concentradores 2 e/ou 5. Portanto, se $z_{ik} z_{jm} = 1$, então o fluxo será transmitido pelo caminho originado em i com destino a j , sendo roteado pelos concentradores k e m . Assim, a maneira como acontece a comunicação pode ser exemplificada através dos nós 3 e 4: a demanda w_{34} primeiramente é coletada do nó origem pelo concentrador 2 por um custo de c_{32} ; após isso, o encaminhamento da mesma é feito para o concentrador 5 pelo custo de c_{25} ; finalmente, o nó 5 envia a demanda ao destino por um custo c_{54} . Portanto, o custo total de transporte de uma unidade de demanda entre os nós 3 e 4, roteada pelos concentradores 2 e 5, é composto por $c_{3425} = c_{32} + \alpha c_{25} + c_{54}$. A comunicação entre os nós 1 e 3 acontece de forma semelhante,

exceto pela inexistência de transmissão entre concentradores, já que ambos nós são associados ao mesmo concentrador (2). Assim sendo, o custo total de transporte da demanda entre os nós 1 e 3 é representada por $c_{321} = c_{32} + c_{21}$.

4 HEURÍSTICA IMPLEMENTADA

O algoritmo proposto é composto por um eficiente método construtivo e uma estrutura sistemática que explora a solução inicial através de perturbações e buscas locais em 4 estruturas de vizinhança. O Algoritmo 1 apresenta a estrutura geral do método implementado.

Algoritmo 1: Estrutura da heurística proposta

Entrada: Matriz de distâncias (C), Matriz de demanda (W), Custo Fixo de Instalação(F)

Saída: Solução S

início

$S \leftarrow$ Construir solução (S, C, W, F);

repita

$S1 \leftarrow$ VND (S, C, W, F);

se $FO(S1) < FO(S)$ **então**

$S \leftarrow S1$;

senão

$S1 \leftarrow$ VNS (S, C, W, F);

fim

$S1 \leftarrow$ Perturbação Troca de Função($S1$, C, W, F);

até Critério de parada;

Retorne S

fim

No Algoritmo 1, primeiramente a solução inicial é construída (Seção 4.1). Após isso, o processo de exploração é estruturado em repetidas iterações com perturbação na vizinhança Troca de Função (Seção 4.2) em conjunto com um processo de busca local realizado por meio da metaheurística Descida em Vizinhança Variável (VND) (Mladenovic and Hansen, 1997). O ciclo de exploração se repete enquanto resultar em melhoria, caso não haja melhora em uma das iterações, a solução é submetida a uma busca em vizinhanças variáveis (VNS) (Mladenovic and Hansen, 1997) e retorna ao processo de perturbação. O algoritmo finaliza após um número pré-estabelecido de iterações sem melhora.

4.1 Processo de Construção

A estratégia de construção de soluções é baseada no processo construtivo do GRASP (Feo and Resende, 1995). Nessa estratégia, a solução é inicializada com somente um concentrador instalado, sendo submetida a um procedimento de melhoria, em que um novo concentrador é adicionado até que o custo total dessa mudança não resulte em melhoria (Gomes et al., 2013).

A estrutura do processo construtivo é mostrado no Algoritmo 2. O processo inicia com uma solução contendo apenas um concentrador, a partir de tal configuração testa-se todas as possibilidades de inserção de um novo concentrador. A cada passo é computado a diferença entre o novo custo gerado e a solução anterior, identificado como Custo Marginal (CM).

Algoritmo 2: Processo de construção.

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

Saída: Solução (s)

L lista de nós candidatos a concentrador

NL lista de nós não candidatos a concentrador

β_j custo marginal de redução de configurar j como concentrador

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

d_{ij} distância entre nós i e j

início

enquanto $L \neq \emptyset$ **faça**

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

 Compute β_j

se $\beta_j > 0$ **então**

$NL \leftarrow NL + j$

fim

fin

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

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

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 $j \in C_s$ **fazer**

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

fin

fim

fim

 retorne s

fim

Após calcular o CM de cada possibilidade de inserção de um concentrador, duas listas são atualizadas: a lista de candidatos a inserção L e a lista de nós não candidatos NL , composta pelos nós que proporcionaram pioras quando transformados em concentrador.

Todo nó que proporcionar CM positivo, ou seja, nós que pioram a solução, será adicionado a NL e não será considerado nos próximos testes. Os demais nós que possuírem CM negativo serão filtrados para realização de um sorteio que decidirá aquele que será configurado como novo concentrador.

