ALGORITMO *SMART* ILS APLICADO AO PROBLEMA DE LOCALIZAÇÃO DE CONCENTRADORES COM MAXIMIZAÇÃO DO LUCRO

Fabricio Alves Oliveira [1,2], Elisangela Martins de Sá [2], Marcone Jamilson Freitas Souza [3], Sérgio Ricardo de Souza [2],

[1] Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais
Rua Afonso Sardinha, 90, CEP: 36420-000, Ouro Branco, MG, Brasil

[2] Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675, CEP: 30510-000, Belo Horizonte, MG, Brasil

[3] Universidade Federal de Ouro Preto
Campus Universitário, Morro do Cruzeiro, CEP: 35400-000, Ouro Preto, MG, Brasil

fabricio.oliveira@ifmg.edu.br, elisangelamartins@cefetmg.br, marcone@ufop.edu.br, sergio@cefetmg.br

RESUMO

Este trabalho aborda o problema de localização de concentradores, que tem, como objetivo, a maximização do lucro. O problema tratado possui alocação múltipla, rede de concentradores incompleta, não permite conexões diretas entre nós não concentradores e possibilita que a demanda da rede seja atendida parcialmente, de modo que seja satisfeita apenas quando for lucrativa. Como esse problema é NP-difícil, foi desenvolvido um algoritmo baseado na metaheurística *Smart* ILS, que utiliza o método RVND como procedimento de busca local. Os resultados dos experimentos computacionais foram validados com valores apontados na literatura e mostraram que o algoritmo proposto produziu soluções de boa qualidade em baixo tempo de processamento, alcançado o valor ótimo para todas as instâncias testadas e *gap* médio inferior a 1% em todos os casos.

PALAVRAS CHAVE. Localização de concentradores. Maximização de lucro. *Smart Iterated Local Search*. Metaheurísticas.

Tópicos: MH - Metaheurísticas, OC – Otimização Combinatória

ABSTRACT

This work deals with a hub location problem that aims to maximize profit. The problem addressed has multiple allocation, an incomplete hub network, does not allow direct connections between non-hubs nodes, and allows the network's demand to be partially met, being satisfied only when it is profitable. As this problem is NP-hard, an algorithm based on the metaheuristic Smart ILS was developed, which applied the RVND method as a local search procedure. The results of the computational experiments were validated with values indicated in the literature and showed that the proposed algorithm yielded good quality solutions in a low processing time. In particular, all tested instances reached the optimum value, and an average gap of less than 1% was achieved in all cases.

KEYWORDS. Hub location. Profit maximization. Smart Iterated Local Search. Metaheuristics.

Paper topics: MH - Metaheuristics, OC - Combinatorial Optimization

1. Introdução

Redes de concentradores são frequentemente usadas no projeto de redes de grande escala. Nessas redes, *commodities* de diferentes origens são enviadas por meio de instalações intermediárias, conhecidas como concentradores. Os concentradores são responsáveis por receber, agregar, transferir e distribuir o fluxo de demanda na rede. Desse modo, é possível conectar os pares de origem e destino, usando uma menor quantidade de conexões, o que permite reduzir o custo total de configuração da rede e também os custos de transporte, uma vez que a consolidação e redistribuição do fluxo pelos concentradores proporciona uma economia de escala nos arcos entre concentradores.

Os problemas de localização de concentradores lidam com as decisões conjuntas de localização e projeto da rede, de modo a otimizar um objetivo específico. Em virtude dessa combinação de elementos da localização e projeto de rede, esses problemas são NP-difíceis [Alumur e Kara, 2008; Campbell e O'Kelly, 2012], o que justifica a utilização de métodos heurísticos em sua resolução. Esse problemas possuem aplicações em diversas áreas, como sistemas de transporte [Lin et al., 2012; Saberi e Mahmassani, 2013], serviços de entrega postal [Ernst e Krishnamoorthy, 1998] e redes de telecomunicações [Bollapragada et al., 2006].

Este trabalho aborda o problema de localização de concentradores com maximização do lucro (PLCML). Esse problema consiste em determinar a quantidade e a localização dos concentradores, selecionar os pares de origem e destino que serão atendidos, estabelecer os arcos que serão ativados, além de determinar os caminhos ótimos pelos quais o fluxo de demanda será roteado, de modo a maximizar o lucro total da rede (dado pela diferença entre a receita total e os custos totais).

Os primeiros trabalhos da área de localização de concentradores com foco na maximização do lucro foram introduzidos por Alibeyg et al. [2016, 2017], nos quais foram propostos, respectivamente, formulações para o problema e um método exato de resolução. Em Taherkhani e Alumur [2018], foram desenvolvidos modelos matemáticos para várias versões do problema, que incluem todos os tipos de estratégia de alocação, permitem conexões diretas entre nós não concentradores e consideram uma rede incompleta, utilizando o CPLEX para avaliar as formulações propostas. Mais recentemente, Taherkhani et al. [2019] abordaram o problema com a maximização do lucro, considerando alocação de capacidade e supondo que a demanda possui origem em diferentes segmentos em uma rede completa. Foram propostos dois algoritmos exatos baseados no método de decomposição de Benders e técnicas para aceleração da convergência dos algoritmos.

