

## **Uma eficiente heurística para o projeto de redes eixo-raio: um estudo de caso para as cidades de Minas Gerais.**

**Rodrigo F. Brito; Bruno N. Gomes**

Instituto Federal de Minas Gerais - IFMG  
Av. Serra da Piedade, 299 - Morada da Serra  
Sabará / MG

contato@rodrigobrito.net; bruno.nonato@ifmg.edu.br

**Ricardo S. de Camargo**

Universidade Federal de Minas Gerais - UFMG  
Av. Antônio Carlos, 6627 - Pampulha  
Belo Horizonte / MG  
rcamargo@dep.ufmg.br

### **RESUMO**

Este artigo aborda o projeto de redes eixo-raio com alocação simples que são estruturas de redes alocando pontos de origem e destino indiretamente por nós concentradores. Tal modelo é muito utilizado em serviços de transportadoras, onde as cargas são alocadas a uma central que concentra o fluxo de demanda e fazem a redistribuição para outras centrais. Por se tratar de um problema NP-Difícil, propõe-se uma heurística para resolução do mesmo, composta por um eficiente fase de construção e 4 funções de vizinhança exploradas em buscas locais e perturbações. Os resultados reportados nos experimentos computacionais mostraram que a heurística proposta supera outro método, considerado estado da arte na literatura, tanto em qualidade de soluções quanto em tempo computacional. O algoritmo desenvolvido também foi aplicado a um estudo de caso considerando as cidades de um estado brasileiro. Resultados mostraram que o método foi capaz de projetar redes aplicáveis na prática.

**PALAVRAS CHAVE. Redes eixo-raio, Otimização combinatória, Heurísticas.**

**Área Principal: Pesquisa Operacional**

### **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 structures explored in local searches and perturbations. 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 practice.

**KEYWORDS. Hub-spoke networks, Combinatorial optimization, Heuristic.**

**Main Area: Operational research**

## 1. Introdução

Redes são utilizadas em várias situações e áreas de conhecimento, elas proporcionam melhorias organizacionais e são ótimos meios para otimização de custos. Atualmente as redes são aplicadas em diversos setores da indústria, principalmente em situações onde se é necessário conectar pontos e distribuir demanda.

Buscando otimizar custos na construção de uma rede, diversas topologias e modelos organizacionais vem sendo abordados, entre eles o modelo de redes eixo-raio (ER) vem se destacando por possuir grandes aplicações nos setores de transporte, telecomunicações, entre outros (Alumur e Kara, 2008).

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. Este formato vem sendo amplamente abordado por ser uma alternativa de configuração mais econômica na construção de redes (O'Kelly e 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.

Um exemplo de aplicação das redes ER são empresas transportadoras de cargas fracionadas, tais empresas organizam suas entregas alocando pontos de distribuição em lugares estratégicos visando a redução de custos. A estratégia comumente adotada consiste em utilizar veículos de pequeno porte para transporte entre central e clientes, e veículos de maior porte para transportes de larga escala entre as centrais instaladas. 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.

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. Pode ser encontrado 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.

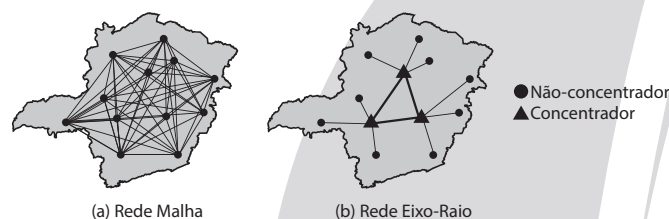


Figura 1: Exemplos de topologias de redes

A Figura 1 ilustra um 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.

Na literatura, as redes ER também são identificadas como um Problema de Localização de Concentradores (PLC), tendo o objetivo de minimizar o custo total de transporte entre entidades de uma rede. Tal problema pode possuir diferentes versões de acordo com as restrições fluxo, número de concentradores, custos 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 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*).

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

O PLC foi introduzido por Goldman (1969), no entanto 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 e 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 e 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 e Kara (2008).

Silva e 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 e 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 igualmente por todo espaço territorial do estado.