O processo de filtragem se resume nos seguintes passos. Inicialmente são computados o CM máximo e CM mínimo, $\forall CM \leq 0$. Após isso, computa-se a Margem de Aceitação (MA) dada pela fórmula $MA = CM_{Min} + \lambda \cdot (CM_{Max} - CM_{Min})$, onde λ é uma constante entre 0 e 1 que define o fator de aleatoriedade do processo de aceitação do CM. Com a MA definida, cria-se a lista L contendo os nós que possuírem $CM \leq MA$. Se um ou mais nós forem inseridos em L , faz-se um sorteio para definir qual nó será configurado como concentrador para tal iteração. Após configurar o novo nó como concentrador, o processo de computação de CM é realizado novamente, agora considerando a solução com o novo concentrador inserido. Esse processo

é repetido até que não se tenha mais nós selecionados em L . Tal procedimento é mostrado no algoritmo 2.

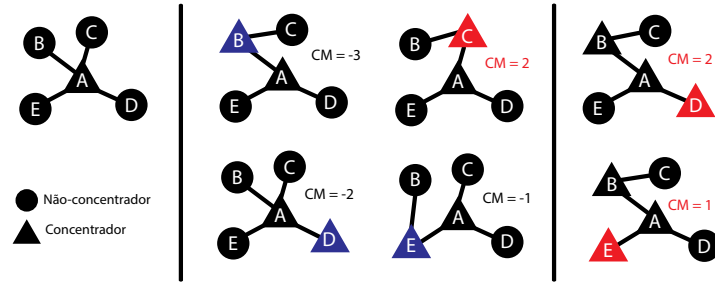


Figura 5: Construção GRASP

A Fig. 5 exemplifica o processo de construção. Inicialmente, considera-se a solução com apenas o nó A configurado como concentrador. Logo após, testa-se as possibilidades de inserir os nós não concentradores como novo concentrador da solução. Note que, ao inserir o nó C gera-se $CM > 0$, ou seja, houve um crescimento no custo de configuração, indicando um movimento de piora. Logo, tal nó será inserido na lista NL e não será testado nas próximas iterações. Depois de calculado todos CM , define-se $CM_{Min} = -3$, $CM_{Max} = -1$. Considerando $\lambda = 0.1$, tem-se $MA = -3 + 0.1[-1 - (-3)] = -2.8$. Logo, apenas o nó B terá $CM \leq MA$. Dessa forma, o nó B é inserido em L , e trivialmente sorteado para ser configurado como concentrador. Tal processo se repete, agora considerando a nova solução com nó B configurado como concentrador. Vê-se pela terceira coluna da Fig. 5 que a configuração dos nós D e E como novos concentradores não acarretou em melhoria no custo da solução ($CM > 0$), logo nenhum nó é adicionado à L e o processo é finalizado.

4.2 Estruturas de vizinhanças

A presente seção descreve as estruturas de vizinhança exploradas nas buscas locais e perturbações. Partindo de uma dada solução inicial o procedimento explora soluções vizinhas buscando melhores resultados. As próximas subseções apresenta as quatro estruturas de vizinhanças exploradas: Troca de alocação (TA), Troca de Função (TF), Insere Concentrador (IC) e Remove Concentrador (RC).

4.3 Vizinhança Troca Alocação - TA

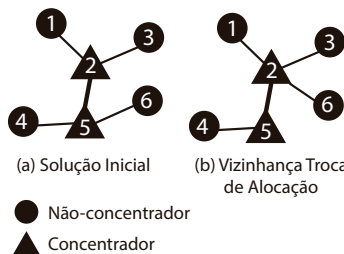


Figura 6: Exemplo de exploração da vizinhança TA

Este movimento de exploração faz modificações nas alocações dos nós não-concentradores, testando todas as possibilidades de alocação para cada nó da solução explorada. A Fig. 6 (b) exemplifica um movimento dessa vizinhança, em que é feita a troca da alocação do nó 6, inicialmente alocado ao concentrador 5, para o concentrador 2.