Devido à complexidade dos problemas de localização de concentradores, diversos tipos de abordagens heurísticas têm sido usadas para resolvê-los. Em Klincewicz [1991] foram apresentadas heurísticas baseadas em troca, agrupamento e enumeração para o problema de localização de p-concentradores. Nesse problema, a quantidade de concentradores é estabelecida previamente. Chen [2007] resolveu o problema de localização de concentradores com alocação única e não capacitado com algoritmos heurísticos híbridos baseados nos métodos *Simulated Annealing* e lista tabu. Em Contreras et al. [2017] foi tratado o problema de localização de concentradores em ciclo, que consiste em localizar uma quantidade fixa de concentradores que estão conectados por meio de um ciclo, alocar nós não concentradores aos concentradores, de forma a minimizar o custo total da rede. Os autores desenvolveram um algoritmo *Branch-and-cut* e propuseram uma metaheurística gulosa baseada em um procedimento de busca adaptativa aleatória. Em Martins de Sá et al. [2018] foram apresentados dois algoritmos especializados, utilizando a decomposição de Benders e um procedimento de pesquisa local estocástico baseado no ILS-VND para lidar com um problema de localização de concentradores, assumindo incertezas sobre a demanda e os custos fixos de instalação de concentradores.

Este trabalho apresenta um algoritmo baseado na metaheurística *Smart Iterated Local Search* (*Smart* ILS) [Reinsma et al., 2018] para lidar com o PLCML. O algoritmo proposto usa, como procedimento de busca local, o método *Random Variable Neighborhood Descent* (RVND) [Souza et al., 2010; Subramanian et al., 2010], que aplica, de forma aleatória, a troca sistemática das estruturas de vizinhança. Esse algoritmo foi desenvolvido tendo em vista a aplicação bem sucedida de suas variantes em problemas combinatórios na literatura, tal como o método ILS-VND proposto por Martins de Sá et al. [2018]. Os resultados dos experimentos computacionais foram comparados com os apontados em Oliveira et al. [2021], em que um algoritmo exato, baseado na decomposição de Benders, foi empregado. Observamos que o PLCML, com as características consideradas neste trabalho, foi abordado apenas por Taherkhani e Alumur [2018] e Oliveira et al. [2021].

O restante deste trabalho está estruturado da seguinte forma. Na Seção 2, o PLCML é definido e suas características são descritas. A Seção 3 detalha o algoritmo proposto. Os experimentos computacionais são expostos na Seção 4. Por fim, na Seção 5, são feitas as considerações finais.

2. Definição e Caracterização do Problema

O problema de localização de concentradores abordado neste artigo tem, como objetivo, maximizar o lucro total da rede. Este valor é dado pela receita total, obtida pelo atendimento da demanda dos pares selecionados, menos a soma dos custos do projeto da rede e dos custos de transporte. Os custos do projeto da rede são dados pelos custos fixos de instalação de concentradores e arcos entre concentradores, enquanto o custo de transporte se refere ao gasto associado para rotear o fluxo de demanda entre os pares de origem e destino. É assumido também que um fator de desconto constante é aplicado no custo de transporte nos arcos entre concentradores.

O problema considerado possui (i) alocação múltipla, de modo que os nós da rede podem ser alocados a mais de um concentrador; (ii) é não capacitado; (iii) possui custos fixos de instalação de concentradores e arcos entre concentradores; (iv) não permite conexões diretas entre nós não concentradores; (v) possui rede de concentradores incompleta, ou seja, os concentradores não precisam ser totalmente interligados entre si; e (vi) não é exigido que todos os pares de origem e destino tenham sua demanda atendida, sendo satisfeita somente quando for lucrativa.

Para apresentar um modelo para o PLCML, considere os parâmetros dados a seguir. Seja N o conjunto de nós que trocam fluxos e que também são potenciais candidatos a concentradores. A quantidade de demanda a ser enviada do nó $i \in N$ para o nó $j \in N$ é denotada por w_{ij}. Para todo $i, j \in N$, tem-se que r_{ij} representa a receita obtida pelo atendimento de uma unidade da demanda do nó de origem i para o nó de destino j. O custo de transporte em um arco ligando os nós $i \in N$ e $j \in N$ é denotado por c_{ij}. O custo fixo de instalação de um concentrador no nó $k \in N$ é dado por s_k, e o custo fixo de instalação de um arco entre os concentradores k e m é denotado por g_{km}, k,$m \in N$. Por fim, denotamos por α ($0 \leq \alpha < 1$) o fator de desconto constante no transporte realizado nas conexões entre concentradores.

Considere, também, o seguinte conjunto de variáveis: (i) variáveis binárias h_k: indicam se um concentrador é localizado no nó $k \in N$; (ii) variáveis binárias z_{km}: indicam se uma conexão entre os concentradores $k \in N$ e $m \in N$ está ativa; (iii) variáveis de fluxo a_{ijk}: representam a fração da demanda entre os nós $i \in N$ e $j \in N$ que é atendida ao longo de um caminho no qual o primeiro concentrador é $k \in N$; (iv) variáveis de fluxo b_{ijm}: representam a fração da demanda entre os nós $i \in N$ e $j \in N$ que é atendida ao longo de uma rota na qual o último concentrador é $m \in N$; (v) variáveis de fluxo x_{ijkm}: determinam a fração da demanda w_{ij} que é roteada no arco entre os concentradores $k \in N$ e $m \in N$.