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

## 2. Formulação Matemática

O modelo matemático proposto por O'Kelly (1987) é baseado nas seguintes definições:  $N$  é o conjunto de  $n$  nós de uma rede, onde o parâmetro  $w_{ij}$  é o fluxo de demanda originado em  $i$  com destino a  $j$  ( $i, j \in N : i \neq j$ ), o qual é roteado por um ou dois concentradores instalados na rede. O custo de transporte por unidade de demanda nesse caminho é definido por  $c_{ijkm} = c_{ik} + \alpha c_{km} + c_{mj}$ , sendo  $0 \leq \alpha \leq 1$  o fator de desconto 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  $z_{kk} = 0$ .

$$\text{Minimize } \sum_{k=1}^n f_k z_{kk} + \sum_{i=1}^n \sum_{j=i+1}^n \sum_{k=1}^n \sum_{m \neq k}^n (w_{ij} c_{ijkm} + w_{ji} c_{ijmk}) z_{ik} z_{jm} \quad (1)$$

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

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

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

A equação (1) 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  $i$  ao nó  $k$ , somente se  $k$  for um concentrador. As equações (4) são restrições de integralidade da variável  $z_{ik}$ .

## 3. 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.

No Algoritmo 1, primeiramente a solução inicial é construída (Seção 3.1). Após isso, o processo de exploração é estruturado em repetidas iterações com perturbação na vizinhança Troca de Função (Seção 3.2) em conjunto com um processo de busca local realizado por meio da metaheurística Descida em Vizinhança Variável (VND) (Mladenovic e Hansen, 1997). O ciclo de

---

**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**


---

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 e Hansen, 1997) e retorna ao processo de perturbação. O algoritmo finaliza após um número pré-estabelecido de iterações sem melhora.

### 3.1. Processo de Construção

A estratégia de construção de soluções é baseada no processo construtivo do GRASP (Feo e 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).

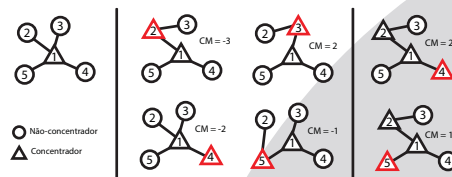


Figura 2: Construção GRASP

A Figura 2 exemplifica tal mecanismo de construção. 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).

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 piores 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 possuem 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  $\lambda$  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 possuem  $CM \leq MA$ . 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  $L$ .

A Figura 2 exemplifica o processo de construção. Inicialmente, considera-se a solução com apenas o nó 1 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 3 gera-se  $CM > 0$ , ou seja, a solução foi piorada. 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 2 terá  $CM \leq MA$ . Dessa forma, o nó 2 é inserido em  $L$ , e escolhido para ser configurado como concentrador. Tal processo se repete, agora considerando a solução com nó 2 como concentrador. Vê-se pela terceira coluna da Figura 2 que a configuração dos nós como concentradores não acarretou em melhoria na solução ( $CM > 0$ ), logo nenhum nó é adicionado à  $L$  e o processo é finalizado.

### 3.2. Estruturas de vizinhanças

A presente seção descreve as estruturas de vizinhança exploradas nas buscas locais e perturbações. As estruturas de vizinhança implementadas são ilustradas na Figura 3.

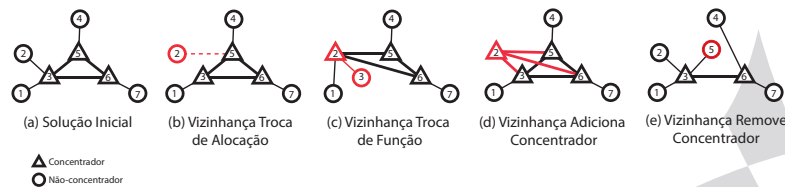


Figura 3: Vizinhanças

**Vizinhança Troca Alocação (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 Figura 3 (b) exemplifica um movimento dessa vizinhança, em que é feita a troca da alocação do nó 2, inicialmente alocado ao concentrador 3, para o concentrador 5.

**Vizinhança Troca Função (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 Figura 3 (c) exemplifica um movimento dessa vizinhança, onde os nós 2 e 3 trocam de função.

**Vizinhança Insere Concentrador (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 Figura 3 (d), onde o nó 2 passa a ser um novo concentrador da solução corrente.

**Vizinhança Remove Concentrador (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 Figura 3 (e), 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 3.3) e VNS (Seção 3.4) desenvolvidos. Além disso, tais vizinhanças são utilizadas como perturbações em que novos vizinhos são gerados de forma aleatória para exploração do espaço de busca.

### 3.3. Descida em Vizinhança Variável - VND

A busca local proposta é baseado na metaheurística VND (Mladenovic e 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.

O Algoritmo 2 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  $k$ .