4.4 Vizinhança Troca Função - TF

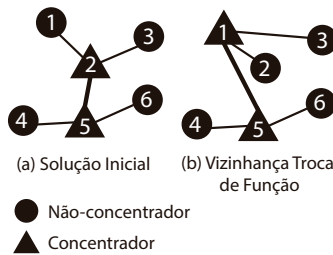


Figura 7: Exemplo de exploração da vizinhança TF

Este movimento de exploração efetua a troca de função entre um nó não-concentrador e o concentrador a qual está alocado, ou seja, os nós invertem seus papéis, passando a função de concentrador e também todas as alocações mantidas por ele. O processo testa tal movimento em todas as combinações possíveis para cada concentrador. A Fig. 7 (b) exemplifica um movimento dessa vizinhança, onde os nós 1 e 2 trocam de função.

4.5 Vizinhança Insere Concentrador - IC

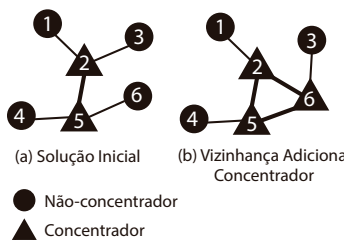


Figura 8: Exemplo de exploração da vizinhança IC

A Vizinhança IC explora o espaço de busca de uma solução através da inserção de um novo concentrador. O método faz a comparação entre as diversas possibilidades de inserção de um novo concentrador, configurando aquele que proporcionar melhor redução de custos na solução corrente. Tal movimento é exemplificado na Fig. 8 (b), onde o nó 6 passa a ser um novo concentrador da solução corrente.

4.6 Vizinhança Remove Concentrador - RC

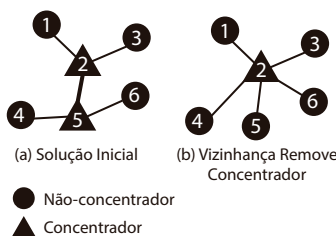


Figura 9: Exemplo de exploração da vizinhança RC

Tal movimento de exploração testa todas as possibilidades de remoção de concentradores, transformando em um nó não concentrador. Um exemplo de remoção é mostrado na Fig. 9 (b), onde o nó 5 que era um concentrador inicialmente, passa a ser um nó não-concentrador.

Todas estruturas de vizinhança são exploradas nos procedimentos VND (Seção 4.7) e VNS (Seção 4.8) desenvolvidos. Além disso, tais vizinhanças são utilizadas como perturbações em que novos vizinhos são gerados de forma aleatória para exploração do espaço de busca.

4.7 Descida em Vizinhança Variável - VND

A busca local proposta é baseado na metaheurística VND (Mladenovic and Hansen, 1997), tal método explora o espaço de soluções através de trocas sistemáticas de vizinhanças. O método analisa todas as soluções vizinhas buscando uma nova combinação que possa resultar em melhoria ao custo.

A mudança de vizinhança acontece quando uma determinada busca não encontra uma solução melhor que a de entrada, com isso uma nova vizinhança passa a ser testada. Caso aconteça um movimento de melhora, volta-se à primeira estrutura de vizinhança considerada, e inicializa-se todo o processo. A busca local finaliza quando todas as estruturas de vizinhança são exploradas sem obtenção de melhoria.

Algoritmo 3: Pseudocódigo do VND

Entrada: Solução (S), Matriz de distância (C), Matriz de demanda (W), Custo Fixo de Instalação(F), Vizinhanças (V)

Saída: Solução S

início

```

     $k \leftarrow 1$ ;
     $n \leftarrow |V|$ ;
    enquanto  $k \leq n$  faça
         $S1 \leftarrow S$ ;
         $S1 \leftarrow$  Busca Local ( $S1, V_k$ );
        se  $FO(S1) < FO(S)$  então
             $S \leftarrow S1$ ;
             $k \leftarrow 1$ ;
        senão
             $k \leftarrow k + 1$ ;
    fim

```

fim

retorna S;

fim

O Algoritmo 3 mostra a estrutura básica de funcionamento do VND, onde se tem n estruturas de vizinhanças $V = \{V_1, V_2, \dots, V_n\}$ controladas pela variável k .

4.8 Pesquisa em Vizinhança Variável - VNS

Durante o processo de busca, uma solução pode ficar presa em um ótimo local, dificultando que o VND encontre melhores soluções no espaço de busca. Por esse motivo, a cada iteração que não resulte em melhoria, a solução é submetida ao método VNS (Mladenovic and Hansen, 1997) com o objetivo de explorar o espaço de soluções através de intercalações entre estruturas de vizinhanças.