Assim, uma formulação para o PLCML, proposta por Oliveira et al. [2021], é dada por:

$$\max \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in N} r_{ij}w_{ij}a_{ijk} - \left[\sum_{i\in N}\sum_{j\in N}\sum_{k\in N} c_{ik}w_{ij}a_{ijk} + \sum_{i\in N}\sum_{j\in N}\sum_{m\in N} c_{mj}w_{ij}b_{ijm} + \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} \alpha c_{km}w_{ij}x_{ijkm} + \sum_{k\in N} s_k h_k + \sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} g_{km}z_{km}\right] \tag{1}$$

sujeito a

$$\sum_{k\in N} a_{ijk} \leq 1 \qquad i,j \in N \tag{2}$$

$$\sum_{m\in N} b_{ijm} \leq 1 \qquad i,j \in N \tag{3}$$

$$a_{ijk} + \sum_{\substack{m\in N\\ m\neq k}} x_{ijmk} = b_{ijk} + \sum_{\substack{m\in N\\ m\neq k}} x_{ijkm} \qquad i,j,k \in N \tag{4}$$

$$a_{ijk} \leq h_k \qquad i,j,k \in N \tag{5}$$

$$b_{ijm} \leq h_m \qquad i,j,m \in N \tag{6}$$

$$x_{ijkm} \leq z_{km} \qquad i,j,k,m \in N, k \neq m \tag{7}$$

$$z_{km} \leq h_k \qquad k,m \in N, k \neq m \tag{8}$$

$$z_{km} \leq h_m \qquad k,m \in N, k \neq m \tag{9}$$

$$x_{ijkm} \geq 0 \qquad i,j,k,m \in N, k \neq m \tag{10}$$

$$a_{ijk} \geq 0 \qquad i,j,k \in N \tag{11}$$

$$b_{ijm} \geq 0 \qquad i,j,m \in N \tag{12}$$

$$h_k \in \{0,1\} \qquad k \in N \tag{13}$$

$$z_{km} \in \{0,1\} \qquad k,m \in N, k \neq m. \tag{14}$$

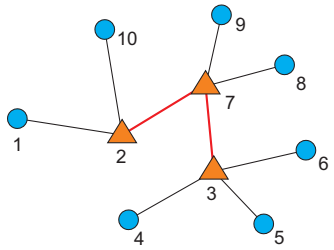
A função objetivo (1) expressa o lucro total da rede, que é obtido pela diferença entre a receita total e o custo total. As restrições (2)-(4) são restrições de conservação do fluxo. As restrições (5) e (6) garantem que a demanda entre os nós de origem e destino pode ser atendida apenas através de concentradores instalados. As restrições (7) asseguram que o fluxo roteado em uma conexão entre os concentradores seja realizado apenas nas conexões ativadas. As restrições (8) e (9) indicam que uma conexão entre concentradores só pode ser operada se os dois nós finais dessa conexão forem concentradores. As restrições (10) - (14) representam o domínio das variáveis de decisão.

3. Descrição do Algoritmo Proposto

Esta seção está organizada da seguinte forma. A Subseção 3.1 mostra como uma solução para o PLCML é representada e avaliada. Um método para construir uma solução inicial é apresentado na Subseção 3.2. A Subseção 3.3 detalha as estruturas de vizinhança. O método de busca local e a perturbação são apresentados nas Subseções 3.4 e 3.5, respectivamente. A Subseção 3.6 apresenta o algoritmo *Smart* ILS aplicado ao problema em questão.

3.1. Representação e Avaliação da Solução

Uma solução para o PLCML será representada por $s = (H^1, Z^1)$, sendo H^1 o conjunto de concentradores instalados e Z^1 o conjunto de arcos entre concentradores ativos. A Figura 1 mostra uma rede de concentradores e a representação da solução associada a ela, em que os círculos são os nós de alocação, os triângulos são os concentradores e os segmentos em vermelho são os arcos entre concentradores.



(a) Rede de concentradores.

$$s = (\{2,3,7\}, \{(2,7),(3,7)\})$$

(b) Representação da solução.

Figura 1: Representação de uma solução.

Observe que, na representação da solução, não é necessário indicar todas as conexões da rede nem a alocação aos concentradores dos pares de nós que foram selecionados para serem atendidos, pois todo o fluxo de demanda deve ser roteado através dos caminhos mais curtos entre os nós de origem e destino. Assim, uma solução s para o PLCML é avaliada por:

$$f(s) = \sum_{i \in N} \sum_{j \in N} w_{ij} P_{ij}(s) - C_{\text{fixos}}(s), \tag{15}$$

em que P_{ij} é o lucro unitário do par $i, j \in N$ associado à configuração da rede determinada por s; C_{fixos} representa os custos fixos de instalação de concentradores e arcos entre concentradores ativos na solução s; e P_{ij} é dado por:

$$P_{ij}(s) = \max\{0, r_{ij} - C_{ij}(s)\}, \tag{16}$$

sendo C_{ij} o custo mínimo de transporte da origem $i \in N$ até o destino $j \in N$ relativo à solução s. A determinação de C_{ij} é feita usando o algoritmo de Floyd-Warshall [Floyd, 1962], conforme a seguir. Ao invés de aplicá-lo em uma rede composta por todos os nós, ele é utilizado para determinar os menores custos de transportes dentre todos os pares de concentradores na rede no nível dos concentradores, assumindo que o custo de transporte nos arcos entre concentradores não ativos é suficientemente grande para não serem considerados. Então, seja C_{km}^{FW} o menor custo unitário de transporte entre os pares de concentradores $k, m \in H^1$. Assim, C_{ij} é determinado por:

$$C_{ij}(s) = \min_{k,m \in H^1} \left\{ c_{ik} + C_{km}^{FW} + c_{mj} \right\}. \tag{17}$$

3.2. Solução Inicial

Uma solução inicial para o PLCML foi gerada considerando a melhor solução obtida com a instalação de um único concentrador na rede. Esse procedimento é descrito no Algoritmo 1, que calcula, para cada nó da rede, o lucro total obtido pelo atendimento dos pares de origem e destino que são selecionados quando esse nó é um concentrador e retém a melhor solução encontrada.