---

### Algoritmo 2: 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 ← 1;
  n ← |V|;
  enquanto k ≤ n faça
    S1 ← S;
    S1 ← Busca Local (S1, Vk);
    se FO(S1) < FO(S) então
      S ← S1;
      k ← 1;
    senão
      k ← k + 1;
  fim
fim
  retorna S;
fim

```

---

### 3.4. 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 e Hansen, 1997) com o objetivo de explorar o espaço de soluções através de intercalações entre estruturas de vizinhanças.

---

### Algoritmo 3: Pseudocódigo do VNS

---

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

**início**

```

  k ← 1;
  n ← |V|;
  enquanto k ≤ n faça
    S1 ← S;
    S1 ← Gerar solução vizinha (S1, Vk);
    S1 ← VND(S1);
    se FO(S1) < FO(S) então
      S ← S1;
      k ← 1;
    senão
      k ← 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 3 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.

#### 4. 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 e 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  $\alpha$  a economia de escala. As instâncias foram 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 vezes em que a solução ótima foi encontrado pelo método para cada instância.

A Tabela 1 reporta a eficiência do método proposto em relação ao AG proposto por Gomes et al. (2013). 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.

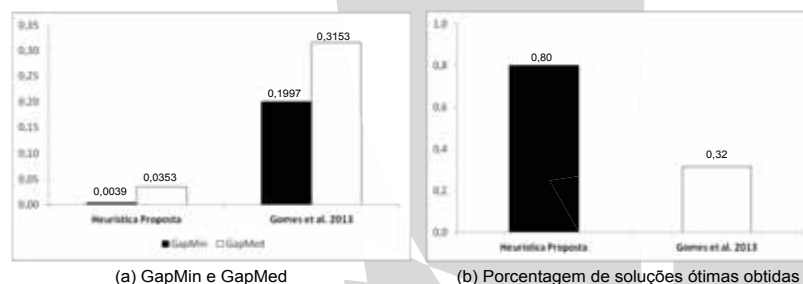


Figura 4: Heurística Proposta x Gomes et al. (2013)