Algoritmo 4: Pseudocódigo do VNS

Entrada: Solução (S), Matriz de distâncias (C), Matriz de demanda (W), Custo Fixo de Instalação(F), Vizinhanças(V)

Saída: Solução S

inicio

```
     $k \leftarrow 1$ ;  
     $n \leftarrow |V|$ ;  
    enquanto  $k \leq n$  faça  
         $S1 \leftarrow S$ ;  
         $S1 \leftarrow$  Gerar solução vizinha ( $S1, V_k$ );  
         $S1 \leftarrow \text{VND}(S1)$ ;  
        se  $FO(S1) < FO(S)$  então  
             $S \leftarrow S1$ ;  
             $k \leftarrow 1$ ;  
        senão  
             $k \leftarrow k + 1$ ;  
    fim  
fim  
retorna S;
```

fim

O VNS é capaz de explorar vizinhanças gradativamente mais distantes da solução de entrada através de perturbações aleatórias. A perturbação é seguida por um processo de busca local responsável por explorar vizinhanças mais próximas da nova solução gerada. A solução corrente é atualizada apenas em casos de melhoria no custo total.

O Algoritmo 4 mostra a estrutura básica do VNS, tem-se n vizinhanças $V = \{V_1, V_2, \dots, V_n\}$ de movimentos aleatórios, sequencialmente controlados por k . Assim como o VND, o algoritmo é finalizado quando uma determinada solução é explorada por todas as vizinhanças e não resulta em nenhuma melhoria.

5 RESULTADOS

Com o objetivo de obter o desempenho computacional do método proposto, realizou-se um estudo comparativo com o algoritmo de Gomes et al. (2013). Os algoritmos foram comparados em relação a eficiência para encontrar a solução ótima. Cabe lembrar que a solução ótima foi obtida pelo algoritmo de Benders do trabalho de Castro (2010).

Os algoritmos foram implementados em C++, e os experimentos computacionais foram executados em um processador Intel Core i7-4510U 2.0GHz, 8GB RAM e sistema operacional Windows 8.1.

Os experimentos foram executados na base de dados Australian Post (AP) (Ernst and Krishnamoorthy, 1996) com instâncias de $|N| = \{10, 20, \dots, 100, 130, 150, 170, 200\}$ e fator de desconto $\alpha = \{0, 2; 0, 4; 0, 6; 0, 8\}$. Assim, 56 problemas teste foram gerados e nomeados por $APN - \alpha$, onde N representa o número de nós e α a economia de escala. As instâncias fo-

ram executadas 10 vezes por cada algoritmo utilizando diferentes sementes gerados de números aleatórios.

Após intenso processo de calibração, os parâmetros do algoritmo foram configurados da seguinte forma. Na fase de construção setou-se $\lambda = 5\%$. O critério de parada do algoritmo principal foi definido por um limite de 20 iterações sem melhoria. A sequência de vizinhanças utilizadas pelo VND e VNS é: $V = \{TA, TF, IC, RC\}$.

Nos experimentos, as seguintes métricas foram computadas: GapMed, GapMin e $\#Best$. Para isso, calcula-se a porcentagem relativa de desvio entre a solução ótima de cada instância e as soluções retornadas pelos métodos. Considerando todas as execuções de cada método, GapMin representa o desvio relativo entre a solução ótima e a melhor solução retornada pelo método e GapMed o desvio relativo entre a solução ótima e a média das soluções retornadas pelo método. Além disso, $\#Best$ denota o número de vezes em que a solução ótima foi encontrada pelo método para cada instância.

Tabela 1: Heurística proposta x Gomes et al. (2013)

Algoritmo	GapMin Médio	GapMed Médio	#Best	Tempo Médio (s)
Heurística Proposta	0,0039	0,0353	448	29,3119
Gomes et al. (2013)	0,1997	0,3153	211	70,1561

A Tabela 1 reporta a eficiência do método proposto em relação ao AG proposto por Gomes et al. (2013), onde GapMin Médio se refere as médias de todos os GapMin computados, assim como o Gap Médio. A heurística proposta se mostrou superior em todas as métricas apresentadas, principalmente em qualidade das soluções para instâncias de maior escala, alcançando os valores ótimos da literatura na maioria dos casos apresentados. O método desenvolvido não alcançou a solução ótima em somente 3 instâncias: AP150-6, AP150-8 e AP170-6.