Algoritmo 1 Solução Inicial: melhor solução com um único concentrador

Entrada: $N, r_{ij}, w_{ij}, c_{ij}, s_k$
Saída: Solução inicial s^0, Valor da solução inicial ϕ

$s^0 \leftarrow (\emptyset, \emptyset)$
$\phi \leftarrow 0$
$\phi_{\text{aux}} \leftarrow 0$
para todo $k \in N$ **faça**
  $\phi_{\text{aux}} \leftarrow -s_k$
  para todo $(i, j) \in N \times N$ **faça**
    $l_{ij} \leftarrow (r_{ij} - c_{ik} - c_{kj}) w_{ij}$
    se $l_{ij} > 0$ **então**
      $\phi_{\text{aux}} \leftarrow \phi_{\text{aux}} + l_{ij}$
    fim se
  fim para
  se $\phi_{\text{aux}} \geq \phi$ **então**
    $\phi \leftarrow \phi_{\text{aux}}$
    $s^0 \leftarrow (\{k\}, \emptyset)$
  fim se
fim para
retorne s^0, ϕ

3.3. Estruturas de Vizinhança

Para explorar o espaço de soluções do problema, foram utilizados seis tipos de movimentos para serem aplicados a uma solução s. Cada movimento dá origem a uma estrutura de vizinhança $\mathcal{N}_i(s), i = 1, \ldots, 6$, definidas a seguir:

(i) $\mathcal{N}_1(s)$: Soluções que podem ser alcançadas pela instalação de um novo concentrador;

(ii) $\mathcal{N}_2(s)$: Soluções que podem ser obtidas removendo um concentrador instalado. Neste caso, os arcos entre concentradores incidentes sobre esse concentrador também são removidos;

(iii) $\mathcal{N}_3(s)$: Soluções que podem ser obtidas ativando um arco entre concentradores instalados;

(iv) $\mathcal{N}_4(s)$: Soluções que podem ser atingidas por meio da remoção de um arco entre concentradores;

(v) $\mathcal{N}_5(s)$: Soluções que podem ser alcançadas instalando um novo concentrador e conectando-o a todos os demais concentradores presentes na rede;

(vi) $\mathcal{N}_6(s)$: Soluções que podem ser obtidas trocando um concentrador instalado por outro não instalado. Neste caso, os arcos entre concentradores incidentes sobre o concentrador removido também são desativados.

3.4. Busca Local

Neste trabalho, o procedimento de busca local é baseado na troca sistemática das estruturas de vizinhança utilizando a estratégia *Best Improvement*. Dada uma ordenação das estruturas de vizinhança, o procedimento de busca local é tal que, sempre que uma vizinhança não melhora a solução atual, a vizinhança seguinte é usada e, sempre que uma solução melhor é encontrada, então o algoritmo reinicia na primeira vizinhança, segundo a ordenação considerada.

A busca local utiliza o método *Random Variable Neighborhood Descent* (RVND) [Souza et al., 2010; Subramanian et al., 2010], que é uma variação do algoritmo *Variable Neighborhood Descent* (VND) [Hansen et al., 2017]. No RVND, apresentado no Algoritmo 2, a ordem de exploração das vizinhanças é aleatória a cada chamada do método, diferentemente do VND, em que essa ordem é estabelecida previamente, geralmente de acordo com a ordem de complexidade das vizinhanças, e fixada durante as execuções do método.

Algoritmo 2 *Random Variable Neighborhood Descent* - RVND

Entrada: Solução s, Conjunto $\mathcal{N}$ de vizinhanças
Saída: Solução s refinada

$\mathcal{NR} \leftarrow \mathcal{N}$ em ordem aleatória
$i \leftarrow 1$
enquanto $i \leq |\mathcal{N}|$ **faça**
    Encontre o melhor vizinho $s' \in \mathcal{NR}_i(s)$
    se $f(s') > f(s)$ **então**
        $s \leftarrow s'$
        $i \leftarrow 1$
    senão
        $i \leftarrow i + 1$
    fim se
fim enquanto
retorne s

3.5. Perturbação

Dado o nível de perturbação $nivel$ em uma determinada iteração, o procedimento de perturbação realizado na solução corrente consiste em selecionar sucessivamente e de modo aleatório $k = nivel$ soluções vizinhas na vizinhança $\mathcal{N}_6$. O Algoritmo 3 detalha esse procedimento.

Algoritmo 3 Perturbação

Entrada: Solução atual s, Vizinhança $\mathcal{N}_6$, Nível de perturbação $nivel$
Saída: Solução perturbada s'

1: $s' \leftarrow s$
2: $k \leftarrow nivel$
3: $i \leftarrow 1$
4: **enquanto** $i \leq k$ **faça**
5: Selecione aleatoriamente uma solução vizinha $s'' \in \mathcal{N}_6(s')$
6: $s' \leftarrow s''$
7: $i \leftarrow i + 1$
8: **fim enquanto**
9: **retorne** s'

3.6. *Smart Iterated Local Search* (*Smart* ILS)

A metaheurística *Smart Iterated Local Search* (*Smart* ILS) [Reinsma et al., 2018] é uma adaptação do método *Iterated Local Search* (ILS) [Lourenço et al., 2003], que consiste em explorar o espaço de soluções por meio de perturbações em ótimos locais. A diferença entre esses dois métodos está na forma como o nível de perturbação é atualizado pelo algoritmo. No ILS, o nível de perturbação é aumentado a cada iteração sem melhoria do método, enquanto no *Smart* ILS isso é feito somente após algumas tentativas frustradas de melhoria.