Instância	Ótimo	Heurística Proposta				Gomes et al. (2013)			
		GapMin	GapMed	#Best	T (s)	GapMin	GapMed	#Best	T (s)
AP10-2	90963539,4763	0,0000	0,0000	10	0,0048	0,0000	0,0000	10	0,0520
AP10-4	95079629,9069	0,0000	0,0086	9	0,0105	0,0000	0,0000	10	0,0960
AP10-6	95161467,5800	0,0000	0,0000	10	0,0019	0,0000	0,0000	10	0,0120
AP10-8	95161467,5800	0,0000	0,0000	10	0,0020	0,0000	0,0000	10	0,0120
AP20-2	91507336,6084	0,0000	0,0000	10	0,0191	0,0000	0,0000	10	0,0440
AP20-4	96673177,8591	0,0000	0,0000	10	0,0193	0,0000	0,0000	10	0,0120
AP20-6	98181949,7142	0,0000	0,0000	10	0,0136	0,0000	0,0000	10	0,0600
AP20-8	98181949,7142	0,0000	0,0000	10	0,0127	0,0000	0,0000	10	0,0560
AP30-2	83576754,7126	0,0000	0,0111	9	0,4346	0,0000	0,1068	5	1,5680
AP30-4	91209293,1037	0,0000	0,1241	1	1,2905	0,0000	0,0075	8	2,2320
AP30-6	95053569,4196	0,0000	0,0000	10	0,0799	0,0000	0,0000	10	0,5440
AP30-8	98520500,5833	0,0000	0,0000	10	0,0989	0,0000	0,0000	10	0,8640
AP40-2	80537210,1476	0,0000	0,0000	10	0,9005	0,0000	0,0000	10	2,5920
AP40-4	88042046,0930	0,0000	0,0000	10	0,1990	0,0000	0,0000	10	0,5120
AP40-6	94523086,4474	0,0000	0,0000	10	0,3601	0,0000	0,0000	10	1,0360
AP40-8	99180263,1029	0,0000	0,0000	10	0,2130	0,0000	0,0000	10	1,0520
AP50-2	71261044,6135	0,0000	0,0000	10	0,7465	0,0000	0,0039	8	1,7680
AP50-4	80325464,9026	0,0000	0,0000	10	0,5816	0,0000	0,3063	3	2,1120
AP50-6	89389885,1918	0,0000	0,0505	4	4,4098	0,0000	0,1607	4	6,7300
AP50-8	95205946,9629	0,0000	0,0000	10	0,3721	0,0000	0,0274	5	1,8240
AP60-2	64790967,0386	0,0000	0,0000	10	1,8103	0,1679	0,2232	0	3,2840
AP60-4	73074656,5752	0,0000	0,0000	10	1,2952	0,5061	0,7665	0	3,2440
AP60-6	80673823,6599	0,0000	0,0331	9	3,3555	0,3309	0,3713	0	5,0940
AP60-8	87285162,0911	0,0000	0,0000	10	2,6693	0,0000	0,0025	8	6,5800
AP70-2	74451085,6010	0,0000	0,0000	10	2,1293	0,0000	0,0000	10	6,8320
AP70-4	83272519,6325	0,0000	0,0000	10	1,6586	0,0000	0,0840	4	8,6400
AP70-6	91741977,1993	0,0000	0,0000	10	3,4637	0,0020	0,2724	0	7,2920
AP70-8	97109841,9802	0,0696	0,0875	0	11,9391	0,0696	0,2614	0	20,5760
AP80-2	70713485,9938	0,0000	0,0000	10	3,3446	0,0000	0,0000	10	9,9400
AP80-4	79704787,9527	0,0000	0,0000	10	2,8910	0,1693	0,3760	0	10,6240
AP80-6	88418089,7530	0,0000	0,0000	10	3,9668	0,6715	1,1481	0	15,6800
AP80-8	95798238,8757	0,0000	0,1546	6	10,6704	0,0577	0,3606	0	24,0160
AP90-2	69223173,9238	0,0000	0,0000	10	6,9534	0,0094	0,0268	0	29,9680
AP90-4	78931357,8060	0,0000	0,0000	10	10,4433	0,0479	0,2330	0	39,9472
AP90-6	87012179,9499	0,0000	0,0000	10	22,7344	0,5047	0,8567	0	33,6184
AP90-8	92780846,2542	0,0000	0,0762	5	25,5123	0,2962	0,5019	0	35,1491
AP100-2	67584119,7648	0,0000	0,0000	10	27,0357	0,0000	0,0127	6	41,7380
AP100-4	77545112,3194	0,0000	0,0000	10	8,1031	0,0732	0,1479	0	22,6640
AP100-6	86371515,8647	0,0000	0,0314	4	50,4625	0,4382	0,7937	0	82,8255
AP100-8	93184508,9692	0,0000	0,1136	4	45,5444	0,5402	0,6440	0	67,8120
AP130-2	60901562,4908	0,0000	0,0417	9	53,4608	0,0029	0,0151	0	85,5370
AP130-4	71560028,4291	0,0000	0,0788	1	148,6842	0,1904	0,2317	0	194,8740
AP130-6	81332451,6280	0,0000	0,1643	1	197,4517	0,5599	0,8899	0	226,2805
AP130-8	89960490,4486	0,0000	0,0050	6	83,9039	0,5219	0,8177	0	180,7930
AP150-2	57397001,4468	0,0000	0,0000	10	59,6084	0,1732	0,1854	0	129,2480
AP150-4	68083743,1244	0,0000	0,0000	10	69,8005	0,3725	0,4969	0	171,0570
AP150-6	77856884,8290	0,0584	0,2142	0	298,8046	0,8627	1,0599	0	266,4472
AP150-8	86068900,6009	0,0522	0,4169	0	260,4720	1,3944	1,5976	0	242,4976
AP170-2	63888877,9054	0,0000	0,0276	6	211,4970	0,0216	0,0404	0	234,7072
AP170-4	73373712,1241	0,0000	0,0000	10	132,0900	0,3676	0,5083	0	191,7234
AP170-6	81744592,0692	0,0360	0,2912	0	404,7596	0,9646	1,2346	0	305,9420
AP170-8	88839517,4393	0,0000	0,0000	10	46,6566	0,1627	0,5003	0	173,8050
AP200-2	62774425,0872	0,0000	0,0440	4	499,2363	0,0490	0,0641	0	337,0540
AP200-4	72333785,1444	0,0000	0,0000	10	137,6671	0,2781	0,4664	0	155,2176
AP200-6	81236269,7266	0,0000	0,0000	10	181,7658	0,9084	1,0512	0	312,3732
AP200-8	89181926,7904	0,0000	0,0000	10	223,8137	0,4694	0,8026	0	222,4810
<b>Média</b>		<b>0,0039</b>	<b>0,0353</b>	<b>448</b>	<b>29,3119</b>	<b>0,1997</b>	<b>0,3153</b>	<b>211</b>	<b>70,1561</b>

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

A Figura 4 (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 Figura 4

(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.

## 5. 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 as cidades do estado de Minas Gerais, considerando dados reais de distância rodoviária e demanda baseada no Produto Interno Bruto (PIB) e população de cada cidade.

### 5.1. 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 cidades. O algoritmo de rotas do *Google Maps* retorna a menor distância entre duas cidades dadas.