Em relação à tempo computacional, a heurística proposta também foi superior ao AG para pequenas e médias instâncias, em alguns casos sendo até 10 vezes mais rápida. Para instâncias de maior escala, ambos os métodos apresentaram valores próximos.

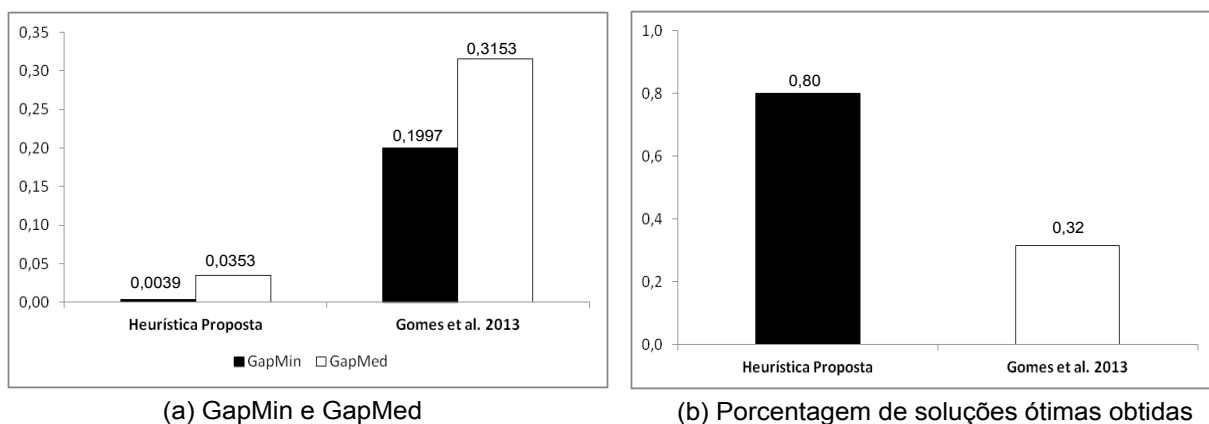


Figura 10: Heurística Proposta x Gomes et al. (2013)

A Fig. 10 (a) apresenta uma comparação acerca da média do GapMin e GapMed alcançado

pelas heurísticas. Nota-se que o algoritmo do presente trabalho possui comportamento extremamente robusto em comparação com o método da literatura, retornando média de GaMed bem menor que a média de GapMin do AG. Isso demonstra a performance efetiva e homogênea diante das 56 instâncias testadas. Afirmando de vez melhor comportamento do algoritmo proposto, a Fig. 10 (b) mostra um comparativo da capacidade das heurísticas em encontrar os valores ótimos considerando todas as execuções. Percebe-se claramente que a heurística proposta se mostrou superior, alcançando a solução ótima em 80% das 560 execuções nas instâncias testadas, ao contrário do AG que alcançou o ótimo em apenas 32% das vezes.

6 ESTUDO DE CASO - MUNICÍPIOS DE MINAS GERAIS

Uma das principais aplicações das redes ER é o sistema de transporte, sendo amplamente utilizado na literatura para experimentos computacionais. Uma das metodologias mais utilizadas é a representação dos nós como pontos cartesianos em que a distância é calculada como uma linha reta entre origem e destino. Nem sempre é possível fazer uma ligação em linha reta entre dois pontos, que é o caso do Brasil, onde a maior parte do tráfego cargueiro é feito por rodovias.

Sabe-se que muitas empresas do ramo de transporte e telecomunicações utilizam redes ER na configuração de suas redes logísticas, porém grande parte dessas empresas projetam e/ou atualizam suas redes de forma empírica. Dessa forma, podem deixar de obter ganhos significativos ao não utilizar métodos computacionais que possuem maior precisão e inteligência para realização de tal tarefa.

Motivado por esse contexto, uma simulação de aplicação real ao cenário brasileiro é proposto. Essa simulação envolve um estudo de caso com os municípios do estado de Minas Gerais, considerando dados reais de distância rodoviária e demanda baseada no Produto Interno Bruto (PIB) e população de cada município.

6.1 Coleta e geração de dados

Os dados utilizados nos testes computacionais foram coletados do sistema cartográfico do *Google Maps* (Google, 2013), através da API de matriz de distâncias, tendo assim uma ótima precisão da distância percorrida por um veículo no trajeto entre municípios. O algoritmo de rotas do *Google Maps* retorna a menor distância entre dois municípios dadas através de trajeto rodoviário, tornando-se uma base de dados prática e muito próxima de uma situação de aplicação real.