O Algoritmo 4 mostra o pseudocódigo do *Smart* ILS implementado para o PLCML, denotado por SILS-RVND. Esse algoritmo possui, como entradas, uma solução inicial s^0, o número máximo de iterações sem melhora $iterMax$ e a quantidade máxima de vezes em um mesmo nível de perturbação $vezesMax$. Inicialmente, na linha 1 é feita uma busca local sobre a solução inicial s^0, utilizando o RVND, e, a solução retornada, é então armazenada em s. Nas linhas 2, 3 e 4 são inicializados a quantidade de iterações sem melhora $iter$, o nível de perturbação $nivel$ e o número de vezes dentro de um mesmo nível de perturbação $nvezes$, respectivamente. Em seguida, entre as linhas 5 e 22, o algoritmo entra em uma laço com a finalidade de melhorar o valor da solução corrente até que a quantidade de iterações sem melhora seja igual a $iterMax$. Na linha 6 o contador de iterações sem melhora é incrementado. Na linha 7, uma solução s' é obtida através do procedimento de perturbação aplicado à solução s, como descrito na Subseção 3.5. Por sua vez, na linha 8 é aplicado o RVND sobre a solução s' e a solução refinada é armazenada em s''. Caso o valor de s'' seja melhor do que o valor de s, então a solução s é atualizada para s'', o contador de iterações sem melhora é reiniciado, bem como o nível de perturbação e a quantidade de vezes em um mesmo nível de perturbação (linhas 9 a 13). Se o valor de s'' não for melhor do que o valor da solução s, então entre as linhas 15 e 20 é verificado se o número de vezes em um mesmo nível de perturbação atingiu o seu valor máximo. Caso afirmativo, então o nível de perturbação é aumentado em uma unidade e a quantidade de vezes no mesmo nível de perturbação é reiniciada. Caso contrário, $nvezes$ é atualizado. Ao final do procedimento, o algoritmo retorna na linha 23, a melhor solução obtida s.

4. Experimentos Computacionais

Esta seção apresenta os resultados dos experimentos computacionais realizados para analisar o desempenho do algoritmo SILS-RVND proposto. A Subseção 4.1 detalha a configuração utilizada dos parâmetros do PLCML. A Subseção 4.2 apresenta os resultados da calibração dos parâmetros do algoritmo. A Subseção 4.3 mostra os resultados dos experimentos.

4.1. Configurações dos Parâmetros do PLCML

Para realização dos experimentos computacionais, foi utilizado o conjunto de dados *Australian Post* (AP) [Ernst e Krishnamoorthy, 1996]. Nesse conjunto de dados são disponibilizadas a distância e a demanda entre cada par de cidades. A matriz de demanda (w_{ij}) não é simétrica e as distâncias entre as cidades são consideradas como os custos de transporte (c_{ij}) da rede.

Algoritmo 4 *Smart Iterated Local Search* - SILS-RVND

Entrada: $s^0, iterMax, vezesMax$
Saída: s
1: $s \leftarrow$ RVND$(s^0, \mathcal{N})$
2: $iter \leftarrow 0$
3: $nivel \leftarrow 1$
4: $nvezes \leftarrow 1$
5: **enquanto** $iter \leq iterMax$ **faça**
6: $iter \leftarrow iter + 1$
7: $s' \leftarrow$ Perturbacao$(s, \mathcal{N}_6, nivel)$
8: $s'' \leftarrow$ RVND$(s', \mathcal{N})$
9: **se** $f(s'') > f(s)$ **então**
10: $s \leftarrow s''$
11: $iter \leftarrow 0$
12: $nivel \leftarrow 1$
13: $nvezes \leftarrow 1$
14: **senão**
15: **se** $nvezes \geq vezesMax$ **então**
16: $nivel \leftarrow nivel + 1$
17: $nvezes \leftarrow 1$
18: **senão**
19: $nvezes \leftarrow nvezes + 1$
20: **fim se**
21: **fim se**
22: **fim enquanto**
23: **retorne** s

Para a receita (r_{ij}) entre cada par de origem e destino, foram tomados três valores: 20, 30 e 50. Esses valores foram definidos por Taherkhani e Alumur [2018]. O conjunto de dados AP também fornece dois conjuntos de valores para os custos fixos de instalação de concentradores, referidos como *loose* (L) e *tight* (T). Foi assumido que o custo de ativação de arcos entre concentradores é 10% dos custos de instalação de concentradores, conforme sugerido em Taherkhani e Alumur [2018]. Não foram considerados custos de coleta e distribuição por unidade de demanda. Em todos os testes, a receita por unidade de demanda também é considerada independente da localização do par de nós a ser atendido e o fator de desconto $\alpha \in \{0{,}2; 0{,}4; 0{,}6; 0{,}8\}$. Foram realizados testes com instâncias de 40, 50, 75 e 100 nós.

O algoritmo foi implementado na linguagem `C++` e executado em um computador Intel Core i7-7500U, 2.70 GHz, com 16 GB de RAM, utilizando o ambiente Linux.

4.2. Calibração dos Parâmetros dos Algoritmos Propostos

Os valores dos parâmetros $iterMax$ e $vezesMax$ do algoritmo implementado foram calibrados pela ferramenta IRACE (*Iterated Racing for Automatic Algorithm Configuration*) [López-Ibáñez et al., 2016]. A Tabela 1 mostra as instâncias utilizadas no treinamento da ferramenta.