A base de dados construída possui 853 cidades pertencentes ao estado de Minas Gerais no Brasil. As informações contidas nessa base são distância e demanda entre cada uma das cidades. Cabe ressaltar que, a demanda entre as cidades foi calculada de acordo dados de PIB e população das mesmas presentes no Instituto Brasileiro de Geografia e Estatística (IBGE, 2011).

### 5.2. Resultados estudo de caso

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

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

Tabela 2: Estudo de caso - Minas Gerais

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 as cidades.

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 das cidades. Considerando a aplicação e a dimensão do problema, o método retornou uma boa solução em tempo computacional viável.

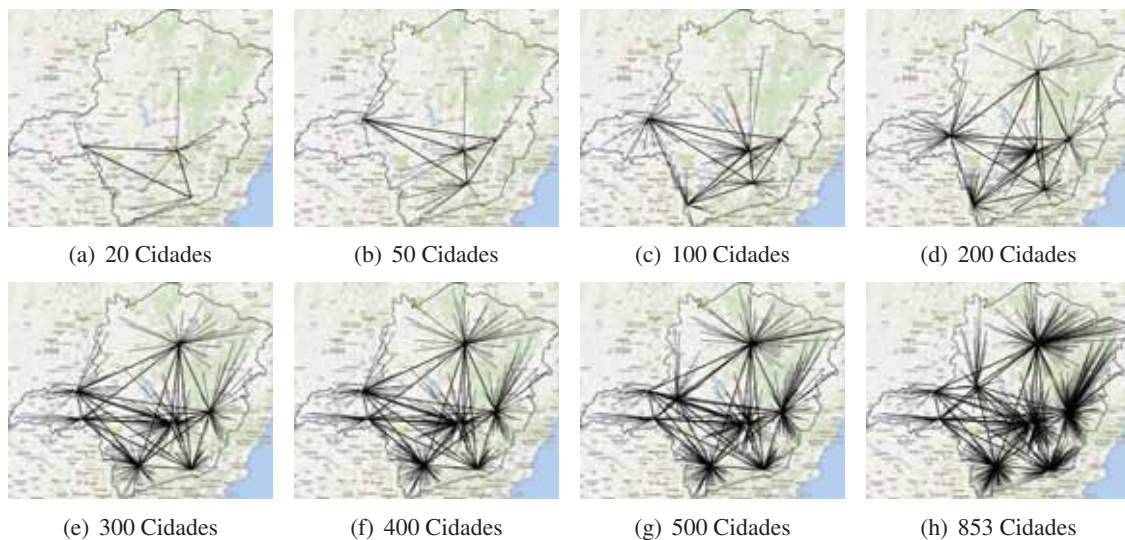


Figura 5: Heurística Proposta em Minas Geras com cidades de maior PIB

Percebe-se pela Figura 5 que as redes projetadas pelo método possuem boa distribuição em relação à instalação dos concentradores e alocação das cidades 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 às 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.

## 6. 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 das cidades do estado. Após aplicação em diversas instâncias geradas, o algoritmo se mostrou capaz projetar boas redes, distribuindo os concentradores e alocando as cidades igualmente por todo o território do estado.

## 7. Agradecimentos

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo à Pesquisa do estado de Minas Gerais (FAPEMIG), pelo suporte financeiro e apoio ao projeto.

## Referências

- Alumur, S. e Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1 – 21.
- Aykin, T. (1994). Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research*, 79(33):501–523.
- Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):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(2):211 – 220.
- Cunha, C. B. e 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(3):747 – 758.
- Ernst, A. e Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4(3):139 – 154.
- Farahani, R. Z., Hekmatfar, M., Arabani, A. B., e Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64:1096–1109.
- Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, first edition.
- Goldman, A. (1969). Optimal location for centers in a network. *Transportation Science*, 3:352–360.
- Gomes, B., Martins, A., de Camargo, R., e Ramirez, J. (2013). An efficient genetic algorithm for the design of hub-and-spoke networks. *Communications Letters, IEEE*, 17(4):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](http://www.sidra.ibge.gov.br).
- Mladenovic, N. e 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(3):393 – 404.
- O'Kelly, M. E. e Miller, H. J. (1994). The hub network design problem: A review e synthesis. *Journal of Transport Geography*, 2(1):31 – 40.
- Silva, M. R. e 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. e Cunha, C. B. (2009). New simple and efficient heuristics for the uncapacitated single allocation hub location problem. *Computers & Operations Research*, 36(12):3152 – 3165.
- Topcuoglu, H., Corut, F., Ermis, M., e Yilmaz, G. (2005). Solving the uncapacitated hub location problem using genetic algorithms. *Computers and Operations Research*, 32 (4):967–984.