A base de dados construída possui 853 municípios pertencentes ao Estado de Minas Gerais no Brasil. As informações contidas nessa base são distância e demanda entre cada um dos municípios. Cabe ressaltar que, a demanda entre os municípios foi calculada de acordo dados de PIB e população dos mesmos presentes no Instituto Brasileiro de Geografia e Estatística (IBGE, 2011).

Minas Gerais é um dos 27 estados do Brasil, sendo o quarto maior em extensão territorial (586528 km^2) e o segundo em número de habitantes (20.73 milhões) IBGE (2011). Somente para se ter uma ideia da dimensão do estado, Fig. 11 mostra a conexão entre um município do extremo sul com outro do extremo norte que possui aproximadamente 1150 km, similarmente, a conexão entre município do oeste com município do leste possui aproximadamente 1200 km.

O estado é maior em extensão territorial e população que muitos países importantes tais como França, Holanda entre outros. Essas características mostram a dificuldade no estabelecimento de uma eficiente rede logística de transporte e a importância de se aplicar métodos computacionais para auxílio na tomada de decisão.

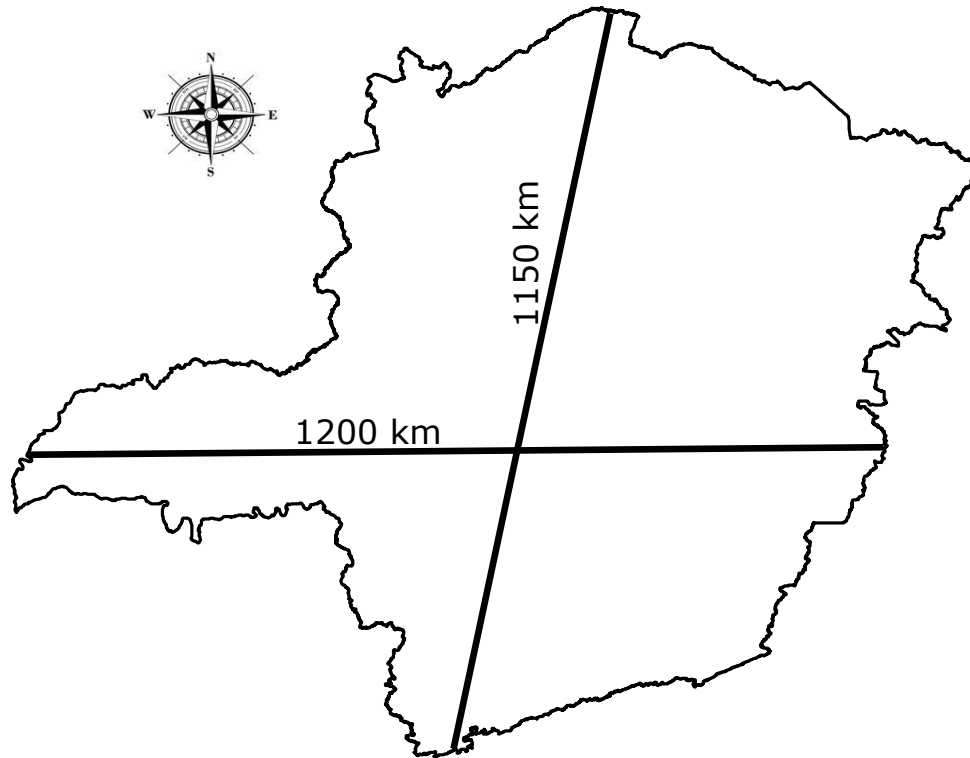


Figura 11: Minas Gerais dimensões leste-oeste e norte-sul

6.2 Resultados do estudo de caso

A presente seção descreve os principais resultados reportados da simulação da heurística proposta no estudo de caso com os municípios de Minas Gerais. Algumas instâncias foram geradas com os dados coletados. Para instâncias menores, os municípios foram selecionados em ordem decrescente de PIB, ou seja, a instância de 20 considera os 20 municípios de Minas Gerais com maior PIB, e assim sucessivamente.

A Tabela 2 reporta os resultados encontrados para as instâncias geradas. Vê-se que a medida que a dimensão da rede é aumentada, mais concentradores são instalados, proporcionando otimização da operação e custo da rede. Além disso, a instalação de mais concentradores permite melhor distribuição das instalações e alocações entre os municípios.