Nesse treinamento, foram consideradas as seguintes faixas de valores para os parâmetros calibrados: $iterMax \in \{5, 10, 15\}$ e $vezesMax \in \{2, 3, 4, 5, 6\}$. Os valores retornados foram

Tabela 1: Instâncias de treinamento submetidas ao IRACE

AP – 40			AP – 50			AP – 75			AP – 100		
Inst	α	Receita	Inst	α	Receita	Inst	α	Receita	Inst	α	Receita
40L	0,2	20	50L	0,6	20	75L	0,2	20	100L	0,6	20
40L	0,4	30	50L	0,4	30	75L	0,6	30	100L	0,2	30
40L	0,8	50	50L	0,2	50	75L	0,8	50	100L	0,4	50
40T	0,4	20	50T	0,2	20	75T	0,8	20	100T	0,4	20
40T	0,8	30	50T	0,6	30	75T	0,4	30	100T	0,2	30
40T	0,2	50	50T	0,8	50	75T	0,6	50	100T	0,8	50

Tabela 2: Resultados para as instâncias com 40 e 50 nós - 30 execuções

Instâncias	Benders		SILS-RVND				
	Valor Ótimo	Tempo (s)	Melhor valor	Valor médio	*Gap* melhor (%)	*Gap* médio (%)	Tempo médio (s)
40L-0,4-20	14635,61	34,67	14635,61	14609,39	0,00	0,18	7,62
40L-0,6-20	14099,19	20,11	14099,19	14099,19	0,00	0,00	1,36
40L-0,8-20	14099,19	12,22	14099,19	14099,19	0,00	0,00	1,26
40T-0,2-20	6404,58	35,10	6404,58	6404,58	0,00	0,00	1,52
40T-0,6-20	6404,58	16,54	6404,58	6404,58	0,00	0,00	1,26
40T-0,8-20	6404,58	15,17	6404,58	6404,58	0,00	0,00	1,21
40L-0,2-30	52299,43	106,70	52299,43	52262,78	0,00	0,07	40,56
40L-0,6-30	44213,55	40,06	44213,55	44213,55	0,00	0,00	16,40
40L-0,8-30	42990,07	27,32	42990,07	42990,07	0,00	0,00	13,35
40T-0,2-30	36135,48	55,02	36135,48	36135,48	0,00	0,00	7,36
40T-0,4-30	33918,82	39,19	33918,82	33918,82	0,00	0,00	5,52
40T-0,6-30	32709,49	30,78	32709,49	32709,49	0,00	0,00	5,11
40L-0,2-50	130906,36	74,92	130906,36	130295,59	0,00	0,47	117,72
40L-0,4-50	124056,22	97,30	124056,22	124042,75	0,00	0,01	54,61
40L-0,6-50	120056,17	49,27	120056,17	120056,17	0,00	0,00	39,26
40T-0,4-50	107847,14	55,77	107847,14	107847,14	0,00	0,00	7,82
40T-0,6-50	105857,00	49,38	105857,00	105857,00	0,00	0,00	5,50
40T-0,8-50	105603,50	32,36	105603,50	105603,50	0,00	0,00	5,38
50L-0,2-20	17666,95	148,99	17666,95	17500,77	0,00	0,94	48,07
50L-0,4-20	14588,82	87,25	14588,82	14588,82	0,00	0,00	9,97
50L-0,8-20	13920,55	29,77	13920,55	13920,55	0,00	0,00	2,64
50T-0,4-20	8001,47	56,35	8001,47	8001,47	0,00	0,00	2,44
50T-0,6-20	8001,47	40,85	8001,47	8001,47	0,00	0,00	2,39
50T-0,8-20	8001,47	37,27	8001,47	8001,47	0,00	0,00	2,53
50L-0,2-30	53200,19	334,89	53200,19	53026,63	0,00	0,33	162,61
50L-0,6-30	43763,67	121,95	43763,67	43756,07	0,00	0,02	91,01
50L-0,8-30	42566,95	227,77	42566,95	42566,95	0,00	0,00	44,57
50T-0,2-30	34513,57	219,72	34513,57	34374,67	0,00	0,40	20,90
50T-0,4-30	31911,87	114,27	31911,87	31911,87	0,00	0,00	11,05
50T-0,8-30	31220,52	58,82	31220,52	31220,52	0,00	0,00	9,89
50L-0,4-50	124770,82	265,10	124770,82	124651,76	0,00	0,10	178,74
50L-0,6-50	119972,77	445,13	119972,77	119889,73	0,00	0,07	162,20
50L-0,8-50	118299,45	82,92	118299,45	118299,45	0,00	0,00	55,16
50T-0,2-50	110962,98	289,18	110962,98	110587,84	0,00	0,34	58,15
50T-0,4-50	106451,33	165,83	106451,33	106451,33	0,00	0,00	18,05
50T-0,6-50	105260,70	115,27	105260,70	105089,41	0,00	0,16	19,98
Média	—	100,92	—	—	0,00	0,09	34,25

$iterMax = 10$ e $vezesMax = 3$. O valor obtido para $vezesMax$ mostra que é mais vantajoso intensificar a exploração em uma mesma região do espaço de soluções durante algumas iterações ao invés de alterá-la a cada iteração do método, apontando que o *Smart* ILS é mais adequado do que o ILS, neste caso.