Nota-se que para a instância com os 853 municípios, em aproximadamente 23 horas de processamento, a heurística projetou uma rede com 9 concentradores que foram capazes de agregar e distribuir os pontos de demanda dos municípios. Considerando a aplicação e a dimensão do problema, o método retornou uma boa solução em tempo computacional viável.

A Fig. 12 mostra o resultado reportado pelo método implementado para os 853 municípios de Minas Gerais, aonde os concentradores são identificados pelos triângulos espalhados pelo

Tabela 2: Resultados para o estudo de caso

Estudo de Caso - Minas Gerais		
Cidades	Concentradores	T (s)
20	3	0,2500
50	4	4,7060
100	5	40,2480
200	6	404,7650
300	8	1802,3890
400	8	4474,0230
500	9	14355,1120
853	9	81398,7230

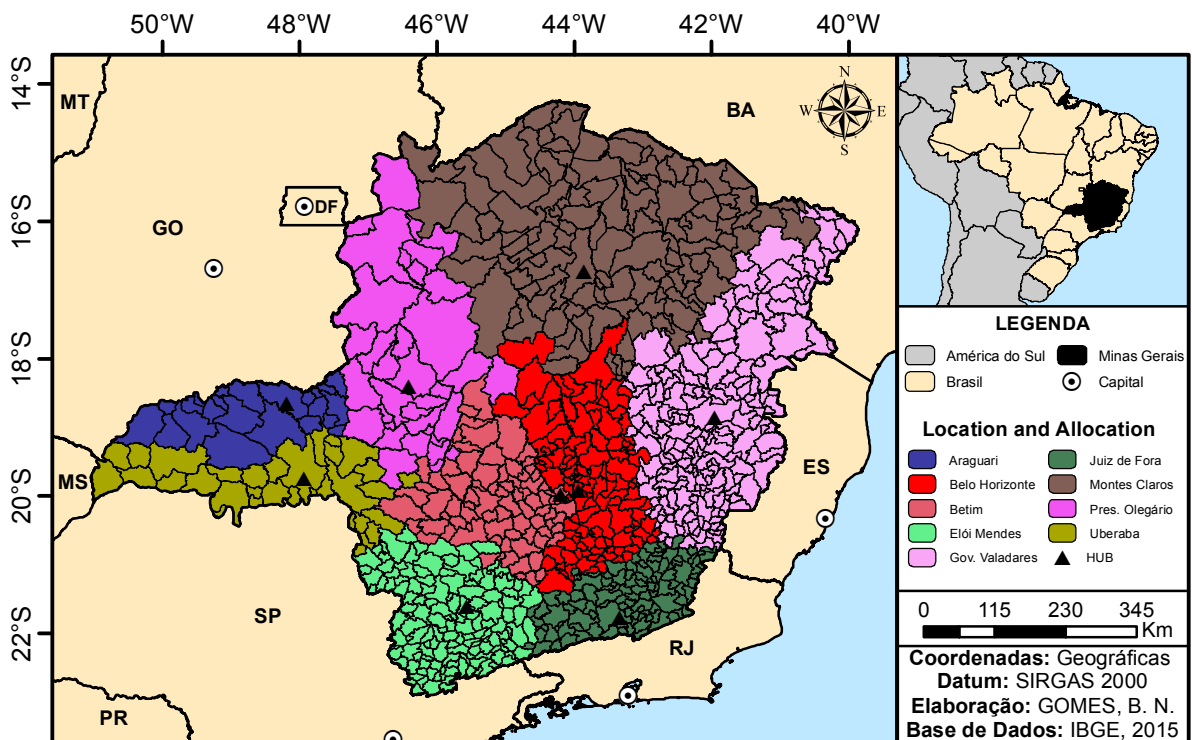


Figura 12: Heurística proposta aplicada ao estado de Minas Gerais

mapa. Os municípios alocados a determinado concentrador são identificados pelas cores definidas na legenda.

Percebe-se pela Fig. 12 que as redes projetadas pelo método possuem boa distribuição em relação à instalação dos concentradores e alocação dos municípios aos concentradores instalados. Pela distribuição das instalações e alocações, vê-se que os projetos das redes podem possuir boa aplicabilidade na prática, principalmente para instâncias de grande dimensão, sendo

uma ótima alternativa em detrimento à soluções empíricas diversas vezes utilizadas.