4.3. Resultados dos Experimentos

Os resultados dos testes computacionais são apresentados nas Tabelas 2 e 3. A Tabela 2 mostra os resultados obtidos com as instâncias de 40 e 50 nós, enquanto a Tabela 3 exibe os resultados das instâncias de tamanho 75 e 100. Nessas tabelas, a primeira coluna indica as instâncias testadas, apresentadas no formato i-α-r, em que i indica a instância AP utilizada, α representa o fator de desconto nos arcos entre concentradores e r representa a receita. Em seguida, são apresentados o valor ótimo e o melhor tempo de resolução, apontados em Oliveira et al. [2021], em que as instâncias foram resolvidas utilizando um algoritmo exato baseado no método de decomposição de Benders (ressaltamos que os testes desse artigo foram executados na mesma máquina utilizada para avaliar o desempenho dos algoritmos deste trabalho). As próximas colunas das tabelas mos-

Tabela 3: Resultados para as instâncias com 75 e 100 nós - 30 execuções

Instâncias	Benders		SILS-RVND				
	Valor Ótimo	Tempo (s)	Melhor valor	Valor médio	*Gap* melhor (%)	*Gap* médio (%)	Tempo médio (s)
75L-0,4-20	15004,99	1348,26	15004,99	14993,32	0,00	0,08	74,89
75L-0,6-20	14058,46	732,49	14058,46	14058,46	0,00	0,00	9,89
75L-0,8-20	14058,46	577,37	14058,46	14058,46	0,00	0,00	8,76
75T-0,2-20	1144,13	931,39	1144,13	1144,13	0,00	0,00	6,74
75T-0,4-20	1144,13	627,04	1144,13	1144,13	0,00	0,00	6,38
75T-0,6-20	1144,13	545,76	1144,13	1144,13	0,00	0,00	6,79
75L-0,2-30	53570,37	3516,02	53570,37	53509,30	0,00	0,11	466,83
75L-0,4-30	48097,7	1988,41	48097,70	48069,08	0,00	0,06	329,31
75L-0,8-30	43645,17	932,62	43645,17	43645,17	0,00	0,00	160,38
75T-0,2-30	25999,62	2298,2	25999,62	25999,62	0,00	0,00	27,08
75T-0,6-30	25999,62	872,35	25999,62	25999,62	0,00	0,00	24,72
75T-0,8-30	25999,62	644,32	25999,62	25999,62	0,00	0,00	22,57
75L-0,2-50	131831,98	3464,41	131831,98	131265,92	0,00	0,43	771,22
75L-0,4-50	125133,91	2366,14	125133,91	125082,02	0,00	0,04	598,43
75L-0,6-50	120693,62	2571,4	120693,62	120680,83	0,00	0,01	385,46
75T-0,2-50	99186,32	2374,03	99186,32	99186,32	0,00	0,00	54,20
75T-0,4-50	99186,32	1561,35	99186,32	99186,32	0,00	0,00	54,50
75T-0,8-50	99186,32	818,08	99186,32	99186,32	0,00	0,00	47,00
100L-0,2-20	17616,73	9427,82	17616,73	17592,67	0,00	0,14	551,44
100L-0,4-20	14211,00	6299,13	14211,00	14211,00	0,00	0,00	92,94
100L-0,8-20	13603,42	1874,36	13603,42	13603,42	0,00	0,00	26,40
100T-0,2-20	2116,57	3470,80	2116,57	2116,57	0,00	0,00	17,68
100T-0,6-20	2116,57	1781,70	2116,57	2116,57	0,00	0,00	17,38
100T-0,8-20	2116,57	1610,26	2116,57	2116,57	0,00	0,00	17,70
100L-0,4-30	47173,46	6411,47	47173,46	47133,38	0,00	0,08	770,18
100L-0,6-30	43747,51	4758,21	43747,51	43747,51	0,00	0,00	494,79
100L-0,8-30	42790,52	3290,31	42790,52	42790,52	0,00	0,00	292,94
100T-0,4-30	25271,32	3435,93	25271,32	25271,32	0,00	0,00	52,91
100T-0,6-30	25271,32	2457,55	25271,32	25271,32	0,00	0,00	58,30
100T-0,8-30	25271,32	2059,14	25271,32	25271,32	0,00	0,00	55,47
100L-0,2-50	130437,81	21249,75	130437,81	129929,78	0,00	0,39	1691,27
100L-0,6-50	119319,31	6389,96	119319,31	119290,06	0,00	0,02	754,96
100L-0,8-50	117676,01	3889,53	117676,01	117676,01	0,00	0,00	372,83
100T-0,2-50	98113,77	8729,69	98113,77	98113,77	0,00	0,00	136,29
100T-0,4-50	98113,77	4295,65	98113,77	98113,77	0,00	0,00	122,20
100T-0,6-50	98113,77	3667,19	98113,77	98113,77	0,00	0,00	124,58
Média	—	3424,11	—	—	0,00	0,04	241,82

tram o melhor valor, o valor médio, o *gap* do melhor valor, o *gap* do valor médio e o tempo médio, registrados em 30 execuções do algoritmo. O *gap* do melhor valor e o *gap* do valor médio foram calculados por:

$$\textit{Gap}\ \text{melhor} = 100 \times \frac{\text{Valor ótimo} - \text{Melhor valor}}{\text{Valor ótimo}} \tag{18}$$

e

$$\textit{Gap}\ \text{médio} = 100 \times \frac{\text{Valor ótimo} - \text{Valor médio}}{\text{Valor ótimo}}. \tag{19}$$

A Tabela 2 indica que o algoritmo proposto apresentou bons resultados para as instâncias de 40 e 50 nós. O valor ótimo foi atingido para todas as instâncias testadas e os valores médios ficaram próximos ou iguais ao valor ótimo para a maioria dos casos. Observe que o maior *gap* médio registrado foi 0,94 para a instância 50L-0,2-20. Quanto ao tempo de resolução, nota-se que os tempos médios do SILS-RVND são inferiores aos tempos gastos pelo método exato, com exceção da instância 40L-0,2-50. Em geral, as instâncias com menores valores de α são mais desafiadoras, pois um maior desconto nos arcos entre concentradores estimula a instalação de uma

maior quantidade de concentradores e arcos entre concentradores na rede.

Os resultados exibidos na Tabela 3, referentes às instâncias de tamanho 75 e 100, mostram que o algoritmo heurístico também apresentou um ótimo desempenho para esse grupo de instâncias. Em relação à qualidade da solução produzida, ele foi capaz de alcançar o valor ótimo para todas as instâncias. Foram obtidos bons valores médios, como pode ser observado pelos pequenos *gaps* médios registrados. Para essas instâncias, vê-se que os tempos computacionais obtidos pelo SILS-RVND são muito inferiores em relação ao método exato, principalmente para as instâncias com um fator de desconto constante baixo e, consequentemente, com uma maior economia de escala. Observe, por exemplo, que a média dos tempos registrada pelo método heurístico é 14 vezes menor do que a obtida com a solução via decomposição de Benders.

5. Considerações Finais

Neste trabalho tratou-se o problema de localização de concentradores com maximização do lucro, que tem por objetivo a determinação do projeto de uma rede de concentradores de modo a maximizar o lucro total. Para resolvê-lo, foi desenvolvido um algoritmo baseado na metaheurística *Smart* ILS, utilizando, para a busca local, o método RVND. O algoritmo proposto, nomeado SILS-RVND, foi testado e os resultados comparados com aqueles apontados em Oliveira et al. [2021], no qual as instâncias foram resolvidas por um método exato. Pelos experimentos, o método apresentou um bom desempenho computacional em relação à qualidade da solução produzida e ao tempo gasto para resolver as instâncias, uma vez que os valores ótimos para todas as 72 instâncias testadas foram atingidos em tempos computacionais baixos. Além disso, a diferença de tempo entre o SILS-RVND e o método exato foi bastante expressiva para as instâncias de tamanho 100.

Para trabalhos futuros, pretende-se investigar as limitações e particularidades do método desenvolvido através da resolução de instâncias maiores e mais desafiadoras do problema.

Agradecimentos

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, código de financiamento 001), à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG, processo PPM CEX 0676/17), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, processo 303266/2019-8), ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) e à Universidade Federal de Ouro Preto (UFOP) pelo apoio ao desenvolvimento deste trabalho.

Referências

Alibeyg, A., Contreras, I., e Fernández, E. (2016). Hub network design problems with profits. *Transportation Research Part E: Logistics and Transportation Review*, 96:40–59.

Alibeyg, A., Contreras, I., e Fernández, E. (2017). Exact solution of hub network design problems with profits. *European Journal of Operational Research*, 266(1):57–71.

Alumur, S. e Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190:1–21.

Bollapragada, R., Li, Y., e Rao, U. S. (2006). Budget-constrained, capacitated hub location to maximize expected demand coverage in fixed-wireless telecommunication networks. *INFORMS Journal on Computing*, 18:422–432.

Campbell, J. F. e O'Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, 46:153–169.

Chen, J.-F. (2007). A hybrid heuristic for the uncapacitated single allocation hub location problem. *Omega*, 35(2):211–220.

Contreras, I., Tanash, M., e Vidyarthi, N. (2017). Exact and heuristic approaches for the cycle hub location problem. *Annals of Operations Research*, 258:655–677.

Ernst, A. T. e Krishnamoorthy, M. (1998). Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research*, 104: 100–112.

Ernst, A. T. e Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4:139–154.

Floyd, R. W. (1962). Algorithm 97: Shortest path. *Communications of the ACM*, 5:345.

Hansen, P., Mladenovic, N., Todosijevic, R., e Hanafi, S. (2017). Variable neighborhood search: basics and variants. *EURO Journal of Computational Optimization*, 5:423–454.

Klincewicz, J. (1991). Heuristics for the p-hub location problem. *European Journal of Operational Research*, 53(1):25–37.

Lin, C. C., Lin, J. Y., e Chen, Y. C. (2012). The capacitated p-hub median problem with integral constraints: An application to a chinese air cargo network. *Applied Mathematical Modelling*, 36: 2777–2787.

Lourenço, H., Martin, O., e Stützle, T. (2003). Iterated local search. In Glover, F. e Kochenberger, G., editors, *Handbook of Metaheuristics*, volume 57, p. 321–353. International Series in Operations Research & Management Science, Boston, MA.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., e Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.

Martins de Sá, E., Morabito, R., e de Camargo, R. S. (2018). Benders decomposition applied to a robust multiple allocation incomplete hub location problem. *Computers and Operations Research*, 89:31–50.

Oliveira, F. A., Martins de Sá, E., e de Souza, S. R. (2021). Benders decomposition applied to profit maximizing hub location problem with incomplete hub network. [Artigo submetido à Computers & Operations Research em Janeiro de 2021].

Reinsma, J. A., Penna, P. H. V., e Souza, M. J. F. (2018). Um algoritmo simples e eficiente para resolução do problema do caixeiro viajante generalizado. In *Anais do* 50^o *Simpósio Brasileiro de Pesquisa Operacional*, Rio de Janeiro. SOBRAPO, Galoá.

Saberi, M. e Mahmassani, H. S. (2013). Modeling the airline hub location and optimal market problems with continuous approximation techniques. *J. of Transport Geography*, 30:68–76.

Souza, M., Coelho, I., Ribas, S., Santos, H., e Merschmann, L. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, 207(2):1041–1051.

Subramanian, A., Drummond, L., Bentes, C., Ochi, L., e Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911. Metaheuristics for Logistics and Vehicle Routing.

Taherkhani, G. e Alumur, S. A. (2018). Profit maximizing hub location problems. *Omega*, 86:1–15.

Taherkhani, G., Alumur, S. A., e Hosseini, S. M. (2019). Benders decomposition for profit maximizing hub location problems with capacity allocation. Disponível em Waterloo's Institutional Repository, `https://uwspace.uwaterloo.ca/handle/10012/14758`, acesso em 14/05/2021.