Muitas empresas projetam suas redes de forma empírica, portanto as redes projetadas pelo algoritmo desenvolvido nesse trabalho, demonstram boa aplicabilidade na prática e podem resultar em melhorias significativas no desempenho logístico e financeiro de empresas que atuam nesse ramo.

Um projeto bem elaborado se torna ainda mais importante, devido à pressão existente em relação à diminuição dos custos, aumento do nível de serviço prestado, horários cada vez mais apertados, além da exigência de melhoria da qualidade de vida dos motoristas e mudanças rápidas nas condições econômicas.

7 CONCLUSÕES

Nesse artigo propôs-se uma eficiente heurística iterativa para o projeto de redes ER com alocação simples. O método se mostrou muito robusto em comparação com o método de Gomes et al. (2013), superando-o tanto em qualidade das soluções quanto em tempo computacional. Além disso, o algoritmo foi capaz de alcançar em 80% das execuções o valor ótimo das instâncias AP testadas.

Uma vez comprovada a eficiência do método, o mesmo foi submetido à um estudo de caso no estado de Minas Gerais. Nesse estudo, simulou-se instâncias reais baseadas nas informações de distância, demanda e PIB dos municípios do estado. Após aplicação em diversas instâncias geradas, o algoritmo se mostrou capaz projetar boas redes, distribuindo os concentradores e alocando os municípios igualitariamente por todo o território do estado.

8 AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo suporte financeiro dado ao projeto.

Bibliografia

- Alumur, S., Kara, B.Y., 2008. Network hub location problems: The state of the art. *European Journal of Operational Research* 190, 1 – 21.
- Aykin, T., 1994. Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research* 79, 501–523.
- Camargo, R.S., 2007. Sistemas eixo-raio de múltipla atribuição: modelos e algoritmos. Ph.D. thesis. Universidade Federal de Minas Gerais.
- Campbell, J.F., 1994. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research* 72, 387 – 405.
- Castro, R.R.M., 2010. Sistemas eixo-raio de alocação simples: Modelos e Algoritmos. Master's thesis. Universidade Federal de Minas Gerais.
- Chen, J.F., 2007. A hybrid heuristic for the uncapacitated single allocation hub location problem. *Omega* 35, 211 – 220.
- Cunha, C.B., Silva, M.R., 2007. 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* 179, 747 – 758.
- Ernst, A., Krishnamoorthy, M., 1996. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science* 4, 139 – 154.
- Farahani, R.Z., Hekmatfar, M., Arabani, A.B., Nikbakhsh, E., 2013. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering* 64, 1096–1109.
- Feo, T.A., Resende, M.G.C., 1995. Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6, 109–133.
- Goldman, A., 1969. Optimal location for centers in a network. *Transportation Science* 3, 352–360.
- Gomes, B., Martins, A., de Camargo, R., Ramirez, J., 2013. An efficient genetic algorithm for the design of hub-and-spoke networks. *Communications Letters, IEEE* 17, 793 – 796.
- Google, 2013. A API da Matriz de distâncias do Google. Disponível em: <https://developers.google.com/maps/documentation/distancematrix/>.
- IBGE, 2011. Sistema IBGE de Recuperação Automática - SIDRA. Disponível em: www.sidra.ibge.gov.br.

- Mladenovic, N., Hansen, P., 1997. A variable neighborhood search. *Computers and Operations Research* 24, 1097–1100.
- O’Kelly, M.E., 1987. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research* 32, 393 – 404.
- O’Kelly, M.E., Miller, H.J., 1994. The hub network design problem: A review and synthesis. *Journal of Transport Geography* 2, 31 – 40.
- Silva, M.R., Cunha, C.B., 2004. Configuração de redes do tipo hub-and-spoke para o transporte rodoviário de carga parcelada no brasil utilizando algoritmos genéticos. *TRANSPORTES* 12, 23–31.
- Silva, M.R., Cunha, C.B., 2009. New simple and efficient heuristics for the uncapacitated single allocation hub location problem. *Computers & Operations Research* 36, 3152 – 3165.
- Skorin-Kapov, D., Skorin-Kapov, J., O’Kelly, M.E., 1996. Tight linear programming relaxations of uncapacitated p-hub median problems. *European Journal of Operational Research* 94, 582–593.
- Topcuoglu, H., Corut, F., Ermis, M., Yilmaz, G., 2005. Solving the uncapacitated hub location problem using genetic algorithms. *Computers and Operations Research* 32 (4), 967–984.